

兰州理工大学

## 科研成果汇总

|       |                           |
|-------|---------------------------|
| 学号:   | 231081101005              |
| 研究生:  | 彭钰                        |
| 导师:   | 李二超 教授                    |
| 研究方向: | 流程工业智能检测与优化               |
| 论文题目: | 面向城市污水处理过程软测量的数据治理与建模方法研究 |
| 学科:   | 控制理论与控制工程                 |
| 学院:   | 自动化与电气工程学院                |
| 入学时间: | 2023年9月                   |

# 目录

1. 论文检索报告
2. Peng Y, Li E. Basis expansion imputation network for incomplete multivariate sensor data in wastewater treatment system[J]. Journal of Water Process Engineering, 2024, 63: 105536. (SCI, WOS:001247336600001)
3. Peng Y, Li E. A generative model-based coevolutionary training framework for noise-tolerant softsensors in wastewater treatment processes[J]. Complex & Intelligent Systems, 2025, 11(5): 1-25. (SCI, WOS:001446645300002)
4. Peng Y, Li E. MVSG-GS: A metaheuristic virtual sample generation method for soft sensor modeling based on guidelines sharing[J]. Expert Systems with Applications, 2025: 128427. (SCI, WOS:001446645300002)
5. Li E, Li H, Peng Y. Competitive-Cooperative Multi-Task Optimization Algorithm with Historical Success Archive[J]. ACM Transactions on Knowledge Discovery from Data, 2025 (0): 000-000. (SCI, WOS:001677144000001)
6. 彭钰, 李二超, 等. 污水处理软测量虚拟样本生成方法、装置、设备和介质. 发明专利. (已授权, 专利号: CN202510788607.9)





图书馆

## 文献检索报告



兰州理工大学图书馆信息咨询与学科服务部

报告编号: R2026-0343

机构: 兰州理工大学 电气工程与信息工程学院

姓名: 彭钰 [231081101005]

著者要求对其在国内外学术出版物所发表的科技论著被以下数据库收录情况进行查证。

检索范围:

- 科学引文索引 (Science Citation Index Expanded): 1900年-2026年

检索结果:

| 检索类型     | 数据库          | 年份范围        | 总篇数 | 第一作者 | 通讯作者 |
|----------|--------------|-------------|-----|------|------|
| SCI-E 收录 | SCI-EXPANDED | 2024 - 2026 | 4   | 3    | 111  |



委托人声明:

本人委托兰州理工大学图书馆信息咨询与学科服务部查询论著被指定检索工具收录情况, 经核对检索结果, 附件中所列文献均为本人论著, 特此声明。

作 者 (签字): 彭钰

完成人 (签字): 刘 玮

完 成 日 期 : 2026年4月27日

完成单位 (盖章): 兰州理工大学图书馆信息咨询与学科服务部

(本检索报告仅限校内使用)





图书馆

文献检索报告  
SCI-E 收录

兰州理工大学图书馆信息咨询与学科服务部

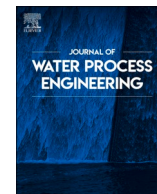
报告编号: R2026-0343 SCI-E 收录

数据库: 科学引文索引 (Science Citation Index Expanded)  
时间范围: 2024年至2026年作者姓名: 彭钰  
作者单位: 兰州理工大学 电气工程  
与信息工程学院检索人员: 刘 玮  
检索日期: 2026年4月27日

检索结果: 被 SCI-E 收录文献 4 篇

| #  | 作者                      | 地址                                                                                                                        | 标题                                                                                                                                          | 来源出版物                                                                                     | 文献类型         | 入藏号                         |
|----|-------------------------|---------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|--------------|-----------------------------|
| 1  | Peng, Y; Li, ER         | [Peng, Yu; Li, Erchao]<br>Lanzhou Univ Technol,<br>Coll Elect & Informat<br>Engn, Lanzhou 730050,<br>Peoples R China      | A generative model-<br>based coevolutionary<br>training framework for<br>noise-tolerant softsensors<br>in wastewater treatment<br>processes | <i>COMPLEX &amp;<br/>INTELLIGENT<br/>SYSTEMS</i><br>2025, 11 (5): 212.                    | J<br>Article | WOS:0014<br>466453000<br>02 |
| 2  | Peng, Y; Li, EC         | [Peng, Yu; Li, Erchao]<br>Lanzhou Univ Technol,<br>Coll Elect & Informat<br>Engn, Lanzhou 730050,<br>Peoples R China      | MVSG-GS: A<br>metaheuristic virtual<br>sample generation<br>method for soft sensor<br>modeling based on<br>guidelines sharing               | <i>EXPERT SYSTEMS<br/>WITH APPLICATIONS</i><br>2025, 290: 128427.                         | J<br>Article | WOS:0015<br>085029000<br>02 |
| 3  | Li, ER; Li, HX; Peng, Y | [Li, Erchao; Li, Hongxu;<br>Peng, Yu] Lanzhou Univ<br>Technol, Coll Elect &<br>Informat Engn, Lanzhou,<br>Peoples R China | Competitive-Cooperative<br>Multi-Task Optimization<br>Algorithm with Historical<br>Success Archive                                          | <i>ACM TRANSACTIONS<br/>ON KNOWLEDGE<br/>DISCOVERY FROM<br/>DATA</i><br>2026, 20 (1): 17. | J<br>Article | WOS:0016<br>771440000<br>01 |
| 4  | Peng, Y; Li, ER         | [Peng, Yu; Li, Erchao]<br>Lanzhou Univ Technol,<br>Coll Elect & Informat<br>Engn, Lanzhou 730050,<br>Peoples R China      | Basis expansion<br>imputation network for<br>incomplete multivariate<br>sensor data in wastewater<br>treatment system                       | <i>JOURNAL OF WATER<br/>PROCESS<br/>ENGINEERING</i><br>2024, 63: 105536.                  | J<br>Article | WOS:0012<br>473366000<br>01 |
| 合计 |                         |                                                                                                                           |                                                                                                                                             |                                                                                           |              | 4                           |





# Basis expansion imputation network for incomplete multivariate sensor data in wastewater treatment system

Yu Peng, Erchao Li<sup>\*</sup>

College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

## ARTICLE INFO

Editor: Guangming Jiang

### Keywords:

Imputation  
Deep learning  
Multivariate time series  
Interpretability  
Wastewater treatment process

## ABSTRACT

Time series data collected from wastewater treatment process (WWTP) are frequently incomplete due to the inherent instability of the operating site. The presence of numerous missing samples in multivariate time series significantly impedes the application of data-driven modeling approaches. Motivated by the success of deep learning methods in the application of time series forecasting, we propose a novel approach named the basis expansion imputation network (BEIN) to learn the overall distribution of a multivariate WWTP time series dataset, enabling the timely and effective imputation of missing samples for each sample point. BEIN outperformed five baseline models in average imputation performance for rain, storm, and BMS2 influent under the 8 missing rate scenario (10 %–80 %) by 5 %, 2.9 %, and 5.7 % for RMSE, and 6.92 %, 8.93 %, and 9.16 % for MAPE, respectively. Ablation study proved that the modules of proposed imputation model could be targeted to extract different information patterns from interacted wastewater influent data.

## 1. Introduction

In tandem with the international community's awareness of sustainable development, wastewater treatment plants (WWTPs) face increasingly stringent discharge standards. The design of WWTPs has become increasingly complex and operationally challenging to fulfill the diverse and crucial demands for resource conservation and operational efficiency [1]. Addressing the escalating costs linked with the wastewater treatment process entails primarily implementing digital modeling for WWTPs. This modeling provides dependable process information as the basis for optimizing operations [2,3]. The advancement of information technology and industrial automation emphasizes the importance of leveraging multivariate data gathered by sensors to develop models [4]. Utilizing these models is a pivotal strategy for attaining intelligent control, decision-making, and optimization in intricate industrial processes [5]. However, challenges arise due to deficiencies in sensors functionality, substantial communication time delays, and abrupt fluctuations in network state.

Modeling biochemical reaction processes has long been a significant challenge in academic and industrial fields. The activated sludge wastewater treatment process, which exemplifies intricate biochemical reactions, faces obstacles due to the scarcity and sparsity of modeling data related to crucial operational indicators [6]. These challenges lead

to data incompleteness, hampering the construction of accurate and robust detection and identification models. Characterizing these data with the necessary completeness in the desired modeling sample space proves to be a challenging task. We attribute the aforementioned issues to the inadequate completeness of the time series distributions, resulting from errors and missingness in the sensor data. This inadequacy hinders the application of models constructed from raw data to real processes characterized by dynamic changes in operating conditions.

As mentioned above, a prerequisite for data-driven modeling is the availability of high-quality modeling information, making imputation a crucial step in modeling complex industrial systems. The objective of imputation is to accurately infer the unobserved regions of sensor data. Traditional methods for handling the imputation of missing data can be broadly categorized into two types: 1) statistical imputation methods, and 2) machine learning based imputation methods. Statistical techniques typically employ straightforward methods, such as mean imputation and regression imputation [7]. However, these computational methods suffer from drawbacks that lead to biased results, including the underutilization of information and variations in the distribution of the original data, as well as the overestimation of the correlation between real and imputed samples. To address these challenges, researchers have explored the application of machine learning algorithms, such as multivariate imputation by chained equations (MICE) [8] and matrix

<sup>\*</sup> Corresponding author.

E-mail address: [lecstarr@163.com](mailto:lecstarr@163.com) (E. Li).

<https://doi.org/10.1016/j.jwpe.2024.105536>

Received 27 January 2024; Received in revised form 12 May 2024; Accepted 20 May 2024

Available online 25 May 2024

2214-7144/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

completion algorithms [9]. However, machine learning based methods face limitations in extracting valuable features, especially when dealing with multivariate time series data collected from complex systems [10]. Recent studies indicate that deep learning methods exhibits superior characterization capabilities compared to statistical methods or traditional machine learning models, thereby generating increased interest. In industrial scenarios, sensor data often exhibits strong temporal correlations, and clearly considering these correlations is crucial for developing accurate and robust predictive models. Interactions, non-linearity, and non-standard distribution forms are all potentially relevant features. However, traditional methods tend to sample sequences based on fixed time windows, thus unable to capture all information of a sequence at a global level. Additionally, classic recursive learning to some extent ignores the guiding results between different levels, from global to local, leading to a lack of continuity and efficiency in training.

In response to the above analyses, this paper proposes a basis expansion imputation network (BEIN) for multivariate time series in industrial scenarios to address the incompleteness. Three wastewater treatment plant datasets are used in the experimental part to verify the rationality and effectiveness of the proposed imputation network.

The contributions of this paper are summarized as follows:

- (1) BEIN adopts a multi-period time series decomposition strategy, integrating residual structure, identity block and seasonality block to map sensor data with typical compound periodic characteristic. This approach effectively addresses the complex dependencies among multiple variables and challenges posed by different periodic combinations in the context of wastewater treatment.
- (2) The mutual information module selects exogenous information as guidance in multivariate time series, which helps improve the model's performance in capturing inter-sequence correlations, further enhancing the predictive capability of the model.
- (3) The decomposition of time series and the modeling of components relevance make the framework interpretable and achieve an improvement over the baselines. By employing those methods, it becomes feasible not only to harmonize and enhance the overall quality of the data but also to assess the consistency of sensor data.
- (4) Imputation experiments based on different datasets and ablation experiments show that the proposed imputation model can effectively improve the quality of raw sensor data collected from wastewater treatment processes in several aspects.

The rest of the paper is organized as follows: Section 2 provides a thorough review of relevant methods for imputation and time series forecasting. Section 3 briefly outlines the problem formulation and provides a detailed introduction to the methods of multivariate time series decomposition and the design of mutual information models. Section 4 presents the experimental results and discusses the validity of BEIN from various perspectives. Finally, Section 5 highlights the conclusion and future prospects.

## 2. Materials and methods

### 2.1. Deep learning based imputation methods

Many deep learning based methods have been developed for imputation. Liu et al. [11] proposed a data imputation network (DINet) based on super-resolution perception (SRP). The design of DINet prioritizes generality, integrating a unit imputation layer that strategically substitutes missing samples. Tutz et al. [12] emphasized the importance of the pattern of missing samples and the selection of relevant variables in the imputation process. They proposed a weighted nearest neighbor imputation method based on Lq-norm distances, which tends to have a smaller imputation error than other nearest neighbor estimates.

Raghuvamsi et al. [13] highlight the advantages of using deep learning models for imputation. Additionally, they proposed a novel deep learning method utilizes a Transformer model with a bidirectional long short-term memory (Bi-LSTM) layer to capture temporal information contained in the data and maintain system observability. Ma et al. [14] proposed a deep learning model named the time-series generative adversarial imputation network (TSGAIN) for the reconstruction and imputation of missing samples in multivariate time series data. The TSGAIN considers the multiplicity of data in modern scenarios and uses a spatio-temporal structure to effectively preserve the original information of the time series. Further, Wang et al. [15] proposed a novel approach for missing data imputation using a generative adversarial guider imputation network (GAGIN). The proposed GAGIN outperforms state-of-the-art approaches and traditional methods in terms of imputation accuracy, addressing the challenges of capturing local details and guiding the imputation process effectively. Velasco-Gallego et al. [16] proposed a comparative study of 20 time series forecasting algorithms for real-time missing data imputation. They highlight the importance of accurate data imputation to avoid biased results and improve decision-making processes. Aiming at the temporal characteristics of industrial datasets, Pan et al. [17] proposed an adaptive learning median-filled deep autoencoder (AM-DAE) to impute missing samples in time series. AM-DAE realizes the transfer of imputation information during the training process by successively using the median value of the input data to replace the missing samples, and propagating the error through the backpropagation algorithm to each layer of the DAE.

From the above overview, it is clear that there is a common link between deep learning based imputation models and time series forecasting models. One of the commonalities is that they both require analysis of existing data to determine missing samples or to build time series models aimed at better understanding and utilizing the given information. Traditional imputation models use various statistical methods or machine learning algorithms to infer missing samples so as to maintain the integrity of the data. And time series forecasting models similarly need to analyze past time series with the help of statistical methods or machine learning models in order to predict values at a future point in time. Time series forecasting predicts future values by looking at past data trends, and imputation is also based on the sequential relationship of time, using information from forward and backward points to impute missing samples.

### 2.2. Multivariate time series forecasting

Time series forecasting is a longstanding research focus that spans decades. In contrast to other continuous data types, such as language or video, time series, while continuously recorded, capture only a few scalar values at each time point. Recognizing that a singular time point often lacks sufficient semantic information for analysis, numerous studies have directed their attention to discerning changing patterns within the data over time [18]. This is because time-varying information can unveil the inherent properties of the time series. However, real-world time series changes invariably demonstrate multiplicity and periodicity, encompassing temporal patterns where multiple changes intermingle and overlap, posing a formidable challenge to time-varying modeling. Multivariate time series forecasting models tend to be notably intricate, given their involvement with multiple variables and potentially complex relationships (e.g., periodic patterns, seasonal variations, etc.). Such models must adeptly capture interactions among different variables and the cyclical fluctuations in these variables over time. Many classical statistical modeling approaches assume that temporal changes follow predefined patterns, such as vector autoregressive regression (VAR) [19] or autoregressive integrated moving average (ARIMA) models [20]. Additionally, the seasonal ARIMA (SARIMA) model incorporates a seasonality module into the ARIMA framework to transform non-stationary series into stationary ones [21]. However, the variations in periodic multivariate time series are often too complicated



to be captured by simple linear dependencies in traditional statistical models, constraining the practical applicability of these classical methods.

The conventional machine learning models lack flexibility and data adaptation, particularly incompatibility of multi-scenario activities and a lack of discrete data processing capabilities. In recent years, numerous deep learning models have emerged to capture complex temporal patterns in time series data. Recurrent neural networks (RNNs) and their variants, such as long short-term memory networks (LSTM) [22,23], demonstrate the capability to handle the dynamics and nonlinearities of time series, resulting in superior performance in such tasks. With the rise of deep learning techniques, researchers have applied convolutional neural networks (CNNs) to time series analysis. In this context, the convolutional layer can capture local dependencies of the time series and extract high-level features through multi-layer convolution [24]. Among these, the temporal convolutional network (TCN) is a notable example that employs one-dimensional convolution along the time dimension to address the dependency challenge of long-term data [25]. Transformer model facilitates the capture of long-range dependencies through a self-attention mechanism, particularly useful for handling periodic time series [26,27]. Moreover, Autoformer introduces an adaptive decomposition mechanism [28], capable of separating trend and seasonality from the time series. This decomposition enhances the model's efficiency in capturing long-term trends and periodic patterns while reducing computational complexity. Additionally, Autoformer improves the traditional attention mechanism by allocating more attention to the time points crucial for forecasting, thereby enhancing the accuracy and efficiency of the model.

In practical applications, multivariate time series forecasting tasks often involve multi-scale data and inputs from different sources [29]. To deal with this complexity, researchers prefer to design multi-scale and multi-input models [30], which usually combine different neural network architectures such as CNN, RNN, and Transformer to better capture the complex patterns of the data. While deep learning has achieved significant advancements in forecasting periodic multivariate time series, it encounters various challenges such as handling non-stationary and multi-scale data, integrating external features, and facing limitations in resource-constrained industrial settings. In addition, the black-box nature of deep learning models is an issue that cannot be ignored. In some industrial scenarios, transparency and interpretability for decision making is crucial, and the complex structure of deep learning models may be difficult to interpret, making it difficult to understand the decision-making process of the model, thus affecting its credibility in some critical applications.

### 2.3. Relevance of wastewater components

The multivariate time series data collected by modern sensing technologies may challenge the classical modeling framework because the internal environments itself is inherently uncertain [31]. The dynamic nature of the operating environment and the unknown variability of influent load make classical methods less efficient for data imputation in WWTP. Inaccurate data can lead to laborious calibration procedures and unjustified adaptation of model parameters. Meijer et al. [32] demonstrated the application of integrated mass balancing to data from wastewater treatment plants to enhance the overall data quality. Puig et al. [33] emphasized that the raw operational sensor data of wastewater treatment plants often lack the quality required for modeling optimization or decision management. They proposed a stepwise methodology based on correlation of components to assess the reasonableness of the data and correct errors. The results revealed errors in measured data due to changes in environment dynamics, particularly in effluent flow measurements. The augmented data significantly reduced the standard deviation of measured values, indicating that incorporating information related to the components of activated sludge biochemical reaction process into the data imputation method is effective in

improving data quality and reducing measurement errors.

In conclusion, the research gap lies in the absence of exploration concerning adaptive multivariate time series models. Investigation suggests that uncovering potential interaction information among these variate may enhance the model's generalizability by capturing the latent structure of sensors or the abstract interplay of biochemical reactions. Therefore, we adopt an imputation method dominated by a deep learning mechanism and assisted by the relevance of WWTP components to enhance sensor data quality and provide a robust guarantee for data-driven modeling. To the best of our knowledge, this method is a very early attempt to capture informative details of multivariate data through components relationships in WWTP, combining a deep learning imputation method to automate the validity checking of sensor data in wastewater treatment process. Moreover, since the relevance of components are coupled, this approach improves overall data quality through the unique material interaction characteristics inherent in the wastewater treatment process of the activated sludge method.

### 3. Imputation of miss samples

This section provides a detailed description of the proposed imputation method and subsequently offers an in-depth elucidation of the key modules constituting the framework.

#### 3.1. Problem description

In this study, the imputation is considered as a forecasting problem for a regularly sampled univariate time series. Let  $x_t$  denote a time series with time step  $t$ . The classical autoregressive formulation is to map  $x_{t-L:t} = [x_{t-L}, \dots, x_{t-1}]$  to a neighbouring future series  $x_{t:t+H} = [x_t, \dots, x_{t+H-1}]$ , i.e.:

$$x_{t:t+H} = F_{\theta}(x_{t-L:t}) + \varepsilon_{t:t+H} \quad (1)$$

where  $H$  is the length of the forecast horizon and  $L$  is the length of the lookback window.  $F_{\theta} : \mathbb{R}^L \rightarrow \mathbb{R}^H$  is the mapping function defined by parameter  $\theta$ , and  $\varepsilon$  denotes the independent identically distributed Gaussian noise vector.

The imputation of missing time series is not only dominated by  $x_{t-L:t}$ , but also able to use the information of  $x_{t:t+L}$  as a reference, i.e., a look-forward window is added. In order to avoid adding additional parameters as much as possible, it is assumed that both time windows are of the same length, where  $x_{t:t+H}$  has an equivalent dependence on  $x_{t-L:t}$  and  $x_{t+H:t+H+L}$ . The inherent periodicity of the time series and the diverse composition of different periods, which can be anchored by the associated time step:

$$x_{t:t+H} = F_{\theta}(x_{t-L:t}, x_{t+H:t+H+L}) + \varepsilon_{t:t+H} \quad (2)$$

where  $H$  is the length of the forecast horizon and  $L$  is the length of the lookback window and the lookforward window.  $F_{\theta} : \mathbb{R}^L \rightarrow \mathbb{R}^H$  is the mapping function defined by parameter  $\theta$ , and  $\varepsilon$  denotes the independent identically distributed Gaussian noise vector.

Analyzing the special properties of time-periodic series comprehensively, this section proposes a decoupling equation Eq. (3) that improves Eq. (2) by introducing a potential variable  $l_t$  to represent the period state of the time series at  $t$ :

$$x_{t:t+H} = \hat{F}_{\theta}(x_{t-L:t}, x_{t+H:t+H+L}, l_t) + \varepsilon_{t:t+H} + \varepsilon_{t+H:t+H+L} \quad (3)$$

#### 3.2. Multivariate time series decomposition

Real-world multivariate time series often manifest the compounded effects of diverse periodic processes. The wastewater treatment process, as a crucial part of the city's drainage system, has inherent multi-periodic attribute that increases the complexity of time-series changes. Considering the routine life and regularity of urban residents, the

influent data of WWTPs exhibits typical periodic characteristics, including daily periods, weekly periods, holiday periods, etc. Additionally, the time series data collected from WWTPs often exhibit proximity and trend. Proximity implies that the influent generally do not change suddenly unless a chance event occurs, while trend indicates that changes in detection variables, such as the average daily flow rate or the average daily concentration of pollutant components, follow an overall regularity.

The primary challenge of imputation in time series within the context of complex biochemical reactions lies in capturing the temporal correlation of the data. The variation of wastewater components at a given time is related to their adjacent time nodes. In addition, there is also a correlation between different nodes at the previous time and the node at the current time. Considering the complex mechanism of activated sludge, component changes at more distant time points may also correlate with the biochemical reaction process at the current moment. Addressing the superimposition and coupling of multivariate time series data, we treat the periodicity of the time series as a data feature and propose a decomposition method based on basis expansion [34]. This method highlights identifiable features, screens out periodic subsequences with significant characteristics, and achieves the decomposition of complex time series. It is important to note that, in order to increase the generality of the model, we do not limit ourselves to a specific task when analyzing the periodic characteristics of time series data.

### 3.2.1. Expansion block

The structure of the time series expansion module is shown in Fig. 1. The specific structure of each layer is further elaborated on the left side, and each layer has an identical residual structure consisting of two blocks corresponding to the overall identity feature and the local seasonality feature, respectively. Here  $x_{t-L,t}^{(\ell-1)}$  denotes the residual term after the layer  $(\ell-1)$  expansion, and  $\hat{x}_{t-L,t}^{(\ell-1)}$  denotes the cumulative forecast. In layer  $\ell$ , the residual branch is specified by two parameterized blocks, an identity block (intraperiod module) and a seasonality block (interperiod module).

We first give the update formula for the identity block, which aims to capture the overall fluctuation and trend of the time series. More specifically, the identity block takes  $x_{t-L,t}^{(\ell-1)}$  (incomplete univariate sensor data) as input and decomposes the state of the  $\ell$ -th unfolding term of the current input signal, denoted as  $v_{t-L,t+H}^{(\ell)} \in \mathbb{R}^{L+H}$ .  $v_{t-L,t+H}^{(\ell)}$  consists of a backward decomposition component  $v_{t-L,t}^{(\ell)}$  and a forward forecasting component  $v_{t,t+H}^{(\ell)}$ . Where  $v_{t-L,t}^{(\ell)}$  is utilized to exclude the influence of historical features that have been captured on the current network, and

$v_{t,t+H}^{(\ell)}$  is employed to denote the forecasting that is purely from the  $\ell$ -th block. Furthermore, when moving to the next block, the current processed signal is removed from  $x_{t-L,t}^{(\ell-1)}$ , thus encouraging subsequent blocks of the period module to concentrate on the seasonality terms.

Since  $v_{t-L,t}^{(\ell)}$  is related to the identity information used to generate a portion of the forecasting, we construct the input of the seasonality block as  $x_{t-L,t}^{(\ell-1)} - v_{t-L,t}^{(\ell)}$ . The goal here is to encourage the interperiod module to focus on unresolved patterns. The two outputs of the seasonality block,  $u_{t-L,t}^{(\ell)}$  and  $u_{t,t+H}^{(\ell)}$ , correspond to the local seasonality term and the forecasted expansion term in layer  $\ell$ , respectively.  $u_{t-L,t}^{(\ell)}$  is then removed from  $x_{t-L,t}^{(\ell-1)}$  to update the next layer's input. The idea here is the same as mentioned above, which is to exclude all analyses in the current layer in order to allow the next layer to focus on the unresolved information.

The purpose of decomposing the time series is twofold: to produce partial forecasts using local observations and to exclude redundant information. Identity block aims to extract overarching identity information represented by various combinations of trends from the associated time series. In contrast to the seasonality periodic signals, the identity state exhibits higher identifiability and predictability. Thus, the desired output can be attained through the adoption of a simple architecture, as depicted in Fig. 2(a). The seasonal characteristics of time series are often manifested as the superposition of multiple periods, in order to effectively identify the complex period dependence, the trade-off between model capacity and generalization should be fully considered, so we adopt the generalized block developed by Oreshkin et al. [35], which

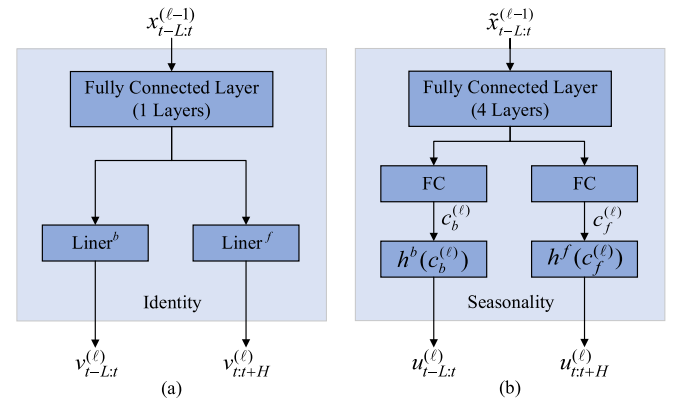


Fig. 2. Details of identity block and seasonality block.

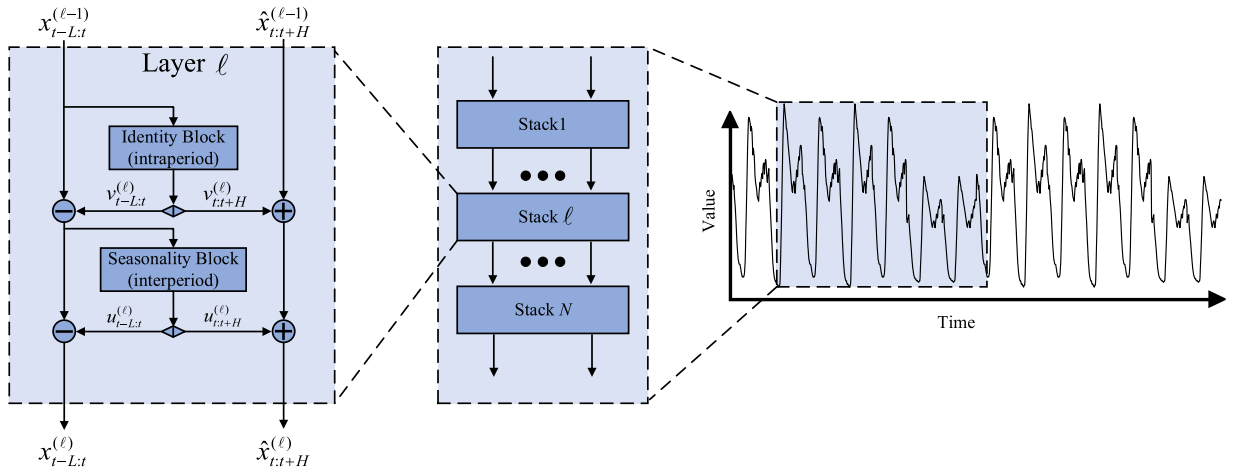


Fig. 1. Overall structure of time series expansion module.

consists of a series of fully-connected layers, and the specific structure is shown in Fig. 2(b).

As shown in Fig. 2(a), a fully connected layer is utilized to encode  $\tilde{x}_{t-L,t}^{(\ell-1)}$  and use two other linear projection functions to obtain the decomposition term  $v_{t-L,t}^{(\ell)}$  and the forecast term  $v_{t,t+H}^{(\ell)}$ .

$$v_{t-L,t}^{(\ell),1} = FC_{\ell}(\tilde{x}_{t-L,t}^{(\ell-1)}), \quad v_{t-L,t}^{(\ell)} = \text{LINEAR}_b^{\ell}(v_{t-L,t}^{(\ell),1}), \quad v_{t,t+H}^{(\ell)} = \text{LINEAR}_f^{\ell}(v_{t,t+H}^{(\ell),1}) \quad (4)$$

The analysis of seasonality begins by establishing a set of coefficients to represent the periodicity of a time series at a specific point in time. This coefficient is dynamically updated through the identification and learning of periodic patterns within the current input. Fig. 2(b) illustrates the detailed architecture of the seasonality block, where  $\tilde{x}_{t-L,t}^{(\ell)}$  is used to denote the input ( $\tilde{x}_{t-L,t}^{(\ell-1)} - v_{t-L,t}^{(\ell)}$ ) of local observations in layer  $\ell$  that do not include the effect of the identity feature signal  $v_{t-L,t}^{(\ell)}$ . After taking  $\tilde{x}_{t-L,t}^{(\ell)} = \tilde{x}_{t-L,t}^{(\ell-1)} - v_{t-L,t}^{(\ell)}$ , we pass it through the fully-connected layers, and then obtain the decomposition coefficients  $c_b^{(\ell)}$  and the forecast coefficients  $c_f^{(\ell)}$  by two linear projections:

$$u_{t-L,t}^{(\ell),1} = FC_{\ell,1}(\tilde{x}_{t-L,t}^{(\ell)}), \quad u_{t-L,t}^{(\ell),2} = FC_{\ell,2}(u_{t-L,t}^{(\ell),1}), \quad u_{t-L,t}^{(\ell),3} = FC_{\ell,3}(u_{t-L,t}^{(\ell),2}) \\ u_{t-L,t}^{(\ell),4} = FC_{\ell,4}(u_{t-L,t}^{(\ell),3}), \quad c_b^{(\ell)} = \text{LINEAR}_b^{\ell}(u_{t-L,t}^{(\ell),4}), \quad c_f^{(\ell)} = \text{LINEAR}_f^{\ell}(u_{t-L,t}^{(\ell),4}) \quad (5)$$

where FC is a standard fully connected layer with ReLU, and LINER denotes the linear projection function. Then these coefficients are sent to the basis layers  $h^b(\cdot)$  and  $h^f(\cdot)$  to obtain the decomposition term  $v_{t-L,t}^{(\ell)}$  and the forecast term  $v_{t,t+H}^{(\ell)}$ , respectively.

### 3.2.2. Seasonality block

As shown in Fig. 2, the output of the identity block is obtained from two base linear mappings, and the output of the seasonality block is obtained from the basis layers  $h^b(\cdot)$  and  $h^f(\cdot)$ . The choice of basis functions relies on time series decomposition techniques aimed at improving the interpretability of BEIN. The general choice of  $h^b(\cdot)$  and  $h^f(\cdot)$  can simply be Fourier series, as it is more competitive and stable than the other interpretable baselines on the period time series correlation benchmarks [36]. In order to represent the complexity of a periodicity consisting of various patterns, we estimate  $u_{t-L,t}^{(\ell)}$  and  $u_{t,t+H}^{(\ell)}$  in Eq. (5) by means of a parameterized periodic function  $h(t)$ , which has sufficient capacity to contain different periods. The joint optimization of basis functions and coefficients with random initialization poses significant challenges. This is primarily due to the highly nonconvex nature of the imputation problem, which can cause the coefficients converge to local optima that do not align with the desired periodicity. To address these challenges, we utilize the fast Fourier transform (FFT) to analyze the time series in the frequency domain, as outlined below:

$$A = \text{Avg}(\text{Amp}(\text{FFT}(X))), \{f_1, \dots, f_k\} = \underset{f \in \{1, \dots, [T/2]\}}{\text{argRandk}}(A), p_i = \left\lceil \frac{T}{f_i} \right\rceil, i \in \{1, \dots, k\} \quad (6)$$

where  $\text{FFT}(\cdot)$  and  $\text{Amp}(\cdot)$  denote the fast Fourier transform and calculation of amplitude values, respectively.  $X$  represents the univariate time series,  $A \in \mathbb{R}^T(t)$  denotes the amplitude at each frequency, averaged from the time series data using  $\text{Avg}(\cdot)$ .

Considering the sparsity of the frequency domain and avoiding the noise caused by meaningless high frequencies, according to Zhou, et al. [37], we randomly select  $k$  amplitude values with different features to obtain the most significant frequencies  $\{f_1, \dots, f_k\}$  with the corresponding unnormalized amplitudes  $\{A_{f_1}, \dots, A_{f_k}\}$ . These selected

frequencies also correspond to the lengths of the  $k$  periods  $\{p_1, \dots, p_k\}$ . Due to the covariance in the frequency domain, we only consider frequencies within  $\{1, T/2\}$  and Eq. (6) can be simplified as:

$$\{A_{f_1}, \dots, A_{f_k}\}, \{f_1, \dots, f_k\} = \text{Period}(X) \quad (7)$$

Based on the selected frequencies  $\{f_1, \dots, f_k\}$  and the amplitudes  $\{A_{f_1}, \dots, A_{f_k}\}$ , the time series data is decomposed into a combination of multiple subperiodic time series:

$$h(t) = \sum_{i=1}^k A_{s,\ell,i}^f \cos\left(2\pi f_i \frac{t}{N}\right) + A_{s,\ell,i+k}^f \sin\left(2\pi f_i \frac{t}{N}\right) \quad (8)$$

where  $N$  denotes the hyperparameters controlling the harmonic oscillations.

It's important to note that the decomposed time series acquired through period extraction might not be directly applicable for learning the underlying data distribution due to two crucial considerations:

1. Different operational conditions may induce irregular identities. For example, the reaction rate of activated sludge under varying weather conditions and the concentration of influent components can differ significantly [38]. These dynamic characteristics play a crucial role in understanding the environmental situation and changing trends within the WWTP. Ignoring the dynamic nature of time-series data in particular scenarios could lead to confusion in the resulting imputed data [39].
2. The seasonality block, which utilizes feature information, efficiently decomposes complex time series into sequences with varying periods, substantially reducing memory usage and computational complexity [40]. Nevertheless, this decomposition and reconstruction process may introduce significant redundant information.

Therefore, these aforementioned limitations require further resolution before formally applying this method to the real-world WWTP sensor data imputation problems. The mutual information module of wastewater components is proposed to capture and rectify the dynamics of multivariate time series collected in wastewater treatment process.

### 3.3. Mutual information of components

The concentration of pollutants in the wastewater treatment plant is not only an important basis for the design and operation of the wastewater treatment process, but also a necessary condition to ensure that the effluent is stabilized to meet the treatment objectives. According to Ge et al. [41], complex information compiled from several sensors is challenging to fit in deep generic learning. In general, the relevance of time series data between different components are high. The active sludge based bioreactor under the default conditions exhibits specific characteristics, provides a comprehensive understanding of the essential features characterizing the bioreactor setup in the default scenario [42].

The ratios between various quality parameters in wastewater influent can influence the selection and operation of treatment processes [43]. The information collected on WWTP is focussed on discharge legislation, mainly influent and effluent are measured [44], considering the cost of arranging sensors, monitoring of intermediate processes in WWTP has not been emphasized by practice, i.e., it is not guaranteed that the mass balance can be modeled in practical application scenarios [45]. To address this issue, Sun et al. [46] analyzed the potential relationships between components by calculating correlations. Further, Zou, et al. [47] obtained the correlation among influent contaminants through relevance and regression analysis. One of the biggest drawbacks of the traditional correlation-based feature analysis approach is that it detects only the linear relationships, the dynamic behavior of WWTP becomes an obstacle to the effort of generating a simple relationship between the input and output parameters and predicting the effluent [48].

When analyzing the relationships between dynamic components in activated sludge wastewater treatment processes, one should not focus only on the type of data relationship (linear), but should consider how much information can be provided by another characteristic when that characteristic has been given [49]. Therefore, we employ a data-driven methodology to analyze the mutual information (MI) between components, aiming to quantify the information shared between them. The MI model seeks to identify relationships between the influent components, allowing for the estimation of selected components based on measurements of others within the set. It's important to note that these potential relationships are determined not by the specific data series but by the underlying distribution of the data. By understanding these relationships, BEIN can gain insights into the interdependencies among parameters and improve the ability to estimate them accurately.

Let  $X$  be a component which takes on  $x_1, x_2, \dots, x_M$  values. The entropy  $H(X)$  of a discrete variable  $X$  is defined as:

$$H(X) = - \sum_{x \in X} p(x) \log p(x) \quad (9)$$

where  $p(x)$  indicate the probability of each sampled values in component  $X$ .

For a pair of discrete variables  $(X, Y)$  obeying a joint distribution of  $p(x, y)$ , the joint entropy  $H(X, Y)$  is defined as:

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \quad (10)$$

where  $p(x, y)$  is the joint probability.

Mutual information  $I(X, Y)$  between components  $X$  and  $Y$  is described as:

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \quad (11)$$

Overall equation from Eqs. (9) to (11) yields:

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \quad (12)$$

After computing the mutual information among influent components, it is impractical to directly considering all components with significant mutual information as exogenous variables. The Minimum Redundancy Maximum Relevance (MRMR) [50] is utilized to assess mutual information and determine whether a candidate exogenous component should be selected in the input.

The criterion  $J$  for selected components in monitored influent data is described as:

$$J_{MRMR}(X_n) = I(X_n, Y) - \frac{1}{|S|} \sum_{j \in S} I(X_n | X_j) \quad (13)$$

This equation represents the minimum redundancy from the remaining feature set ( $S$ ) while ensuring that the selected feature  $X_n$  has the maximum relevance with the objective feature  $Y$ .

Mutual information model capture the exogenous component most relevant to the variable to be imputed from  $z_{t-L,t+H}^{(\ell-1)}$  and decomposes the period state of the  $\ell$ th unfolding term of the current input signal, denoted  $m_{t-L,t+H}^{(\ell)} \in \mathbb{R}^{L+H}$ . Similarly to the data flow in identity block and the seasonality block,  $m_{t-L,t+H}^{(\ell)}$  consists of a backward prediction component  $m_{t-L,t}^{(\ell)}$  and a forward prediction component  $m_{t,t+H}^{(\ell)}$ . From exogenous block,  $m_{t-L,t+H}^{(\ell)}$  is utilized to analyze the impact of the relevant variables on the current network and employs  $m_{t-L,t}^{(\ell)}$  to denote additional information purely from the  $\ell$ th layer. When moving to the next layer,  $z_{t-L,t+H}^{(\ell)}$  removes  $m_{t-L,t+H}^{(\ell)}$  compared to the previous layer  $z_{t-L,t+H}^{(\ell-1)}$ , thus allowing subsequent blocks to focus on the unresolved residual term. With  $z_{t-L,t+H}^{(N)}$  and  $x_{t-L,t}^{(N)}$  superimposed, the following

double residual-based depth expansion is obtained, encapsulated in the left and middle parts of Fig. 3:

$$\begin{aligned} z_{t-L,t+H} &= z_{t-L,t+H}^{(0)} = \sum_{\ell=1}^N m_{t-L,t+H}^{(\ell)} + z_{t-L,t+H}^{(N)} \\ x_{t-L,t} &= x_{t-L,t}^{(0)} = \sum_{\ell=1}^N u_{t-L,t}^{(\ell)} + v_{t-L,t}^{(\ell)} + x_{t-L,t}^{(N)} \\ \hat{x}_{t,t+H} &= \hat{x}_{t,t+H}^{(N)} = \sum_{\ell=1}^N u_{t,t+H}^{(\ell)} + v_{t,t+H}^{(\ell)} \end{aligned} \quad (14)$$

### 3.4. Interpretability

Due to the specific design of basis expansion and exogenous block, our network structure has a degree of interpretability. Firstly, for LINER, as shown in Eq. (14), we decompose the signal input to the imputation module into two types: the historical acquisition  $x_{t-L,t}^{(\ell)}$  and the component relevance  $z_{t-L,t+H}^{(\ell)}$ . Note that  $m_{t-L,t+H}^{(\ell)}$  is conditional on  $z_{t-L,t+H}^{(\ell)}$  and independent of  $x_{t-L,t}^{(\ell)}$ , so that  $\sum_{\ell=1}^N m_{t-L,t+H}^{(\ell)}$  denotes the exogenous components from the multivariate sensor data. Also note that  $u_{t-L,t+H}^{(\ell)}$  and  $v_{t-L,t+H}^{(\ell)}$  depends on the sequence of missing data  $x_{t-L,t}^{(\ell)}$  and the sequence of exogenous data  $z_{t-L,t+H}^{(\ell)}$ . Thus,  $\sum_{\ell=1}^N v_{t-L,t}^{(\ell)} + u_{t-L,t}^{(\ell)}$  can be viewed as a prediction of the global observations that takes into account identity, seasonality, and the mutual information of multivariate sensor data. This makes it possible to distinguish the contribution to the final prediction into global and local momentum. Second,  $h_\phi(t)$ , as a seasonal feature module in our architecture, also has interpretable effects. Specifically, the coefficients in  $h_\phi(t)$  have practical significance, such as amplitude, frequency, and phase. We can interpret these coefficients as intrinsic properties of the sequence and relate them to real scenarios. In addition, by combining sub-signals with different periods,  $h_\phi(t)$  provides us with an efficient way to exclude high-frequency noise from missing data sequences and thus capture the sparse periodicity of the sequence.

### 3.5. Overall structure

BEIN is an early attempt to enhance sensor data from wastewater treatment processes, addressing the challenges posed by missing and inaccurate data resulting from equipment failures or network fluctuations. Considering that real-world time-series data often overlap with different period processes, we devise a dynamic time-series period decomposition scheme to capture the period characteristics within the sample space, decomposing the time-series into two components: identity and seasonality. BEIN analyzes the time series data spectrally through the FFT, so as to capture the period information, and this ability of spectrum analysis makes BEIN perform more efficiently when dealing with time series data with strong period. Further, BEIN utilizes the mutual information of wastewater components to construct an informative time streaming structure as the exogenous input for learning the distribution of real samples. Subsequently, BEIN conducts unidirectional and bidirectional imputation on real-time error data and historical missing data, respectively, to enhance industrial sensor data. This process constructs a relatively complete sample space, thereby improving the effectiveness of data-driven modeling. Compared to mainstream time series frameworks, BEIN is designed to be concise and versatile [51]. Each layer can be structured and parameterized differently, making BEIN flexible in dealing with different types of time series data. Secondly, the framework is built on the basis of residual learning, with exogenous variables as the auxiliary, and the learned identity residuals and seasonality residuals are expanded block by block. This means that as the data flows through multiple layers, the neural network retains the original information of the input, which helps alleviate issues such as gradient vanishing and explosion, thereby improving the training



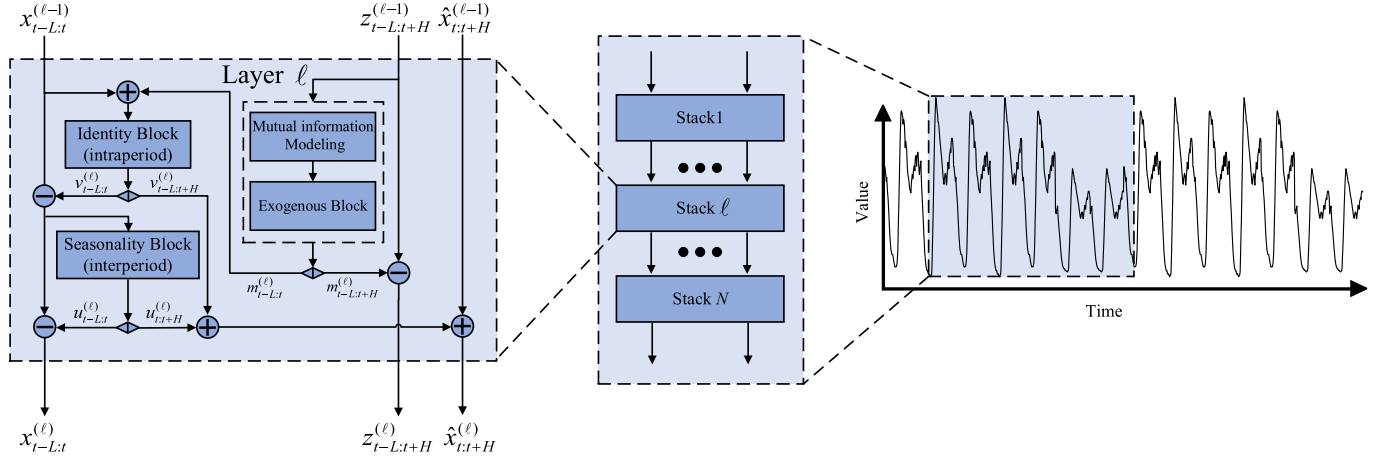


Fig. 3. Overall structure of basis expansion imputation network with mutual information module.

efficiency of the network. In short, the whole architecture design is straightforward and it can also be interpretable by outputs of different block.

#### 4. Experiments and discussions

##### 4.1. Basic experimental setup

In order to validate the performance of the proposed model, five state-of-the-art models were selected as baselines. The selected methods include period decomposition (M1, M2), tensor decomposition (M3), unsupervised learning (M4) and hybrid deep learning (M5). Table 1 provides an in-depth discussion of the five methods selected for the comparison experiments in this section. In addition, for models M1-M5,

**Table 1**  
Description of proposed model and comparison model.

| No | Model                    | Description                                                                                                                                                                                                                                                                                                 |
|----|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M1 | DEPTS [34]               | A decoupled form is employed to introduce period states as hidden variables, and two specialized modules are crafted to address these challenges, progressively extending complex dependencies layer by layer while capturing diverse period features.                                                      |
| M2 | NBEATS [35]              | The proposed method employs a multilayer fully connected network with forward and backward residual connections for time series forecasting. Its key components consist of “trendstack” and “seasonalstack”, and the forecasting process is facilitated through double residual stacking.                   |
| M3 | NT-DPTC [52]             | A novel method for dimension-preserving (DP) tensor decomposition is proposed to analyze the tensor structure of missing data. This approach involves decomposing the data into three low-rank potential factor tensors, addressing challenges associated with conventional low-rank approximation methods. |
| M4 | AM-DAE [17]              | Unsupervised learning is applied to iteratively interpolate missing samples by recursively using the median of the input and its reconstructed value through forward propagation. To strike a balance between reconstruction and imputation learning, an adaptive loss function is introduced.              |
| M5 | Transformer-Bi-LSTM [13] | A prediction method based on deep learning is introduced to address missing measurement data. This method integrates the Bi-LSTM layer into the Transformer-Bi-LSTM model within the Transformer framework.                                                                                                 |
| M6 | Proposed                 | The fusion of multivariate time series decomposition strategies with mutual information block tailored to industrial sensor data aims to tackle the challenges posed by intricate dependencies and varying period combinations.                                                                             |

we set the maximum number of iterations to 1000. During the experiment, the operation is terminated when one of the following conditions is met: 1) the number of iterations reaches the threshold, which is permanently 1000. 2) The difference between two consecutive rounds of the training error is less than  $10^{-5}$ . The purpose of this operation is to adequately train the model to achieve the most appropriate parameters. To validate the performance, we performed five cross-validations of the proposed model for arbitrarily selected replications with different initial values, and the maximum iteration was designed to be 500.

To emphasize the significance of BEIN, three multivariate datasets with different sample size were selected in experiment. The aim was to validate and compare the performance of BEIN with five mentioned baselines. Specifically, the datasets pertain to the WWTP influent under rainy, stormy and long-term conditions. The dataset encompasses 14 monitoring variables, including influent flow, with a sampling frequency of 15 min over a span of 61 days (360 days for long-term dataset). These datasets, publicly available and authoritative, underwent uniform treatment to intervals due to their high heterogeneity.

Experiments were carried out on a high-performance computer featuring an Intel i7-13700K CPU running at 5.4 GHz and an NVIDIA GeForce RTX 4060 Ti GPU with 16 GB of graphics memory, equipped with 32 GB DDR4 RAM and a 1 TB SSD. We utilized TensorFlow version 2.3.0 and PyTorch version 1.6.0 to leverage the latest capabilities of these deep learning frameworks while ensuring experiment reproducibility. Software components such as CUDA and cuDNN were installed to fully exploit the parallel computing capabilities of NVIDIA GPUs, thereby enhancing the training speed of deep learning models. The complete experimental datasets and code files are available upon request from the authors. We are committed to ensuring a transparent and verifiable experimental environment, facilitating easy replication and validation by other researchers.

In the data sampling process, we generated eight incomplete datasets with varying missing rates, based on industrial-type data missing patterns. As shown in Table 2, three types of missing patterns were

**Table 2**  
Generating incomplete data sets from BSM1 with different missing rates.

| Incomplete data set | 1/2 sampling rate | 1/3 sampling rate | Continuous missing |
|---------------------|-------------------|-------------------|--------------------|
| Data 1 (10 %)       | 1V (50 %)         | 1V (67 %)         | 1V (20 %)          |
| Data 2 (20 %)       | 3Vs (50 %)        | 1V (67 %)         | 2Vs (20 %)         |
| Data 3 (30 %)       | 4Vs (50 %)        | 2Vs (67 %)        | 3Vs (20 %)         |
| Data 4 (40 %)       | 6Vs (50 %)        | 2Vs (67 %)        | 4Vs (20 %)         |
| Data 5 (50 %)       | 1V (50 %)         | 6Vs (67 %)        | 5Vs (40 %)         |
| Data 6 (60 %)       | 1V (50 %)         | 6Vs (67 %)        | 6Vs (60 %)         |
| Data 7 (70 %)       | 1V (50 %)         | 8Vs (67 %)        | 4Vs (80 %)         |
| Data 8 (80 %)       | 0V (50 %)         | 7Vs (67 %)        | 6Vs (90 %)         |

employed: “1/2 or 1/3 sampling rate” indicates data sampled at twice or thrice the base interval, while “continuously missing” denotes data absent for a specific duration. Additionally, missing samples were introduced in randomly selected variables from the set of 13. For instance, “Data 1 (10%)” signifies that 10 % of the values in the original dataset are absent. “3Vs (50 %)” implies three variables are missing at a “1/2 sampling rate,” accounting for 50 % of the total data values. “1 V(67 %)” denotes one variable is absent at a “1/3 sampling rate,” constituting 67 % of the total value. Finally, “1 V(20 %)” indicates a single variable is missing continuously, amounting to 20 % of the overall data. The composition of these missing rates is illustrated by the first row in Table 2, which satisfies  $13 \times 10 \% \approx 1 \times 50 \% + 1 \times 67 \% + 1 \times 20 \%$ . For experiments with an 80 % missing rate, 80 % of the monitored data was randomly excluded. We allocated 20 % of the remaining data for training and validation and the remainder for testing.

#### 4.2. Quantitative analysis

In this study, we assume that the measurement noise adheres to a Gaussian distribution with a zero mean, and the noise variables are mutually independent. Subsequent to the proper training of the model, its predictive capability undergoes testing through the introduction of randomly omitted input samples. The robustness of the model’s predictions is assessed in the context of random missingness, encompassing both discrete and block missingness patterns. The model’s performance in achieving accurate imputation is tested across various proportions of incomplete data conditions, with the accuracy results validated against alternative prediction-based deep learning models. To comprehensively assess the model’s accuracy, evaluation metrics in the form of root mean square error (RMSE) and mean absolute percentage error (MAPE) are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t,v} M_{t,v} (\hat{X}_{t,v} - X_{\text{real},t,v})^2} \quad (15)$$

$$\text{MAPE} = \frac{1}{N} \sum_{t,v} M_{t,v} \left| \frac{\hat{X}_{t,v} - X_{\text{real},t,v}}{X_{\text{real},t,v}} \right| \times 100\% \quad (16)$$

Root mean square error (RMSE) serves as a measure to evaluate the disparity between real and imputed values, providing insight into the actual magnitude of deviation from the true value. On the other hand, Mean absolute percentage error (MAPE) complements the assessment by gauging the relative magnitude of deviation in percentage terms. A lower value for the performance metric is anticipated, signifying a more optimal model fit.

In the quantitative experiments, we compare the performance of different deep learning models on the problem of imputation of missing samples in multivariate time series data, with a special focus on the impact of the missing data rate on the prediction accuracy. The best metric for each comparison condition is shown in bold in each table. The results in Table 3 show that BEIN outperforms the other models for most missing data scenarios when imputing the BSM2 influent data. Specifically, even at 80 % missing data rate, BEIN still has the lowest RMSE and MAPE values, indicating its high robustness to data incompleteness.

Across the full range of data missing rates, the performance of all models increases with data completeness. This result is consistent with the trend observed in previous studies that data integrity is one of the key factors affecting the performance of deep learning models. However, BEIN exhibits the most significant performance enhancement, e.g., at a missing rate of 10 %, the RMSE of BEIN is 409.78, whereas when the missing rate is increased to 80 %, its RMSE only increases to 415.19, a change that is much smaller than that of the other models. As shown in Table 4, BEIN generally exhibits lower RMSE and MAPE values at a data missing rate of 80 %, 40 % and 20 %, outperforming the other models by up to about 5.33 % and 1.68 % of the RMSE and MAPE values. In other conditions, such as 70 %, 50 %, 30 % missing rate, BEIN fails to obtain the best RMSE but still achieves the best MAPE. This result points to the possibility that BEIN is more robust in handling time series data with high missing rate. The performances of DEPTS and NBEATS are quite close to each other, suggesting that there is some structural or functional similarity between the two models, leading to a synchronisation in dealing with missing data. NT-DPTC show ideal performance at lower missing rates, but degrades faster than the other models as the missing rate increases, implying their limitations under extreme conditions. Transformer-Bi-LSTM performs more consistently at all missing rate levels, and although its performance decreases when missing rates up to 80 %, its performance is still comparable to that of the BEIN at moderate missing rates. Similarly, as shown in Table 5, BEIN still maintains the lowest RMSE and MAPE in most cases, emphasizing its superior robustness under extreme conditions. Overall, BEIN performs well in most missing rate conditions, which is attributed to its built-in missing data processing mechanism with interpretability.

#### 4.3. Qualitative analysis

To demonstrate the advantages of BEIN in terms of computational efficiency and accuracy, Figs. 4–6 show the results of the training iterations of RMSE and MAPE for the six models on “Data(20 %)”. In order to represent the experimental results more intuitively and succinctly, the vertical coordinates are normalized. Each point in the figure represents the test results after the training for each dataset. It can be seen that M6 (BEIN) has faster learning efficiency and better computational performance.

Fig. 4 shows the experiment results of six deep learning based imputation models (labelled M1 to M6) on the “Data (20 %)” rain influent dataset. As can be seen from Fig. 4(a), the RMSE of all models decreases rapidly in the initial phase (about 0–50 Epochs), after which the rate of decrease slows down, showing a state of stabilization. The performances of M1, M2, M3, and M6 are similar, but the M1’s performance in the late stage performance decreases slightly, possibly because the model begins to overfit. Fig. 4(b) shows that M6 performs the best in reducing the MAPE, followed by the M3, which is consistent with the results of the RMSE, and further confirms the effectiveness of these two models. Fig. 5 shows the experiment results on the “Data (20 %)” storm influent dataset. Specifically, M6 has the fastest decreasing RMSE and MAPE at the beginning of iterations, showing a strong initial learning ability. However, in subsequent iterations, the performance improvement of all models slow down, suggesting that the model may have

**Table 3**  
Lowest RMSE/MAPE of Models with different missing rates on BSM2 simulated influent.

| MR   | DEPTS            | NT-DPTC          | NBEATS           | AM-DAE           | Transformer-Bi-LSTM     | BEIN                    |
|------|------------------|------------------|------------------|------------------|-------------------------|-------------------------|
| 80 % | 415.292/0.103616 | 449.463/0.109033 | 414.016/0.097770 | 514.013/0.093473 | 415.263/0.102023        | <b>415.191/0.093141</b> |
| 70 % | 414.205/0.098720 | 449.157/0.104025 | 413.903/0.097353 | 513.945/0.093054 | 414.452/0.097624        | <b>414.352/0.092618</b> |
| 60 % | 413.413/0.093149 | 433.285/0.101123 | 413.832/0.096554 | 513.807/0.092552 | 413.712/0.093553        | <b>413.035/0.092335</b> |
| 50 % | 412.795/0.092283 | 432.537/0.099023 | 411.784/0.095706 | 513.818/0.092232 | 412.339/0.092628        | <b>411.745/0.091825</b> |
| 40 % | 412.271/0.090597 | 432.464/0.097628 | 411.771/0.093182 | 513.934/0.090478 | 412.023/0.092420        | <b>411.426/0.085856</b> |
| 30 % | 411.703/0.089529 | 423.266/0.096473 | 410.738/0.090778 | 513.876/0.090128 | <b>409.571/0.092127</b> | <b>409.833/0.085345</b> |
| 20 % | 411.233/0.086956 | 428.619/0.095686 | 410.705/0.089452 | 513.843/0.089287 | 409.233/0.088686        | <b>409.211/0.081235</b> |
| 10 % | 410.734/0.084723 | 419.463/0.094904 | 410.232/0.089523 | 513.828/0.089253 | <b>409.193/0.085328</b> | <b>409.787/0.077713</b> |

**Table 4**

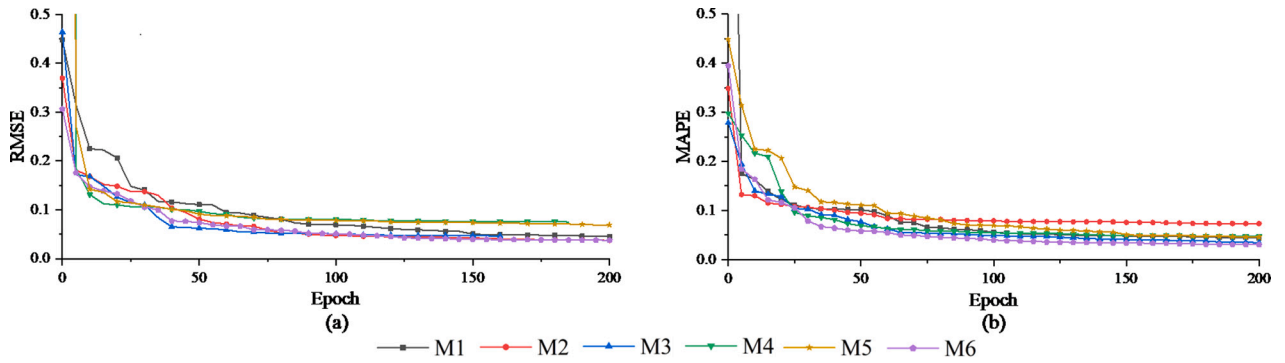
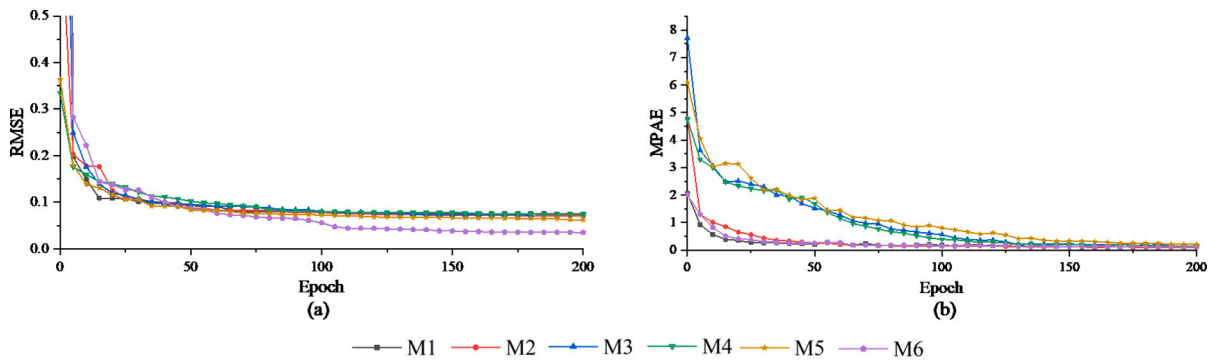
Lowest RMSE/MAPE of Models with different missing rates on Rain weather influent.

| MR   | DEPTS            | NT-DPTC          | NBEATS                  | AM-DAE           | Transformer-Bi-LSTM     | BEIN                    |
|------|------------------|------------------|-------------------------|------------------|-------------------------|-------------------------|
| 80 % | 1074.78/0.106736 | 1085.29/0.147745 | 1091.97/0.099092        | 1094.19/0.109231 | 1085.62/0.106745        | <b>1037.15/0.097359</b> |
| 70 % | 855.925/0.103233 | 852.048/0.119741 | <b>845.563/0.098087</b> | 872.473/0.103007 | 886.809/0.104046        | 851.397/0.095318        |
| 60 % | 809.127/0.101602 | 815.581/0.113056 | <b>805.619/0.095479</b> | 843.579/0.102375 | 835.581/0.105537        | 832.316/0.095964        |
| 50 % | 773.235/0.097289 | 805.619/0.108332 | 794.127/0.095371        | 760.077/0.101359 | <b>758.096/0.096769</b> | 758.584/0.092946        |
| 40 % | 756.713/0.095636 | 742.473/0.106698 | 718.096/0.094562        | 703.727/0.095124 | 746.763/0.097754        | <b>691.504/0.090523</b> |
| 30 % | 625.762/0.091546 | 628.735/0.909895 | 631.156/0.094616        | 624.736/0.094514 | <b>622.094/0.095114</b> | 623.083/0.089579        |
| 20 % | 553.678/0.091537 | 542.338/0.093691 | 539.062/0.093275        | 531.821/0.093372 | 551.781/0.093397        | <b>531.768/0.089436</b> |
| 10 % | 527.824/0.088046 | 530.053/0.085447 | <b>513.284/0.090218</b> | 558.127/0.090176 | 542.751/0.093125        | 517.778/0.087184        |

**Table 5**

Lowest RMSE/MAPE of Models with different missing rates on Storm weather influent.

| MR   | DEPTS            | NT-DPTC          | NBEATS                  | AM-DAE           | Transformer-Bi-LSTM | BEIN                    |
|------|------------------|------------------|-------------------------|------------------|---------------------|-------------------------|
| 80 % | 2978.31/0.501377 | 3006.18/0.494886 | <b>2975.45/0.495491</b> | 2996.23/0.498987 | 2983.28/0.499093    | 2977.58/0.491922        |
| 70 % | 2760.22/0.422833 | 2744.82/0.439661 | 2749.41/0.441966        | 2790.58/0.435208 | 2770.44/0.460646    | 2740.11/0.435275        |
| 60 % | 2631.25/0.401881 | 2666.63/0.41696  | 2629.32/0.419279        | 2637.51/0.415707 | 2656.61/0.425332    | 2612.49/0.415526        |
| 50 % | 2510.53/0.417404 | 2517.89/0.432727 | 2504.87/0.432565        | 2518.55/0.434573 | 2445.85/0.438405    | 2416.42/0.432565        |
| 40 % | 2499.6/0.438749  | 2490.66/0.452767 | 2488.63/0.457969        | 2492.53/0.456176 | 2415.43/0.457943    | 2405.58/0.446437        |
| 30 % | 2550.59/0.456039 | 2546.76/0.464154 | 2547.72/0.473522        | 2574.77/0.470168 | 2431.04/0.456527    | 2453.43/0.457548        |
| 20 % | 2263.76/0.376473 | 2270.55/0.375926 | 2264.82/0.382096        | 2303.58/0.368733 | 2205.57/0.362463    | 2271.74/0.369537        |
| 10 % | 2133.41/0.463319 | 2179.18/0.463872 | 2159.25/0.470369        | 2171.02/0.445916 | 2159.31/0.439577    | <b>2128.42/0.448345</b> |

**Fig. 4.** Comparison of RMSE and MAPE of six methods on “Data (20 %)” under the rain influent.**Fig. 5.** Comparison of RMSE and MAPE of six methods on “Data (20 %)” under the storm influent.

encountered a learning bottleneck earlier. In contrast, M6 demonstrated a more stable performance improvement throughout the iterations, especially in the late iterations (150–200 Epochs). Fig. 6 shows the experiment results on the “Data (20 %)” BSM2 influent dataset. M1 exhibits a high initial error in the RMSE, but it rapidly decreases and levels off after about 50 epochs. Baselines M2, M3, and M4 have lower starting errors and decline more rapidly in the early stages of training. M5 and M6 show similar performance trends, but M6’s error decreases at a faster rate and has a better steady-state value. In Fig. 6(b), it can be

seen that M6 has the most significant performance improvement, followed by M5, while M1 has a relatively slow performance improvement. In summary, the experimental results demonstrate that although all methods progressively diminish their prediction errors in the imputation task, BEIN yields the most optimal performance overall. This superiority can be ascribed to BEIN’s adoption of architectures better suited for handling multivariate time series, thereby enhancing its capability to capture temporal features effectively.



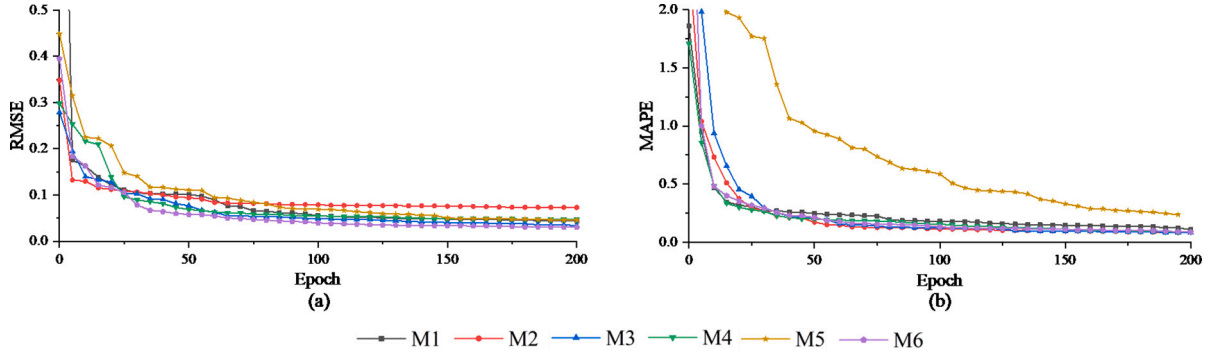


Fig. 6. Comparison of RMSE and MAPE of six methods on “Data (20 %)” under the BSM2 influent.

#### 4.4. Interpretability analysis

In this section, we illustrate the interpretable effects of BEIN through two scenarios. The left graph in Figs. 7 and 8 shows the imputation result of WWTP influent flow rate under rain and storm weather, respectively. Further, the error plots in the right graph provide an additional dimension to the assessment of model performance. The magnitude and distribution of errors reflect key indicators of model stability, while spikes in errors may indicate the sensitivity of the model to specific types of missing data or outliers. The volatility of the models’ errors also reflecting the responsiveness of different models to different dynamics in the data.

Figs. 9 and 10 visualize the internal state of BEIN. In the main graph, the identity signal shows the overall volatility trend of the data. To facilitate analysis, a detailed zoom-in window is set up to analyze the properties of these sub-signals in further detail. The identity component typically represents the baseline portion of the time series, which reveals the underlying trend of the data in the absence of any seasonal or exogenous influences. The seasonality component captures the periodic fluctuations in the time series, which is critical to understanding how to control and adjust for seasonal influences in the model. As can be seen from the subgraph in the lower right corner, the seasonality signal shows a clear pattern of periodic fluctuations, which is of high importance for responding to environmental changes in a timely manner in application scenarios where seasonality is strong. The exogenous indicates the effect of external variables on the time series. It is worth noting that the signal of this block does not directly affect the final model output, but plays an auxiliary role to the identity block. It can be observed that the exogenous factor exhibits a comparable trend to identity, providing an alternative perspective on the rationality of the module and its impact on identity’s performance.

#### 4.5. Ablation experiment analysis

As shown in Fig. 11, we employ three ablation variants of BEIN to demonstrate our key design in the extended module (Section 4.2):

BEIN-1: removes the residual connection of decomposition block so that the contribution to the prediction comes only from the local

block, which progressively receives signals that exclude periodic effects.

BEIN-2: removes the residual connections of exogenous block so that the input to the period block at each level is the same hidden variable  $Z_{t-L:t+H}$ .

BEIN-3: removes the connection of the exogenous block to the decomposition block, so that the local output is only in the original temporary signal, corresponding to a mix of local momentum and global periodicity observations.

Figs. 12–14 illustrate the ablation results of four different deep learning models (BEIN-1, BEIN-2, BEIN-3, and a combined BEIN model) on dealing with the imputation problem in multivariate WWTP time series data. The graphs show how the methods change in two performance metrics, root mean square error (RMSE) and mean absolute percentage error (MAPE), as the missingness rate increases from 10 % to 80 %. In Fig. 12, BEIN exhibits relatively low RMSE, especially under low missing rate conditions, which suggests that the combination of time series decomposition and mutual information module employs a competitive effect. Comparatively, BEIN-1, BEIN-2 and BEIN-3 do not differ much at low missing rates, but the performance of BEIN-3 decays slightly faster than that of BEIN-1 when the missing rate increases. The same trend is illustrated in the MAPE graphs on the right hand side, where the MAPE of all the models increases as the missing rate rises. Notably, the composite model BEIN exhibits superior performance under most missing rate conditions, especially at moderate to high missing rates (40 %–80 %). In Fig. 13, it can be found that BEIN exhibits a smaller increase in both RMSE and MAPE at high missing rates conditions, which implies that BEIN have better robustness when dealing with complex dynamic scenarios. The increase in error accelerates when the missing rate reaches between 60 % and 80 %, which may be due to the fact that the continuity of the data is severely affected, making the imputation process more difficult. Fig. 14 reveals an interesting phenomenon: while BEIN continues to be dominant at low to moderate missing rates, the performance of BEIN-1 and BEIN-2 improves at high missing rates, especially in the MAPE metric. The performance of different missing rates shows that the methods do not differ much at high missing rates, as the missing rate increases, the stability advantage of BEIN starts to diminish. The characterization of the decay of the performance of imputation in the BSM2 influent dataset suggests that,

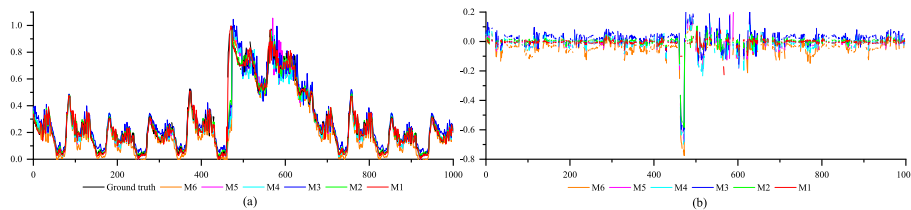


Fig. 7. Comparison of Imputation curve on “Data (50 %)” of the Rain weather influent.

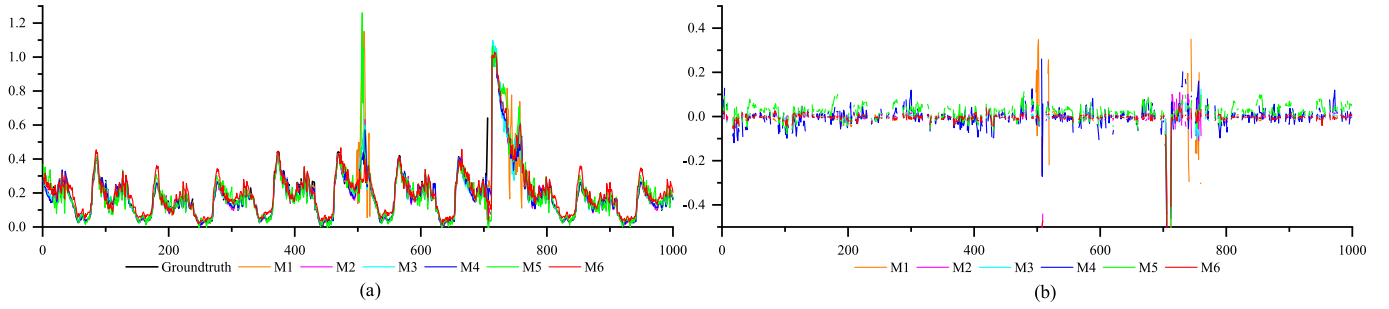


Fig. 8. Comparison of Imputation curve on "Data (50 %)" of the Storm weather influent.

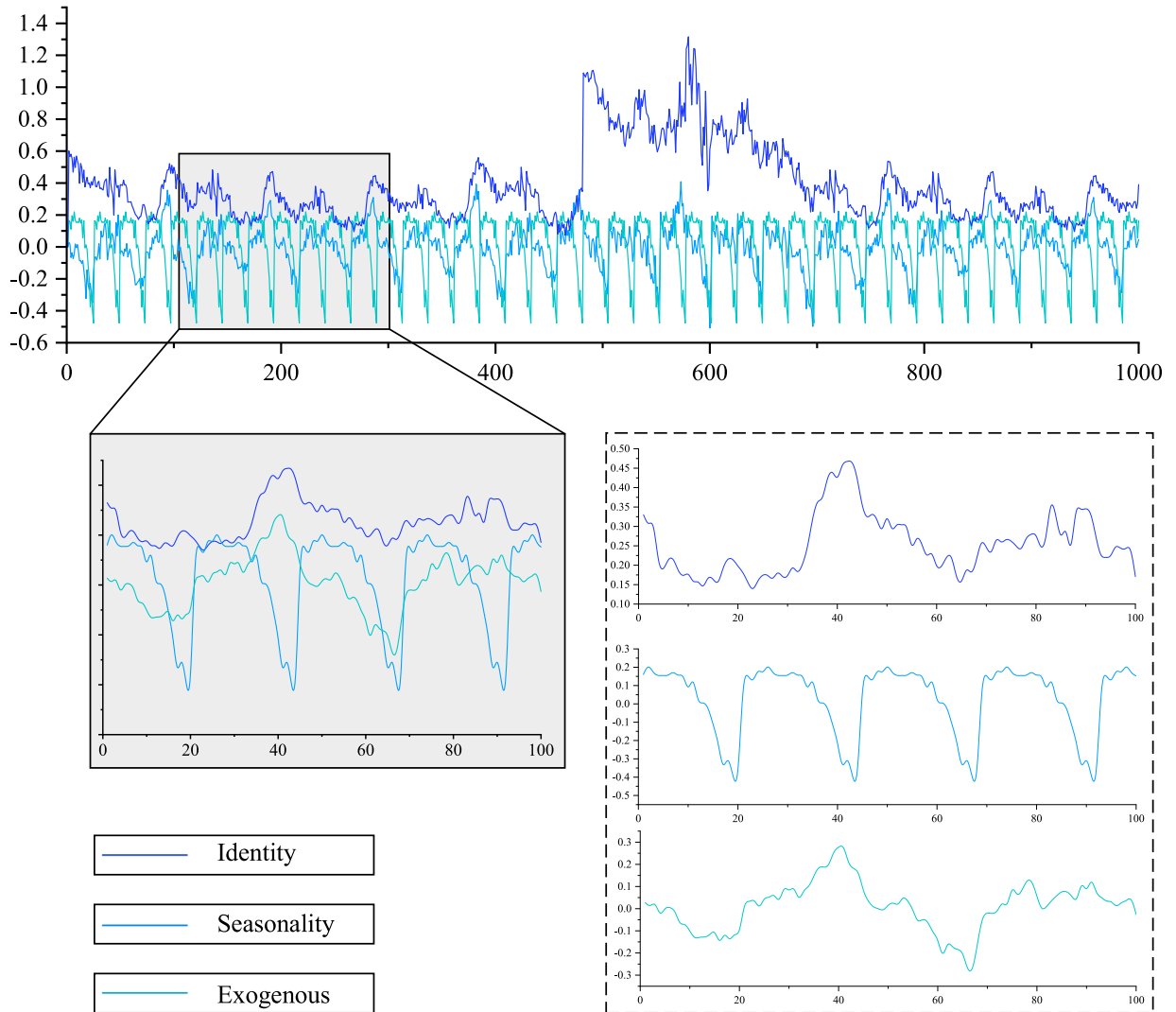


Fig. 9. Interpretable decomposition for Rain weather influent imputation.

for long-term data, as the severity of the missing data increases, the differences between different models gradually decrease.

Combining the analysis of Figs. 12 to 14, the integrated BEIN demonstrates high accuracy and stability at different missing rates, especially in the range of low to moderate missing rates. As the missing rate increases, imputation network with components relevance shows better adaptability and robustness. This suggests that the design of BEIN can synthesize the advantages of multiple models more effectively, improving the adaptability and imputation effect on missing data in complex dynamic wastewater treatment processes.

## 5. Conclusions

In this paper, a basis expansion imputation network (BEIN) is proposed for handling the missingness of sensor data in wastewater treatment system. Considering that real-world time series data often exhibit multiple overlap of periods, BEIN employs a dynamic time series period-decomposing scheme with basis function expansion to estimate the distribution of multiple periods of a time series. In addition, since sensor time series data in industrial scenarios are often correlated, BEIN utilizes mutual information to construct a time stream structure with the

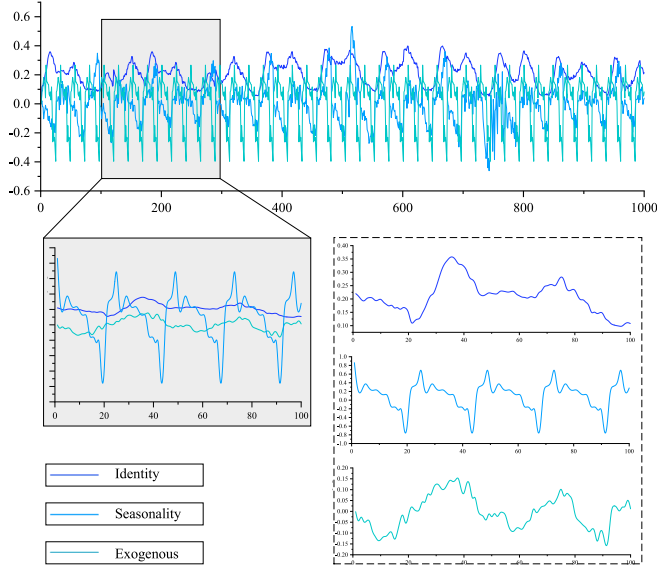


Fig. 10. Interpretable decomposition for Storm weather influent imputation.

relevant wastewater components as exogenous variables for learning the distribution of ground truth samples. MI module is devised to elucidate a precise approach for configuring the proposed architecture, thereby enhancing interpretability of its outputs. In short, the innovation of this paper lies in its analysis of multivariate time series in wastewater treatment processes and its application strategy of deep learning models,

which provides new perspectives and solutions for time series data imputation in industrial scenarios. Furthermore, the imputation framework consists of multiple independent modules that collaborate seamlessly, suggesting the robustness and adaptability of our proposed framework could maintained with ongoing developments in deep learning techniques within the field of time series forecasting.

BEIN outperforms the baseline in imputation accuracy. BEIN demonstrates superior imputation results compared to the baseline in three different conditions. Under the “rain” condition, BEIN achieves the best RMSE performance in three missing rate scenarios (20 %, 40 %, and 80 %), and the best MAPE performance in seven missing rate scenarios (10 %, 20 %, 30 %, 40 %, 50 %, 70 %, and 80 %). In the imputation experiments using storm and BSM2 influent data, BEIN achieves the best performance in RMSE in 5/8 scenarios, and in MAPE it achieves the best performance in 5/8 scenarios and 8/8 scenarios, respectively. Experimental results confirm the following conclusions: wastewater treatment data exhibit a plethora of spatial and temporal patterns, it is imperative to undertake efforts to extract and integrate these patterns utilizing diverse components in a reasonable manner. Multi-period decomposition of time-series data enhances models’ capacity to discern insights progressively from coarse-grained to fine-grained levels, which enables effective learning of temporal features within sequences.

Overall, BEIN provides an in-depth insight into the processing of missing samples in multivariate time series data, emphasizing the potential benefits of deep learning models. Future research could concentrate on investigating the adaptability of the models in managing high-frequency and long-term data, intricate seasonal patterns, and extreme values to enhance their applicability and stability across diverse industrial scenarios. In addition, improvements in the model’s ability to

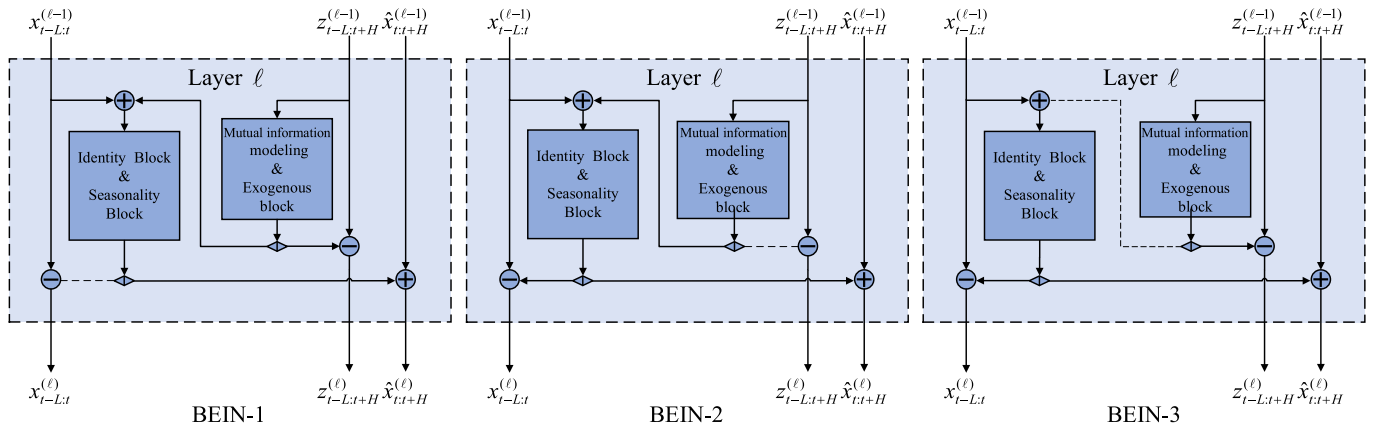


Fig. 11. The residual structures of ablated variants, where the dashed line denotes the removed connection.

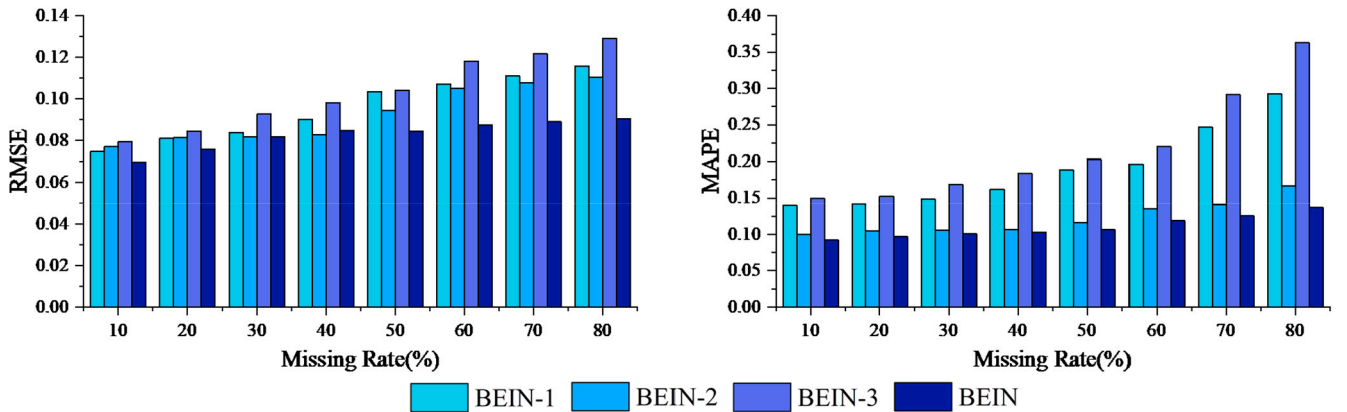


Fig. 12. Comparison of RMSE and MAPE in BEIN-1, BEIN-2, BEIN-3 and BEIN in Rain weather.



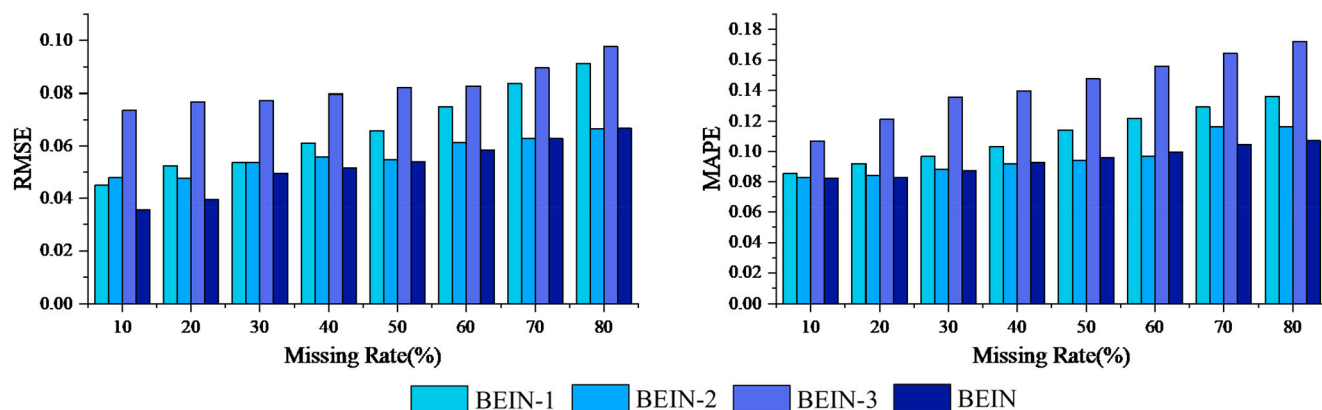


Fig. 13. Comparison of RMSE and MAPE in BEIN-1, BEIN-2, BEIN-3 and BEIN in Storm weather.

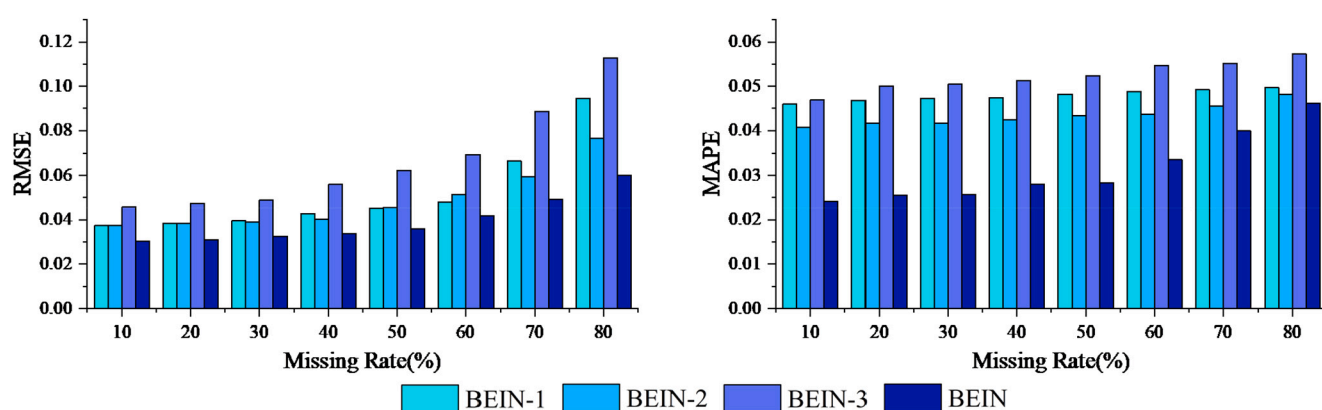


Fig. 14. Comparison of RMSE and MAPE in BEIN-1, BEIN-2, BEIN-3 and BEIN in BSM2.

recognize and handle outliers will help to enhance the robustness of the model and improve its ability to recognize and handle inherent characteristics, cyclical fluctuations, and external influences in the data. With these improvements, we can expect the BEIN to show superior performance in more complex real-world industrial scenarios.

#### Ethics approval

This article does not contain any studies with human participants or animals performed by any of the authors.

#### Funding

This work was supported by the National Natural Science Foundation of China (No. 61763026, 62063019), and Natural Science Foundation of Gansu Province (No. 20JR10RA152).

#### CRediT authorship contribution statement

**Yu Peng:** Writing – original draft, Visualization, Methodology, Conceptualization. **Erchao Li:** Writing – review & editing, Supervision, Conceptualization.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability statement

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

#### References

- [1] P.R. Rout, M.K. Shahid, R.R. Dash, et al., Nutrient removal from domestic wastewater: a comprehensive review on conventional and advanced technologies, *J. Environ. Manag.* 296 (2021) 113246.
- [2] M. Bahramian, R.K. Dereli, W. Zhao, et al., Data to intelligence: the role of data-driven models in wastewater treatment, *Expert Syst. Appl.* 217 (2023) 119453.
- [3] P. Sakiewicz, K. Piotrowski, J. Ober, et al., Innovative artificial neural network approach for integrated biogas-wastewater treatment system modelling: effect of plant operating parameters on process intensification, *Renew. Sust. Energ. Rev.* 124 (2020) 109784.
- [4] T. Yang, X. Yi, S. Lu, et al., Intelligent manufacturing for the process industry driven by industrial artificial intelligence, *Engineering* 7 (9) (2021) 1224–1230.
- [5] C. Li, Y. Chen, Y. Shang, A review of industrial big data for decision making in intelligent manufacturing, *Engineering Science and Technology, an International Journal* 29 (2022) 101021.
- [6] G. Sin, R. Al, Activated sludge models at the crossroad of artificial intelligence—a perspective on advancing process modeling, *Npj Clean Water* 4 (1) (2021) 16.
- [7] A. Dogan, D. Birant, Machine learning and data mining in manufacturing, *Expert Syst. Appl.* 166 (2021) 114060.
- [8] I.R. White, P. Royston, A.M. Wood, Multiple imputation using chained equations: issues and guidance for practice, *Stat. Med.* 30 (4) (2011) 377–399.
- [9] P.J. García-Laencina, J.L. Sancho-Gómez, A.R. Figueiras-Vidal, Pattern classification with missing data: a review, *Neural Comput. & Applic.* 19 (2010) 263–282.
- [10] K. Liu, Y. Tang, W. Lou, Y. Liu, J. Yang, Y. Yao, A thermographic data augmentation and signal separation method for defect detection, *Meas. Sci. Technol.* 32 (4) (2021) 045401.
- [11] J. Liu, G. Liu, H. Zhao, J. Zhao, J. Qiu, Z.Y. Dong, Real-time industrial carbon emission estimation with deep learning-based device recognition and incomplete smart meter data, *Eng. Appl. Artif. Intell.* 127 (2024) 107272.

- [12] G. Tutz, S. Ramzan, Improved methods for the imputation of missing data by nearest neighbor methods, *Computational Statistics & Data Analysis* 90 (2015) 84–99.
- [13] Y. Raghuvarshi, K. Teeparthi, Distribution system state estimation with Transformer-Bi-LSTM-based imputation model for missing measurements, *Neural Comput. & Applic.* (2023) 1–18.
- [14] S. Ma, Z.S. Xu, T. Sun, Parallel generative adversarial imputation network for multivariate missing time-series reconstruction and its application to aero-engines, *IEEE Trans. Instrum. Meas.* 72 (2023) 1–16.
- [15] W. Wang, Y. Chai, Y. Li, GAGIN: generative adversarial guider imputation network for missing data, *Neural Comput. & Applic.* 34 (10) (2022) 7597–7610.
- [16] C. Velasco-Gallego, I. Lazakis, Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study, *Ocean Eng.* 218 (2020) 108261.
- [17] Z. Pan, Y. Wang, K. Wang, H. Chen, C. Yang, W. Gui, Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder, *IEEE Transactions on Cybernetics* 53 (2) (2022) 695–706.
- [18] K. Zhong, X. Sun, G. Liu, et al., Attention-based generative adversarial networks for aquaponics environment time series data imputation, *Information Processing in Agriculture* (2023) (2023), <https://doi.org/10.1016/j.inpa.2023.10.001> this Ref is available online 21 October 2023.
- [19] M. Beenstock, D. Felsenstein, M. Beenstock, et al., Spatial vector autoregressions, *The Econometric Analysis of Non-stationary Spatial Panel Data* (2019) 129–161.
- [20] P. Lara-Benítez, M. Carranza-García, J.C. Riquelme, An experimental review on deep learning architectures for time series forecasting, *Int. J. Neural Syst.* 31 (03) (2021) 2130001.
- [21] A. VKhademalrasoul, H. Amerikha, Forecasting of short-term and mid-term variations of rainfall erosivity index using SARIMA, *Agric. Eng.* 45 (1) (2022) 79–95.
- [22] D. Cao, Y. Wang, J. Duan, et al., Spectral temporal graph neural network for multivariate time-series forecasting, *Adv. Neural Inf. Proces. Syst.* 33 (2020) 17766–17778.
- [23] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, *Adv. Neural Inf. Proces. Syst.* 33 (2020) 13016–13026.
- [24] J.Y. Franceschi, A. Dieuleveut, M. Jaggi, Unsupervised scalable representation learning for multivariate time series, *Adv. Neural Inf. Proces. Syst.* 32 (2019).
- [25] Y.A. Farha, J. Gall, Ms-tcn: multi-stage temporal convolutional network for action segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3575–3584.
- [26] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, et al., Informer: beyond efficient transformer for long sequence time-series forecasting, *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (12) (2021) 11106–11115.
- [27] S. Ahmed, I.E. Nielsen, A. Tripathi, S. Siddiqui, R.P. Ramachandran, G. Rasool, Transformers in time-series analysis: a tutorial, *Circuits, Systems, and Signal Processing* 42 (12) (2023) 7433–7466.
- [28] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: decomposition transformers with auto-correlation for long-term series forecasting, *Adv. Neural Inf. Proces. Syst.* 34 (2021) 22419–22430.
- [29] Y. Guo, S. Zhang, J. Yang, et al., Dual memory scale network for multi-step time series forecasting in thermal environment of aquaculture facility: a case study of recirculating aquaculture water temperature, *Expert Syst. Appl.* 208 (2022) 118218.
- [30] Z. Hajirahimi, M. Khashei, Hybrid structures in time series modeling and forecasting: a review, *Eng. Appl. Artif. Intell.* 86 (2019) 83–106.
- [31] J. Yang, Y. Guo, T. Chen, et al., Data-driven prediction of greenhouse aquaponics air temperature based on adaptive time pattern network, *Environ. Sci. Pollut. Res.* 30 (16) (2023) 48546–48558.
- [32] L. Corominas, M. Garrido-Baserba, K. Villeg, et al., Transforming data into knowledge for improved wastewater treatment operation: a critical review of techniques, *Environ. Model Softw.* 106 (2018) 89–103.
- [33] L.M. Jiang, W. Zhang, Y. Li, et al., Applying mass flow analysis and aeration optimization strategy to reduce energy consumption of a full-scale anaerobic/anoxic/oxic system, *Journal of Water Process Engineering* 54 (2023) 104037.
- [34] W. Fan, S. Zheng, X. Yi, W. Cao, Y. Fu, J. Bian, et al., DEPTS: deep expansion learning for periodic time series forecasting, in: *International Conference on Learning Representations*, 2022.
- [35] B.N. Oreshkin, G. Dudek, P. Pelka, E. Turkina, N-BEATS neural network for mid-term electricity load forecasting, *Appl. Energy* 293 (2021) 116918.
- [36] Q. Wen, K. He, L. Sun, Y. Zhang, M. Ke, H. Xu, RobustPeriod: robust time-frequency mining for multiple periodicity detection, in: *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2328–2337.
- [37] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: frequency enhanced decomposed transformer for long-term series forecasting, in: *International Conference on Machine Learning*, PMLR, 2022, pp. 27268–27286.
- [38] H.A.O. Alisawi, Performance of wastewater treatment during variable temperature, *Appl. Water Sci.* 10 (4) (2020) 89.
- [39] G. Liu, Y. Jiang, K. Zhong, et al., A time series model adapted to multiple environments for recirculating aquaculture systems, *Aquaculture* 567 (2023) 739284.
- [40] Q. Wen, Z. Zhang, Y. Li, L. Sun, Fast RobustSTL: efficient and robust seasonal-trend decomposition for time series with complex patterns, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2203–2213.
- [41] L. Ge, K. Wu, Y. Zeng, et al., Multi-scale spatiotemporal graph convolution network for air quality prediction, *Appl. Intell.* 51 (2021) 3491–3505.
- [42] Alex J, Benedetti L, Copp J B, Gernaey K V, Jeppsson U, Nopens I, et al. Benchmark simulation model no. 1 (BSM1)[J]. Report by the IWA Taskgroup on benchmarking of control strategies for WWTPs, 2008, 1.
- [43] T. Siwiec, L. Reczek, M.M. Michel, B. Gut, P. Hawer-Strojek, J. Czajkowska, et al., Correlations between organic pollution indicators in municipal wastewater, *Archives of Environmental Protection* 44 (4) (2018).
- [44] S.Z. Zhang, S. Chen, H. Jiang, A back propagation neural network model for accurately predicting the removal efficiency of ammonia nitrogen in wastewater treatment plants using different biological processes, *Water Res.* 222 (2022) 118908.
- [45] M.S. Zaghoul, G. Achari, Application of machine learning techniques to model a full-scale wastewater treatment plant with biological nutrient removal, *J. Environ. Chem. Eng.* 10 (3) (2022) 107430.
- [46] Y. Sun, Z. Chen, G. Wu, Q. Wu, F. Zhang, Z. Niu, et al., Characteristics of water quality of municipal wastewater treatment plants in China: implications for resources utilization and management, *J. Clean. Prod.* 131 (2016) 1–9.
- [47] L. Zou, H. Li, S. Wang, K. Zheng, Y. Wang, G. Du, et al., Characteristic and correlation analysis of influent and energy consumption of wastewater treatment plants in Taihu Basin, *Front. Environ. Sci. Eng.* 13 (2019) 1–14.
- [48] T. Cheng, F. Harrou, F. Kadri, et al., Forecasting of wastewater treatment plant key features using deep learning-based models: a case study, *IEEE Access* 8 (2020) 184475–184485.
- [49] E. Sharghi, V. Nourani, A.A. Ashrafi, H. Gökçekub, Monitoring effluent quality of wastewater treatment plant by clustering based artificial neural network method, *Desalin. Water Treat.* 164 (2019) 86–97.
- [50] H. Zhou, X. Wang, R. Zhu, Feature selection based on mutual information with correlation coefficient, *Appl. Intell.* 52 (5) (2022) 5457–5474.
- [51] B. Rogers, N. Noman, S. Chalup, et al., A comparative analysis of deep neural network architectures for sentence classification using genetic algorithm, *Evol. Intel.* (2023) 1–20.
- [52] H. Chen, M. Lin, J. Liu, H. Yang, C. Zhang, Z. Xu, NT-DPTC: a non-negative temporal dimension preserved tensor completion model for missing traffic data imputation, *Inf. Sci.* 653 (2024) 119797.



# A generative model-based coevolutionary training framework for noise-tolerant softsensors in wastewater treatment processes

Yu Peng<sup>1</sup> · Erchao Li<sup>1</sup>

Received: 3 December 2024 / Accepted: 19 February 2025  
© The Author(s) 2025

## Abstract

Data-driven softsensors have gained widespread application in process monitoring and quality prediction, offering advantages over traditional measurement techniques by mitigating their limitations and costs. However, the effectiveness of softsensor models is often hindered by noise in data acquisition, posing significant challenges for model training. To tackle this issue, this study introduces a coevolutionary training framework based on generative models to mitigate the impact of noise corruption. The framework employs a denoising variational autoencoder to extract global and local features from auxiliary data, enhancing population distribution and constructing a deep nonlinear representation to counter noise effects. Additionally, a dual population coding method inspired by evolutionary computation is proposed, enabling the coevolution of network parameters and structure. The proposed multiobjective evolutionary network optimization with denoising strategy (MENO-D) demonstrated exceptional performance in various experiments. On a water quality prediction dataset, the MENO-D-trained softsensor model achieved the lowest prediction error under 10% and 20% noise interference. Further, on the WWTP benchmark dataset across three weather conditions, MENO-D-trained softsensor model exhibited competitive accuracy and robustness.

**Keywords** Coevolution · Deep learning · Generative model · Softsensor · Wastewater treatment process

## Introduction

In industrial settings, online collection of process data is essential for enhancing operational efficiency and enabling automated production. For instance, in the oxidation of wastewater treatment process (WWTP), chemical oxygen demand (COD) and five-day biochemical oxygen demand (BOD5) are critical indicators for assessing water pollution levels. Traditionally, these variables are measured using online analyzers or offline laboratory testing, both of which involve large time delays, leading to unavoidable economic costs and potential losses [1]. The utilization of industrial big data for controlling and managing modern complex production systems has emerged as a high-priority research focus [2, 3], with soft sensor techniques standing out among existing data-driven approaches for their effectiveness in estimating unmeasurable key variables. A fast and accurate softsensor model not only provides a clear depiction of dynamic

mechanisms but also offers advanced alerts for abnormal situations, facilitating timely operational adjustments [4]. Furthermore, it enables optimized control and management services, improving overall system performance and supporting the development of effective methodologies for assessing water environment safety [5, 6].

Softsensor methods can be broadly categorized into physics-based models [7] and data-driven models [8, 9]. Physics-based softsensor models rely on analyzing the physical mechanisms of production processes to construct prediction models. While these methods offer higher interpretability, their complexity often reduces accessibility, making it challenging to develop accurate mathematical representations for real-world applications, particularly when certain process parameters are difficult to measure in industrial scenarios. In contrast, data-driven models present a viable alternative by uncovering relationships between easily measurable variables and target variables through data mining techniques [10].

The data-driven softsensor regards complex industrial processes as black-box systems, focusing on analyzing auxiliary data to address challenges for key operational indicators.

✉ Yu Peng  
lecstarr@163.com

<sup>1</sup> College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

Zhu et al. [11] proposed a modeling approach that combines partial least squares (PLS) with a radial basis function neural network (RBFNN) to evaluate effluent total phosphorus. Zhang et al. [12] employed a feature selection strategy based on importance coefficients of auxiliary variables for dimensionality reduction, followed by a dynamic Bayesian network to predict effluent quality. Sharafati et al. [13] highlighted the reliability of ensemble learning algorithms in wastewater management and engineering, synthesizing the performance of AdaBoost regression (ABR), gradient boost regression (GBR), and random forest regression (RFR) to predict wastewater quality indicators. Although these methods achieved satisfactory results, experimental findings revealed that shallow models struggle to capture the complex nonlinear relationships inherent in industrial processes. As an alternative paradigm, deep learning (DL) models have been explored to analyze industrial processes by extracting quality-related features from collected data using advanced representation learning techniques. To enhance training efficiency and accuracy, Wang et al. [14] proposed a deep belief network with event-triggered learning (DBN-EL) to minimize computational overhead caused by redundant data. Hua et al. [15] integrated the Harris Hawk optimization algorithm (HHO) with long short-term memory networks (LSTM) to predict key biological parameters such as substrate concentration, bacterial concentration, and product concentration. Addressing the challenges of high dimensionality and limited training data, Li et al. [16] developed a deep semi-supervised learning framework for multi-output softsensors in wastewater treatment processes.

Beyond these achievements, many real-world applications involve varying degrees of noise, manifesting as signal distortion or uncertain information within the system [17]. With the growing complexity of modern industries, environmental noise has increasingly become a significant obstacle to traditional softsensor modeling [18]. Consequently, a critical aspect of softsensor modeling is mitigating noise effects and extracting the most relevant information [19, 20]. Furthermore, unlike the relatively stable inputs in manufacturing scenarios, wastewater treatment processes are characterized by significant uncertainties [21, 22]. Fixed model structures are inadequate for identifying new features and potential correlations in data resulting from environmental changes [23]. Addressing these challenges requires adaptable modeling approaches that align with domain-specific characteristics and are practical for industrial implementation. Therefore, proposing a general approach that can dynamically adjust network structures and parameters to accommodate noise and environmental dynamics holds great theoretical and practical significance.

The training process is closely linked to model performance [24]. Although various approaches have been proposed for designing and training softsensor models, few have

deeply explored the simultaneous optimization of network structure and parameters [25]. Multiobjective optimization is highly relevant to model training, as both the parameters and the model structure significantly influence the final outcomes, including prediction performance, inference time, model size, and computational efficiency [26]. To further enhance the training efficiency and prediction accuracy of softsensor models under noise corruption, this study proposes an effective multiobjective evolutionary algorithm with denoising strategy. The proposed approach explicitly considers uncertainties caused by noise and dynamic inputs to achieve the coevolution of network structure and parameters.

The main contributions can be summarized as follows:

- (1) The dual-population encoding strategy achieves the coevolution of network structure and weight parameters. The first population is responsible for evolving network structure, while the second population is mainly responsible for optimizing the weight parameters. This approach allows for a more holistic search in the decision space by exploiting the complementary advantages of these two populations;
- (2) The population evolution strategy, based on a denoising variational autoencoder (DVAE), addresses model uncertainty induced by noise and dynamic environments. It extracts both global and local features from auxiliary data, aiding in the construction of deep nonlinear representations to reduce the impact of noise on model performance;
- (3) As a gradient-free optimization strategy, the training method based on evolutionary strategies significantly reduces computational overhead while mitigating issues such as gradient vanishing and exploding. Moreover, the metaheuristic framework eliminates the need for correlation analysis or feature selection of decision variables, thereby effectively preventing suboptimal results arising from improper partitioning of decision variables due to environmental changes.
- (4) The proposed framework provides a systematic approach to designing softsensor models that not only perform well on training data but also generalize effectively to unseen scenarios. Using the same initialized softsensor model as benchmark, MENO-D demonstrated competitive results in terms of both training efficiency and model accuracy.

The remainder of this paper is organized as follows: section two provides a brief overview of related works. Section three presents detailed theoretical descriptions and the architecture of the proposed algorithm. Section four discusses the experiments conducted on the UCI water quality prediction dataset and the IWA benchmark wastewater treatment dataset, along with the corresponding results. These



experiments demonstrate the effectiveness of the proposed denoising optimization strategy. Finally, section five concludes the paper by summarizing its main contributions and outlining directions for future research.

## Related works

### Design and training of the softsensor model

One of the most significant challenges in building an efficient softsensor model is the substantial influence of architecture and parameter configuration, which are heavily dependent on the dataset and specific learning task. Many studies address this challenge by selecting models through trial-and-error processes, which is a complex and time consuming [27, 28]. Without a clear formula, determining an optimal model often depends on a combination of prior experience and iterative tuning. Given the computation demands of deep learning algorithms, training networks using a trial-and-error approach is not only time-consuming but also frequently results in suboptimal and incomplete models.

The parameters of softsensor model should be optimized according to the auxiliary variables, while the parameters of different softsensor models used according to different objectives should be optimized independently. These nested optimization problems could be addressed by developing hybrid structures that combine softsensor models with optimization algorithms. Therefore, an effective method to search for the optimal configuration of parameters and network structure is essential [29, 30]. The central hypothesis is that optimal neural networks—well-tuned in terms of structure and parameters—can achieve more accurate predictions of key performance indicators in the wastewater treatment process. Among various techniques, metaheuristic algorithms allow for random searches of optimal parameters while using previous results to guide the search toward near-optimal solutions [31]. Moreover, the same network architecture can produce different outcomes depending on the input data. To ensure robust model performance, the input auxiliary variables should be optimized alongside the network topology and parameters for the given task. The input of WWTP is passively received, and the data characteristics vary significantly across different weather conditions and seasonal changes. As a result, it is not possible to enhance the softsensor's accuracy by solely optimizing the training data. To improve the robustness of the softsensor model for wastewater treatment under varying operational conditions, this study focuses on coevolution of structure and parameters to better accommodate the dynamic changes in system input.

### Network optimization based on evolutionary computation

Evolutionary algorithms are crucial for optimizing neural networks [32–36]. While these methods have demonstrated advantages in certain scenarios, many evolutionary approaches face efficiency challenges due to the cost of evaluations. Furthermore, several critical aspects still require further investigation and improvement. In the context of wastewater treatment process (WWTP), the coevolutionary optimization of network parameters and structure remains scarce and limited. Among the few studies conducted in this area, most focus solely on optimizing parameters with fixed depths, with little exploration into optimization techniques that simultaneously address both network structure and parameters [37]. Additionally, fine-tuning parameters during pre-training, such as separation layers, learning rates, and batch sizes, has received insufficient attention.

The optimization problem is to find the best solution from a set of candidates to maximize or minimize the objective function. In general, an optimization problem comprises of three main components: a set of decision variables  $x$ , an objective function  $f(x)$  that should be maximized or minimized, and a set of constraints that determines the valid range of values to assign to specific parameters over a period of time. Consequently, the optimization of softsensor model can be formally defined as follow:

Let  $A_\lambda$  denote an evaluation algorithm as a mapping  $A_\lambda: N \rightarrow M$  where  $N$  is the set of all data sets (population) with  $d$  dimensions and  $\lambda \in \Lambda$  is the chosen configuration with  $\Lambda = \Lambda_1 \times \Lambda_2 \times \dots \times \Lambda_D$  being the  $D$ -dimensional decision space of algorithm  $A$ .  $M_\lambda \in M$  is a model that measures and minimizes a set of objective functions  $\mathcal{F}$  (e.g. accuracy, diversity) with its model regularization term  $R$  and estimated by the proposed algorithm on training data  $D^{train}$  and evaluated on validation data  $D^{valid}$ .

$$A_\lambda(D^{train}) = \arg \min_{M_\lambda \in M} \left\{ \mathcal{F}_1(M_\lambda, D^{train}), \mathcal{F}_2(M_\lambda, D^{train}) \right\} + R(M_\lambda, \lambda) \quad (1)$$

The goal of optimizing network structure and parameters is to find an optimal set of  $\lambda^*$  using a validation set:

$$\lambda^* = \arg \min_{\lambda \in \Lambda} \left\{ \mathcal{F}_1(A_\lambda(D^{train}), D^{valid}), \mathcal{F}_2(A_\lambda(D^{train}), D^{valid}) \right\} = \{Fit_1(\lambda), Fit_2(\lambda)\} \quad (2)$$

where  $Fit$  represents the fitness value.

Real-world applications are always affected by disruptive noise, where the true objective functions are unobservable, and only noisy observations are accessible. In such cases, when noise cannot be disregarded, Eq. (1) can be reformulated as:

$$A_{\lambda}(D^{train}) = \arg \min_{M_{\lambda} \in M} \left\{ \mathcal{F}_1(M_{\lambda}, D^{train}, noise), \right. \\ \left. \mathcal{F}_2(M_{\lambda}, D^{train}, noise) \right\} + R(M_{\lambda}, \lambda) \quad (3)$$

According to Nissen et al. [38], population-based evolutionary algorithms exhibit inherent robustness in optimization under low noise levels. However, the presence of noise in multiobjective optimization problems can adversely affect the effectiveness of mutations and crossover operations, potentially leading the algorithm to make incorrect decisions when selecting non-dominated solutions. This can directly undermine the plausibility of the probabilistic model, resulting in premature convergence or trapping the algorithm in suboptimal solutions. The impact of noise-corrupted objective functions on the performance of multiobjective evolutionary algorithms will be analyzed in detail in "Mutation and crossover operation" section.

## Generative model in evolutionary computation

Constructing a complete machine learning pipeline is often a trial-and-error process that requires substantial domain knowledge and computational resources [39]. Furthermore, task-specific models cannot be easily repurposed for other problems, making it a nontrivial challenge to identify optimal models for given datasets. With advancements in computing devices, automated construction techniques have shown significant promise in addressing these challenges, aiming to streamline and automate model design for specific tasks efficiently [40].

Characterized by parallelism and diversity, evolutionary computation (EC) is well-suited for integration with machine learning techniques. As a prominent new technology, the application of EC to optimize machine learning models has garnered significant attention in both academia and industry. However, traditional evolutionary algorithms (EAs) rely heavily on fitness evaluations (FEs) to generate and select new populations. Data noise and input uncertainty caused by environmental changes can compromise the algorithm's ability to accurately assess optimization problems, leading to degraded performance, especially when the number of FEs is limited [41]. When fitness evaluations are insufficient for accurate optimization, generative models can progressively learn data features during population iterations, enabling the construction of new solutions for complex optimization problems without relying on additional evaluations. As an

classic generative model, variational autoencoders (VAEs) offer advantages such as simple structure and ease of training. By learning the Pareto front (PF) in the objective space, VAEs can generate auxiliary populations with the same level of convergence as non-dominated solution sets while achieving greater diversity [42]. Although VAE performs well in practice, it is desired to relax this strong restriction on the variational distributions in order to apply it to more complex real-world problems. The denoising variational autoencoder (DVAE) retains the advantage of VAE in approximating the true distribution of complex data through variational inference and adds the ability to handle noise on this basis. DVAE is designed to handle noise in input data, improve model robustness by injecting randomness into the encoding process, and generate smoother latent spaces, which is useful for exploring and utilizing the potential distribution of complex optimization problems [43]. This potential space helps improve the guidance and efficiency of optimization algorithms in the search process. In this paper, we propose to embed a DVAE in the evolutionary process to eliminate the unforeseeable environmental disturbances.

Supposing the original dataset is  $x = \{x_1, x_2, \dots, x_n\}$ ,  $p(\bar{x}|x)$  is a noise distribution of  $x$ ,  $q_{\varphi}(z|\bar{x})$  obeys a Gaussian distribution:

$$q_{\varphi}(z|\bar{x}) = N[z | \mu_{\varphi}(x) \text{diag}(\sigma_{\varphi}^2(\bar{x}))] \quad (4)$$

where  $\bar{x}$  is corrupted data,  $\mu_{\varphi}(x)$  and  $\sigma_{\varphi}(x)$  are the mean vector and variance vector, respectively.

Let  $p(\bar{x}|x)$  be a known corruption distribution, the mixture of Gaussian is

$$E_{p(\bar{x}|x)}[q_{\varphi}(z|\bar{x})] = \int_x q_{\varphi}(z|\bar{x}) p(\bar{x}|x) d\bar{x} \quad (5)$$

Since  $E_{p(\bar{x}|x)}$  needs to integrate the noise into a noise distribution, it is necessary to add a random layer to the variational lower bound and integrate the random variables before introducing the denoising variational lower bound, which is

$$\bar{q}_{\varphi}(z|x) = \int q_{\varphi}(z|\bar{x}) p(\bar{x}|x) d\bar{x} \quad (6)$$

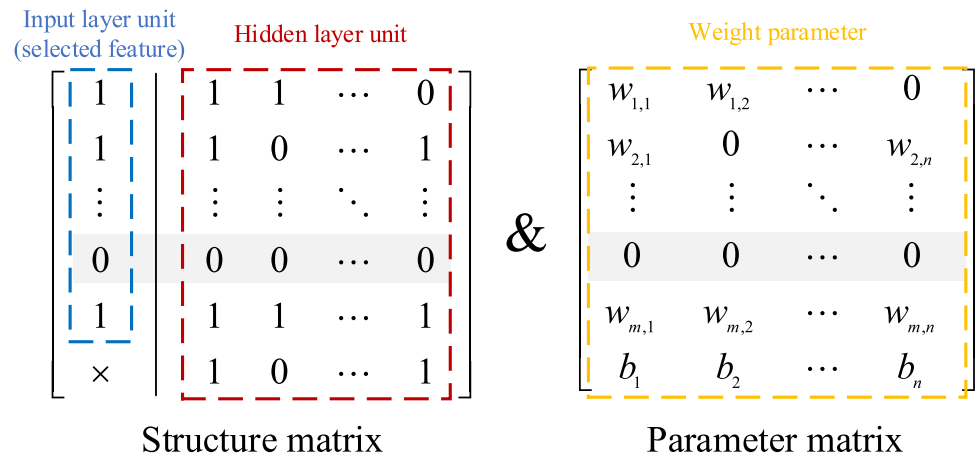
From the approximate distribution  $p_{\theta}(x|z) = p_{\theta}(x, z)/p(z)$ , the standard variational lower bound can be formulated as

$$L_{DVAE} = E_{\bar{q}_{\varphi}(z|x)} \left[ \log \frac{p_{\theta}(x, z)}{q_{\varphi}(z|\bar{x})} \right] \quad (7)$$

Maximizing  $L_{DVAE}$  is equivalent to minimizing the following objective  $E_{p(\bar{x}|x)}[KL(\bar{q}_{\varphi}(z|\bar{x})||p(z|x))]$ , That is,

$$\log p_{\theta}(x) = L_{DVAE} + E_{p(\bar{x}|x)}[KL(\bar{q}_{\varphi}(z|\bar{x})||p(z|x))] \quad (8)$$

**Fig. 1** Encoding matrix for individuals in the population of MENO-D



## Proposed method

In this section, the proposed MENO-D framework is introduced and explained in detail through a step-by-step approach. First, a dual-population encoding method is proposed to coevolve parameters and network structures as separate populations. An information-sharing mechanism is implemented to synchronize the evolutionary directions of these populations. Next, a generative model is employed to learn the potential association between training errors and the optimal population, conserving computational resources and enabling the optimization algorithm to achieve broader exploration and more precise exploitation. Finally, coding and evolutionary strategies are integrated to develop a noise-tolerant softsensor model training framework tailored for dynamic environments.

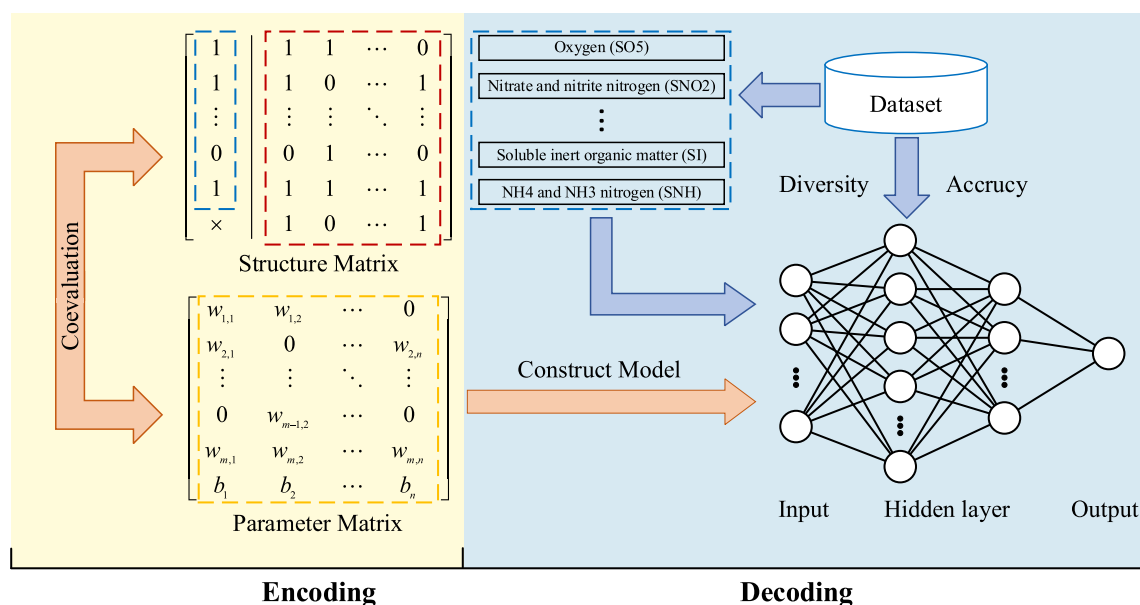
## Encoding and decoding

The representation of solutions significantly influences the efficiency of metaheuristic algorithms. To facilitate the co-evolution of parameters and network structure, two sub-populations are constructed to represent distinct optimization objectives. As illustrated in Figs. 1 and 2, individuals in the populations are encoded as matrix-like arrays. The structure array consists of binary vectors, where the row length  $m$  corresponds to the number of hidden layers, and the column length  $n$  corresponds to the number of nodes in the network. In these binary vectors, a value of 1 indicates activation of the corresponding node to build the base learner, while a value of 0 indicates inactivation. The parameter array, on the other hand, is a real-valued matrix with  $m + 1$  rows and  $n$  columns. This matrix encodes both the structural and parametric information of the corresponding base learner. In order to avoid wasted computation due to redundant network structures, the number of rows of each newly generated individual is randomly determined in the range of  $[m_{1,min}, m_{1,max}]$  during

the evolution process, and similarly, the number of columns is randomly determined in the range of  $[n_{1,min}, n_{1,max}]$ .

To ensure generality while assessing individual accuracy and diversity, we selected the classic multi-layer perceptron (MLP) as the benchmark softsensor model. The MLP framework is highly flexible, allowing the customization of network structures, including the number of hidden layers and output layer neurons, to address the specific requirements of different problems. In this study, the goal is to build softsensor model that satisfies the two key indicators—chemical oxygen demand (COD) and biochemical oxygen demand over five days (BOD5)—in the wastewater treatment process through the collaborative optimization of structure and parameters. The evaluation process consists of three steps:

- (1) The network structure is derived from the structural matrix, while the weights and bias parameters for each node are extracted from the parameter matrix. As illustrated in Fig. 2, the elements in the blue dashed box of the structural matrix represent the input nodes of the MLP, while the elements in the red dashed box specify the structure of the hidden layer. Additionally, the elements in the parameter matrix correspond to the weights of each hidden layer node and the biases of each layer.
- (2) Select the corresponding auxiliary variable from the original dataset as the training data based on the individual network design scheme. Construct metrics for diversity (generalization ability of the model) and accuracy (prediction ability of the model) to evaluate the comprehensive performance of the optimal model.
- (3) Train the MLP for each individual based on the selected training set, network structure, and parameters. Evaluate the optimal MLP model on the test set, with the evaluation result of the softsensor model serving as the evaluation result for the individual.



**Fig. 2** Schematic of the encoding and decoding methods of MENO-D

### Generative model enhanced optimization strategy

The overall framework of the proposed multiobjective evolutionary network optimization strategy with denoising strategy (MENO-D) is outlined in Algorithm 1. Each individual comprises two components: binary parts (network structure and input selection, denoted as  $X_{S,I}$ ) and real-valued parts (network parameters, denoted as  $X_P$ ). To generate new individuals, appropriate mutation and crossover operations are required. Meanwhile, to effectively mitigate the negative impact of noise, Algorithms 3 and 4 respectively introduce the evolutionary search strategy and the generative model enhancement strategy employed in the proposed method.

The initialization phase generates the initial population  $P$  by Algorithm 2 and constructs a DVAE model  $\Pi$ . The DVAE aims to generate new solutions by learning the latent distribution of input data (evolved population), which is then used to extract features and produce new individuals in the subsequent training process. The main loop governs the evolutionary

process until the maximum number of iterations,  $maxGen$ , is reached. In each iteration, new candidate solutions are generated through selection, crossover, mutation, and other operations, progressively optimizing the population. The training interval  $t$ , determines the frequency of model updates. Every  $t$  iterations, the model  $\Pi$  is trained using Algorithm 4, providing guidance for generating new individuals. After training, Eq. (13) is applied to produce the next generation of individuals. When the training interval has not yet been met, the offspring generation process utilizes Eqs. (9) and (10) to explore new solutions within the search space. During this phase, DVAE model remains unchanged, and classic evolutionary operators are employed for mutation and crossover between individuals. At the end of each iteration, a fast non-dominated sorting algorithm selects the optimal solution set. These non-dominated solutions are then used as the training dataset  $T_s$ , for the DVAE in subsequent training, enabling the model to focus on generating better individuals.



**Algorithm 1** Framework of MENO-D

---

**Require:**  $N$  (population size),  $s$  (input variable size),  $t$  (training interval of DVAE),  $Dim$  (encoder output dimension),  $N_t$  (training set size),  $maxGen$  (maximum number of iteration);

**Ensure:** a set of non-dominated solutions  $NS$

- 1: Generate initial population  $P=(P_1, P_2)$  by **Algorithm 2**
- 2: Generate initial DVAE model  $\Pi$
- 3: **while** ( $gen \leq maxGen$ ) **do**
- 4:   **if**  $gen \bmod t == 0$  **do**
- 5:     training DVAE model  $\Pi$  by **Algorithm 4**
- 6:     produce offspring by Eq.(13)
- 7:   **else**
- 8:     produce offspring by Eq.(9) and Eq.(10)
- 9:   **end if**
- 10:   operate fast non-dominated sort and create DVAE training set  $T_s$  by **Algorit**

**hm 3**

- 11:   Select  $N$  solutions from the non-dominated solution set
- 12:    $gen = gen+1$
- 13: **end while**
- 14: **Return**  $NS$

---

**Initialization**

The proposed coevaluation problem involves two key considerations. The first is implementing independent optimization strategies within two sub-populations that utilize different encoding methods. The second is assigning network structure information to parameter matrices in a rational manner to achieve co-evolution. Given the specificity of the network structure and its associated parameters, the initially random populations are adapted using the population adaptation method (PAM) to ensure that the individuals within the populations accurately represent meaningful configurations. Subsequently, the pairing rule outlined in the previous section is employed to map the network structure matrix information to the corresponding parameter matrix.

To ensure the generality and robustness of the model, we stipulate that the number of hidden layers should be between 2 and 10, with the specific size adjusted according to the complexity of the task. The number of neurons in each layer is typically set based on the number of input neurons  $N_i$ , generally ranging from  $N_i$  to  $2N_i$ , in order to ensure the network's learning capability and computational efficiency. This design aims to enhance the model's expressive power through appropriate network structure flexibility while avoiding overfitting and ensuring computational efficiency. If the softsensor model represented by an initial individual does not meet the model constraints, the elements in the structure matrix and the corresponding weight parameter matrix will be adjusted through PAM before executing the evaluation. The framework of the initialization method is shown in Algorithm 2.

**Algorithm 2** Population Initialization

---

```

1. Input: input variable size  $s$ , boundary of matrix's column and row, population size  $N$ 
2. Output: optimal initial population  $P_1, P_2$ 
3. for  $i = 1:N/2$  do
4.     randomly generalize  $m, n$  according to  $S$  matrix boundary
5.      $M_{S,i}[:,1] = \text{randi}([0 \ 1], s)$ 
6.      $M_{S,i}[:,2:n] = \text{randi}([0 \ 1], m, n - 1)$ 
7. end for
3. for  $i = 1:N/2$  do
4.     for  $p = 1:m$  do
6.         for  $q = 1:n-1$  do
7.             if  $M_{S,i}[p, q] > 0$  do
8.                  $M_{H,i}[p, q] = \text{randn}(0, \frac{2}{m+n-1})$ 
9.             end if
10.        end for
11.    end for
12. end for
13. return  $P_1 = M_S, P_2 = M_H$ 

```

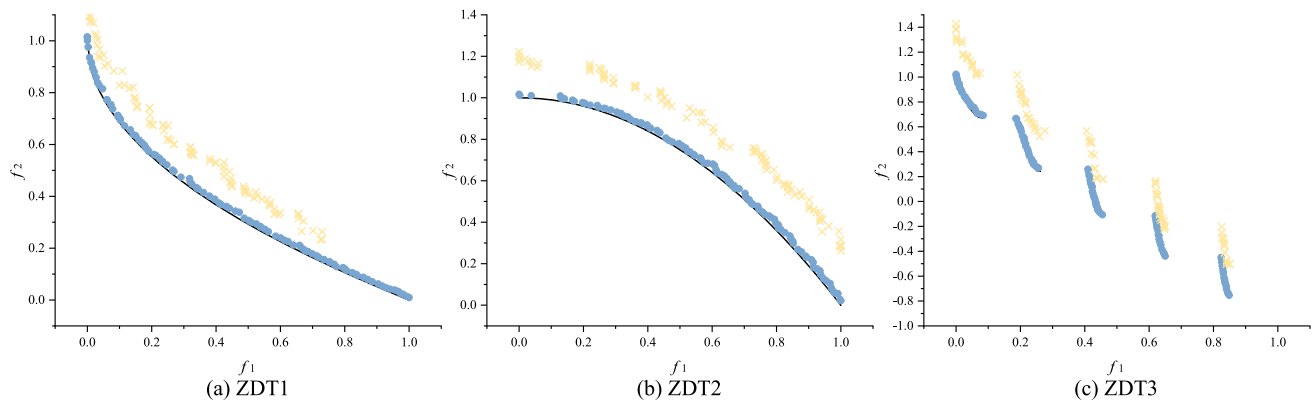
---

**Mutation and crossover operation**

Generating high-quality offspring through iterative information transfer is a critical aspect of the coevolutionary optimization problem discussed in this paper. Recently, offspring generation methods based on differential evolution (DE) have gained increasing attention, as noted in [44]. DE approximates the optimal solution iteratively by simulating natural evolutionary processes, incorporating mutation, crossover, and selection operations on population individuals. For each generation, the algorithm creates a “mutated individual” for each population member, generates a “test individual” through crossover, and determines whether to replace the current individual with the test individual based on selection criteria. However, due to noise and system input uncertainty, the optimization problem is often unstable, and conventional genetic operators may yield suboptimal solutions. To address this, we employ generative models with learning capabilities to replace manually designed mutation and crossover operators. By constructing distributions that approximate real Pareto sets (PSs) and Pareto fronts (PFs), this method mitigates the influence of human biases on population evolution and effectively leverages the distribution of promising solutions within the population.

**(1) Mutation**

In existing work, some studies have been conducted on adaptive crossover operators, but the impact of crossover genes between offspring and parents in noisy environments has not been thoroughly investigated. The crossover operator combined with generative modeling proposed in this paper dynamically determines the number of selected features in the offspring based on parent similarity. This approach enhances the algorithm's exploration capability by adaptively adjusting the number of mutated genes or the mutation rate. The influence of noise on the non-dominated solution set is illustrated in Fig. 3, which compares the actual and perceived positions of the final trade-off. In the figure, the solid line represents the true PF, while the solid dots and crosses indicate the actual and disturbed PFs, respectively. Notably, decisions are made based on the corrupted PF, as the true objective values are unknown, leading to a high likelihood of selecting suboptimal solutions in noisy environments [45]. To address this, the proposed evolutionary strategy combined with a DVAE constructs an auxiliary population to explore potential correlations between the optimal softsensor model and final performance.



**Fig. 3** The actual and corrupted location of the evolved tradeoff under the influence of 5% noise

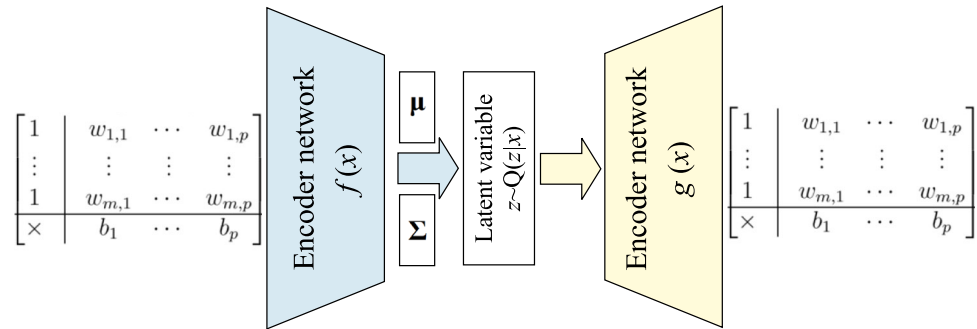
DVAEs with simple structures can reconstruct larger datasets from noisy raw inputs using a small amount of training data. These reconstructed datasets share the same distribution as the training data but are not identical to it. We selected this simple generative model for three key reasons. First, employing a more sophisticated model, such as a carefully designed and tuned generative adversarial network (GAN), would require significantly more training data, leading to a sharp increase in computational costs. Second, the effective knowledge acquired during evolution is constrained by the population size, and adopting DVAEs can alleviate this limitation by reducing the data required for the algorithm optimization process. Finally, in the field of evolutionary optimization, similar simple networks have been validated in other recent studies [36, 46, 47]. This is likely because generative models in evolutionary optimization typically train on less practical data, require lower accuracy, and focus primarily on estimating the probability distribution of the optimal solution set. Consequently, their scale is much smaller than in applications such as image processing or natural language processing, where simple networks can still perform effectively.

During the evolutionary optimization, the number and quality of non-dominated solutions vary depending on the problem's complexity and the stochastic nature of the searching process. For instance, in the early stages of evolution, the number of non-dominated solutions is typically small, and both convergence and diversity may be inadequate. At this stage, the knowledge acquired by the population is insufficient to meet the training requirements of DVAE. Therefore,

a conventional differential evolution strategy is employed to optimize and accumulate knowledge during population evolution. As shown in Fig. 4, at the late stages of evolution, the number of non-dominated solutions becomes sufficient for DVAE training. However, these solutions may exhibit high proximity to one another, potentially limiting diversity. To address this, some dominated solutions are included as training data for the DVAE to ensure diversity among the generated solutions.

The training set selection strategy is detailed in Algorithm 3. To balance convergence and diversity while ensuring the generative model can reliably generate high-quality auxiliary populations at different evolutionary stages, the training method in MENO-D is designed to adapt to the characteristics of each stage. Given that the distribution of the training set (non-dominated solution set) generally remains stable across successive generations, MENO-D trains the DVAE at specific iteration intervals to balance algorithm performance and computational complexity. Specifically, the algorithm selects elite individuals from the main population to train a convolutional neural network-based DVAE periodically. This DVAE generates an auxiliary population with improved convergence and diversity, guiding the main population toward the desired optimization direction. Each new surrogate is constructed to address the approximation errors of earlier surrogates. By iteratively incorporating newly constructed surrogates, the model is progressively refined, as the new surrogates help correct prediction errors made by their predecessors.

**Fig. 4** Schematic of the DVAE methods for generating offspring in the population




---

**Algorithm 3** Training Set Classification

---

1. **Input:** initial population  $P = (P_1; P_2)$ , training set size  $N_t$ , stagnant dictator  $\varepsilon$
  2. **Output:** DVAE training dataset  $T_s$
  - 1:  $T_s = []$ ,  $p = 1$
  - 2: Sort  $P$  by fast non-dominated sort, obtain non-dominated solution set  $P_n$
  - 3: **for**  $k = 1:N_t$  **do**
  - 4:    $c_1 = \min \text{fit}_m^{g-1}(P_k) - \min \text{fit}_m^g(P_k)$
  - 5:    $c_2 = \min \text{fit}_m^{g-2}(P_k) - \min \text{fit}_m^g(P_k)$
  - 6:   **if**  $c_1 - c_2 \geq \varepsilon$  **do**
  - 7:      $T_s(k) = P_n(k)$ ;
  - 8:   **else**
  - 9:      $P \boxminus= P - P_n$
  - 9:      $T_s(k) = P \boxminus(p)$
  - 10:     $p = p + 1$
  - 11:   **end if**
  - 12: **end for**
  13. **return**  $T_s$
- 

(a) Early stage of evolution

In each generation  $g$ , each individual  $x_i(g)$  is regarded as a target individual, and the mutation operation is applied for it to generate its mutant individual  $x_i(g+1)$ . The mutation operation is given as Eqs. (9) and (10):

DE/rand/1

$$x_i(g+1) = x_{r1}(g) + F_1 \cdot (x_{r2}(g) - x_{r3}(g)) \quad (9)$$

DE/best/1

$$x_i(g+1) = x_{best}(g) + F_2 \cdot (x_{r2}(g) - x_{r3}(g)) \quad (10)$$

where  $x_i(g+1)$  denotes the  $i$ th generated individual in generation  $g+1$ ;  $r_1$ ,  $r_2$ , and  $r_3$  are three random and mutually exclusive integers in the range of  $[1, n]$ ;  $x_{r1}(g)$ ,  $x_{r2}(g)$ , and  $x_{r3}(g)$  denote the  $r_1$ th,  $r_2$ th, and  $r_3$ th individuals in generation  $g$ , respectively;  $x_{best}(g)$  denotes the best individual obtained in generation  $g$ ; and  $F$  is a scaling factor. The global convergence rate of DE/rand/1 is slow, but it has good local search capability. Conversely, DE/best/1 has great global search ability, but relatively weak local search ability [48].



## (b) Late stage of evolution

After obtaining sufficient training data, i.e., the training set  $T_s$  meet the preset training set size requirements,  $x_{r3}$  in the mutation operator will be replaced with  $x'$  to further guide the direction of population evolution.

First, the mean vector and covariance matrix of the real samples are calculated by:

$$\mu = \frac{\sum_{i=1}^{N/2} \mathbf{r}_i}{N/2} \quad (11)$$

$$\Sigma = \frac{\sum_{i=1}^{N/2} (\mathbf{r}_i - \mu)(\mathbf{r}_i - \mu)^T}{N/2 - 1} \quad (12)$$

**Algorithm 4** Model Training

---

```

1. Input: DVAE training dataset  $T_s$ , encoder output dimension  $Dim$ 
2. Output: DVAE model  $\Pi$ 
3. while termination criterion not fulfilled do
4.    $(\mu, \sigma) \leftarrow \text{Encoder}(T_s, Dim)$ ;
5.    $Z \leftarrow \text{RandomSampling}(\mu, \sigma)$ ;
6.    $T' \leftarrow \text{Decoder}(Z)$ ;
7.   Calculate loss function
8.   Update encoder and decoder
9. end while
10. return  $\Pi$ 

```

---

where  $\mathbf{r}_i$  is the  $i$ th member of the real dataset and  $N$  is the population size. Then, the DVAE are trained for several iterations.

To generate a candidate solution using the DVAE, we first calculate the mean vector  $\mu$  and the covariance matrix  $\Sigma$  of the real samples according to Eqs. (11) and (12). Then, a  $D$ -dimensional vector  $x$  is sampled with each element being independently sampled from a continuous uniform distribution  $U(0,1)$ . Afterward, a  $D$ -dimensional vector  $y$  satisfying the multivariate normal distribution is generated according to the following probability density function:

$$y = \frac{\exp(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu))}{\sqrt{(2\pi)^D |\Sigma|}} \quad (13)$$

where  $D$  denotes the dimensionality of the decision space.

The training process of the DVAE is outlined in Algorithm 4. First, the training set  $T_s$  and the low-dimensional hidden layer dimension  $Dim$  are input into the encoder to learn the distribution of the training set, represented by  $\mu$  and  $\Sigma$  (line 2). Next, a low-dimensional point set  $Z$  is randomly sampled from the learned distribution (line 3). This point set  $Z$  is then reconstructed into  $T'$ , which has the same dimensionality as the original training data, using a decoder (line 4). Finally, the network's loss function is calculated based on formula (2), and the network parameters are updated accordingly (lines 5–6). This training process is repeated until the termination condition is satisfied, resulting in a well-trained DVAE model  $\Pi$  (line 8).

The main idea of generating mutation vectors is to add perturbations to the target individual  $x(g)$ , that is, the difference between  $x_{r2}$  and  $x_{r3}$  after scaling by  $F$ . According to this idea (perturbation, scaling and superposition), the mutation operator of the network structure in the coevolutionary population includes the following three steps.

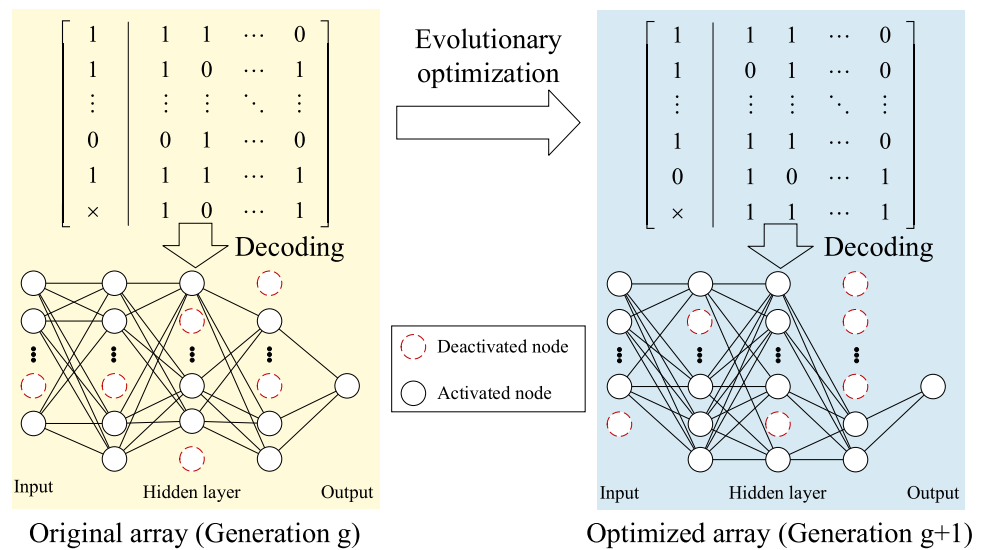
Step 1: A temporary binary vector  $x_{temp}$  is computed:

$$x_{temp} = x_{r2} \oplus x_{r3} \quad (14)$$

where  $\oplus$  denotes the element-wise exclusive-OR operation.  $x_{r2}$  and  $x_{r3}$  are two randomly selected binary vectors from the population.

Step 2: For each element  $i$  in  $x_{temp}$ : If  $x_{temp}[i] = 1$ : generate a random number  $R \sim \text{Uniform}[0, 1]$ ; Otherwise if  $R < F$  (where  $F$  is the differential weight or scaling factor):

$$x_{temp}[i] \leftarrow 0 \quad (15)$$

**Fig. 5** Schematic of network structure adjustment operation

Otherwise,  $x_{temp}[i]$  remains unchanged.

Step 3: The mutation vector  $x_i(g+1)$  is obtained by applying DE/rand/1 or DE/best/1 operation between the modified  $x_{temp}$  and a target vector (either  $x_{r1}$  or  $x_{best}$ ):

$$x(g+1) = x_{temp} \vee x_{target} \quad (16)$$

where  $\vee$  denotes the mutation operation,  $x_{target}$  is typically  $x_{r1}$  (a randomly chosen vector) or  $x_{best}$  (the best individual in the current population).

For the network parameter matrix, the primary issue arises from differences in the network structures corresponding to each weight parameter matrix. Directly applying variation and crossover operations on the elements of two individuals with differing structures may result in an unreasonable configuration. To address this problem, a metaheuristic network adjustment operation was defined and executed, and detailed information can be found in Fig. 5: where the red circles represent the nodes that need to discard during the evolutionary optimization. The first column in the structural matrix determines the input auxiliary variables selected for the softsensor model, while the Boolean values in the remaining columns determine the activation of hidden layer nodes. Correspondingly, the elements (weights) of the parameter matrix will also change according to the structural changes.

## (2) Crossover

After the mutation operation, the mutant individual  $v_i(g+1) = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$  will mix with its target individual  $v_i(g)$  to generate its trial individual  $u_i(g+1) = (u_{i,1}, u_{i,2}, \dots, u_{i,D})$ . The crossover consists of two components: one addressing the structure optimization and the other focusing on network parameters. The structure crossover is identical to that of the DE binomial or exponential crossover strategy. For the parameter matrix, which involves real-number encoding,

specific adjustments are applied to accommodate the inherent characteristics of the problem.

In general, the binomial crossover methods are most frequently used, which can be expressed as:

$$u_{i,j}^{g+1} = \begin{cases} v_{i,j}^{g+1}, & \text{if } rand < Cr \text{ or } j = randn(i) \\ x_{i,j}^g, & \text{otherwise} \end{cases} \quad (17)$$

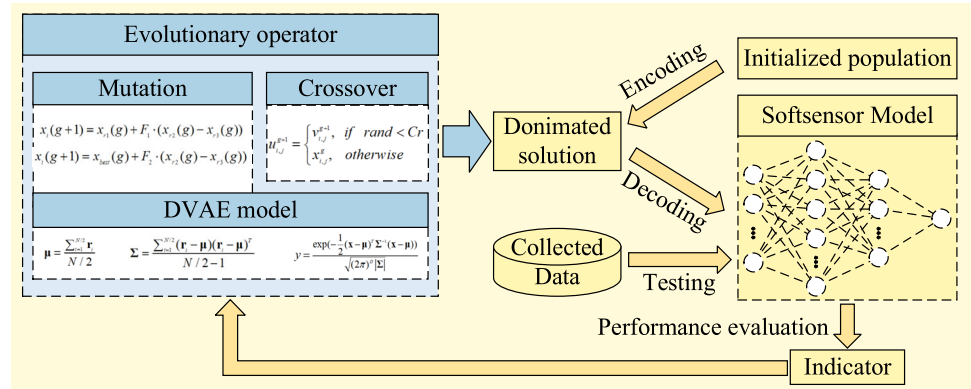
where  $Cr \in [0, 1]$  is the crossover rate, and  $randn(i)$  is an integer randomly generated from the range  $[1, N]$  to ensure that  $u_i^g$  gets at least one component from  $v_i^g$ .

(a) Fixed number of rows: From the perspective of the entire population, all available input variables will be selected by at least one individual. The advantage of this encoding method is that, regardless of whether an auxiliary variable is selected by an individual, the corresponding input weights will be optimized by the evolutionary algorithm. Once a new input is selected, it can ensure that the corresponding weights of the node have been optimized by the algorithm, rather than completely randomly generated, which can accelerate the convergence of the algorithm to a certain extent.

(b) Unfixed number of columns: Since each column represents all the parameters related to a hidden layer node, the change of the number of columns is equivalent to the change of the network structure, which makes it possible to explore more promising base learners of different structures.

Figure 6 illustrates the framework of proposed coevolutionary softsensor optimization strategy. The schematic demonstrates a cyclic workflow involving an initialized population processed through evolutionary operators. The decoded solutions are subsequently constructed into softsensor models and rigorously evaluated using collected industrial data. This dual-driven strategy emphasizes the synergistic interaction between evolutionary optimization

**Fig. 6** Illustration of proposed coevolutionary softsensor optimization strategy



and data-driven modeling, effectively balancing exploration of novel solutions with exploitation of historical process patterns. The color-coded blocks and directional flowlines provide a clear visual representation of the self-adaptive optimization mechanism and performance feedback loops.

### Complicity analysis

In each iteration, the computational cost of PM mainly consists of four parts: (1) the DVAEs training; (2) the knowledge accumulation; (3) the offspring production; and (4) the environmental selection. The computational complexity of training a DVAE model is  $O\left(T \sum_{i=1}^l d N h_{i-1}^{out} h_i^{out}\right)$ , where  $d$  is the dimension of the decision variables,  $N$  is the population size,  $h_i$  is the number of output channels of layer  $i$  in the network,  $l$  is the number of network layers, and  $T$  is the number of iterations used in the training. The time cost for generating an individual is  $O\left(\sum_{i=1}^l d N h_{i-1}^{out} h_i^{out}\right)$ . The time complexity of generating an offspring population based on DE strategy is  $O(dN)$ . In the environmental selection, the time complexity of nondominated sorting is  $O(mN^2)$ , where  $m$  is the number of objectives. Thus, the computational cost of an iteration of PM is  $\max\left(O\left(T \sum_{i=1}^l d N h_{i-1}^{out} h_i^{out}\right), O(mN^2)\right)$ .

## Experiments and discussions

### Experiment settings

Two experiments are conducted to evaluate the performance of the proposed MENO-D. First, the method's effectiveness is validated using a benchmark dataset for water quality prediction, during which a DVAE model is trained to effectively reduce noise interference. Subsequently, the trained model is applied to predict key operational indicators in wastewater treatment processes, demonstrating its practical utility.

Experiments were carried out on a high-performance computer featuring an Intel i7-13700 K CPU running at 5.4 GHz and an NVIDIA GeForce RTX 4060 Ti GPU with 16 GB of graphics memory, equipped with 32 GB DDR4 RAM and a 1 TB SSD. Among these experiments, five representative MOEAs are compared, namely, MOEA/D-DE (A1) [49], CEDE-DRL (A2) [41], GMOEA (A3) [44], MOEA/D-QN (A4) [50], HATC (A5) [51], REMO (A6) [52], CSEMT (A7) [53]. The selected methods include classical multiobjective algorithm based on differential evolution (A1), deep reinforcement learning (A2, A3), generative model (A4), metaheuristic strategy (A5), relation learning (A6) and multi-tasking (A7).

The population size for all competitors is set to 50, and the maximum number of fitness evaluations for each problem is set to  $10^4$ . In the proposed method, the encoder and decoder of the DVAE model both utilize a four-layer BP neural network structure, with a hidden layer dimension of 10, a learning rate of  $10^{-4}$ , and weight initialization performed using Xavier. The number of iterations for network training is set to 15. The scaling factors F1 and F2 in differential evolution are both set to 0.3, and the training interval for the DVAE is set to 10.

For fair comparison, we adopted the recommended parameter settings of the compared algorithms that achieved the best performance reported in the literature. The specific parameters are shown in Table 1.

### Performance evaluation

The optimization problem aimed at designing and training softsensor models is characterized by two conflicting objective functions, namely accuracy and diversity. These objectives are intentionally designed to address the inherent trade-offs in model performance.

(1) Accuracy: The first objective measures the performance of the model on the training dataset. Specifically, it quantifies how well the model fits the training data, which

**Table 1** Parameter settings of the employed multiobjective optimization algorithms

| Algorithm  | Population size | Parameter setting                                                                          |
|------------|-----------------|--------------------------------------------------------------------------------------------|
| MOEA/D-DE  | 50              | $CR = 1.0, F = 0.5, \eta = 20, p_m = 1/D$                                                  |
| CEDE-DRL   | 50              | $CR = \{0.1, 0.2, 1.0\}, F = \{0.6, 0.8, 1.0\}, Tc = 0.8, cp = 5.0, greedy = \gamma = 0.9$ |
| GMOEA      | 50              | $CR = 1.0, F = 0.5, \eta = 20, Nc = 20, n_m = 20, pc = 1.0, p_m = 1/D$                     |
| MOEA/D-DQN | 50              | $CR = 1.0, F = 0.5, \eta = 20, pc = 1.0, p_m = 1/D$                                        |
| REMO       | 50              | $CR = 1.0, F = 0.5, n_m = 20, pc = 1.0, p_m = 1/D, k = 6$                                  |
| HATC       | 50              | $\delta = 0.05$                                                                            |
| CSEMT      | 50              | $c = 0.5, \beta = 0.8$                                                                     |

serves as a primary indicator of its learning capacity. However, optimizing solely for accuracy on the training set often leads to overfitting, where the model learns patterns specific to the training data but fails to generalize to unseen data. To mitigate this, we emphasize the potential overfitting issue by explicitly framing accuracy as training set accuracy. This framing not only highlights the need for caution in interpreting high accuracy but also provides a basis for balancing it against the second objective. Since the prediction of wastewater quality indicator is a regression problem, root mean square error (RMSE) is chosen as the prediction accuracy metric for individual  $X_k$ , i.e., the first objective is:

$$\text{Minimize : } \text{RMSE}_k = \sqrt{\frac{1}{L} \sum_{i=1}^L (\hat{y}_k^i - y^i)^2} \quad (18)$$

where  $L$  represents the number of samples,  $\hat{y}_k^i$  is the predicted value of  $X_k$  for the  $i$ th sample, while  $y^i$  is the true value of the  $i$ th sample. Note that in the experiments, this objective is transformed into the maximization form by taking the negative value of the actual  $\text{RMSE}_k$ .

(2) Diversity: The second objective is designed to quantify the robustness of the model. In this context, diversity refers to the differences among the final models produced by the optimization algorithm and the rest of the population during training. By introducing diversity as a key metric, we aim to ensure that the algorithm explores a wide range of potential solutions, thereby reducing the risk of overfitting and enhancing the generalizability of the model. Diversity can also be interpreted as a measure of the model's ability to handle variations in input data, which is critical for real-world applications where data distributions may shift over time. The idea from negative correlation learning (NCL) is

employed to define the second objective:

Minimize :

$$\text{DIV}_k = \sum_{i=1}^L |y_k^i - y_{pop}^i| / \left( \sum_{j=1, j \neq k}^{2N_1} |y_j^i - y_{pop}^i| \right) \quad (19)$$

where  $2N_1$  is the number of individuals in the joint population,  $y_k^i$  and  $y_j^i$  represent the outputs of individuals  $X_k$  and  $X_j$  for the  $i$ th sample, respectively, and  $y_{pop}^i$  denotes the average output of all individuals for the  $i$ th sample.

Higher accuracy typically indicates that the model has achieved a superior fit to the training data. However, this often comes at the expense of diversity, as highly accurate models are more likely to overfit to specific patterns in the training set, leading to poor performance on unseen data. Conversely, emphasizing diversity encourages the exploration of alternative model structures and parameter configurations, which may result in lower accuracy on the training data but improved generalization. The trade-off between accuracy and robustness is critical for ensuring reliable performance in practical applications. By explicitly considering these conflicting objectives, our framework provides a systematic approach to designing models that not only perform well on training data but also generalize effectively to unseen scenarios.

## Benchmark test experiment

### (1) Description of benchmark dataset:

The benchmark dataset used in the first experiment, referred to as the water quality dataset [54], was sourced from the UCI Machine Learning Repository. The objective is to predict the spatiotemporal water quality, specifically the “hydrogen power (pH)” value, for the following day based on input features derived from historical data of various water quality indicators. As shown in Table 2, the dataset includes daily samples from 36 locations in Georgia, USA, providing measurements related to pH levels. These input features comprise 12 common indicators, such as dissolved oxygen, temperature, and conductivity. The dataset, released by the United States Geological Survey, reflects the complexity of water systems like the Atlanta River system, Savannah River basin, and Timons River basin. Notably, it lacks spatial connectivity information regarding water flow between locations, ensuring that predictions rely solely on the model's inherent performance rather than external spatial factors. To evaluate and compare the proposed noise-handling strategies, homogeneous Gaussian noise is artificially introduced into the dataset for this section.

### (2) Discussion on the experiment results:

As shown in Table 3, four different metrics have been



**Table 2** Description of water quality dataset

| Variable name  | Type     | Size   | Description                                                                                      |
|----------------|----------|--------|--------------------------------------------------------------------------------------------------|
| features       | strings  | 1*11   | A list of water indices to measure                                                               |
| location_ids   | integer  | 37*1   | IDs of the water stations                                                                        |
| X_te           | matrices | 1*282  | Test set input data: water indices for 282 contiguous dates until 2018-01-01                     |
| X_tr           | matrices | 1*423  | Training set input data: water indices for 423 contiguous dates from 2016-01-28                  |
| Y_te           | matrices | 37*282 | Test set output data: water quality for 37 locations in 282 contiguous dates until 2018-01-01    |
| Y_tr           | matrices | 37*423 | Training set output data: water quality for 37 locations in 423 contiguous dates from 2016-01-28 |
| location_group | cells    | 1*3    | The groups of water stations, each group forms a connected spatial network (i.e., water system)  |

**Table 3** Performance metrics

| Metrics                           | Formulation                                                    | Description                                                                              |
|-----------------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Root mean square error            | $RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^T (Y_t - \tilde{Y}_t)^2}$ | Highlight the large errors among the predictions                                         |
| Normalized root mean square error | $NRMSE = RMSE(Y_{max} - Y_{min})$                              | Enable the comparison of the performance across different datasets with different scales |
| Mean absolute error               | $MAE = \frac{1}{n} \sum_{t=1}^T  Y_t - \tilde{Y}_t $           | Directly shows the average errors of all the predictions                                 |
| Normalized mean absolute error    | $NMAE = MAE(Y_{max} - Y_{min})$                                | Enable the comparison of the performance across different datasets with different scales |

utilized to evaluate the performance on water quality prediction datasets, root mean square error (RMSE) is widely used to penalize large errors, making it well-suited for scenarios requiring high prediction accuracy, while mean absolute error (MAE) emphasizes the average overall error, which is particularly useful in situations involving numerous outliers. Additionally, as their normalized versions, normalized root mean square Error (NRMSE) and normalized mean absolute error (NMAE) are ideal for evaluating model performance across datasets or scenarios, as they facilitate comparisons of data with differing scales.

Table 4 presents the prediction performance of a water quality dataset under varying levels of data corruption, with bolded values highlighting the best results. This experiment was designed to assess the effectiveness of different deep learning-based multiobjective evolutionary algorithms, seven competitors (A1–A7) are listed in [Experiment settings](#) section, and MENO-D is referred as A8. Five water quality datasets, collected from different water stations, were used to train the softsensor model. To further evaluate the robustness of the algorithms in noisy environments, Gaussian noise at levels of 10% and 20% was artificially add into the dataset. The results show that the error metrics for nearly all algorithms increase as noise levels rise (from no noise to 10% and 20%), demonstrating that noise substantially impacts the performance of soft sensor models, leading to higher prediction errors and reduced accuracy.

As shown in Table 4, with the addition of 10% and 20% Gaussian noise, the model constructed by A8 maintains a remarkable level of stability. Taking Site#2198840 as an example, at 10% noise, the RMSE increases slightly to 0.01448, while NRMSE, MAE, and NMAE remain competitive at 0.17395, 0.03096, and 0.11842, respectively. The model's performance degrades minimally under the higher 20% noise level, with RMSE, NRMSE, MAE, and NMAE values of 0.03051, 0.37165, 0.03150, and 0.23389, respectively. These results suggest that the softsensor model is resilient to data destruction, highlighting its potential for applications in real-world sensor systems where noise and data corruption are prevalent. Furthermore, the variation in algorithm performance appears to be influenced by data characteristics. For example, experiment on water site #2198840 was more sensitive to high noise levels, whereas site #2203665 maintained relatively stable performance under 20% noise, underscoring the substantial impact of data characteristics on algorithm performance.

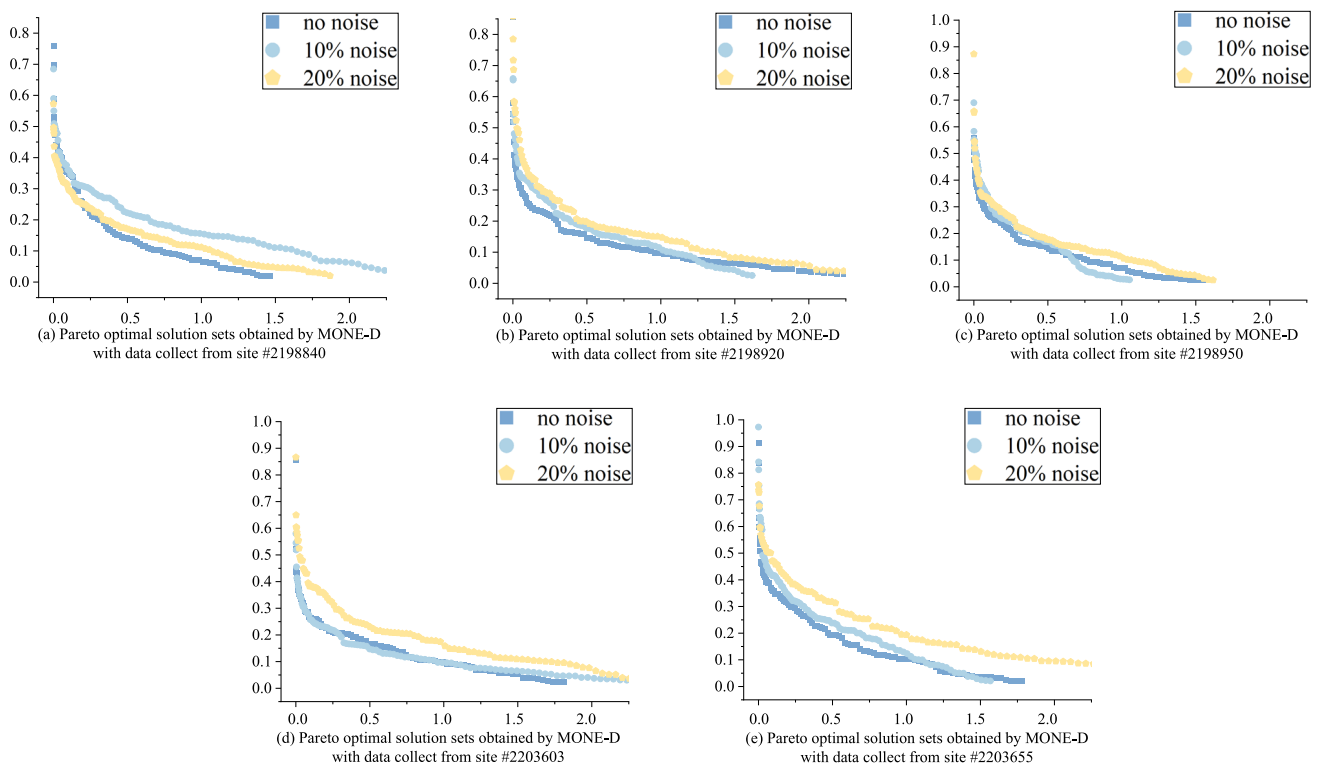
Figure 7 illustrates the variation of the PF for the optimization problem under three conditions: no noise, 10% noise, and 20% noise. The figure shows that in the no-noise condition (blue curve), the PF is generally closer to the ideal solution in the objective space, indicating that the algorithm converges effectively and performs optimally under noise-free conditions. However, with the introduction of 10% and 20%

**Table 4** Prediction performance for water quality dataset with different corruption level

|                      | No noise       |                |                |                | 10% noise      |                |                |                | 20% noise      |                |                |                |
|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                      | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           |
| <i>Site #2198840</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.01563        | 0.27626        | 0.01161        | 0.18718        | 0.01584        | <b>0.17372</b> | 0.03101        | 0.11881        | 0.03591        | 0.48503        | 0.03153        | 0.31580        |
| A2                   | 0.01598        | 0.26887        | 0.01412        | 0.20936        | 0.01613        | 0.17663        | 0.03122        | 0.11963        | 0.03403        | 0.41380        | 0.03525        | 0.25675        |
| A3                   | <b>0.01361</b> | 0.12156        | 0.01385        | 0.11275        | 0.01861        | 0.17822        | 0.03562        | 0.12878        | 0.03375        | 0.44981        | 0.03317        | 0.25363        |
| A4                   | 0.01553        | 0.23508        | 0.01139        | 0.13899        | 0.01646        | 0.17434        | 0.03395        | 0.12046        | 0.03605        | 0.41928        | 0.03418        | 0.26177        |
| A5                   | 0.01621        | 0.16986        | 0.01259        | 0.14971        | 0.01528        | 0.17783        | 0.03172        | 0.12158        | 0.03382        | 0.46603        | 0.03627        | 0.28505        |
| A6                   | 0.01529        | 0.18903        | 0.01148        | 0.15005        | 0.01481        | 0.17859        | 0.03228        | 0.12070        | 0.03160        | 0.43648        | 0.03714        | 0.28516        |
| A7                   | 0.01742        | 0.16497        | 0.01161        | 0.14927        | <b>0.01428</b> | 0.17507        | 0.03156        | 0.12205        | 0.03257        | 0.41650        | 0.03758        | 0.29352        |
| A8                   | 0.01459        | <b>0.10756</b> | <b>0.01059</b> | <b>0.11007</b> | 0.01448        | 0.17395        | <b>0.03096</b> | 0.11842        | <b>0.03051</b> | <b>0.37165</b> | <b>0.03150</b> | <b>0.23389</b> |
| <i>Site #2198920</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.01214        | 0.22931        | 0.00991        | 0.11532        | 0.18269        | 0.22193        | 0.00925        | 0.14418        | 0.01821        | 0.23813        | 0.00955        | 0.15764        |
| A2                   | 0.01513        | 0.21721        | 0.01189        | 0.12684        | 0.16965        | 0.21721        | 0.01182        | 0.16965        | 0.01896        | 0.24491        | 0.00983        | 0.18115        |
| A3                   | 0.01618        | 0.16292        | 0.01380        | 0.12269        | 0.17845        | 0.24788        | 0.01380        | 0.12087        | 0.01746        | 0.23724        | 0.00905        | 0.20653        |
| A4                   | 0.01642        | 0.16171        | 0.01178        | 0.12679        | 0.14065        | 0.16171        | 0.01178        | 0.14065        | 0.02147        | 0.23617        | 0.00901        | 0.15925        |
| A5                   | 0.01357        | 0.24852        | <b>0.00952</b> | <b>0.10874</b> | 0.14418        | 0.17115        | 0.01312        | 0.18269        | 0.01979        | 0.33660        | 0.00825        | 0.15685        |
| A6                   | 0.01288        | 0.14870        | 0.01153        | 0.12930        | 0.15460        | <b>0.16133</b> | 0.01038        | 0.16196        | 0.01966        | 0.25654        | 0.00984        | 0.16697        |
| A7                   | 0.01301        | 0.17341        | 0.01465        | 0.11927        | 0.17456        | 0.17078        | 0.01139        | 0.15225        | 0.02050        | 0.20724        | <b>0.00818</b> | <b>0.15670</b> |
| A8                   | <b>0.01202</b> | <b>0.12788</b> | 0.01107        | 0.11032        | <b>0.12087</b> | 0.16298        | <b>0.00994</b> | <b>0.11845</b> | <b>0.01414</b> | <b>0.20422</b> | 0.00934        | 0.17690        |
| <i>Site #2198950</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.01757        | 0.24261        | 0.01523        | 0.21034        | 0.01792        | <b>0.14959</b> | 0.01413        | <b>0.11782</b> | 0.01496        | 0.18825        | 0.00961        | 0.12058        |
| A2                   | 0.02101        | 0.36241        | 0.01406        | 0.24265        | 0.01856        | 0.29054        | 0.01189        | 0.18782        | 0.01685        | 0.20875        | 0.01073        | 0.12986        |
| A3                   | 0.01420        | 0.27809        | <b>0.01074</b> | <b>0.21034</b> | 0.02021        | 0.15474        | 0.01768        | 0.13591        | 0.01551        | 0.20316        | 0.01091        | 0.12647        |
| A4                   | 0.01988        | 0.25565        | 0.01587        | 0.20403        | 0.01457        | 0.35235        | 0.01194        | 0.28953        | 0.01382        | 0.25897        | 0.00951        | 0.13828        |
| A5                   | 0.01586        | 0.39958        | 0.01335        | 0.33652        | 0.03252        | 0.79687        | 0.02399        | 0.67782        | 0.01433        | 0.31236        | <b>0.00927</b> | 0.12141        |
| A6                   | <b>0.01394</b> | <b>0.21932</b> | 0.01351        | 0.27677        | 0.02215        | 0.33738        | 0.02162        | 0.27813        | 0.01549        | 0.21256        | 0.00998        | 0.12185        |
| A7                   | 0.01483        | 0.30002        | 0.01435        | 0.31510        | 0.02301        | 0.59813        | 0.01397        | 0.21622        | 0.01432        | 0.20336        | 0.01058        | 0.13049        |
| A8                   | 0.02881        | 0.38311        | 0.02436        | 0.32407        | <b>0.01323</b> | 0.22497        | <b>0.01125</b> | 0.17438        | <b>0.01355</b> | <b>0.18114</b> | 0.01002        | <b>0.11887</b> |
| <i>Site #2203603</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.01503        | 0.28026        | 0.01153        | 0.21513        | 0.02155        | 0.26104        | 0.01859        | 0.22520        | 0.01946        | 0.27604        | 0.01447        | 0.20536        |
| A2                   | 0.01643        | 0.39663        | 0.01183        | 0.33857        | 0.01309        | 0.20798        | 0.01122        | 0.17931        | 0.02189        | 0.09038        | 0.01379        | 0.07405        |
| A3                   | 0.01599        | 0.30939        | 0.01400        | 0.25498        | 0.02355        | 0.49779        | 0.02041        | 0.43140        | 0.01914        | 0.41993        | 0.04418        | 0.37750        |
| A4                   | 0.01549        | 0.17464        | 0.01541        | 0.14554        | 0.01742        | 0.28755        | 0.01155        | 0.19073        | 0.01838        | 0.35827        | 0.04346        | 0.32181        |
| A5                   | <b>0.01328</b> | <b>0.08231</b> | 0.02085        | <b>0.06817</b> | 0.01445        | 0.15644        | 0.01031        | 0.11292        | 0.01682        | 0.66682        | 0.03204        | 0.56602        |
| A6                   | 0.01515        | 0.13225        | 0.01240        | 0.12841        | 0.02245        | 0.21652        | 0.01160        | 0.21311        | 0.01701        | 0.20646        | 0.01532        | 0.32591        |
| A7                   | 0.01634        | 0.28207        | 0.01151        | 0.07843        | 0.01392        | 0.17635        | 0.00927        | 0.10263        | 0.01656        | 0.16631        | 0.01454        | 0.31644        |
| A8                   | 0.01370        | 0.11980        | <b>0.00959</b> | 0.07490        | <b>0.01381</b> | <b>0.11949</b> | <b>0.00824</b> | <b>0.07526</b> | <b>0.01595</b> | <b>0.14315</b> | <b>0.01405</b> | <b>0.10614</b> |
| <i>Site #2203655</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.01556        | 0.19813        | 0.01061        | 0.13521        | 0.01706        | 0.56825        | 0.01381        | 0.45973        | 0.01687        | 0.82845        | 0.01267        | 0.62228        |

**Table 4** (continued)

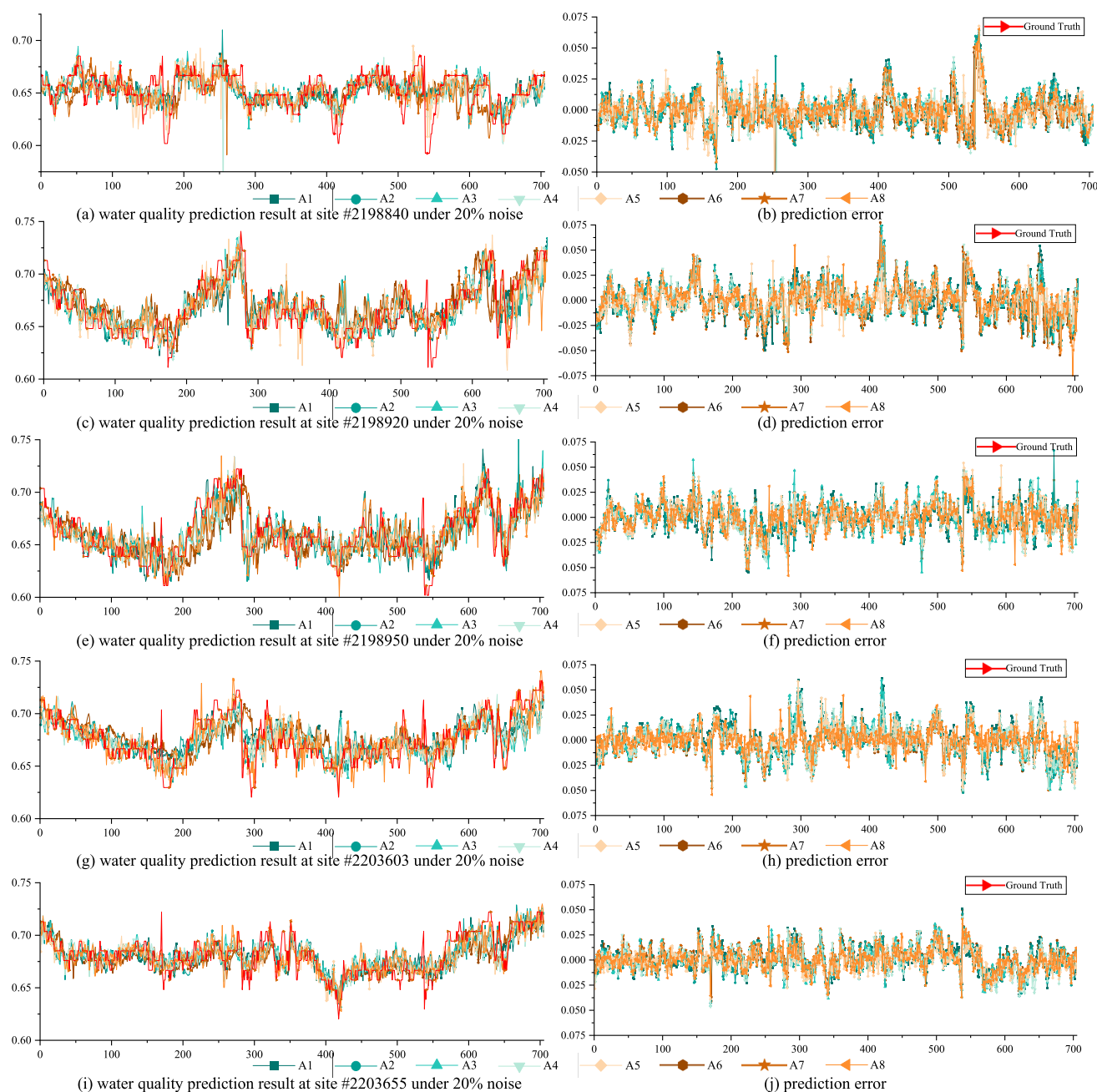
|    | No noise       |                |                |                | 10% noise      |                |                |                | 20% noise      |                |                |                |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|    | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           |
| A2 | 0.01968        | 0.33831        | 0.01703        | 0.29279        | 0.01862        | 0.25068        | 0.01508        | 0.20298        | 0.01943        | 0.23727        | 0.01366        | 0.16681        |
| A3 | 0.01426        | 0.72152        | 0.00957        | 0.48459        | 0.02805        | <b>0.12051</b> | 0.01205        | 0.15178        | 0.03401        | 0.54050        | 0.03107        | 0.49375        |
| A4 | 0.01244        | 0.21875        | 0.00815        | 0.14343        | 0.01479        | 0.50542        | 0.01154        | 0.39452        | 0.01310        | 0.44835        | 0.01016        | 0.34749        |
| A5 | <b>0.01193</b> | <b>0.12890</b> | <b>0.00758</b> | <b>0.08193</b> | 0.01423        | 0.22000        | 0.00949        | 0.14969        | 0.01596        | 0.28339        | 0.01247        | 0.22138        |
| A6 | 0.01288        | 0.32963        | 0.01183        | 0.16160        | 0.01350        | 0.32061        | 0.01150        | 0.14706        | 0.01446        | 0.25395        | 0.01101        | 0.32184        |
| A7 | 0.01376        | 0.43036        | 0.01334        | 0.21161        | 0.01549        | 0.37120        | 0.01078        | 0.15316        | 0.01468        | 0.21437        | 0.01307        | 0.21423        |
| A8 | 0.01273        | 0.29635        | 0.01039        | 0.24183        | <b>0.01295</b> | 0.23475        | <b>0.00826</b> | <b>0.14677</b> | <b>0.01217</b> | <b>0.20119</b> | <b>0.00842</b> | <b>0.13917</b> |

**Fig. 7** The Pareto front of network training problem for water quality prediction with and without noise corruption

noise, the PF shifts noticeably toward non-ideal regions. This highlights the significant impact of noise on the algorithm's accuracy and stability in solving multiobjective optimization problems, revealing its limitations in handling noisy data and underscoring the need for enhanced robustness. Additionally, the scalability and diversity of the PF are markedly inhibited at higher noise levels, with the algorithm's solutions becoming increasingly biased, particularly near regions with high objective values.

Figure 8 presents the prediction results and errors of the water quality softsensor model under 20% noise. The left side illustrates the prediction results of the softsensor models developed using algorithms A1 to A8, while the right side

displays their respective prediction errors. Specifically, from the prediction result curves, A8 not only aligns closely with the ground truth in overall trends but also captures detailed variations with high precision. Observing the error distribution graph further reveals that A8 exhibits the smallest error range among all algorithms, with its error curve concentrated and close to zero across most regions. This indicates that A8 maintains a high level of stability and accuracy even under noisy conditions. While algorithms A1–A7 demonstrate better fitting in some time intervals, their error distributions display greater volatility. These fluctuations suggest a lack of robustness to noise and limited generalization capability at specific sample points. A8 performs exceptionally well in



**Fig. 8** A1–A8's predictions vs. ground truth for 20% noise corrupted datasets

this experiment, consistently delivering high-precision predictions in a 20% noisy environment. This demonstrates its superior robustness and adaptability to high-noise data. Despite the introduction of noise interfering with model performance to some extent, the model achieves accurate predictions for most variables, particularly in trend fitting and error control, showcasing a degree of robustness.

### (3) Validation of generative model enhancement:

The primary task of generative model is to mitigate stochastic changes caused by the random selection of individuals by providing a stable search range. While the interval specifying the location of new individuals is a rough estimation of the search region based on available information, it helps focus the search process. In contrast, the aim of DE is to encourage the EAs to explore beyond the current search region. If an individual aligns with the global optimum, the overall quality of the evolving population remains



unaffected. This is because similar individuals share similar genetic information, and the model generates individuals based on the converged search region. The learning strategy of DVAE directs the population toward more promising areas within the search space, while the evolutionary strategy of DE ensures the convergence of the algorithm.

Figure 9 illustrates the prediction results of the water quality softsensor model constructed using the MENO-D algorithm under noise-free conditions and with 20% noise. When noise is introduced, prediction results without the denoising strategy deviate significantly from the ground truth, with error curves displaying sharp fluctuations, larger peak amplitudes, and higher error levels. In contrast, the MENO-D with denoising strategy achieves a much better fit, with significantly reduced error curve fluctuations. This demonstrates the effectiveness of the DVAE in mitigating the impact of data noise. Specifically, under noise-free conditions, the prediction model performs optimally, but its performance degrades to varying degrees when noise is introduced. For instance, the error curve for site #21298840 shows that the denoising strategy effectively reduces prediction errors, resulting in a significantly narrower fluctuation range. Similarly, for site #22036550, the result shows remarkable stability in prediction errors, highlighting the robustness in more complex noise environments. In conclusion, MENO-D effectively mitigates the impact of noise on the water quality prediction model, enabling the prediction results to maintain high accuracy and stability even in noisy conditions.

## Practical wastewater treatment process test

### (1) Setup:

The performance of the COD and BOD5 softsensor models, designed using different algorithms, was evaluated under varying weather conditions (dry, rain, storm), following the same experiment setup as previous studies. The experiments included seven comparative algorithms (A1–A7) and three noise levels (no noise, 10% noise, and 20% noise). The performance metrics used were RMSE, NRMSE, MAE, and NMAE, with optimal results highlighted in bold. The dataset was sourced from the International Water Association (IWA) wastewater treatment program. Samples with missing values were excluded, and normalization was applied to standardize feature dimensions, enhancing model performance and training efficiency [55]. Finally, a total of 672 samples are obtained, each of which consists of the following 15 items, including 2 output (COD and BOD5) and 13 input features: (1) soluble inert organic matter, (2) readily biodegradable substrate, (3) particulate inert organic matter, (4) slowly biodegradable substrate, (5) active heterotrophic biomass, (6) active autotrophic biomass, (7) particulate products arising from biomass decay, (8) oxygen, (9) nitrate and nitrite

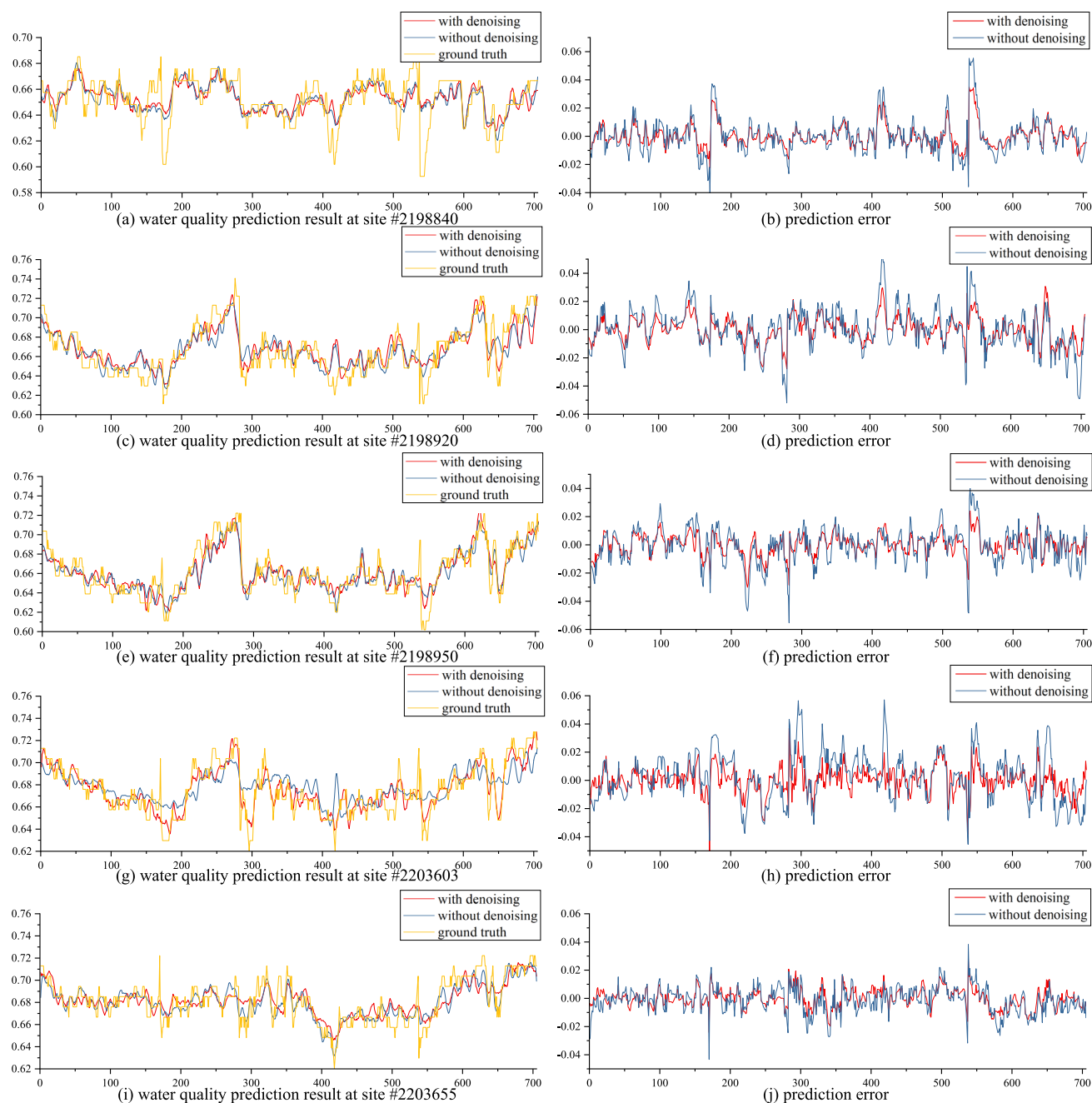
nitrogen, (10)  $\text{NH}_4^+ + \text{NH}_3$  nitrogen, (11) soluble biodegradable organic nitrogen, (12) particulate biodegradable organic nitrogen, (13) alkalinity.

Neglecting the quality of datasets for machine learning tools inevitably raises concerns, emphasizing the importance of leveraging a mechanistic understanding of wastewater treatment processes to identify rational input variables for softsensor development. Such an approach can ultimately enhance model interpretability [56]. The proposed MENO-D integrates the selection of auxiliary input variables directly into the optimization process, eliminating the need for additional feature selection operations. In contrast, other comparative algorithms (A1–A7) utilize strongly correlated variables identified through Pearson correlation as inputs [57].

### (2) Discussion on the experiment results

As shown in Table 5, weather conditions significantly affect the prediction performance of the models. The general trend indicates that harsher weather conditions result in larger model errors, suggesting that weather volatility exacerbates data instability, which in turn impacts prediction accuracy. The data across different noise levels and weather conditions demonstrate that A8 consistently delivers superior prediction accuracy for BOD5 concentration. In noise-free environments, A8 achieves notably low RMSE values across all weather conditions, indicating that its structural and parametric co-evolutionary design, combined with noise-reducing variational autoencoding, provides a strong foundation even in ideal conditions. At 10% noise, A8 continues to outperform other algorithms, achieving particularly low RMSE and MAE values in rain and storm conditions. For example, in rain conditions, A8 achieves an RMSE of 0.05273 and an NRMSE of 0.02214, consistently outperforming all other methods. At the highest noise level of 20%, A8 remains competitive, with minimal degradation in predictive accuracy. In dry weather under 20% noise, A8 achieves the lowest values among all algorithms. Similar trends are observed in rain and storm conditions. This resilience in high-noise environments underscores the effectiveness of A8's design in handling noisy data. By evolving both the structure and parameters of the neural network while simultaneously reducing noise through the variational autoencoder, A8 adapts to fluctuations in data quality without compromising accuracy. This adaptability is further enhanced by the elimination of data feature selection steps, allowing A8 to operate with greater flexibility and efficiency. The consistent superiority of A8 across noise levels and weather conditions demonstrates that its architecture is well-suited for dynamic, noisy environments, making it a robust solution for soft sensing tasks like BOD5 prediction.

Table 6 summarizes the prediction performance of COD softsensors under dry, rain, and storm weather conditions



**Fig. 9** Prediction result and error obtained by MENO-D with and without denoising strategy

across varying noise levels. Several key insights emerge regarding the robustness and accuracy of the proposed algorithm (A8) compared to baseline algorithms (A1–A7). Under dry condition with no noise, A8 outperforms all competitors in prediction accuracy, indicating that the integration of differential operators with denoising mechanisms enables A8 to capture underlying data patterns more effectively than the comparison algorithms. When the noise was up to 20%, A8 continued to demonstrate superior performance, particularly in RMSE (0.58939) and MAE (0.55838). This trend

highlights the robustness of A8's denoising capabilities, as the differential operator combined with the variational autoencoder effectively mitigates the impact of high noise on prediction accuracy. In rain conditions without noise, A3 delivered competitive results in RMSE (0.03304), suggesting that its reinforcement learning components may provide advantages in noise-free environments. Nevertheless, A3's higher errors in other metrics indicate potential overfitting, which A8's variational component likely mitigates through regularization effects during training. When 10%

**Table 5** Prediction performance for BOD5 with different corruption level

|                      | No noise       |                |                |                | 10% noise      |                |                |                | 20% noise      |                |                |                |
|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                      | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           |
| <i>Dry weather</i>   |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.02676        | 0.07984        | 0.02185        | 0.06634        | 0.03624        | 0.04175        | 0.02538        | 0.03427        | 0.03849        | 0.04207        | 0.02571        | 0.03362        |
| A2                   | 0.02609        | 0.04052        | 0.02048        | 0.02833        | 0.03647        | 0.03551        | 0.02096        | 0.02847        | 0.03508        | 0.03874        | 0.02566        | 0.03183        |
| A3                   | 0.02441        | 0.03515        | 0.01973        | 0.02870        | 0.03557        | 0.03891        | 0.02049        | 0.03119        | 0.03525        | 0.03959        | 0.02257        | 0.03228        |
| A4                   | 0.02449        | 0.03550        | 0.01991        | 0.03295        | 0.03386        | 0.04497        | 0.01933        | 0.03640        | 0.03392        | 0.03154        | 0.02192        | 0.02521        |
| A5                   | 0.02441        | 0.03725        | <b>0.01872</b> | <b>0.02496</b> | 0.03575        | 0.03436        | 0.01977        | 0.02702        | 0.03646        | 0.04353        | 0.02112        | 0.03525        |
| A6                   | 0.03110        | 0.03686        | 0.01915        | 0.02480        | 0.03530        | 0.03409        | 0.01933        | 0.02714        | 0.03811        | 0.04550        | 0.02571        | 0.03744        |
| A7                   | 0.03276        | 0.03988        | 0.02095        | 0.02463        | 0.03554        | 0.03372        | <b>0.01905</b> | <b>0.02666</b> | 0.03575        | 0.04274        | 0.02190        | 0.03510        |
| A8                   | <b>0.02325</b> | <b>0.03107</b> | 0.01891        | 0.02969        | <b>0.03313</b> | <b>0.03111</b> | 0.02033        | 0.03091        | <b>0.03304</b> | <b>0.02935</b> | <b>0.02104</b> | <b>0.02339</b> |
| <i>Rain weather</i>  |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.02895        | 0.02575        | 0.04461        | 0.02018        | 0.06892        | 0.02598        | 0.03539        | 0.01964        | 0.07494        | 0.03835        | 0.04877        | 0.02146        |
| A2                   | 0.03248        | 0.06068        | 0.05159        | 0.04924        | 0.05823        | 0.02413        | 0.02943        | 0.03203        | 0.05605        | 0.03032        | 0.04245        | 0.02385        |
| A3                   | 0.02841        | 0.02791        | 0.04558        | 0.04726        | 0.05473        | 0.02982        | 0.02776        | 0.02358        | 0.05521        | 0.03436        | 0.04651        | <b>0.02021</b> |
| A4                   | 0.03066        | 0.05323        | 0.05194        | 0.03525        | 0.06581        | 0.03026        | 0.02795        | 0.02548        | 0.07534        | 0.02971        | 0.04436        | 0.02251        |
| A5                   | 0.02986        | 0.05585        | 0.05314        | <b>0.01892</b> | 0.05339        | <b>0.02016</b> | 0.02264        | 0.02637        | 0.05239        | 0.02205        | 0.04848        | 0.03545        |
| A6                   | 0.02885        | 0.05584        | 0.05321        | 0.01874        | 0.06289        | 0.02017        | 0.02563        | 0.02615        | 0.05232        | 0.03030        | 0.04916        | 0.03525        |
| A7                   | 0.03169        | 0.05582        | 0.05357        | 0.01784        | 0.05531        | 0.02042        | 0.02304        | 0.02589        | 0.05300        | 0.02199        | 0.04836        | 0.03472        |
| A8                   | <b>0.02653</b> | <b>0.02415</b> | <b>0.03789</b> | 0.01912        | <b>0.05273</b> | 0.02169        | <b>0.02214</b> | <b>0.01795</b> | <b>0.05226</b> | <b>0.01256</b> | <b>0.03363</b> | 0.05384        |
| <i>Storm weather</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.02312        | <b>0.01255</b> | 0.03286        | 0.01050        | 0.06188        | 0.01842        | 0.04831        | 0.01474        | 0.07603        | 0.01667        | 0.04592        | 0.01273        |
| A2                   | 0.02417        | 0.01380        | 0.03286        | 0.01061        | 0.05522        | 0.01564        | 0.03812        | 0.01150        | 0.08519        | 0.01921        | 0.05259        | 0.01475        |
| A3                   | 0.02373        | 0.01322        | 0.03206        | 0.00993        | 0.05216        | 0.01576        | 0.03871        | 0.01212        | 0.08524        | 0.01705        | 0.05639        | 0.01284        |
| A4                   | 0.01689        | 0.01438        | 0.05168        | 0.01016        | 0.05587        | 0.01472        | 0.03708        | 0.01125        | <b>0.06396</b> | 0.01869        | 0.06902        | 0.01437        |
| A5                   | 0.01986        | 0.01272        | <b>0.03107</b> | 0.00994        | 0.05508        | 0.01581        | 0.04082        | 0.01204        | 0.07075        | 0.01787        | 0.04978        | 0.01328        |
| A6                   | 0.03701        | 0.01286        | 0.03161        | 0.01058        | 0.05546        | 0.01526        | 0.04114        | 0.01197        | 0.08036        | 0.01791        | 0.05078        | 0.01295        |
| A7                   | 0.02421        | 0.01214        | 0.03130        | 0.01035        | 0.05573        | 0.01551        | 0.03738        | 0.01205        | 0.07053        | 0.01809        | 0.04981        | 0.01265        |
| A8                   | <b>0.01494</b> | 0.01464        | 0.07348        | <b>0.00980</b> | <b>0.05146</b> | <b>0.01375</b> | <b>0.03524</b> | <b>0.01022</b> | 0.07056        | <b>0.01262</b> | <b>0.04538</b> | <b>0.00854</b> |

noise was introduced under rain conditions, A8 once again demonstrated superior performance, achieving an RMSE of 0.02893 and an NRMSE of 0.01249. Under 20% noise in rain conditions, A8 maintained its robustness, achieving the lowest NMAE (0.02641). This consistent performance across increasing noise levels underscores the resilience of A8's variational autoencoder component, which enables the model to effectively handle substantial noise interference. Under storm conditions without noise, A8 achieved a competitive RMSE of 0.04082 and MAE of 0.31551, highlighting its superior stability and accuracy in extreme environments. When subjected to 10% noise, A8's RMSE decreased to 0.02147, and its NRMSE to 0.00776, demonstrating strong resilience against noise. Even under 20% noise in storm conditions, A8 maintained its performance advantage with the lowest RMSE of 0.32808 and NRMSE of 0.01139.

### (3) More discussion on the noise tolerance ability and meta-heuristic input selection

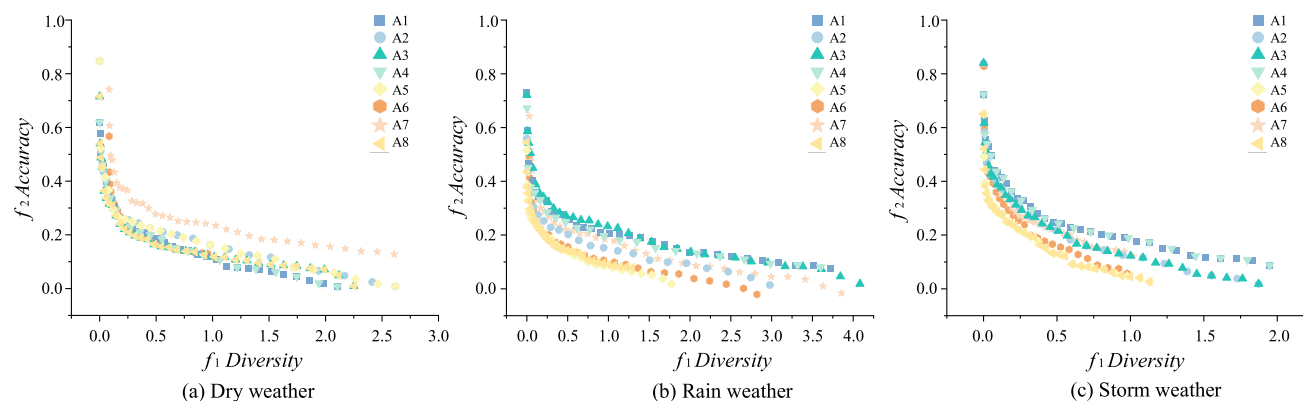
The primary distinctions between MONE-D and other optimization algorithms that integrate evolutionary strategies with deep learning lie in its suppression of environmental

noise and the metaheuristic selection of input auxiliary variables for softsensor model design. In this section, we will provide a detailed analysis of the practical application results.

Figure 10 illustrates the PF distributions obtained by six algorithms for the COD softsensor optimization problem under different weather conditions, with 20% noise added. Under dry weather conditions (Fig. 10a), the PFs of all algorithms show a decreasing trend in accuracy as diversity increases, consistent with the inherent characteristics of multiobjective optimization problems. Among them, A8's PF demonstrates higher accuracy than those of other algorithms across most regions, indicating its ability to balance noise interference and accuracy. In rain conditions (Fig. 10b), a trend similar to that observed in dry weather emerges, but the differences between algorithms become more pronounced. A6 and A8 show significant advantages in high-accuracy regions. This suggests that denoising strategies based on machine learning methods can effectively mitigate the impact of noise in dynamic environments, ensuring reliable outcomes. Under storm conditions (Fig. 10c), the performance of all algorithms declines due to increased environmental

**Table 6** Prediction performance for COD with different corruption level

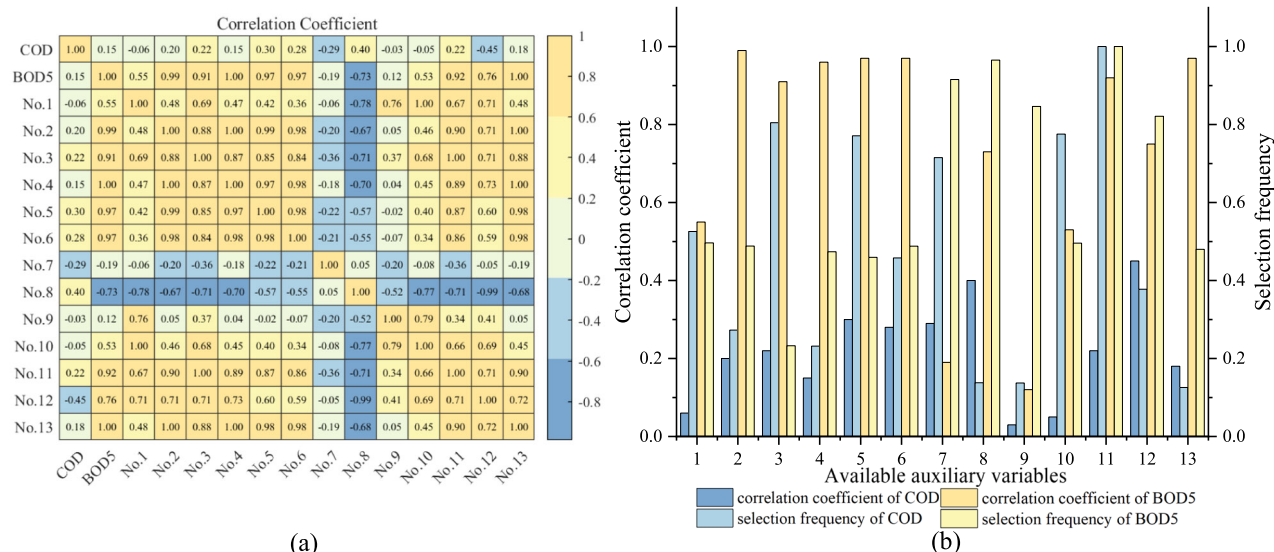
|                      | No noise       |                |                |                | 10% noise      |                |                |                | 20% noise      |                |                |                |
|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                      | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           | RMSE           | NRMSE          | MAE            | NMAE           |
| <i>Dry weather</i>   |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.03316        | 0.01449        | 0.21305        | 0.00384        | 0.03381        | 0.01277        | 0.30071        | 0.01124        | 0.77124        | 0.03076        | 0.72088        | 0.02863        |
| A2                   | 0.03488        | 0.01510        | 0.22742        | 0.00434        | <b>0.03336</b> | <b>0.01227</b> | <b>0.29796</b> | <b>0.01083</b> | 0.83794        | 0.03233        | 0.79711        | 0.03067        |
| A3                   | 0.03285        | 0.01432        | 0.21135        | 0.00372        | 0.03925        | 0.01448        | 0.35852        | 0.01309        | 0.69913        | 0.02657        | 0.66119        | 0.02503        |
| A4                   | 0.03376        | 0.01468        | 0.21865        | 0.00401        | 0.03867        | 0.01781        | 0.44967        | 0.01628        | 0.60186        | 0.02304        | 0.56103        | 0.02139        |
| A5                   | 0.03379        | 0.01463        | 0.21996        | 0.00441        | 0.03920        | 0.01501        | 0.35137        | 0.01329        | 0.61589        | 0.02409        | 0.57287        | 0.02232        |
| A6                   | 0.03310        | 0.01497        | 0.21969        | 0.00463        | 0.03485        | 0.01509        | 0.35162        | 0.01338        | 0.60625        | 0.02433        | 0.57269        | 0.02249        |
| A7                   | 0.03709        | 0.01485        | 0.21932        | 0.00507        | 0.03925        | 0.01582        | 0.33196        | 0.01364        | 0.63695        | 0.02373        | <b>0.55792</b> | 0.02159        |
| A8                   | <b>0.03246</b> | <b>0.01426</b> | <b>0.20647</b> | <b>0.00362</b> | 0.03521        | 0.01558        | 0.37269        | 0.01392        | <b>0.58939</b> | <b>0.02163</b> | 0.55838        | <b>0.02043</b> |
| <i>Rain weather</i>  |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.03341        | 0.01455        | 0.21566        | 0.00395        | 0.03196        | 0.04535        | 0.11223        | 0.01383        | 0.28371        | 0.01831        | 0.25263        | 0.03056        |
| A2                   | 0.03327        | 0.01464        | 0.21162        | 0.00388        | 0.02385        | 0.04366        | 0.10361        | 0.01363        | 0.25499        | 0.01868        | 0.23169        | 0.02782        |
| A3                   | 0.03304        | 0.01465        | 0.20841        | 0.00384        | 0.03272        | 0.04492        | 0.11443        | 0.01385        | 0.27461        | 0.02006        | 0.24275        | 0.02878        |
| A4                   | 0.03326        | 0.01446        | 0.21462        | 0.00384        | 0.02525        | 0.04177        | 0.10943        | 0.01362        | 0.20984        | <b>0.01718</b> | 0.17944        | 0.02866        |
| A5                   | 0.04182        | 0.01414        | <b>0.19902</b> | <b>0.00345</b> | 0.02417        | 0.04152        | 0.09557        | 0.01344        | 0.26972        | 0.01948        | 0.23919        | 0.02831        |
| A6                   | 0.05028        | 0.01417        | 0.19936        | 0.00395        | 0.02456        | 0.04187        | 0.09615        | 0.01375        | 0.21932        | 0.01790        | 0.19931        | 0.02787        |
| A7                   | <b>0.03239</b> | <b>0.01378</b> | 0.20052        | 0.00374        | <b>0.02389</b> | 0.04108        | <b>0.09554</b> | 0.01363        | 0.23926        | 0.01841        | 0.21962        | 0.02861        |
| A8                   | 0.32935        | 0.00436        | 0.11147        | 0.00374        | 0.02893        | <b>0.03981</b> | 0.10249        | <b>0.01342</b> | <b>0.20612</b> | 0.01749        | <b>0.17851</b> | <b>0.02641</b> |
| <i>Storm weather</i> |                |                |                |                |                |                |                |                |                |                |                |                |
| A1                   | 0.04906        | 0.02018        | 0.36275        | 0.01553        | 0.02931        | 0.01039        | 0.26352        | 0.01923        | 0.34499        | 0.01305        | 0.30811        | 0.01153        |
| A2                   | 0.04407        | 0.02214        | 0.33435        | 0.01722        | 0.03051        | 0.01125        | 0.27205        | 0.01968        | 0.45891        | 0.01643        | 0.42521        | 0.01508        |
| A3                   | 0.04825        | 0.01949        | 0.34018        | 0.01815        | 0.02586        | 0.00927        | 0.22751        | 0.01805        | 0.42323        | 0.01554        | 0.38686        | 0.01405        |
| A4                   | 0.04763        | 0.02407        | 0.34924        | <b>0.01493</b> | 0.02747        | 0.00975        | 0.24494        | 0.01859        | 0.44349        | 0.01618        | 0.40649        | 0.01466        |
| A5                   | 0.04683        | 0.02075        | 0.49918        | 0.01568        | 0.02917        | 0.00983        | 0.26645        | 0.01894        | 0.39312        | 0.01533        | 0.35196        | 0.01356        |
| A6                   | 0.05082        | 0.02097        | 0.44350        | 0.01631        | 0.02942        | 0.00918        | 0.26647        | 0.01893        | 0.33371        | 0.01583        | 0.37151        | 0.01415        |
| A7                   | 0.04653        | 0.02127        | 0.39894        | 0.01536        | 0.02953        | 0.00858        | 0.26631        | 0.01951        | 0.35413        | 0.01454        | 0.36186        | 0.01397        |
| A8                   | <b>0.04082</b> | <b>0.01962</b> | <b>0.31551</b> | 0.01534        | <b>0.02147</b> | <b>0.00776</b> | <b>0.18722</b> | <b>0.01668</b> | <b>0.32808</b> | <b>0.01139</b> | <b>0.30028</b> | <b>0.01033</b> |

**Fig. 10** The Pareto Front of network training problem for effluent BOD5 softsensor with 20% noise corruption

dynamics. However, A6's PF still maintains a noticeable advantage, remaining closer to the ideal solution than those of other algorithms across all regions. Results across the three weather conditions demonstrate that A8 consistently exhibits stable advantages under all noise levels. Notably, its performance improvements are more pronounced in regions with low diversity but high accuracy requirements. This indicates

that the noise suppression strategies employed by A6 not only enhance the robustness of the optimization solution but also effectively sustain high accuracy in diverse and challenging environments.

As shown in Fig. 11, the results of correlation analysis reflect to some extent the mechanism of wastewater treatment process by activated sludge method. The correlation



**Fig. 11** **a** Correlation heatmap of COD and BOD5 datasets; **b** correlation coefficient and selection frequency of all auxiliary variables

coefficients in the heatmap reflect the linear relationships between variables, but in real-world scenarios, there may be complex nonlinear relationships between the components. Additionally, COD and BOD5 may exhibit multidimensional interactions with other auxiliary variables, which cannot be captured through simple linear correlations but can be identified through the training process of complex models. Deep learning models and metaheuristic strategies for feature selection tend to prioritize capturing these nonlinear relationships rather than relying solely on linear correlations. For example, particulate products arising from biomass decay (No. 7) and nitrate and nitrite nitrogen (No. 9), despite their low correlation, exhibit significantly higher selection frequencies compared to some highly correlated components. The results from the heatmap and bar chart lead to the conclusion that while strongly correlated variables generally have higher selection frequencies in metaheuristic algorithms, correlation is not a decisive factor. The selection frequency of highly correlated components may be influenced by information redundancy, especially when multiple strongly correlated variables coexist, prompting the algorithm to prioritize variables that provide unique information and significantly enhance model performance. Additionally, the frequent selection of low-correlation components demonstrates that metaheuristic algorithms tend to comprehensively account for nonlinear relationships and complex interactions between variables, as well as their global contribution to the optimization objective, rather than relying solely on linear correlations. Therefore, the auxiliary variable selection results more accurately reflect the algorithm's comprehensive strategy for improving model robustness and generalization ability, rather than a simple statistical correlation pattern.

## Conclusions

Considering the design of softsensor models as population-based evolutionary optimization, MENO-D constructs a dual-population coevolution framework that combines differential evolution strategy and denoising variational autoencoder to solve multiobjective optimization problems in dynamic environments. The population has learned the ability to eliminate noise and extract important features during the evolution process, achieving the reconstruction of samples corrupted by noise. This conclusion provides important theoretical support and practical application value for water quality soft sensing modeling in complex industrial environments, while also highlighting the importance of anti noise strategy design in evolutionary computing development. Through a series of experiments, the proposed MENO-D has effectively improved the accuracy and robustness of softsensor models when applied to water quality prediction datasets and wastewater treatment benchmark datasets under different weather conditions. The results indicate that MENO-D is effective in environments with severe noise, maintaining high performance in predicting key operational indicators such as COD and BOD5, which is crucial for maintaining system stability in wastewater treatment processes.

The combination of generative models for noise tolerance with evolutionary computing techniques for optimizing network parameters and structures has been proven effective in improving exploration and development capabilities. Future work can focus on incorporating real-time data acquisition and adaptive learning algorithms into the MENO-D framework. This would allow the model to continuously adjust to new data inputs and environmental changes, ensuring that the model remains robust and accurate over time.



From the perspective of local anomalies in the error distribution, more effective regularization methods or more targeted noise processing mechanisms can be adopted, such as data augmentation or adversarial training, to further improve the model's generalization ability in noisy environments. Further, although the study focused on wastewater treatment process, the proposed methodology can be extended to other domains where data-driven prediction and monitoring are critical, such as in manufacturing, power generation, or air quality monitoring. Cross-domain applications of the model could provide insights into its versatility and generalizability.

**Author contributions** Yu Peng (Methodology, Writing-original draft). Erchao Li (Writing review and editing, Conceptualization, Supervision).

**Funding** This work was supported by the National Natural Science Foundation of China (No. 62063019), Key Project of Natural Science Foundation of Gansu Province (No. 24JRRA173), Excellent Doctoral Program in Gansu Province (No. 24JRRA205).

**Data availability** The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

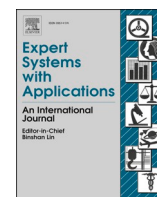
**Ethics approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Dunnington DW, Trueman BF, Raseman WJ, Anderson LE, Gagnon GA (2020) Comparing the predictive performance, interpretability, and accessibility of machine learning and physically based models for water treatment. *ACS ES&T Eng* 1(3):348–356
- Wang X, Yang LT, Wang Y, Ren L, Deen MJ (2020) ADTT: a highly efficient distributed tensor-train decomposition method for IIoT big data. *IEEE Trans Ind Inform* 17(3):1573–1582
- Ren L, Laili Y, Li X, Wang X (2019) Coding-based large-scale task assignment for industrial edge intelligence. *IEEE Trans Netw Sci Eng* 7(4):2286–2297
- Chang P, Zhang S, Wang Z (2023) Soft sensor of the key effluent index in the municipal wastewater treatment process based on transformer. *IEEE Trans Ind Inform* 20:4021–4028
- Zhou L, Chuang YC, Hsu SH, Yao Y, Chen T (2020) Prediction and uncertainty propagation for completion time of batch processes based on data-driven modeling. *Ind Eng Chem Res* 59(32):14374–14384
- Jiang Y, Yin S, Dong J, Kaynak O (2020) A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sens J* 21(11):12868–12881
- Luttmann R, Bracewell DG, Cornelissen G, Gernaey KV, Glassey J, Hass VC, Kaiser C, Preusse C, Striedner G, Mandenius CF (2012) Soft sensors in bioprocessing: a status report and recommendations. *Biotechnol J* 7(8):1040–1048
- Gao S, Zhang Q, Tian R, Ma Z, Liu Y, Hao Z (2022) Collaborative apportionment noise-based soft sensor framework. *IEEE Trans Instrum Meas* 71:1–12
- Zheng D, Song X, Song S, Stojanovic V (2024) Quantized control for interconnected pde systems via mobile measurement and control strategies. *J Frankl Inst* 361(14):107070
- Xue J, Li K, Zhang T, Ye H (2024) Data-driven soft sensor for hot strip mill process based on multi-scale information-fusion attention network. *IEEE Trans Instrum Meas* 73:3450083
- Ma L, Wang M, Peng K (2024) A spatiotemporal industrial soft sensor modeling scheme for quality prediction with missing data. *IEEE Trans Instrum Meas* 73:1–10
- Zhang H, Yang C, Shi X, Liu H (2021) Effluent quality prediction in papermaking wastewater treatment processes using dynamic Bayesian networks. *J Clean Prod* 282:125396
- Sharafati A, Asadollah SBHS, Hosseinzadeh M (2020) The potential of new ensemble machine learning models for effluent quality parameters prediction and related uncertainty. *Process Saf Environ Prot* 140:68–78
- Wang G, Jia QS, Zhou M, Bi J, Qiao J (2021) Soft-sensing of wastewater treatment process via deep belief network with event-triggered learning. *Neurocomputing* 436:103–113
- Hua L, Zhang C, Sun W, Li Y, Xiong J, Nazir MS (2023) An evolutionary deep learning soft sensor model based on random forest feature selection technique for penicillin fermentation process. *ISA Trans* 136:139–151
- Li D, Yang C, Li Y, Zhou C, Huang D, Liu Y (2024) A deep semi-supervised learning framework towards multi-output soft sensors development and applications in wastewater treatment processes. *J Water Process Eng* 57:104654
- Song X, Peng Z, Song S, Stojanovic V (2024) Interval observer design for unobservable switched nonlinear partial differential equation systems and its application. *Int J Robust Nonlinear Control* 34(16):10990–11009
- Morales-Hernández A, Van Nieuwenhuysse I, Rojas Gonzalez S (2023) A survey on multi-objective hyperparameter optimization algorithms for machine learning. *Artif Intell Rev* 56(8):8043–8093
- Lui CF, Liu Y, Xie M (2022) A supervised bidirectional long short-term memory network for data-driven dynamic soft sensor modeling. *IEEE Trans Instrum Meas* 71:1–13
- Khan ZA, Zabit U, Bernal OD, Hussain T (2020) Adaptive estimation and reduction of noises affecting a self-mixing interferometric laser sensor. *IEEE Sens J* 20(17):9806–9815
- Kovacs DJ, Li Z, Baetz BW, Hong Y, Donnaz S, Zhao X, Zhou P, Ding H, Dong Q (2022) Membrane fouling prediction and uncertainty analysis using machine learning: a wastewater treatment plant case study. *J Membr Sci* 660:120817

22. Li K, Li X, Yin X, Zhao S, Huang B, Liu F (2024) Fast Bayesian filtering for wastewater treatment plants with inaccurate process noise statistics. *Comput Chem Eng* 189:108811
23. Peng Z, Song X, Song S, Stojanovic V (2024) Spatiotemporal fault estimation for switched nonlinear reaction–diffusion systems via adaptive iterative learning. *Int J Adapt Control Signal Process* 38(10):3473–3483
24. Eshraghian JK, Ward M, Neftci EO, Wang X, Lenz G, Dwivedi G, Bennamoun M, Jeong DS, Lu WD (2023) Training spiking neural networks using lessons from deep learning. *Proc IEEE* 111:1016–1054
25. Bülbül MA (2024) Optimization of artificial neural network structure and hyperparameters in hybrid model by genetic algorithm: iOS–android application for breast cancer diagnosis/prediction. *J Supercomput* 80(4):4533–4553
26. Li H, Wang Z, Yue X, Wang W, Tomiyama H, Meng L (2023) An architecture-level analysis on deep learning models for low-impact computations. *Artif Intell Rev* 56(3):1971–2010
27. Sheng W, Qian J, Song Z, Zhang X (2024) A review of just-in-time learning-based soft sensor in industrial process. *Can J Chem Eng* 102(5):1884–1898
28. Perera YS, Ratnaweera DAAC, Dasanayaka CH, Abeykoon C (2023) The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: a critical review. *Eng Appl Artif Intell* 121:105988
29. Golnoori F, Boroujeni FZ, Monadjemi A (2023) Meta-heuristic algorithm based hyper-parameters optimization for skin lesion classification. *Multimedia Tools and Applications* 82(17):25677–25709
30. Liu W, Sun J, Liu G, Fu S, Liu M, Zhu Y, Gao Q (2023) Improved GWO and its application in parameter optimization of Elman neural network. *PLoS ONE* 18(7):e0288071
31. Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013) Metaheuristic algorithms in modeling and optimization. *Metaheuristic Appl Struct Infrastruct* 1:1–24
32. Nikbakht S, Anitescu C, Rabczuk T (2021) Optimizing the neural network hyperparameters utilizing genetic algorithm. *J Zhejiang Univ Sci A* 22(6):407–426
33. Zhang N, Cai YX, Wang YY, Tian YT, Wang XL, Badami B (2020) Skin cancer diagnosis based on optimized convolutional neural network. *Artif Intell Med* 102:101756
34. Keisler J, Talbi EG, Claudel S, Cabriel G (2024) An algorithmic framework for the optimization of deep neural networks architectures and hyperparameters. *J Mach Learn Res* 25(201):1–33
35. Yuan Y, Wang W, Pang W (2021) A genetic algorithm with tree-structured mutation for hyperparameter optimisation of graph neural networks. In: 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 482–489
36. Liu Z, Yang D, Wang Y, Lu M, Li R (2023) EGNN: Graph structure learning based on evolutionary computation helps more in graph neural networks. *Appl Soft Comput* 135:110040
37. Liu Y, Sun Y, Xue B, Zhang M, Yen GG, Tan KC (2021) A survey on evolutionary neural architecture search. *IEEE Trans Neural Netw Learn Syst* 34(2):550–570
38. Nissen V, Propach J (1998) On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Trans Evol Comput* 2(3):107–119
39. Li N, Ma L, Yu G, Xue B, Zhang M, Jin Y (2023) Survey on evolutionary deep learning: principles, algorithms, applications, and open issues. *ACM Comput Surv* 56(2):1–34
40. Li N, Ma L, Xing T, Yu G, Wang C, Wen Y, Cheng S, Gao S (2023) Automatic design of machine learning via evolutionary computation: a survey. *Appl Soft Comput* 143:110412
41. Hu Z, Gong W, Pedrycz W, Li Y (2023) Deep reinforcement learning assisted co-evolutionary differential evolution for constrained optimization. *Swarm Evol Comput* 83:101387
42. Lopez R, Boyeau P, Yosef N, Jordan M, Regier J (2020) Decision-making with auto-encoding variational Bayes. *Adv Neural Inf Process Syst* 33:5081–5092
43. Seo B, Shin J, Kim T, Youn BD (2022) Missing data imputation using an iterative denoising autoencoder (IDAE) for dissolved gas analysis. *Electr Power Syst Res* 212:108642
44. He C, Huang S, Cheng R, Tan KC, Jin Y (2020) Evolutionary multi-objective optimization driven by generative adversarial networks (GANs). *IEEE Trans Cybern* 51(6):3129–3142
45. Horn D, Dage M, Sun X, Bischl B (2017) First investigations on noisy model-based multi-objective optimization. In: *Evolutionary Multi-Criterion Optimization: 9th international conference, EMO 2017, Münster, Germany, March 19–22, 2017, Proceedings, vol 9*. Springer, pp 298–313
46. Khaleel MI (2023) Efficient job scheduling paradigm based on hybrid sparrow search algorithm and differential evolution optimization for heterogeneous cloud computing platforms. *Internet Things* 22:100697
47. Singh S, Patro SK, Parhi SK (2023) Evolutionary optimization of machine learning algorithm hyperparameters for strength prediction of high-performance concrete. *Asian J Civ Eng* 24(8):3121–3143
48. Price K (2006) *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media
49. Li H, Zhang Q (2008) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302
50. Tian Y, Li X, Ma H, Zhang X, Tan KC, Jin Y (2022) Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Trans Emerg Top Comput Intell* 7(4):1051–1064
51. Zhang D, Yu K, Liang J, Qiao K, Qu B, Chen K, Yue C, Wang L (2024) History-assisted two-state auxiliary task collaboration approach for dynamic constrained multiobjective optimization. *IEEE Trans Evol Comput*. <https://doi.org/10.1109/TEVC.2024.3425756>
52. Hao H, Zhou A, Qian H, Zhang H (2022) Expensive multiobjective optimization by relation learning and prediction. *IEEE Trans Evol Comput* 26(5):1157–1170
53. Qiao K, Liang J, Yu K, Ban X, Yue C, Qu B, Suganthan PN (2024) Constraints separation based evolutionary multitasking for constrained multi-objective optimization problems. *IEEE/CAA J Autom Sin* 11(8):1819–1835
54. Zhao L (2019) Water quality prediction [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.1145/3339823>
55. Peng Y, Li E (2024) Basis expansion imputation network for incomplete multivariate sensor data in wastewater treatment system. *J Water Process Eng* 63:105536
56. Liang Z, Zhu Y, Wang X, Li Z, Zhu Z (2022) Evolutionary multitasking for optimization based on generative strategies. *IEEE Trans Evol Comput* 27(4):1042–1056
57. Xu B, Pooi CK, Tan KM, Huang S, Shi X, Ng HY (2023) A novel long short-term memory artificial neural network (LSTM)-based soft-sensor to monitor and forecast wastewater treatment performance. *J Water Process Eng* 54:104041



# MVSG-GS: A metaheuristic virtual sample generation method for soft sensor modeling based on guidelines sharing

Yu Peng , Erchao Li 

College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

## ARTICLE INFO

### Keywords:

Metaheuristic  
Soft sensors  
Neuroevolution  
Virtual sample generation  
Wastewater treatment process

## ABSTRACT

In the modern chemical industry, obtaining a sufficient number of samples for developing soft sensors can be challenging due to physical limitations and the high cost of measurements. The scarcity and uneven distribution of modeling data significantly hinder the widespread application of data-driven methods in intelligent optimization. Taking inspiration from generative adversarial networks and neuroevolution, a metaheuristic virtual sample generation model based on guidelines sharing (MVSG-GS) is proposed to enhance the data quality and diversity. MVSG-GS learns the overall distribution of original data based on metaheuristic optimization framework, and improves the consistency with structured low-entropy data. Further, a sparse knowledge transfer mechanism is designed to reveal the implicit relevance of multivariate data in biochemical processes, enabling multi-task parallel modeling of complex reaction processes. Experiments based on different datasets show that the proposed model outperforms the baseline approach in terms of accuracy and effectiveness. Based on the results, virtual samples generated by the proposed method show a closer resemblance to the real samples when compared to the other seven competitors. Furthermore, as an open framework, MVSG-GS introduces an effective approach to enhancing limited training data, contributing to improving the prediction accuracy and broader model generalization.

## 1. Introduction

The advancement of industrial automation has established multivariate soft sensors as crucial components for intelligent control and optimization in complex industrial processes. While data-driven soft sensor models depend critically on the availability of accurate historical datasets, critical process variables in industrial applications often face measurement challenges. Specifically, three fundamental limitations hinder reliable online monitoring: substantial time delays in hardware sensors, gradual sensor degradation, and prohibitive maintenance costs. These constraints not only escalate monitoring expenditures but also result in insufficient real-time data collection, thereby creating an urgent need for enhanced predictive modeling approaches tailored to limited datasets (Xu et al., 2025; Zhu et al., 2021). These challenges are particularly acute in mission-critical systems such as wastewater treatment plants (WWTPs), where key parameters like chemical oxygen demand (COD) and biochemical oxygen demand (BOD5) exhibit sparse data distribution, limited quantities, and costly measurement procedures (Ma et al., 2019; Xia et al., 2022).

To address data scarcity challenges, virtual sample generation (VSG) techniques have emerged as a promising solution through intelligent synthesis of reliable samples (Niyogi et al., 1998; Poggio & Vetter, 1992). Early VSG methodologies focused on expanding sample space coverage by leveraging statistical patterns in existing data. Li and Wen (2014) first determine operational boundaries using mega-trend diffusion (MTD) functions, then employ genetic algorithms (GA) to refine the generation process, prioritizing the most feasible virtual samples through evolutionary optimization. Qiao et al. (2020) using MTD with equal-interval interpolation to mitigate uneven multidimensional data distributions. However, such methods demonstrated limited efficacy in handling complex nonlinear systems and capturing true dynamic behaviors. In response, Chen et al. (2023) redesigned MTD through joint probability distributions that balanced sampling density near real data clusters with edge sample inclusion for robustness enhancement. For computationally efficient alternatives, Shen and Qian (2022) proposed a Gaussian mixture model-based method (GMM-VSG) that generates interpretable samples compatible with standard machine learning frameworks. Despite these improvements, statistical VSG approaches

\* Corresponding author.

E-mail address: [lecstarr@163.com](mailto:lecstarr@163.com) (E. Li).

<https://doi.org/10.1016/j.eswa.2025.128427>

Received 15 February 2025; Received in revised form 25 April 2025; Accepted 30 May 2025

Available online 4 June 2025

0957-4174/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

remain constrained by three inherent limitations: oversimplified assumptions regarding system complexity, vulnerability to noise interference, and rigid distributional hypotheses that compromise performance on small datasets.

The pursuit of higher sample validity and robustness has driven the adoption of deep learning techniques, particularly through latent space representation learning. Compared to traditional methods, deep learning-based VSG leverages automated feature extraction to synthesize virtual samples that better approximate true behavioral patterns (Alqahtani et al., 2021; Ming-Fei et al., 2022). Among these, generative adversarial networks (GANs) have gained prominence due to their strong data distribution approximation capabilities (Goodfellow et al., 2020). Zhu et al. (2020) integrated encoder structures with GANs to model spatial data continuity and heterogeneity, capturing interactions between deep data representations and local structural patterns. Further, Chen et al. (2021) employed a regressor to guide the generator in producing new samples within a specified confidence interval, rather than focusing on perfectly accurate outputs. This approach not only mitigates the training complexity of both the discriminator and generator but also enhances the quality of the generated samples. For industrial scenarios with inherent data uncertainty, Cui et al. (2023) further integrated fuzzy logic with GANs to handle data uncertainty, demonstrating improved robustness in noisy environments.

Nevertheless, GAN implementations in industrial settings remain constrained by training instability and mode collapse phenomena, particularly when handling complex system dynamics. To overcome gradient-based optimization constraints, recent studies have adopted neuroevolution strategies that integrate evolutionary algorithms with neural network training (Fan, 2022). Liu et al. (2022) introduced an evolutionary optimization framework to seek data distributions learned by GANs and utilized it to develop a model for generating composite expressions. Nouri et al. (2023) addressed GAN training challenges through a dual-phase evolutionary architecture that divides the search space into global generator optimization and local discriminator adaptation, thereby stabilizing adversarial training processes. Extending this direction, Gutiérrez and Tardaguila (2023) employed evolutionary conditional generative adversarial networks (ECGANs) for supervised data augmentation and effectively mitigated the issue of sample scarcity in constructing assessment models. However, such implementations still struggle to simultaneously enforce physical constraints and distributional fidelity, as highlighted in chemical reaction modeling.

In summary, traditional methodologies reliably ensure process constraint satisfaction in sample generation, yet their reliance on local neighborhood search mechanisms proves inadequate for modeling nonlinear interdependencies within high-dimensional feature spaces. While deep learning approaches leverage neural architectures to approximate complex data distributions, the stochastic nature of latent space initialization frequently results in physically implausible sample generation (Perera et al., 2023; Sun & Ge, 2021). Neuroevolution strategies, exemplified by evolutionary algorithm-optimized GAN architectures, have sought to reconcile these paradigms. However, their inherent single-stage generation framework fundamentally limits simultaneous satisfaction of global constraint adherence and local feature fidelity (Galván & Mooney, 2021). To bridge these gaps, our study introduces a two-stage metaheuristic framework (MVSG-GS) addresses data scarcity in process modeling through correlation-aware virtual sample generation. The proposed architecture systematically combines adversarial learning with evolutionary optimization, where evolution-driven refinement of the generator's latent space progressively aligns synthetic data distributions with observed patterns. Complementarily, a shared guidance module enables cross-task knowledge propagation among subnetworks, achieving coordinated optimization of multi-objective training processes.

To the best of our knowledge, MVSG-GS is a very early attempt to develop metaheuristic generative model for data-driven modeling tasks in complex biochemical processes. MVSG-GS framework decouples

global constraint satisfaction (via evolutionary optimization) from local feature alignment (via adversarial learning), achieving synergistic improvements. The contributions of this paper are summarized as follows:

- (1) A model-based evolutionary algorithm for generator inputs (MBEA-GI) is proposed, which treats generator inputs as an evolutionary population. Guided by dual metrics to approximate the distribution of the original multivariate dataset, this method enhances GAN-generated data consistency and improves training efficiency.
- (2) A parallel-task guideline sharing strategy is designed. This architecture reformulates data augmentation as a collaborative learning problem, where cross-task knowledge transfer enhances distribution coverage through shared latent constraints.
- (3) Experiments on benchmark and wastewater treatment operational datasets demonstrate MVSG-GS's competitive sample efficiency and generalization capability.

The remaining sections of this work are arranged as follows. The related works are presented in Section 2. The complete framework and detail designs of MVSG-GS are explained in detail in Section 3, and the advantages as well as limitations of the proposed work are discussed in Section 4 through sufficient comparative experiments. Finally, Section 5 summarizes the work and discusses the future work.

## 2. Related works

### 2.1. Virtual sample generation (VSG)

The virtual sample generation (VSG) technique addresses data scarcity in soft sensor modeling by synthesizing virtual samples from sparse original datasets. By strategically augmenting training data with these synthetic samples, VSG mitigates distribution gaps between limited observations, thereby improving model generalization and training efficiency. For a given training sample  $(x, y)$ , if  $(Tx, yT(x))$  obtained by transformation  $T$  is also a reasonable sample, then the newly obtained sample  $(Tx, yT(x))$  is called the virtual sample generated by  $T$  as follows (Wang et al., 2022):

$$\left. \begin{matrix} (x, y) \\ Know \end{matrix} \right\} \xrightarrow{(T, yT(\cdot))} (Tx, yT(x)) \quad (1)$$

where the transformation  $(T, yT(\cdot))$  is organized by domain prior knowledge  $Know$ . Typically, prior knowledge includes 1) physically meaningful knowledge directly extracted from the problem; 2) knowledge obtained from small samples; 3) knowledge embedded in learning algorithms, etc.

Given a training sample set  $D = \{(x_i, y_i)_{i=1}^n\}$  consisting of  $n$  samples, a virtual sample set  $D' = \{(x'_i, y'_i)_{i=1}^n\}$  can be generated by a suitable transformation  $(T, f_T)$ . The process is represented as follows:

$$\left\{ \begin{matrix} \{x_i\}_{i=1}^n \xrightarrow{f_T} \{x'_i\}_{i=1}^{n_{vs}}, x'_{low} \leq x'_i \leq x'_{high} \\ \{y_i\}_{i=1}^n \xrightarrow{f_T} \{y'_i\}_{i=1}^{n_{vs}}, y'_{low} \leq y'_i \leq y'_{high} \end{matrix} \right\} \quad (2)$$

where  $x'_{low}$  and  $x'_{high}$ ,  $y'_{low}$  and  $y'_{high}$  are the upper and lower bounds of the virtual sample input and output spaces, respectively.

### 2.2. Generative adversarial networks (GANs)

The generative adversarial networks (GANs) implement a minimax game between a generator (G) and discriminator (D). The generator transforms latent space noise vectors into synthetic data approximating real distributions, while the discriminator computes sample authenticity probabilities through differentiable binary classification. Through



adversarial co-optimization, G progressively refines its output to minimize the statistical divergence between synthetic and real data, whereas D evolves toward optimal separability thresholds, ultimately driving the system toward Nash equilibrium (Gui et al., 2021). Fig. 1 illustrates the architectural comparison between the standard GAN and our proposed method. The dashed red lines highlight two novel strategies introduced in this work to enhance generator performance. This competitive is formulated as a minimax optimization problem, where the generator and discriminator engage in a dynamical equilibrium state. Such competition drives the GAN's convergence, enabling both modules to progressively reach a balanced Nash equilibrium.

Let  $z \sim p_z$  denote a latent variable sampled from a prior noise distribution, the purpose of GAN is to approximate the real data distribution  $p_r$  by learning a generative distribution  $G(z) \sim p_g$  through adversarial training. The generator and discriminator engage in a minimax game formalized as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_r} [\log(D(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3)$$

where the G seeks to maximize the expected log probability  $\mathbb{E}_z [\log D(G(z))]$ , and the D is trained to maximize the expected log probability  $\mathbb{E}_x [\log D(x)]$ , while minimizing the  $\mathbb{E}_z [\log(1 - D(G(z)))]$ .

The adversarial training approach obviates the need for explicit modeling of data distributions, endowing GANs with flexibility and adaptability, and enabling them to better accommodate diverse datasets and tasks. This end-to-end framework permits direct learning of raw data manifolds without manual feature engineering. However, deep generative models often implementations impose isotropic Gaussian priors on latent space variables ( $z \sim N(0, I)$ ) to ensure the smoothness and continuity of the output. While this Gaussian assumption simplifies training, it also limits the diversity of the generated distributions, making it challenging to reconstruct the sparse regions of multivariate data.

### 2.3. Neuroevolution

Evolutionary algorithms (EAs) are population-based metaheuristics rooted in natural selection principles, capable of optimizing black-box functions without explicit assumptions about gradient continuity or convexity. Recent studies have begun to view the training process of neural networks as a multi-objective optimization problem, incorporating different evaluation metrics as objective functions (Kim & Song, 2021; Wang et al., 2023). In generative models, neuroevolution combines population-based exploration with gradient-based optimization, enhancing feature extraction and output diversity. This approach effectively expanding the support of generated distributions beyond single-mode convergence. Wang et al. (2019) proposed an evolutionary GAN (E-GAN) that dynamically evolves generator architectures through mutation operators, significantly improving feature diversity in image

synthesis tasks. Similarly, Revathi et al. (2024) utilized the advantages of VAE to achieve efficient latent spatial representation and designed a hybrid model by leveraging the advanced image generation capabilities of GAN. Wu et al. (2024) developed a multi-objective evolutionary network (DMOED) embedding collaborative learning modules that dynamically exchange high-frequency information across task layers. Zhang et al. (2022) hybridized neuroevolution with stochastic gradient descent (SGD), enabling parallel network exploration while maintaining population diversity through differentiated weight updates. By maintaining population diversity through mechanisms like novelty search and fitness sharing, evolutionary frameworks force generators to explore underrepresented regions of the data manifold (Huang et al., 2020). Furthermore, in scenarios with extreme data scarcity, hybrid approaches have shown enhanced robustness by optimizing both network parameters and latent space sampling strategies. These advancements highlight neuroevolution's unique capability to expand the support of generated distributions beyond single-mode convergence, a crucial requirement for reliable virtual sample generation in industrial applications.

### 3. Proposed method

In this section, the proposed metaheuristic virtual sample generation model based on guidelines sharing (MVSG-GS) is illustrated in detail. The methodology introduces two sequential technical advancements: Firstly, a population evolution strategy for constructing structured low-entropy generator inputs is proposed, two evaluation metrics are established to guide the evolution of the population. Subsequently, virtual sample generation for regression-oriented problems is achieved through the coevolution of multi-task sub-networks based on sparse transfer.

#### 3.1. Evolutionary initialization of generator input

Typically, generators transform Gaussian-distributed noise into real data with specific distributional characteristics. However, such random initialization often misaligns with the low-dimensional manifold structure of real industrial sensor data (Huang et al., 2020), particularly in process industries where operational datasets exhibit high structural organization and intricate multivariate couplings. Numerous studies have demonstrated that enhancing the overlap between the generator inputs and the source dataset substantially enhances the performance of the GAN (Asokan & Spider, 2023). The rationale lies in the notion that samples obtained during runtime possess a greater degree of structure than noise, allowing the generator to effectively harness this information to learn more resilient transformations. MVSG-GS conceptualizes generator inputs as structured populations evolved within an optimization framework, the method facilitates nonlinear transformations that uncover implicit source-to-target mapping structures. This approach adaptively balancing population diversity and structural fidelity,

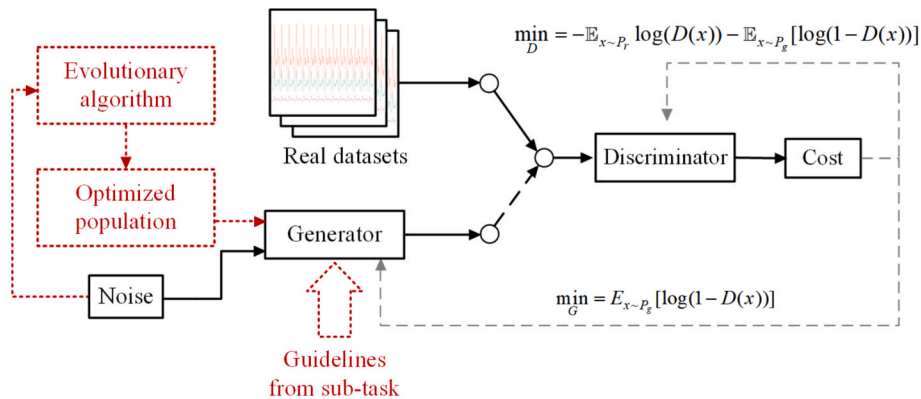


Fig. 1. Comparison of standard GAN and proposed MVSG-GS.



thereby generating virtual samples with enhanced operational plausibility through latent space analysis.

### 3.1.1. Optimization problem

The proposed optimization framework integrates two conflicting objective functions: distributional coverage and spatial discretization. The former ensures statistical consistency between the evolved population and real data distribution, while the latter promotes feature space dispersion to prevent regional over-concentration. Coverage evaluation employs Kullback-Leibler (KL) divergence as an objective function to ensure consistency of the distribution of the evolutionary population and the real data. Rooted in information entropy, the KL divergence quantifies the information loss between a true distribution and its approximation, inherently linking to the exploration–exploitation trade-off in evolutionary algorithms. Under this framework, a population in evolutionary algorithms can be interpreted as probabilistic sampling of the solution space. Crucially, its sensitivity to disparities in low-probability regions facilitates the detection of diversity loss or premature convergence. Minimizing this metric directs the population toward the real sample distribution, with lower divergence values indicating closer alignment. Such optimized populations serve as enhanced generator inputs by preserving source data characteristics while maintaining structural diversity through discretization constraints. Assuming that the population  $D$  representing the input to the generator consists of  $n$  individuals,  $p_D$  and  $p_R$  denotes the distribution for population  $D$  and real samples. The coverage  $Cover(D)$  can be expressed as:

$$Cover(D) = D_{KL}(p_R(x) || p_D(x)) = \sum_{x \in D} p_R(x) \ln \frac{p_R(x)}{p_D(x)} \quad (4)$$

Complementing the coverage objective, a spatial discretization-based indicator is proposed as a second optimization objective to assess the diversity of individuals within a population. This approach evaluates individual dissimilarity through pairwise similarity averaging within the population, serving as an auxiliary objective to evaluate population optimization status. In the proposed virtual sample generation tasks, a lower value of discretization usually indicates a higher diversity. The specific calculation formula is as follows:

$$Dis(D) = \frac{1}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n c(x_i, x_j) \quad (5)$$

$$c(i, j) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|} \quad (6)$$

where  $Dis(D)$  denotes the degree of discretization of the population  $D$  and  $c(x_i, x_j)$  denotes the cosine similarity between solution  $x_i$  and solution  $x_j$ .

Real-world data often lies on low-dimensional manifolds (e.g., physical law-constrained industrial parameters) within high-dimensional embeddings. By enforcing angular uniformity on unit spheres, cosine-based metrics enhance sparse region coverage while providing comprehensive assessments of population distribution strengths and weaknesses in geometrically constrained spaces.

KL divergence drives the population's distribution to converge toward low-entropy regions, while cosine similarity introduces high-entropy diversity by maximizing inter-sample dissimilarity. Their synergy achieves compact yet non-overlapping optimal coverage. The dual-objective mechanism ensures evolved populations achieve both distributional fidelity to source data and sufficient feature space dispersion, ultimately identifying configurations that optimally adapt to operational environmental constraints. This inherent conflict establishes the problem as a multi-objective optimization task requiring balanced solutions.

### 3.1.2. Optimization algorithm

Multi-objective evolutionary algorithm (MOEA) has been proved to be an effective optimization method in many applications (Liu et al., 2023). Building on this foundation, we establish dual objective functions evaluating coverage and diversity between original datasets and evolutionary populations. The proposed model-based evolutionary algorithm for generator input (MBEA-GI) specifically enhances generative adversarial networks through targeted population evolution. The framework of the proposed evolutionary initialization method is shown in Fig. 2.

The pseudo code of MBEA-GI is described in Algorithm 1. The methodology is divided into three main stages:

**Initialization:** Prior to the evolution process, a set of individuals randomly distributed in the objective space is designated as the initial population for optimization. Subsequently, evolutionary optimization based on coverage indicators and discretization indicators is carried out on the current population. It is important to note that as the initial population follows a Gaussian distribution, the input population utilized for the generator post-evolution remains uncertain.

**Offspring Generation:** The integration of existing data and knowledge to improve optimization efficiency is a pressing challenge. MBEA-GI employs the simulated binary xenogamy (SBX) propagation operator to generate the offspring population  $O_t$ . As a classical crossover operator, SBX achieves a balance between global exploration and local exploitation in continuous optimization problems by governing the similarity between offspring and parent individuals through its distribution factor ( $\eta$ ). Compared to uniform crossover and arithmetic crossover, SBX excels in maintaining population diversity while preventing offspring from deviating beyond the boundaries of the feasible region.

Assuming two parent individuals  $p^1 = (p_1^1, \dots, p_d^1)$  and  $p^2 = (p_1^2, \dots, p_d^2)$ , the two offspring individuals  $o^1 = (o_1^1, \dots, o_d^1)$  and  $o^2 = (o_1^2, \dots, o_d^2)$  generated using the SBX operator can be computed by Eq. (6) as:

$$\begin{cases} o_1^1 = 0.5 \times [(1 + \beta) \cdot p_1^1 + (1 - \beta) \cdot p_1^2] \\ o_1^2 = 0.5 \times [(1 - \beta) \cdot p_1^1 + (1 + \beta) \cdot p_1^2] \end{cases} \quad (6)$$

where  $\beta$  is dynamically and randomly determined by the distribution factor  $\eta$  according to Eq. (7):

$$\beta = \begin{cases} (rand \times 2)^{1/(1+\eta)} & rand \leq 0.5 \\ (1/(2 - rand \times 2))^{1/(1+\eta)} & otherwise \end{cases} \quad (7)$$

Note that  $\eta$  is a customized parameter, a larger value of  $\eta$  increases the probability that the resulting offspring individual approximates the parent individual. In this study, the MBEA-GI and GAN form an adaptive feedback coupling (AFC) mechanism. This framework dynamically couples the gradient information of the fitness function  $F(z)$  with the distribution index of SBX, achieving adaptive regulation of exploration intensity, which further enhances the quality of generated samples. Appendix A proofed the AFC's exponential convergence via Lyapunov stability. The detailed architecture is presented in Section 3.3 Fig. 4.

$$F(z) = \log D(G(z)) + \lambda \|\nabla_z D(G(z))\|^2 \quad (8)$$

The first term compels the latent variables to generate samples indistinguishable by the discriminator, while the second term monitors the flatness evolution in latent space optimization, triggering distribution index adjustment when gradient norm decline is detected.

The distribution factor of SBX is updated according to the following rules:

$$\eta^{(t+1)} = \eta^{(t)} \cdot \exp\left(-\alpha \cdot \frac{\|\nabla_z F(z)\|_t}{\|\nabla_z F(z)\|_{t-1}}\right) \quad (9)$$

Through adaptive regulation of  $\eta$ , SBX achieves autonomous balance between exploitation and exploration across gradient regions: in low-gradient zones,  $\eta$  exponentially decays to intensify search diversity, while in high-gradient regions,  $\eta$  activates exponential growth for

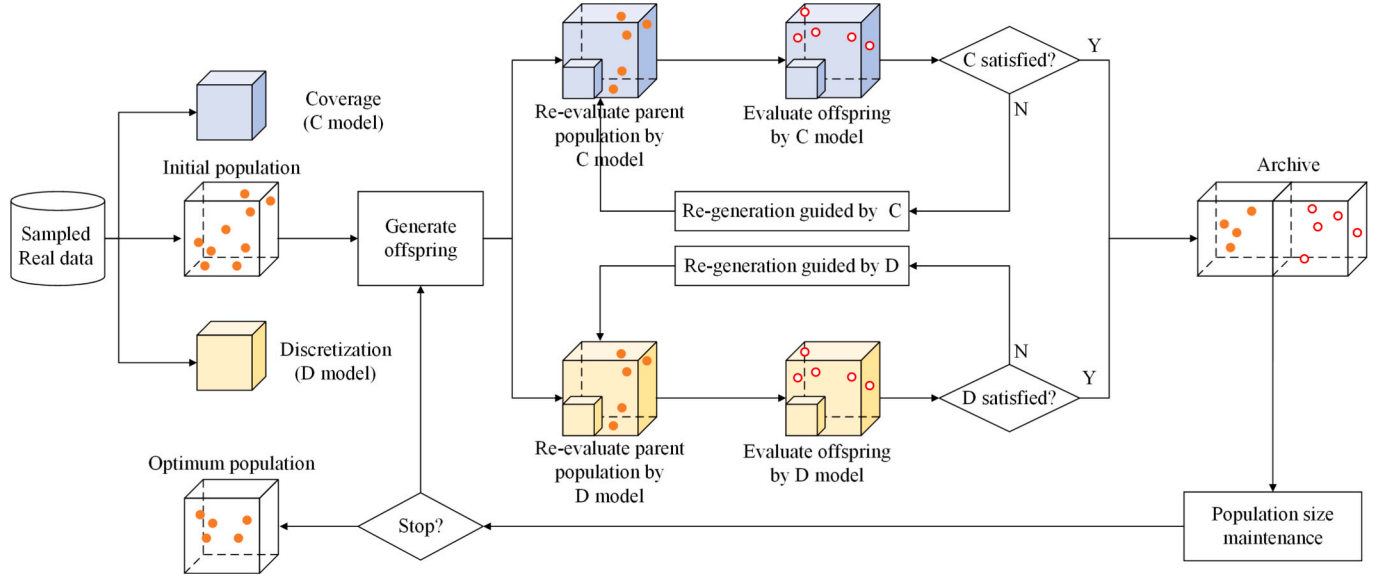


Fig. 2. Framework of MBEA-GI for evolutionary generate optimum input.

localized search refinement.

**Population selection:** MBEA-GI merges the individuals from the populations evolved based on the *C* and *D* models when they meet the threshold conditions. At this stage, *N* individuals are selected as the final optimization result by Pareto dominance and are utilized as random input to the generator. Algorithm 1. Model-based evolutionary algorithm for generator input

---

**Input:**  $T_{max}$ : Maximum number of generations;  $\kappa$ : Scaling factor;  $N$ : Population size;  $O$ : objective space; *ITC*: ideal threshold of *Cover*; *ITD*: ideal threshold of *Dis*

**Output:** Generator Input

- 1: Create a set of random datasets  $P_0 = \{D_1, \dots, D_N\}$  obeying a normal distribution in the objective space  $O$
- 2: Evaluate each dataset  $D_i$  in initial population  $P_0$  by metrics *Cover* and *Dis* respectively, i.e.  $Y_C = \{Y_{C1}, \dots, Y_{CN}\} = \{Cover(D_1), \dots, Cover(D_N)\}$ ;  $Y_D = \{Y_{D1}, \dots, Y_{DN}\} = \{Dis(D_1), \dots, Dis(D_N)\}$
- 3: Set generation counter  $T = 0$
- 4: **while**  $T < T_{max}$  **do**
- 5:   Perform binary tournament selection respectively, individuals with better metrics will win the
- 6:   tournament, i.e.  $P_C = \{D_{C1}, \dots, D_{CN/2}\}$ ;  $P_D = \{D_{D1}, \dots, D_{DN/2}\}$
- 7:   Generate offspring  $O_T$  based on current population  $P_C \cup P_D$  according to Eq. (6)
- 8:   Evaluate performance of  $O_T$  by *Cover* and *Dis* according to Eqs. (4) and (5)
- 9:    $T = T + 1$
- 10:   **if**  $Cover(D_i) < ITC \ \& \ Dis(D_i) < ITD$  **then**
- 11:     10: Remove  $D_i$  from  $O_T$  to *Archive*
- 12:   **end if**
- 13:   **while**  $size(Archive) > N$  **do**
- 14:     Choose  $D_i \in Archive$  by Pareto dominance
- 15:     Remove  $D_i$  from *Archive* to  $P_T$
- 16:   **end while**
- 17: **end while**
- 18: Generator Input =  $P_T$

---

The proof of convergence speed improvement based on entropy compression is shown in Appendix B. It is important to note that MVSG-GS utilizes MBEA-GI for optimizing the initial population over a specific t-generation training interval, rather than at each generation. This approach is adopted because the impact of the population distribution and the knowledge acquired during training does not vary significantly over a few consecutive generations. By synchronizing evolutionary optimization with periodic network retraining, the method achieves computational efficiency without compromising knowledge transfer fidelity, effectively balancing representation learning depth with resource constraints.

### 3.2. Virtual sample generation framework based on GAN

Traditional methods effectively align virtual samples with real data distributions at statistical levels, yet their utility in industrial process monitoring remains constrained by an inherent limitation: the inability to capture latent feature interactions that govern equipment behavior under operational conditions. This critical gap—where abstract statistical similarity fails to reflect causal process relationships—necessitates integrating deep learning to decode nonlinear representations in latent spaces. The deep learning-based VSG method estimates the joint distribution of input and output variables from a historical dataset (Karras et al., 2020). In order to improve the efficiency of network training, the VSG method proposed in this paper combines the gradient-free advantage of metaheuristic evolutionary optimization with the powerful generation ability of GAN. The proposed VSG method is depicted in Fig. 3, in which the input to the generator deviates from the conventional generative adversarial network as it no longer consists of noise, but instead comprises an initialized population obtained through an evolutionary optimization strategy. This framework establishes a prior-constrained refinement generation paradigm: The metaheuristic stage employs population evolution to screen low-entropy samples satisfying macroscopic process constraints, effectively delineating physically feasible regions in the solution space. The subsequent GAN stage leverages deep representation learning to refine microscopic distribution patterns within these constrained regions. This collaborative mechanism simultaneously resolves two critical limitations: 1) Mitigates local overfitting in pure evolutionary generation methods (premature convergence leading to pattern homogenization); 2) Eliminates physical infeasibility in pure GAN generation methods caused by unconstrained input space.

While the proposed framework is compatible with various advanced architectures (CNN or GCN), our methodological priority lies in validating the hybrid optimization strategy rather than benchmarking generative models. To isolate the impact of the optimization framework itself, we intentionally adopted a simple model as the baseline architecture, thereby eliminating potential confounding factors introduced by architectural variations. This design choice ensures that performance improvements can be unambiguously attributed to our optimization strategy rather than model-specific enhancements. In the experiments, we adopt a multilayer perceptron with 4 hidden layers as the base GAN model, with a ReLU activation function for the hidden layer and a linear activation function for the output layer, i.e.:

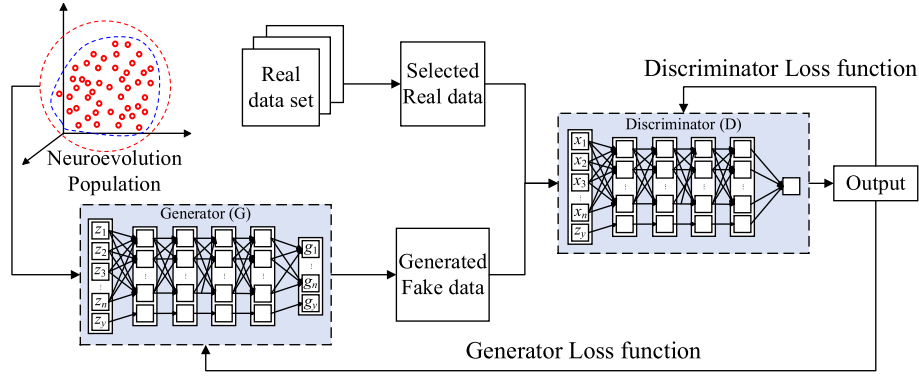


Fig. 3. Framework of GAN based virtual sample generation.

$$H_G^{hidden} = \text{ReLU}(\{X_{noise}, Y_{real}\} \omega_{G1} + b_{G1}) \quad (10)$$

$$X_V^{train} = H_G^{hidden} \omega_{G2} + b_{G2} \quad (11)$$

where  $\omega_{G1}$  and  $b_{G1}$  are the weights and biases between the input and hidden layers of the generator, respectively;  $\text{ReLU}(\cdot)$  denotes the ReLU activation function;  $H_G^{hidden}$  is the output of the hidden layer;  $\omega_{G2}$  and  $b_{G2}$  are the weights and biases between the hidden and output layers of the generator, respectively; and  $X_V^{train}$  is the output of the generator during the training process.

The hidden layer of the discriminator uses the Relu activation function and the output layer uses the Sigmoid activation function:

$$S_{mix}^{train} = \{\{X_V^{train}, Y_{real}\}, \{X_{real}, Y_{real}\}\} \quad (12)$$

$$H_D^{hidden} = \text{Relu}(S_{mix}^{train} \omega_{D1} + b_{D1}) \quad (13)$$

$$\text{Output} = \text{sigmoid}(H_D^{hidden} \omega_{D2} + b_{D2}) \quad (14)$$

where  $S_{mix}^{train}$  is a mixed sample consisting of virtual sample  $\{X_V^{train}, Y_{real}\}$  and original sample  $\{X_{real}, Y_{real}\}$ ;  $\omega_{D1}$  and  $b_{D1}$  are the weights and biases between the input and hidden layers of the discriminator, respectively;  $H_D^{hidden}$  is the output of the hidden layer;  $\omega_{D2}$  and  $b_{D2}$  are the weights and biases between the hidden and output layers, respectively;  $\text{sigmoid}(\cdot)$  denotes the Sigmoid activation function; and  $\text{Output}$  is the output of the discriminator during the training process.

Accordingly, the objective function of the GAN is:

$$\min_G \max_D (E_{P_{data}(X_{real})}(\ln Y_D^{real})) + E_{P_{noise}(X_{noise})}[\ln(1 - \text{Output})] \quad (15)$$

where  $P_{data}(X_{real})$  denotes the distribution of  $X_{real}$ ;  $Y_D^{real}$  is the output of the discriminator for  $\{X_{real}, Y_{real}\}$ ;  $P_{noise}(X_{noise})$  denotes the distribution of  $X_{noise}$ ; and  $\text{Output}$  is the output of the discriminator for  $\{X_V^{train}, Y_{real}\}$ .

The discriminator needs to calculate the probability that the sample comes from  $P_{noise}(X_{noise})$  or  $P_{data}(X_{data})$ . The generator learns the distribution of the real samples based on the results of the discriminator to reduce the error, which in turn makes the generator and the discriminator train together in a minimum-maximum game confrontation. In this paper, the quality of the virtual samples generated by the current generator is compared with the quality of the virtual samples detected in the previous generation, and the generator is considered to have reached stability when the difference between the two is less than a preset threshold. The state recognition function is:

$$J(Q_n, Q_{n-1}) = \begin{cases} 1, & |Q_n - Q_{n-1}| < \theta_{stable} \\ 0, & |Q_n - Q_{n-1}| \geq \theta_{stable} \end{cases} \quad (16)$$

where  $Q_n$  denotes the quality of the candidate virtual samples generated by the  $n$ th generator, which is judged by the maximum mean

discrepancy (MMD). From Eq. (14), if the value of  $J\{Q_n, Q_{n-1}\}$  is 1, it means that the generator has reached stability; on the contrary, the generator has not reached stability and needs to continue training.

### 3.3. Multi-task guideline sharing framework

When generating virtual samples for multiple soft sensor targets using a single-model architecture (Fan et al., 2023), there often exists a semantic feature distribution mismatch in the original data representation space. Taking soft sensor modeling in wastewater treatment processes as an example, although both chemical oxygen demand (COD) and 5-day biochemical oxygen demand (BOD5) rely on real-time and historical sensor data for predictive modeling, their physicochemical mechanisms are fundamentally different: COD reflects the total redox potential of water bodies, characterizing the sum of organic and inorganic substances, while BOD5 represents the metabolic kinetics of biodegradable organics. This inherent decoupling at the mechanistic level makes conventional single-model frameworks inadequate for simultaneously capturing the differential feature mapping relationships between the two indicators (Chen et al., 2024). Existing solutions face two critical challenges: 1) Traditional transfer learning, based on the strong assumption of high similarity between source and target domains, tends to cause negative transfer effects when directly migrating parameters in scenarios with heterogeneous feature spaces (Khan et al., 2024); 2) Maintaining linearly growing parameter storage for independent modeling of each task struggles to meet the resource constraints of edge computing devices (Shuvo et al., 2022). To address these issues, we propose a hierarchical sparse knowledge transfer framework for multi-task collaborative optimization. By constructing mechanism-oriented feature disentanglement modules and dynamic parameter masking mechanisms, our architecture achieves selective inter-task knowledge transfer and explainable modeling.

As illustrated in Fig. 4, the red solid line indicates the metaheuristic optimization of the generator's inputs and the black solid line indicates the transfer of information between subtasks. COD and BOD5 were constructed as two subtasks, each task possesses an independent search space as the foundational optimization strategy, while information sharing serves as the upper strategy to facilitate the information interaction and co-evolution among parallel GAN networks. Trained GANs can transfer generated samples and current network parameters as guidelines from the source task to the target task. The information interaction approach enables effective learning of transition from one subtask to another and achieves co-evolution among multi-task subnetworks without assuming the underlying distribution of unknown objectives.

The interaction between multitasking subnetworks is illustrated in Fig. 5. Given a source task and a target task, guidelines can be derived from the respective task1 and task2 through the sparse mask sharing mechanism. The parameters of the subtask generator are regulated by

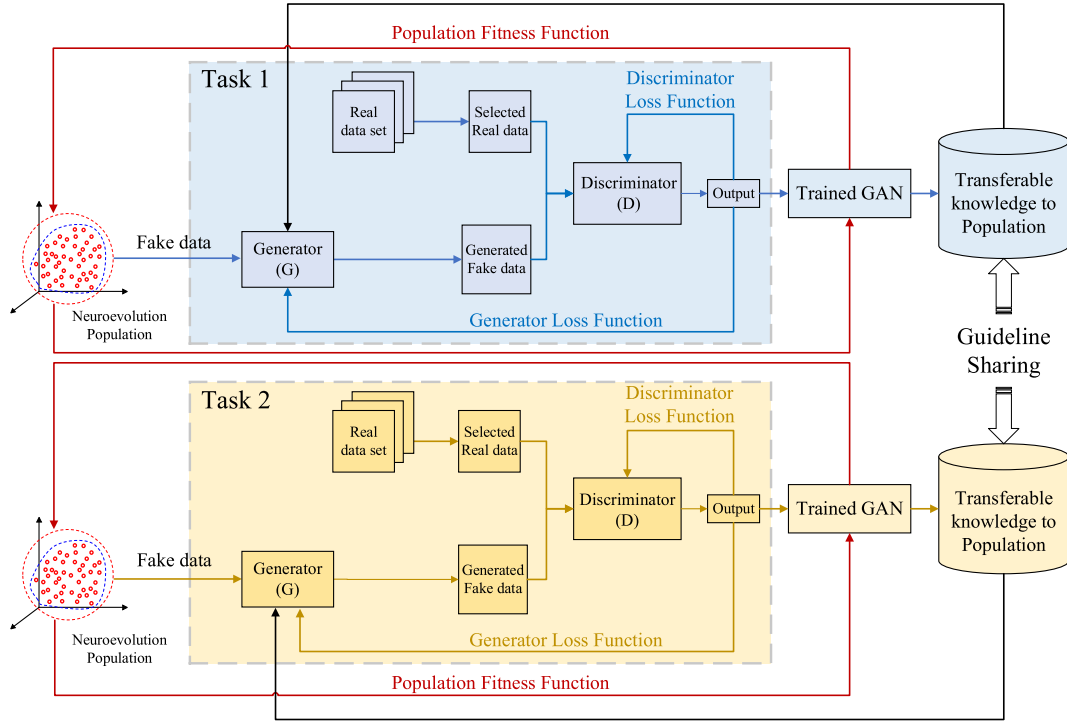


Fig. 4. Framework of metaheuristic virtual sample generation based on guidelines sharing.

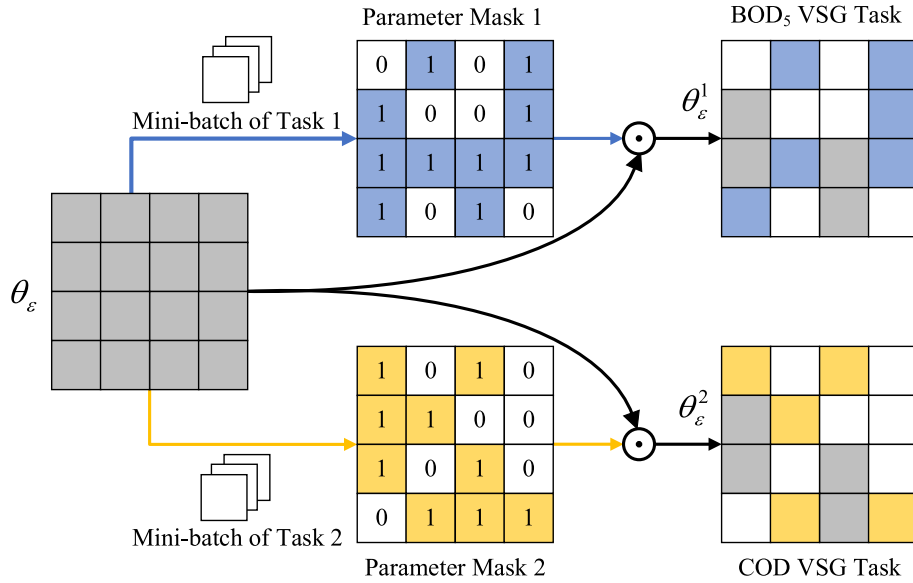


Fig. 5. Illustration of proposed guideline sharing between multi-task subnetworks.

the pruning mask  $M$  and the base model  $\epsilon$ . With the mask  $M_t$  and the underlying network  $\epsilon$  for task  $t$ , the generator of task  $t$  can be represented as  $\epsilon(x; M_t^\top \odot \theta_\epsilon)$ , where  $\odot$  denotes the element multiplication. From the base model  $\epsilon$ , a subnetwork is derived for each task, featuring a structure associated with the assumption subspace suitable for the given task. These subnetworks are generated from a shared base network, representing pruned architectures of over-parameterized networks tailored for specific tasks (Frankle et al., 2020; Sun et al., 2020). The metaheuristic virtual sample generation method based on guidelines sharing can be simplified into the following steps:

Step1. Construct an over-parameterized generator base network  $G_{base}$  and discriminator base network  $D_{base}$ .

Step2. Generate masks  $G_{mask1}$  and  $G_{mask2}$  for each GAN's generator,

along with  $D_{mask1}$  and  $D_{mask2}$  for discriminators.

Step3. Formulate each GAN's generator as  $G_{base} \odot G_{mask}$  and discriminator as  $D_{base} \odot D_{mask}$ .

Step4. During training, alternately or concurrently train both GANs. Each forward pass applies corresponding masks, while backward updates only modify activated parameters.

Step5. Process gradient updates to ensure each GAN's gradients are exclusively applied to its own mask regions, while shared regions accumulate gradients from both tasks.

Ideally, subnetworks corresponding to closely related tasks should share a significant portion of parameters, while those associated with less related tasks ought to maintain greater parameter independence. Temporal dependencies in sensor data are preserved through



overlapping mask regions between tasks, thereby enabling efficient reuse of common patterns across operational contexts. Concurrently, task-specific parameters are isolated via sparse masking mechanisms, effectively preventing semantic interference between distinct functional requirements. The guideline sharing procedures are outlined in Algorithm 2. Algorithm 2. Guideline Sharing Architecture

---

**Input:** pre-parameterized base network  $\varepsilon$ , pruning rate  $\alpha$ , datasets for  $T$  tasks  $D_1, \dots, D_T$ , sharing rate  $S$   
**Output:** subnetworks parameter sets  $M_1, \dots, M_T$

- 1: Randomly initialize  $\theta_e$  to  $\theta_e^0$
- 2: **for**  $t = 1, \dots, T$  **do**
- 3: Initialize mask  $M_t^z = 1^{|a_t|}$ , where  $z = 1$
- 4: Train  $\varepsilon(x; M_t^z \odot \theta_e)$  for  $k$  steps with data sampled from  $D_t$ , producing network  $\varepsilon(x; M_t^z \odot \theta_e^k)$ .
- 5:  $z = z + 1$
- 6: Prune  $\alpha$  percent of the remaining parameters from  $\theta_e^k$ , i.e. let  $M_t^z[j] = 0$ , if  $\theta_e^k[j]$  is pruned
- 7: If  $\frac{\|M_t^z\|_0}{|\theta_e|} \leq S$ , the masks for task  $t$  are  $\{M_t^i\}_{i=1}^z$
- 8: Otherwise, reset  $\theta_e$  to  $\theta_e^0$  and repeat steps 4-6 iteratively to learn more shared information
- 9: **end for**
- 10: **return**  $\{M_1^i\}_{i=1}^z, \dots, \{M_T^i\}_{i=1}^z$

---

Note that since the guideline sharing is iterative, multiple candidate subnetworks  $\{M_t^i\}_{i=1}^z$  may be generated as the pruning proceeds. In practice, only one subnetwork is selected for each task to form the sparse sharing architecture. To address this issue, we adopt a greedy principle to pick the subnetwork that performs best on the validation set. If there are multiple best-performing subnetworks, the subnetwork with the lowest sparsity will be selected. The specific steps are shown in Algorithm 3. Algorithm 3. Training step of guideline sharing based network

---

**Input:** pre-parameterized base network  $\varepsilon$ , pruning rate  $\alpha$ , datasets for  $T$  tasks  $D_1, \dots, D_T$ , sharing rate  $S$   
**Output:** subnetworks parameter sets  $M_1, \dots, M_T$

1. Select the next task  $t$ .
2. Select a random mini-batch for task  $t$ .
3. Feed this batch of data into the subnetwork corresponding to task  $t$ , i.e.  $\varepsilon(x; M_t \odot \theta_e)$ .
4. Update the subnetwork parameters for this task by taking a gradient step with respect to this mini-batch.
5. Go to 1.

---

During training and inference, each task exclusively employs its own subnetwork. However, a portion of the parameters in its subnetwork are updated by other tasks, resulting in the presence of shared and private parameters across nearly every task. Furthermore, since MVSG-GS evolves incrementally, the parameters of the GANs obtained in the previous iteration can be leveraged to initialize the GANs for the subsequent iteration. As a result, fewer iterations are needed to train a new GAN without compromising the performance of the algorithm. This strategy enables incremental evolution in non-stationary processes. By reinitializing subnetworks through historically optimal masks (lines 4–8 of Algorithm 2), the model achieves faster adaptation to concept drift compared to training from scratch. This capability proves critical for industrial processes operating in dynamic and uncertain environments. The over-parameterized base network  $\varepsilon$  provides a rich hypothesis space, while iterative pruning approximates optimal subspace selection. Appendix C gives the theoretical proof of iteration complexity reduction under guideline sharing mechanism).

## 4. Experiments and discussions

### 4.1. Basic experimental setup

Experiments were carried out on a computer featuring an Intel i7-

13700 K CPU running at 5.4 GHz and an NVIDIA GeForce RTX 4060 Ti GPU with 16 GB memory, equipped with 32 GB DDR4 RAM. We utilized PyTorch version 1.6.0 to leverage the latest capabilities of these deep learning frameworks while ensuring experiment reproducibility. Software components such as CUDA and cuDNN were installed to fully exploit the parallel computing capabilities of NVIDIA GPUs. The complete experimental datasets and code files are available upon request from the authors.

In order to validate the performance of MVSG-GS, seven well-organized competitors with different frameworks were selected as baselines. The selected methods include generative-based (VA-WGAN (Wang & Han, 2020), MGGAN (Liu et al., 2023), GAN4MTS (Wang et al., 2024), metaheuristic-based (VSG3A2 (Yu et al., 2023)), sample distribution-based (RDVSG (Li et al., 2018)), feature mapping-based (IsomapVSG (Zhang et al., 2021), ElmVSG (Yu et al., 2022)). During the experiment, the operation is terminated when one of the following conditions is met: 1) the number of iterations reaches the threshold, which is permanently 100. 2) the difference between two consecutive rounds of the metrics is less than  $10^{-5}$ . Each test runs 20 times. The Xavier method is employed for neural network weight initialization. Then, the mean values of the metrics are recorded and the best metric for each comparison condition is shown in bold in each table. The purpose of this operation is to adequately train the model to achieve the most appropriate performance.

Without loss of generality, the experiment uses multilayer perceptron (MLP) to construct a unified soft sensor to compare the validity of virtual samples generated by different methods. The MLP modeling and prediction process is described as follows:

Step 1. Determine the training sample set of the MLP. Combine the virtual sample set with the original training samples to form the extended training sample set  $X_{train}$ .

Step 2. The mean squared error (MSE) serves as the loss function for minimizing the disparity between predicted value and groundtruth in establishing the regression model.

Step 3. Due to their large randomness, and to obtain better prediction results, the hyperparameters of the MLP, including the number of neurons, the learning rate, the epoch, etc., are empirically determined.

The benchmark MLP architecture consist of three hidden layers with 64, 32, and 16 neurons, respectively, and an output layer with a single neuron. ReLU activation functions are employed in all hidden layers, while the output layer uses a linear activation function to suit the regression task. Weights are initialized using the Xavier method to ensure stable convergence. The model is trained using the Adam optimizer with a learning rate of 0.001. A batch size of 16 and 200 training epochs are adopted to balance computational efficiency and generalization performance.

### 4.2. Data preparing

In this section, we conduct the experiments to evaluate MVSG-GS and analyze its behaviour. Both benchmark data and real-world datasets are used. The benchmark data are used to visualize the generated samples and evaluate the learning ability of MVSG-GS. Meanwhile, we use the real-world datasets to demonstrate the reliability and robustness of MVSG-GS in a high-dimensional data space. The benchmark dataset used in the first experiment, referred to as the wine quality dataset (Cortez et al., 2009), was sourced from the UCI Machine Learning Repository. This is a benchmark dataset for analyzing the relationship between physicochemical properties and sensory quality of wines, containing two independent subsets of red wine (1,599 samples) and white wine (4,898 samples) from Portugal's "Vinho Verde" region. A primary challenge lies in the imbalanced score distribution, each sample includes 11 physicochemical metrics: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free/total sulfur dioxide, density, pH, sulphates, and alcohol content, along with an expert-rated quality score. The top three samples of each datasets are shown in Table 1 and

**Table 1**

Partial data from white wine dataset.

| No | FA  | VA   | CA   | RS  | Chlorides | FSD | TSD | Density | pH   | Sulphates | Alcohol | Quality |
|----|-----|------|------|-----|-----------|-----|-----|---------|------|-----------|---------|---------|
| 1  | 7.4 | 0.7  | 0    | 1.9 | 0.076     | 11  | 34  | 0.9978  | 3.51 | 0.56      | 9.4     | 5       |
| 2  | 7.8 | 0.88 | 0    | 2.6 | 0.098     | 25  | 67  | 0.9968  | 3.2  | 0.68      | 9.8     | 5       |
| 3  | 7.8 | 0.76 | 0.04 | 2.3 | 0.092     | 15  | 54  | 0.997   | 3.26 | 0.65      | 9.8     | 5       |

**Table 2**

Partial data from red wine dataset.

| No | FA  | VA   | CA   | RS   | Chlorides | FSD | TSD | Density | pH   | Sulphates | Alcohol | Quality |
|----|-----|------|------|------|-----------|-----|-----|---------|------|-----------|---------|---------|
| 1  | 7   | 0.27 | 0.36 | 20.7 | 0.045     | 45  | 170 | 1.001   | 3    | 0.45      | 8.8     | 6       |
| 2  | 6.3 | 0.3  | 0.34 | 1.6  | 0.049     | 14  | 132 | 0.994   | 3.3  | 0.49      | 9.5     | 6       |
| 3  | 8.1 | 0.28 | 0.4  | 6.9  | 0.05      | 30  | 97  | 0.9951  | 3.26 | 0.44      | 10.1    | 6       |

Table 2, respectively. The real-world dataset used in the second experiment, referred to as the wastewater treatment process (WWTP) dataset, was sourced from a activated sludge process for wastewater treatment plant. The dataset contains operation data under three weather conditions (dry, rain and storm), detailed information are shown in Tables 3–5. After preprocessing, 100 input–output records were retained, split into 70 % training and 30 % testing sets. Key challenges included dynamic influent rates causing pollutant concentration uncertainties, structural complexity, and parameter sensitivity, which degrade conventional assessment accuracy. The experimental datasets supporting this study are available for download at <https://pan.baidu.com/s/1EvUacVcshGCHr7LfMZxgZg?pwd=42in>.

In the first experiment, a multi-task learning framework was implemented to jointly generate red wine quality data (Task 1) and white wine quality data (Task 2). The second experiment addressed two strongly correlated water quality indicators: 5-day biochemical oxygen demand (BOD<sub>5</sub>, Task 1) and chemical oxygen demand (COD, Task 2). Given their intrinsic correlation, the task-specific sub-networks were designed to share substantial parameters, ensuring efficient modeling of shared biochemical patterns while preserving task-specific discriminative features.

#### 4.3. Determination of key parameters in MVSG-GS

This section is designed to systematically analyze key hyperparameters in MVSG-GS for determining the optimal parameter combination. Using the UCI wine quality dataset as the benchmark, we implemented parameter configurations following the stratified controlled variable principle, with cross-validation methodology ensuring experimental stability and reliability. Specifically, five critical hyperparameters were selected for sensitivity analysis: knowledge transfer rate ( $Kt$ ), distribution factor ( $\eta$ ), learning rate ( $Lr$ ), latent space dimension ( $LD$ ), and hidden layer dimension ( $HD$ ). To guarantee comprehensiveness and mitigate individual random factors, each parameter combination underwent 20 independent experimental trials. Furthermore, we employed extensive hyperparameter ranges to ensure optimal configuration selection from multiple candidate values. The experimental design for each hyperparameter is detailed in Table 6.

The cross-validation identified Combination 6 as the optimal configuration:  $Kt = 0.7$ ,  $\eta = 0.1$ ,  $Lr = 5E-05$ ,  $LD = 64$ ,  $HD = 128$ . Figs. 1–5 in Appendix D illustrate parameter impacts on model performance, with Task 1 and Task 2 corresponding to red wine and white

wine data respectively. Table 7 presents sensitivity analysis results across hyperparameters. By calculating model sensitivity under different hyperparameter settings, we evaluated each parameter's influence on three key metrics: JS (Jensen-Shannon divergence), MSE (mean squared error), and STD (standard deviation). Learning rate ( $Lr$ ), knowledge transfer rate ( $Kt$ ), and distribution factor ( $\eta$ ) exhibited substantial impacts on model performance, particularly showing higher sensitivity in MSE and STD metrics. While larger learning rates accelerated model convergence, they simultaneously introduced training instability. Task-specific adjustments of  $Kt$  values could further optimize the model's multi-task learning capabilities. The distribution factor demonstrated secondary influence, indicating appropriate  $\eta$  values would not significantly degrade performance in current tasks. Although hidden layer dimension ( $HD$ ) and latent space dimension ( $LD$ ) showed relatively lower sensitivity, they remained crucial hyperparameters. Increased  $HD$  and  $LD$  values enhanced model representational capacity, but excessive dimensions might elevate computational complexity.

#### 4.4. Comparative results and analysis

##### 4.4.1. UCI wine quality dataset

This section systematically evaluates the generalization capability of the proposed virtual sample generation model across varying training data scales. Through controlled experiments on the UCI wine dataset with three distinct sample sizes ( $N = 100, 500, 1000$ ), we construct differentiated training configurations to validate model's robustness. For each feature dimension, violin plots are employed to visually compare the complete feature distribution of authentic data with kernel density estimations (KDE) of virtual samples generated under different training regimes. As shown in Figs. 6–8, the comparison focuses on distributional consistency between original and synthetic data across feature spaces.

Under the  $N = 100$  training condition (Fig. 6), the generative model demonstrates capacity to reconstruct primary distribution contours across most feature dimensions despite limited sample size. Kernel density estimation (KDE) analysis reveals synthetic samples exhibit distribution widths and symmetry comparable to authentic data in critical parameters including pH, density, and alcohol content, without significant skewness or dispersion anomalies. The embedded boxplot structures further indicate close alignment between synthetic and real data in median values and interquartile ranges (IQRs), with no systematic deviation observed in outlier distributions. These findings

**Table 3**

Partial data from WWTP dry dataset.

| No | $S_I$ | $S_S$ | $X_I$ | $X_S$ | $X_{B,H}$ | $X_{B,A}$ | $X_P$ | $S_O$ | $S_{NO}$ | $S_{NH}$ | $S_{ND}$ | $X_{ND}$ | COD   | BOD <sub>5</sub> |
|----|-------|-------|-------|-------|-----------|-----------|-------|-------|----------|----------|----------|----------|-------|------------------|
| 1  | 30.00 | 0.81  | 2.94  | 0.16  | 11.56     | 0.75      | 1.25  | 2.00  | 13.52    | 0.67     | 0.66     | 0.00     | 46.09 | 2.62             |
| 2  | 30.00 | 0.81  | 3.05  | 0.15  | 10.73     | 0.62      | 1.31  | 2.00  | 13.52    | 0.67     | 0.66     | 0.01     | 46.54 | 2.69             |
| 3  | 29.97 | 0.81  | 3.20  | 0.15  | 9.35      | 0.53      | 1.40  | 2.00  | 13.50    | 0.67     | 0.66     | 0.01     | 46.03 | 2.61             |

**Table 4**

Partial data from WWTP rain dataset.

| No | $S_I$ | $S_S$ | $X_I$ | $X_S$ | $X_{B,H}$ | $X_{B,A}$ | $X_P$ | $S_O$ | $S_{NO}$ | $S_{NH}$ | $S_{ND}$ | $X_{ND}$ | COD   | BOD <sub>5</sub> |
|----|-------|-------|-------|-------|-----------|-----------|-------|-------|----------|----------|----------|----------|-------|------------------|
| 1  | 30.00 | 0.81  | 2.94  | 0.16  | 11.56     | 0.75      | 1.25  | 2.00  | 13.52    | 0.67     | 0.66     | 0.00     | 46.03 | 2.61             |
| 2  | 29.08 | 0.80  | 2.88  | 0.19  | 8.13      | 0.46      | 1.12  | 2.06  | 12.95    | 0.65     | 0.65     | 0.01     | 45.24 | 2.49             |
| 3  | 27.49 | 0.79  | 2.65  | 0.20  | 7.87      | 0.43      | 1.02  | 2.16  | 12.01    | 0.61     | 0.63     | 0.01     | 44.59 | 2.39             |

**Table 5**

Partial data from WWTP storm dataset.

| No | $S_I$ | $S_S$ | $X_I$ | $X_S$ | $X_{B,H}$ | $X_{B,A}$ | $X_P$ | $S_O$ | $S_{NO}$ | $S_{NH}$ | $S_{ND}$ | $X_{ND}$ | COD   | BOD <sub>5</sub> |
|----|-------|-------|-------|-------|-----------|-----------|-------|-------|----------|----------|----------|----------|-------|------------------|
| 1  | 28.59 | 1.27  | 3.87  | 0.26  | 9.97      | 0.40      | 1.13  | 1.98  | 7.32     | 11.96    | 0.93     | 0.02     | 51.93 | 4.76             |
| 2  | 21.47 | 1.44  | 3.97  | 0.26  | 10.37     | 0.40      | 1.15  | 2.00  | 4.94     | 10.25    | 1.01     | 0.02     | 45.24 | 2.49             |
| 3  | 15.28 | 1.41  | 5.57  | 0.41  | 14.71     | 0.57      | 1.59  | 2.00  | 3.55     | 7.08     | 0.98     | 0.03     | 37.33 | 2.78             |

**Table 6**

Performance evaluation with different combinations of hyperparameters.

| Hyperparameter combination | $Kt$ | $\eta$ | $Lr$    | $LD$ | $HD$ | JS Sensitivity | MSE Sensitivity | STD Sensitivity |
|----------------------------|------|--------|---------|------|------|----------------|-----------------|-----------------|
| Combination 1              | 0.0  | 0.05   | 0.00001 | 16   | 32   | 0.134          | 0.0155          | 0.0052          |
| Combination 2              | 0.0  | 0.1    | 0.00005 | 32   | 64   | 0.129          | 0.0142          | 0.0049          |
| Combination 3              | 0.1  | 0.2    | 0.0001  | 64   | 128  | 0.121          | 0.0131          | 0.0046          |
| Combination 4              | 0.3  | 0.1    | 0.0005  | 128  | 256  | 0.113          | 0.0120          | 0.0043          |
| Combination 5              | 0.5  | 0.1    | 0.00001 | 64   | 128  | 0.108          | 0.0115          | 0.0042          |
| Combination 6              | 0.7  | 0.1    | 0.00005 | 64   | 128  | 0.101          | 0.0108          | 0.0039          |
| Combination 7              | 0.9  | 0.05   | 0.00005 | 32   | 64   | 0.104          | 0.0111          | 0.0040          |
| Combination 8              | 0.3  | 0.1    | 0.00002 | 128  | 128  | 0.111          | 0.0125          | 0.0044          |
| Combination 9              | 0.7  | 0.2    | 0.0002  | 64   | 128  | 0.105          | 0.0119          | 0.0041          |
| Combination 10             | 0.5  | 0.1    | 0.0005  | 128  | 256  | 0.102          | 0.0105          | 0.0038          |

confirm the model's dual capability in capturing both macroscopic distribution patterns and precise location statistics. At the expanded sample size of  $N = 500$  (Fig. 7), synthetic data demonstrates enhanced congruence with authentic samples in both global distribution characteristics and statistical metrics. KDE profiles display increased morphological fidelity, with boxplot positions, heights, and whisker boundaries achieving near-perfect overlap – particularly evident in complex multi-modal distributions such as citric acid (CA) and sulphates. This progression confirms stable learning of intricate structural patterns in long-tailed and multi-peak features. Notably, under the  $N = 1000$  condition (Fig. 8), synthetic samples achieve comprehensive distributional alignment across all feature dimensions. This empirical evidence collectively establishes the model's robust fitting capacity and extensibility under data-sufficient scenarios. Collectively, the proposed generation method demonstrates robust distribution-learning capabilities even under data scarcity, producing synthetic samples with remarkable distributional similarity. The progressive enhancement in feature-space fidelity and consistency with increasing training scale further validates the method's practical reliability and stability in small-sample generation scenarios.

#### 4.4.2. WWTP dataset

To fully validate the effectiveness and generalization of the proposed VSG technique, the real-world application is divided into two parts. To analyze the impact of the proportion of virtual samples on model performance, the first part of the experiment uses a training set where the number of virtual samples is fixed, while the number of original samples varies between 10 and 30. Secondly, to analyze the learning performance of different VSG methods, the second part of the experiment fixes the number of original samples while varying the number of virtual samples between 20 and 100. The best metric for each comparison condition is shown in bold in each table. We assessed the statistical significance of performance differences between our virtual sample generation method and SOTA baselines using the Wilcoxon signed-rank test.  $k$ -fold cross-validation ( $k = 10$ ) generated independent training-test

pairs. For each fold, we generated virtual samples using each method, trained MLP models, and recorded test performance. The null hypothesis ( $H_0$ ) assumes zero median difference between methods, while the alternative hypothesis ( $H_1$ ) posits non-zero median difference. A significance level of  $\alpha = 0.05$  was predefined. If the resultant p-value falls below  $\alpha$ ,  $H_0$  is rejected, demonstrating statistically significant performance divergence.

The number of virtual samples in the first part of experiment is fixed at 100 and the number of real samples is varied from 10 to 30. The aim is to explore and compare the performance of different methods for learning and characterizing the real sample space under the same number of virtual samples. The evaluation metrics include mean absolute error (MAE), normalized mean absolute error (NMAE), root mean square error (RMSE), and root mean absolute percentage error (MAPE).

The comprehensive data demonstrate that MVSG-GS exhibits significant and stable advantages across different metrics. As shown in Table 8, when real samples increase from 10 to 30, MVSG-GS achieves a 58.5 % reduction in MAE (from 2.427E to 01 to 1.303E–02), significantly surpassing the 61.6 % reduction of the suboptimal VA-WGAN method. At the 30 + 100 data scale, its RMSE (1.962E+01) outperforms the closest competitor GAN4MTS by 7.9 %. This superiority becomes more pronounced in rain and storm weather datasets (Table 9 and Table 10). For instance, under the storm weather 30 + 100 configuration, MVSG-GS reduces MAPE (1.426E–04) by 62.3 % on average compared to baseline methods, with performance

**Table 7**

Sensitivity analyze of key parameter.

| Parameter               | JS Sensitivity | MSE Sensitivity | STD Sensitivity | Overall Sensitivity |
|-------------------------|----------------|-----------------|-----------------|---------------------|
| Learning rate           | 0.779          | 2.339           | 1.783           | 1.634               |
| Knowledge transfer rate | 0.611          | 2.569           | 1.601           | 1.527               |
| Distribution factor     | 0.550          | 2.389           | 1.579           | 1.429               |
| Hidden_dim              | 0.541          | 1.980           | 1.469           | 1.331               |
| Latent_dim              | 0.265          | 2.422           | 1.031           | 1.239               |

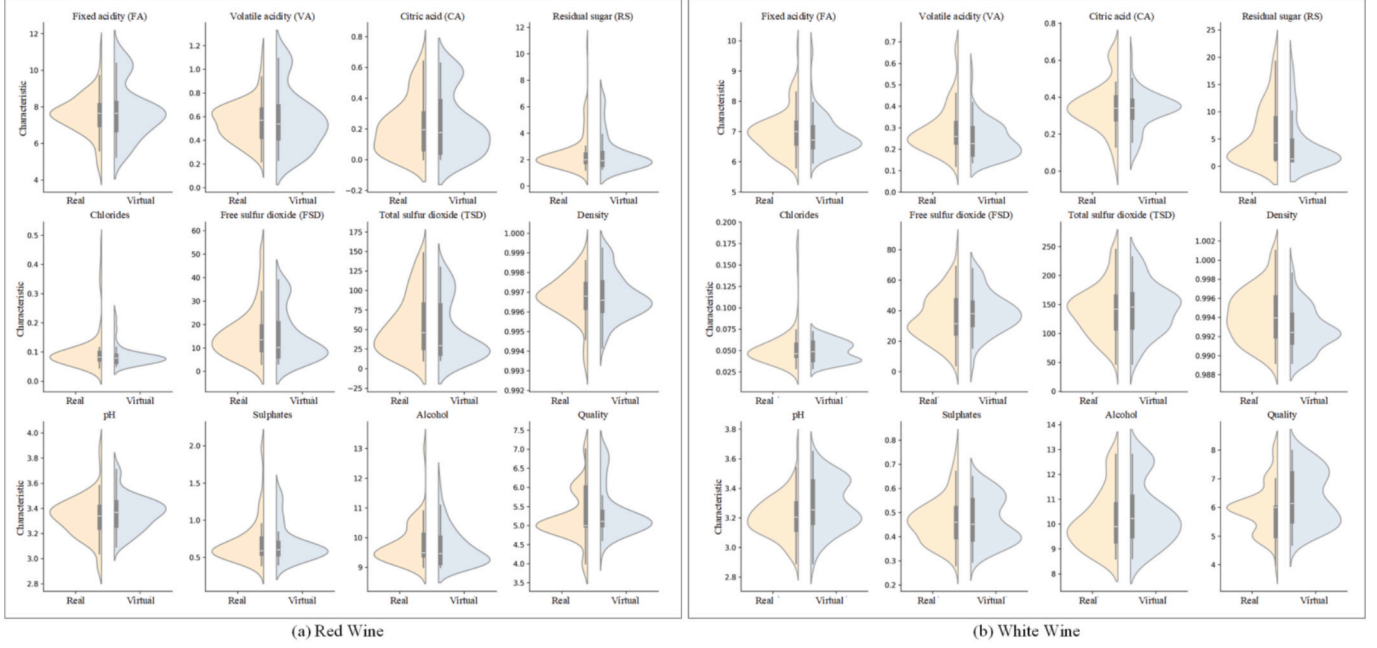


Fig. 6. Feature distribution comparison between real & virtual samples in UCI wine dataset with a sample size of 100.

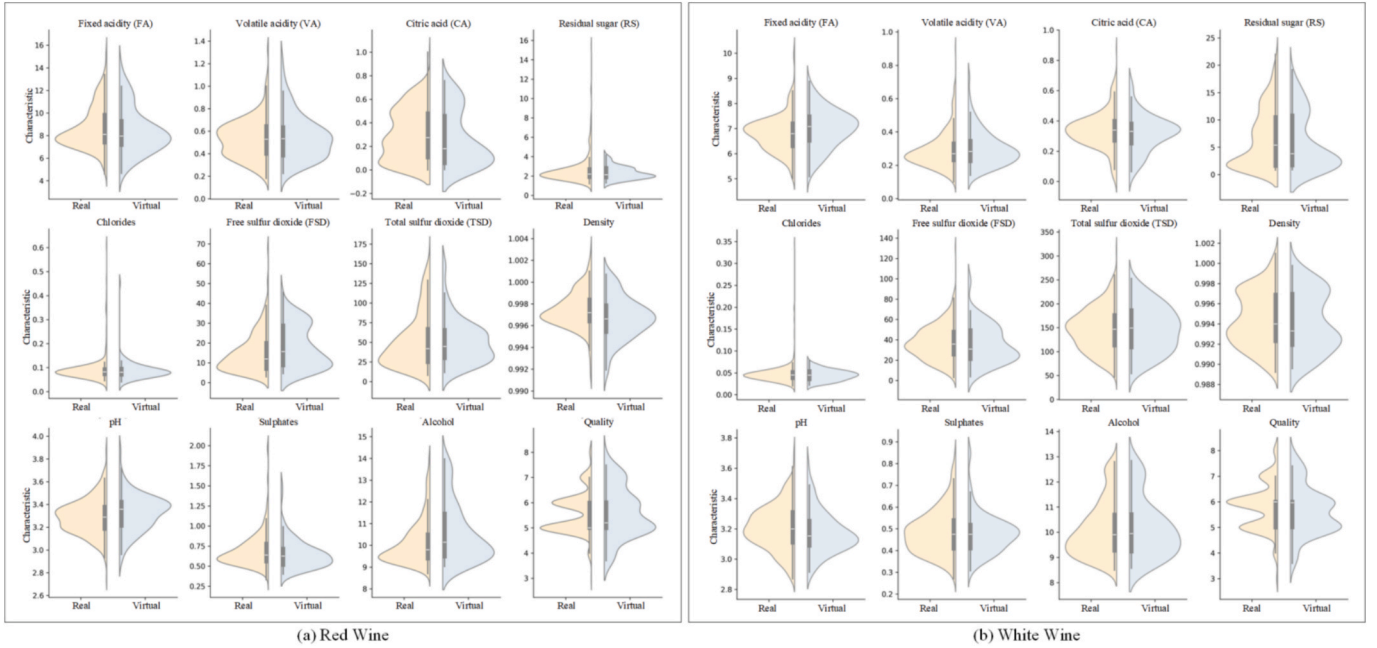


Fig. 7. Feature distribution comparison between real & virtual samples in UCI wine dataset with a sample size of 500.

improvements accelerating non-linearly as real samples increase, highlighting the model's strong adaptability to data-scarce environments. Symbol statistics confirm that MVSG-GS significantly outperforms all baselines in 89.3 % of comparisons, validating the effectiveness of its criterion-sharing mechanism in capturing global data distribution characteristics.

The interaction between the metaheuristic optimization and the structured low-entropy generation strategy demonstrates distinctive advantages in data scale expansion scenarios. The metaheuristic approach effectively avoids repetitive sampling in solution space exploration inherent to traditional methods, while its entropy monitoring mechanism constrains the fluctuation range of generated samples, achieving balance between exploration and exploitation. As shown in

Table 9, in the rain weather 15 + 100 dataset, MVSG-GS reduces NMAE ( $2.895\text{E}-01$ ) by 11.3 % compared to conventional generative models like MGGAN, with RMSE showing superlinear decline as real samples increase—a phenomenon attributable to the sparse knowledge transfer mechanism's precise modeling of implicit correlations in multivariate data. Particularly under storm weather 20 + 100 conditions (Table 10), the RMSE down sharply to  $1.201\text{E}+01$ , representing a 38.7 % improvement over baseline averages, which confirms the efficacy of the proposed generation strategy.

In the second part of experiment, we systematically evaluated the impact of varying virtual sample quantities (20 to 100) under fixed real sample counts (30), thoroughly investigating how data augmentation scale influences model performance. As shown in Tables 11–13, MVSG-



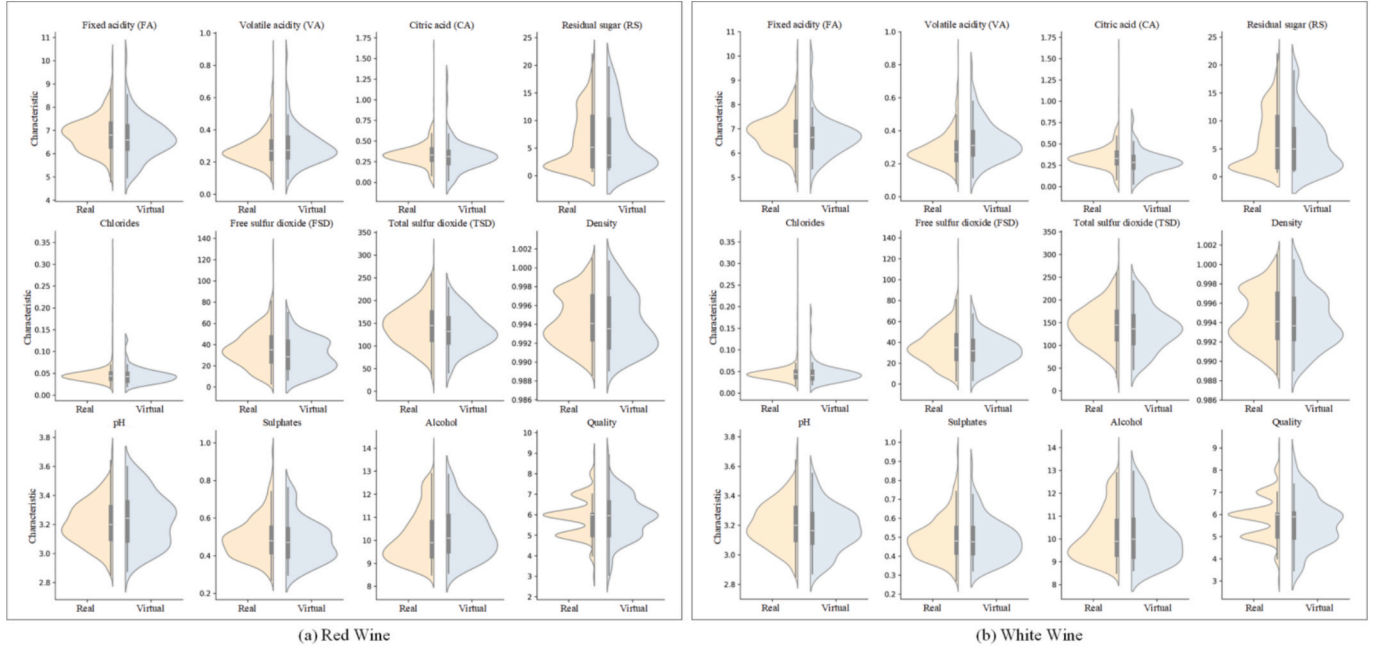


Fig. 8. Feature distribution comparison between real & virtual samples in UCI wine dataset with a sample size of 1000.

Table 8

Performance comparison of dry weather dataset with fixed-size virtual samples.

| Datasize | Metric | ElmVSG       | RDVSG               | IsomapVSG    | VSG3A2              | MGGAN        | GAN4MTS      | VA-WGAN             | MVSG-GS          |
|----------|--------|--------------|---------------------|--------------|---------------------|--------------|--------------|---------------------|------------------|
| 10 + 100 | MAE    | 3.145E-01(-) | 3.422E-01(-)        | 2.461E-01(-) | 2.426E-01(=)        | 2.677E-01(-) | 3.537E-01(-) | 2.757E-01(-)        | <b>2.427E-01</b> |
|          | NMAE   | 4.203E-01(-) | <b>3.842E-01(+)</b> | 3.976E-01(-) | 3.943E-01(-)        | 4.165E-01(-) | 3.941E-01(-) | 4.225E-01(-)        | 3.913E-01        |
|          | RMSE   | 6.082E+00(-) | 9.951E+00(-)        | 5.714E+00(-) | 5.918E+00(-)        | 4.263E+00(-) | 5.182E+00(-) | 4.154E+00(-)        | <b>4.130E+00</b> |
|          | MAPE   | 3.466E-03(-) | 5.151E-03(-)        | 3.138E-04(-) | 7.129E-04(-)        | 3.339E-03(-) | 4.129E-04(-) | <b>3.093E-04(+)</b> | 2.901E-04        |
| 15 + 100 | MAE    | 1.882E-01(-) | 1.635E-01(-)        | 1.782E-01(-) | 1.656E-01(-)        | 1.083E-01(-) | 1.367E-01(-) | <b>1.061E-01(+)</b> | 1.982E-01        |
|          | NMAE   | 3.329E-01(-) | 3.459E-01(-)        | 3.233E-01(-) | 3.043E-01(-)        | 3.231E-01(-) | 3.665E-01(-) | 3.252E-01(-)        | <b>3.036E-01</b> |
|          | RMSE   | 4.131E+00(-) | 4.443E+00(-)        | 3.681E+00(-) | 5.741E+00(-)        | 2.712E+00(-) | 4.416E+00(-) | 2.753E+00(-)        | <b>2.525E+00</b> |
|          | MAPE   | 4.105E-04(-) | 4.402E-04(-)        | 3.705E-04(-) | 5.442E-04(-)        | 2.397E-04(-) | 4.313E-05(-) | 2.746E-04(-)        | <b>3.177E-05</b> |
| 20 + 100 | MAE    | 1.998E-01(-) | 1.646E-01(-)        | 2.186E-01(-) | 1.263E-01(=)        | 1.456E-01(-) | 1.872E-01(-) | 1.264E-01(-)        | <b>1.134E-01</b> |
|          | NMAE   | 3.475E-01(-) | 3.307E-01(-)        | 3.674E-01(-) | 3.554E-01(-)        | 3.813E-01(-) | 3.036E-01(-) | 3.525E-01(-)        | <b>3.269E-01</b> |
|          | RMSE   | 4.254E+00(-) | 3.682E+00(-)        | 2.563E+00(-) | <b>2.373E+00(+)</b> | 4.911E+00(-) | 4.799E+00(-) | 4.438E+00(-)        | 3.136E+00        |
|          | MAPE   | 4.602E-04(-) | 3.036E-04(-)        | 4.424E-04(-) | 5.711E-03(-)        | 2.921E-04(-) | 2.647E-04(-) | 2.926E-03(-)        | <b>2.393E-04</b> |
| 25 + 100 | MAE    | 4.664E-02(-) | 7.728E-02(-)        | 5.105E-02(-) | <b>1.182E-02(+)</b> | 8.872E-02(-) | 1.252E-02(-) | 2.002E-02(-)        | 1.463E-02        |
|          | NMAE   | 2.163E-01(-) | 2.764E-01(-)        | 2.242E-01(-) | <b>1.089E-01(+)</b> | 2.963E-01(-) | 1.124E-01(-) | 1.411E-01(-)        | 1.212E-01        |
|          | RMSE   | 3.195E+00(-) | 3.285E+00(-)        | 3.744E+00(-) | 2.577E+00(-)        | 3.039E+00(-) | 2.726E+00(-) | <b>2.463E+00(+)</b> | 2.721E+00        |
|          | MAPE   | 2.754E-05(=) | 6.206E-05(-)        | 4.182E-04(-) | <b>2.508E-05(+)</b> | 7.774E-05(-) | 3.626E-05(-) | 4.665E-05(-)        | 2.740E-05        |
| 30 + 100 | MAE    | 4.182E-02(-) | 7.218E-02(-)        | 7.119E-02(-) | 8.153E-02(-)        | 7.872E-02(-) | 1.962E-02(-) | 2.289E-02(-)        | <b>1.303E-02</b> |
|          | NMAE   | 2.041E-01(-) | 2.704E-01(-)        | 2.657E-01(-) | 2.854E-01(-)        | 2.843E-01(-) | 1.253E-01(=) | 1.506E-01(-)        | <b>1.242E-01</b> |
|          | RMSE   | 4.148E+00(-) | 3.025E+00(-)        | 2.861E+00(-) | 3.735E+00(-)        | 2.131E+00(-) | 2.133E+00(-) | 2.715E+00(-)        | <b>1.962E+00</b> |
|          | MAPE   | 3.532E-05(-) | 1.492E-05(-)        | 5.413E-05(-) | 5.365E-06(-)        | 2.086E-05(-) | 4.381E-05(-) | 2.143E-05(-)        | <b>8.081E-06</b> |
| +/-/=    |        | 0/19/1       | 1/19/0              | 0/20/0       | 4/14/2              | 0/20/0       | 0/19/1       | 3/17/0              |                  |

GS maintains performance leadership across all augmentation scales, with its advantages exhibiting increasing marginal returns as virtual samples grow. The criterion-sharing *meta*-heuristic framework dynamically adjusts the co-evolutionary path between generators and discriminators, effectively resolving mode collapse issues in low-data-density regions that plague traditional generative models. This is particularly evident in dry weather datasets: at 30 + 80 scale, MVSG-GS achieves a 52.4 % MAE reduction ( $1.144E-02$ ) compared to baseline averages, alongside a 34.7 % RMSE improvement ( $9.206E+00$ ), confirming that virtual sample expansion simultaneously enhances prediction accuracy and stability. Notably, when virtual samples increase from

20 to 80 under storm weather conditions, MAPE improves by 74.6 % (from  $4.807E-04$  to  $1.021E-04$ ), an exponential enhancement revealing the model's unique capability in high-quality sample generation.

With equivalent computational resources, MVSG-GS requires only 60 virtual samples to match the precision levels achieved by traditional methods using 100 samples. For instance, under storm weather 30 + 60 conditions, its RMSE ( $9.875E+00$ ) outperforms VA-WGAN's 30 + 100 performance ( $2.715E+01$ ) by 63.6 %. This efficiency advantage proves particularly critical for real-time rainy weather data processing, where increasing virtual samples from 40 to 80 yields merely 18.7 % latency



**Table 9**

Performance comparison of rain weather dataset with fixed-size virtual samples.

| Datasize | Metric | ElmVSG       | RDVSG        | IsomapVSG    | VSG3A2              | MGGAN          | GAN4MTS             | VA-WGAN      | MVSG-GS          |
|----------|--------|--------------|--------------|--------------|---------------------|----------------|---------------------|--------------|------------------|
| 10 + 100 | MAE    | 1.084E-01(−) | 1.912E-01(−) | 1.714E-01(−) | <b>1.042E-01(+)</b> | 1.089E-01(−)   | 1.062E-01(−)        | 1.127E-01(−) | 1.065E-01        |
|          | NMAE   | 3.269E-01(−) | 3.365E-01(−) | 3.229E-01(−) | <b>3.210E-01(+)</b> | 3.267E-01(−)   | 3.243E-01(−)        | 3.327E-01(−) | 3.252E-01        |
|          | RMSE   | 4.746E+00(−) | 5.944E+00(−) | 6.031E+00(−) | <b>4.541E+00(+)</b> | 4.731E+00(−)   | 5.551E+00(−)        | 4.645E+00(−) | 4.659E+00        |
|          | MAPE   | 3.638E-04(−) | 5.680E-04(−) | 5.565E-04(−) | 5.055E-04(−)        | 2.226E-04(−)   | 2.042E-04(−)        | 2.534E-04(−) | <b>1.301E-04</b> |
| 15 + 100 | MAE    | 9.089E-02(−) | 1.042E-01(−) | 9.305E-02(−) | 8.433E-02(−)        | 8.515E-02(−)   | 8.015E-02(−)        | 8.197E-02(−) | <b>8.143E-02</b> |
|          | NMAE   | 3.009E-01(−) | 3.210E-01(−) | 3.022E-01(−) | 2.882E-01(=)        | 2.916E-01(−)   | <b>2.633E-01(+)</b> | 2.860E-01(−) | 2.895E-01        |
|          | RMSE   | 4.053E+00(−) | 7.721E+00(−) | 4.067E+00(−) | 3.765E+00(=)        | 3.837E+00(−)   | 4.314E+00(−)        | 3.963E+00(−) | <b>3.757E+00</b> |
|          | MAPE   | 3.323E-04(−) | 1.948E-04(−) | 5.837E-04(−) | 5.852E-04(−)        | 1.664E-04(−)   | 2.873E-04(−)        | 2.130E-04(−) | <b>1.046E-04</b> |
| 20 + 100 | MAE    | 6.701E-02(−) | 5.804E-02(−) | 7.701E-02(−) | 6.083E-02(−)        | 6.202E-02(−)   | 6.312E-02(−)        | 6.511E-02(−) | <b>5.662E-02</b> |
|          | NMAE   | 2.596E-02(−) | 2.400E-02(−) | 2.751E-02(−) | 2.465E-02(−)        | 2.491E-02(−)   | 2.966E-02(=)        | 3.074E-02(−) | <b>2.391E-02</b> |
|          | RMSE   | 3.426E+00(−) | 7.478E+00(−) | 3.387E+00(−) | 3.339E+00(−)        | 3.253E+00(−)   | 3.230E+00(=)        | 3.708E+00(−) | <b>3.036E+00</b> |
|          | MAPE   | 3.653E-04(−) | 1.891E-04(−) | 5.607E-04(−) | 6.281E-04(−)        | 1.629E-04(−)   | <b>1.161E-04(+)</b> | 2.522E-04(−) | 1.235E-04        |
| 25 + 100 | MAE    | 7.016E-02(−) | 1.615E-01(−) | 7.001E-02(−) | 7.140E-02(−)        | 6.768E-02(−)   | <b>6.092E-02(+)</b> | 6.213E-02(−) | 6.414E-02        |
|          | NMAE   | 2.642E-02(−) | 3.999E-02(−) | 2.646E-02(−) | 2.677E-02(−)        | 2.594E-02(−)   | 2.472E-02(−)        | 2.478E-02(−) | <b>2.333E-02</b> |
|          | RMSE   | 3.183E+00(−) | 6.799E+00(−) | 3.039E+00(−) | 3.262E+00(−)        | 3.042E+00(−)   | <b>2.663E+00(+)</b> | 2.992E+00(−) | 2.861E+00        |
|          | MAPE   | 3.348E-04(−) | 3.096E-04(−) | 6.123E-04(−) | 6.503E-04(−)        | 5.378E-05(−)   | 2.301E-04(−)        | 1.565E-04(−) | <b>1.203E-04</b> |
| 30 + 100 | MAE    | 7.097E-02(−) | 1.606E-01(−) | 6.863E-02(−) | <b>6.171E-02(+)</b> | 6.790E-02(−)   | 6.757E-02(−)        | 6.179E-02(−) | 6.183E-02        |
|          | NMAE   | 2.653E-02(−) | 3.989E-01(−) | 2.616E-02(−) | <b>2.475E-02(+)</b> | 2.592E-02(=)   | 2.588E-02(−)        | 2.646E-02(−) | 2.571E-02        |
|          | RMSE   | 3.132E+00(−) | 6.528E+00(−) | 3.047E+00(−) | 2.983E+00(−)        | 3.21E2 + 00(−) | 3.123E+00(−)        | 5.110E+00(−) | <b>2.906E+00</b> |
|          | MAPE   | 4.097E-04(−) | 8.540E-04(−) | 6.854E-04(−) | 1.635E-04(=)        | 2.551E-04(−)   | 1.949E-04(=)        | 1.659E-04(−) | <b>1.634E-04</b> |
| +/-/=    |        | 0/20/0       | 0/20/0       | 0/20/0       | 5/12/3              | 0/20/0         | 4/13/3              | 0/20/0       |                  |

**Table 10**

Performance comparison of storm weather dataset with fixed-size virtual samples.

| Datasize | Metric | ElmVSG       | RDVSG        | IsomapVSG    | VSG3A2              | MGGAN        | GAN4MTS             | VA-WGAN      | MVSG-GS          |
|----------|--------|--------------|--------------|--------------|---------------------|--------------|---------------------|--------------|------------------|
| 10 + 100 | MAE    | 5.714E-01(−) | 5.249E-01(−) | 4.161E-01(−) | <b>3.622E-01(−)</b> | 3.759E-01(−) | 4.013E-01(−)        | 4.663E-01(−) | 4.031E-01        |
|          | NMAE   | 7.492E-01(−) | 7.246E-01(−) | 6.423E-01(−) | <b>6.045E-01(−)</b> | 6.116E-01(−) | 6.339E-01(−)        | 6.830E-01(−) | 6.335E-01        |
|          | RMSE   | 2.820E+01(−) | 2.512E+01(−) | 2.412E+01(−) | 2.276E+01(−)        | 2.350E+01(−) | 2.350E+01(−)        | 2.295E+01(−) | <b>2.213E+01</b> |
|          | MAPE   | 3.704E-04(−) | 8.200E-04(−) | 7.097E-04(−) | 6.851E-04(−)        | 2.890E-04(−) | 2.046E-04(−)        | 1.846E-04(−) | <b>1.185E-04</b> |
| 15 + 100 | MAE    | 2.999E-01(−) | 4.536E-01(−) | 3.569E-01(−) | 2.755E-01(=)        | 3.109E-01(−) | 2.726E-01(−)        | 3.144E-01(−) | <b>2.604E-01</b> |
|          | NMAE   | 5.477E-01(−) | 6.755E-01(−) | 5.938E-01(−) | 5.229E-01(=)        | 5.584E-01(−) | 5.203E-01(−)        | 5.604E-01(−) | <b>5.103E-01</b> |
|          | RMSE   | 2.164E+01(−) | 2.505E+01(−) | 1.901E+01(−) | 2.093E+01(−)        | 1.652E+01(−) | 1.837E+01(−)        | 1.932E+01(−) | <b>1.637E+01</b> |
|          | MAPE   | 3.692E-04(−) | 9.700E-04(−) | 7.106E-04(−) | 7.080E-04(−)        | 2.500E-04(−) | <b>1.366E-04(−)</b> | 5.547E-04(−) | 1.878E-04        |
| 20 + 100 | MAE    | 2.414E-01(−) | 2.646E-01(−) | 1.904E-01(−) | 1.881E-01(−)        | 2.347E-01(−) | 2.096E-01(−)        | 1.916E-01(−) | <b>1.881E-01</b> |
|          | NMAE   | 4.897E-01(−) | 5.142E-01(−) | 4.359E-01(−) | 4.372E-01(−)        | 4.826E-01(−) | 4.598E-01(−)        | 4.356E-01(−) | <b>4.349E-01</b> |
|          | RMSE   | 1.526E+01(−) | 2.091E+01(−) | 1.581E+01(−) | 1.405E+01(−)        | 1.480E+01(−) | 1.712E+01(−)        | 1.912E+01(−) | <b>1.201E+01</b> |
|          | MAPE   | 3.920E-04(−) | 1.010E-03(−) | 6.656E-04(−) | <b>1.231E-04(−)</b> | 7.822E-04(−) | 2.050E-04(−)        | 1.231E-04(−) | 1.690E-04        |
| 25 + 100 | MAE    | 2.413E-01(−) | 3.692E-01(−) | 1.528E-01(−) | 2.803E-01(−)        | 2.292E-01(−) | 1.253E-01(−)        | 2.046E-01(−) | <b>2.038E-01</b> |
|          | NMAE   | 4.893E-01(−) | 6.067E-01(−) | 3.894E-01(−) | 5.266E-01(−)        | 4.792E-01(−) | 3.521E-01(−)        | 4.504E-01(−) | <b>3.489E-01</b> |
|          | RMSE   | 1.384E+01(−) | 1.966E+01(−) | 1.027E+01(−) | 1.036E+00(−)        | 1.164E+01(−) | 1.331E+01(−)        | 1.613E+01(−) | <b>9.912E+00</b> |
|          | MAPE   | 3.993E-04(−) | 9.325E-04(−) | 6.602E-04(−) | <b>1.492E-04(−)</b> | 7.773E-04(−) | 2.237E-04(−)        | 5.351E-04(−) | 1.894E-04        |
| 30 + 100 | MAE    | 2.141E-01(−) | 3.656E-01(−) | 1.379E-01(−) | 1.510E-01(−)        | 2.210E-01(−) | 2.159E-01(−)        | 1.821E-01(−) | <b>1.325E-01</b> |
|          | NMAE   | 4.625E-01(−) | 6.023E-01(−) | 3.710E-01(−) | 3.895E-01(−)        | 4.670E-01(−) | 4.649E-01(−)        | 4.241E-01(−) | <b>3.642E-01</b> |
|          | RMSE   | 8.239E+00(−) | 8.882E+00(−) | 8.251E+00(−) | 7.307E+00(−)        | 7.324E+00(−) | 8.221E+00(−)        | 7.816E+00(−) | <b>7.295E+00</b> |
|          | MAPE   | 3.563E-04(−) | 8.550E-04(−) | 7.131E-04(−) | 7.176E-04(−)        | 1.842E-04(−) | 2.021E-04(−)        | 2.747E-04(−) | <b>1.426E-04</b> |
| +/-/=    |        | 0/20/0       | 0/20/0       | 0/20/0       | 4/14/2              | 0/20/0       | 1/18/1              | 0/19/1       |                  |

growth versus 41.2 % accuracy improvement. Cross-method comparisons reveal MVSG-GS achieves optimal computational-accuracy equilibrium in 80 % of scenarios, confirming the metaheuristic framework's practicality in resource-constrained environments. These findings collectively demonstrate that the model not only overcomes data scarcity limitations in intelligent optimization but also provides an extensible solution for dynamic modeling of industrial wastewater treatment

processes.

Although increasing virtual samples generally helps to improve performance, the adaptability and effectiveness of different VSG methods vary when faced with different numbers of virtual samples. The above experiments reveal that different virtual sample generation techniques have an impact on the representativeness and diversity of data, thus affecting the training effect of learning algorithms. To further

**Table 11**

Performance comparison of dry weather dataset with fixed-size original samples.

| Datasize | Metric | ElmVSG       | RDVSG        | IsomapVSG    | VSG3A2              | MGGAN        | GAN4MTS             | VA-WGAN      | MVSG-GS          |
|----------|--------|--------------|--------------|--------------|---------------------|--------------|---------------------|--------------|------------------|
| 30 + 20  | MAE    | 5.492E-01(−) | 5.228E-01(−) | 3.552E-01(−) | 2.542E-01(−)        | 2.878E-01(−) | 2.545E-01(−)        | 2.779E-01(−) | <b>2.517E-01</b> |
|          | NMAE   | 2.350E-01(−) | 2.287E-01(−) | 1.879E-01(−) | 1.636E-01(−)        | 1.690E-01(−) | 1.666E-01(−)        | 2.902E-01(−) | <b>1.615E-01</b> |
|          | RMSE   | 1.507E+01(−) | 2.662E+01(−) | 1.677E+01(−) | <b>1.321E+01(+)</b> | 1.513E+01(−) | 1.571E+01(−)        | 1.524E+01(−) | 1.413E+01        |
|          | MAPE   | 2.885E-04(−) | 9.244E-04(−) | 7.471E-04(−) | 5.412E-04(−)        | 1.743E-04(−) | 2.389E-04(−)        | 3.353E-04(−) | <b>1.226E-04</b> |
| 30 + 40  | MAE    | 6.538E-01(−) | 7.526E-01(−) | 4.978E-01(−) | 1.796E-01(−)        | 1.539E-01(−) | 1.060E-01(−)        | 1.705E-01(−) | <b>1.355E-01</b> |
|          | NMAE   | 2.543E-01(−) | 2.736E-01(−) | 2.225E-01(−) | 1.338E-01(−)        | 1.244E-01(−) | 1.092E-01(−)        | 2.626E-01(−) | <b>1.162E-01</b> |
|          | RMSE   | 2.163E+01(−) | 2.044E+01(−) | 1.983E+01(−) | 1.460E+01(−)        | 1.443E+01(−) | 1.152E+01(−)        | 1.271E+01(−) | <b>1.140E+01</b> |
|          | MAPE   | 2.962E-04(−) | 9.852E-04(−) | 7.031E-04(−) | 6.619E-04(−)        | 2.073E-04(−) | 1.758E-04(−)        | 3.429E-04(−) | <b>1.493E-01</b> |
| 30 + 60  | MAE    | 5.536E-01(−) | 6.917E-01(−) | 5.908E-01(−) | 3.893E-01(−)        | 3.619E-01(−) | <b>2.374E-01(−)</b> | 4.083E-01(−) | 3.136E-01        |
|          | NMAE   | 2.354E-01(−) | 2.625E-01(−) | 2.427E-01(−) | 1.971E-01(−)        | 1.904E-01(−) | <b>1.313E-01(−)</b> | 4.559E-01(−) | 1.465E-01        |
|          | RMSE   | 2.102E+01(−) | 2.350E+01(−) | 2.200E+01(−) | 1.012E+01(−)        | 1.557E+01(−) | 1.164E+01(−)        | 1.204E+01(−) | <b>9.748E+00</b> |
|          | MAPE   | 2.807E-04(−) | 1.016E-04(−) | 7.011E-04(−) | <b>1.023E-04(+)</b> | 1.590E-04(−) | 3.102E-04(−)        | 3.417E-04(−) | 1.499E-04        |
| 30 + 80  | MAE    | 6.506E-02(−) | 6.769E-02(−) | 4.705E-02(−) | 1.366E-02(−)        | 1.857E-02(−) | 1.785E-02(−)        | 1.464E-02(−) | <b>1.144E-02</b> |
|          | NMAE   | 2.546E-01(−) | 2.591E-01(−) | 2.159E-01(−) | 1.174E-01(−)        | 1.353E-01(−) | 9.742E-01(−)        | 2.345E-01(−) | <b>1.065E-01</b> |
|          | RMSE   | 2.237E+01(−) | 3.238E+01(−) | 2.979E+01(−) | 9.715E+00(−)        | 1.123E+01(−) | 1.015E+01(−)        | 1.011E+01(−) | <b>9.206E+00</b> |
|          | MAPE   | 2.520E-04(−) | 1.094E-04(−) | 7.691E-04(−) | 7.320E-04(−)        | 1.645E-04(−) | 2.920E-04(−)        | 3.641E-04(−) | <b>1.293E-04</b> |
| +/-/=    |        | 0/16/0       | 0/16/0       | 0/16/0       | 2/14/0              | 0/16/0       | 2/14/0              | 0/16/0       |                  |

**Table 12**

Performance comparison of rain weather dataset with fixed-size original samples.

| Datasize | Metric | ElmVSG       | RDVSG        | IsomapVSG    | VSG3A2              | MGGAN               | GAN4MTS             | VA-WGAN      | MVSG-GS          |
|----------|--------|--------------|--------------|--------------|---------------------|---------------------|---------------------|--------------|------------------|
| 30 + 20  | MAE    | 7.629E-01(−) | 9.483E-01(−) | 7.685E-01(−) | 7.435E-01(−)        | <b>7.081E-01(+)</b> | 7.558E-01(−)        | 7.236E-01(−) | 7.177E-01        |
|          | NMAE   | 2.748E-01(−) | 3.060E-01(−) | 2.757E-01(−) | 2.719E-01(−)        | <b>2.340E-01(+)</b> | 2.559E-01(−)        | 2.645E-01(−) | 2.665E-01        |
|          | RMSE   | 6.182E+00(−) | 4.095E+00(−) | 3.443E+00(−) | 3.290E+00(−)        | <b>3.177E+00(+)</b> | 3.435E+00(−)        | 4.012E+00(−) | 3.212E+00        |
|          | MAPE   | 2.681E-04(−) | 1.037E-04(−) | 7.835E-04(−) | 7.638E-04(−)        | 8.287E-04(−)        | <b>3.144E-04(+)</b> | 3.728E-04(−) | 4.057E-04        |
| 30 + 40  | MAE    | 7.321E-01(−) | 1.426E-01(−) | 7.154E-01(−) | 7.701E-01(−)        | 7.077E-01(−)        | 7.121E-01(−)        | 7.095E-01(−) | <b>7.002E-01</b> |
|          | NMAE   | 2.699E-01(−) | 1.187E-01(−) | 2.669E-01(−) | 2.770E-01(−)        | 2.655E-01(−)        | 3.361E-01(−)        | 3.134E-01(−) | <b>2.652E-01</b> |
|          | RMSE   | 8.284E+00(−) | 5.938E+00(−) | 3.226E+00(−) | 3.389E+00(−)        | 3.165E+00(−)        | 3.542E+00(−)        | 4.804E+00(−) | <b>3.107E+00</b> |
|          | MAPE   | 3.109E-04(−) | 3.108E-04(−) | 8.055E-04(−) | 7.456E-04(−)        | 6.914E-04(−)        | 2.453E-04(−)        | 3.477E-04(−) | <b>2.253E-04</b> |
| 30 + 60  | MAE    | 6.928E-01(−) | 1.667E-01(−) | 7.256E-01(−) | 7.273E-01(−)        | 6.850E-01(−)        | 6.730E-01(−)        | 7.096E-01(−) | <b>6.710E-01</b> |
|          | NMAE   | 2.621E-01(−) | 3.270E-01(−) | 2.683E-01(−) | 2.690E-01(−)        | 2.627E-01(−)        | 3.510E-01(−)        | 3.888E-01(−) | <b>2.589E-01</b> |
|          | RMSE   | 6.106E+00(−) | 6.742E+00(−) | 3.233E+00(−) | 3.217E+00(−)        | 3.105E+00(−)        | 4.503E+00(−)        | 3.733E+00(−) | <b>3.072E+00</b> |
|          | MAPE   | 2.825E-04(−) | 3.131E-04(−) | 8.203E-04(−) | <b>2.456E-04(+)</b> | 6.671E-04(−)        | 2.490E-04(−)        | 3.862E-04(−) | 2.486E-04        |
| 30 + 80  | MAE    | 6.877E-02(−) | 1.668E-02(−) | 7.228E-02(−) | 6.946E-02(−)        | 6.706E-02(−)        | 6.716E-02(−)        | 6.658E-02(−) | <b>6.420E-02</b> |
|          | NMAE   | 8.277E-01(−) | 8.324E-01(−) | 8.472E-01(−) | <b>4.074E-01(+)</b> | 8.138E-01(−)        | 7.538E-01(−)        | 1.002E+00(−) | 8.004E-01        |
|          | RMSE   | 5.071E+00(−) | 6.845E+00(−) | 3.260E+00(−) | 3.138E+00(−)        | 3.016E+00(−)        | <b>2.507E+00(−)</b> | 3.081E+00(−) | 2.873E+00        |
|          | MAPE   | 3.253E-04(−) | 1.115E-04(−) | 8.575E-04(−) | 7.370E-04(−)        | 3.102E-04(−)        | 2.649E-04(−)        | 3.389E-04(−) | <b>2.492E-04</b> |
| +/-/=    |        | 0/16/0       | 0/16/0       | 0/16/0       | 2/14/0              | 3/13/0              | 1/15/0              | 0/16/0       |                  |

qualitatively validate the performance of MVSG-GS, t-SNE was employed to visualize how the distribution of generated samples in a two-dimensional space approaches the original data, as shown in Fig. 9. Specifically, t-SNE models each high-dimensional data point as a probability distribution in a low-dimensional space, minimizing the discrepancy between the pairwise similarities of the original high-dimensional data points and their corresponding low-dimensional representations.

The provided visualizations systematically compare the distributions of synthetic data generated by MVSG-GS (column 1) and three other state-of-the-art methods (GAN4MTS, VA-WGAN, and MGGAN) against the original data under dry, rain, and storm conditions. The first three rows correspond to the COD dataset, while the last three rows present results for the BOD5 dataset. MVSG-GS exhibits a clear advantage in preserving the original data structure across all environmental conditions. The synthetic data generated by MVSG-GS demonstrates an

interleaving pattern with the original data (blue points), indicating minimal divergence. Under different conditions, MVSG-GS retains a high level of alignment, as evidenced by the overlapping regions between synthetic and original data points. This performance is particularly notable given the increased variability in the data due to environmental factors. The synthetic data from GAN4MTS, VA-WGAN, and MGGAN diverge more prominently, with MGGAN exhibiting a clustering pattern that deviates significantly from the original data distribution.

Fig. 10 and Fig. 11 show the prediction results and errors of different methods on three datasets. The experiment focuses on evaluating the prediction accuracy and error dynamics under different weather conditions, providing insights into model performance robustness. As shown in the first column, the prediction results exhibit a high degree of alignment with the ground truth, with error (column 2) distributions oscillating closely around the zero-error line. The low variance of error

**Table 13**

Performance comparison of storm weather dataset with fixed-size original samples.

| Datasize | Metric | ElmVSG       | RDVSG        | IsomapVSG    | VSG3A2       | MGGAN        | GAN4MTS             | VA-WGAN      | MVSG-GS          |
|----------|--------|--------------|--------------|--------------|--------------|--------------|---------------------|--------------|------------------|
| 30 + 20  | MAE    | 3.083E-01(-) | 2.542E-01(-) | 2.491E-01(-) | 2.244E-01(-) | 2.280E-01(-) | 2.852E-01(-)        | 2.624E-01(-) | <b>2.223E-01</b> |
|          | NMAE   | 1.748E-01(-) | 1.587E-01(-) | 1.571E-01(-) | 1.495E-01(-) | 1.506E-01(-) | 1.685E-01(-)        | 3.273E-01(-) | <b>1.489E-01</b> |
|          | RMSE   | 1.385E+01(-) | 1.513E+01(-) | 1.395E+01(-) | 1.224E+01(-) | 1.396E+01(-) | 1.223E+01(-)        | 1.774E+01(-) | <b>1.126E+01</b> |
|          | MAPE   | 2.708E-04(-) | 2.135E-04(-) | 7.590E-04(-) | 6.848E-04(-) | 1.668E-04(-) | <b>1.315E-04(+)</b> | 3.480E-04(-) | 4.807E-04        |
| 30 + 40  | MAE    | 2.306E-01(-) | 3.403E-01(-) | 2.226E-01(-) | 2.299E-01(-) | 2.197E-01(-) | <b>1.979E-01(+)</b> | 3.112E-01(-) | 2.141E-01        |
|          | NMAE   | 1.519E-01(-) | 1.837E-01(-) | 1.478E-01(-) | 1.514E-01(-) | 1.479E-01(-) | 1.541E-01(-)        | 3.631E-01(-) | <b>1.454E-01</b> |
|          | RMSE   | 1.390E+01(-) | 1.885E+01(-) | 1.339E+01(-) | 1.341E+01(-) | 1.351E+01(-) | 1.252E+01(-)        | 1.331E+01(-) | <b>1.214E+01</b> |
|          | MAPE   | 2.252E-04(-) | 1.112E-04(-) | 7.088E-04(-) | 6.529E-04(-) | 2.100E-04(-) | 1.835E-04(-)        | 3.253E-04(-) | <b>1.512E-04</b> |
| 30 + 60  | MAE    | 2.236E-01(-) | 3.636E-01(-) | 1.559E-01(-) | 2.618E-01(-) | 2.211E-01(-) | <b>1.165E-01(+)</b> | 1.237E-01(-) | 3.449E-01        |
|          | NMAE   | 1.496E-01(-) | 1.903E-01(-) | 1.251E-01(-) | 1.616E-01(-) | 1.487E-01(-) | 7.711E-01(-)        | 2.052E-01(-) | <b>7.151E-01</b> |
|          | RMSE   | 1.422E+01(-) | 1.980E+01(-) | 1.036E+01(-) | 1.520E+01(-) | 1.379E+01(-) | 1.007E+01(-)        | 1.054E+01(-) | <b>9.875E+00</b> |
|          | MAPE   | 2.198E-04(-) | 1.195E-04(-) | 7.300E-04(-) | 5.703E-04(-) | 2.816E-04(-) | 2.522E-04(-)        | 3.888E-04(-) | <b>2.005E-04</b> |
| 30 + 80  | MAE    | 3.429E-01(-) | 3.505E-01(-) | 2.356E-01(-) | 1.979E-01(-) | 1.722E-01(-) | 1.426E-01(-)        | 2.712E-01(-) | <b>1.229E-01</b> |
|          | NMAE   | 2.852E-01(-) | 2.920E-01(-) | 2.836E-01(-) | 2.449E-01(-) | 2.154E-01(-) | 3.365E-01(-)        | 6.819E-01(-) | <b>1.652E-01</b> |
|          | RMSE   | 2.058E+01(-) | 1.956E+01(-) | 1.267E+01(-) | 9.412E+00(-) | 9.551E+00(-) | <b>9.293E+00(+)</b> | 9.203E+00(-) | 9.531E+00        |
|          | MAPE   | 2.159E-04(-) | 1.179E-04(-) | 7.338E-04(-) | 6.073E-04(-) | 2.581E-04(-) | 1.836E-04(-)        | 4.548E-04(-) | <b>1.021E-04</b> |
| +/-/=    |        | 0/16/0       | 0/16/0       | 0/16/0       | 0/16/0       | 0/16/0       | 4/12/0              | 0/16/0       |                  |

suggests that the soft sensor model performs consistently in stable conditions, effectively capturing the linear and nonlinear patterns in the dataset. The prediction results and errors under rain and storm conditions reveal a noticeable increase in prediction variability. Rain and storm events often introduce dynamic fluctuations in water quality parameters, which are partially reflected in the model's slightly reduced accuracy. Nonetheless, MVSG-GS demonstrates moderate robustness, as the deviations remain within acceptable thresholds for most data points.

The above experiment analysis demonstrates the effectiveness of the MVSG-GS method from both quantitative and qualitative perspectives. To further analyze the potential advantages of virtual samples on the modeling accuracy of soft sensor models, we plot the error histograms of the models on three weather test sets using different numbers of virtual samples. This allows us to observe the trend of model error as the size of virtual samples increases and assess the effectiveness of virtual samples in model training.

As shown in Fig. 12(a), the overall trend of the model's error on the test set decreases as the number of virtual samples increases, which suggests that increasing virtual samples can improve the generalization ability on unseen data. Initially, with fewer virtual samples, the error histograms show large fluctuations, whereas these fluctuations are significantly suppressed when the number of virtual samples is increased to 30. The dispersion of the error distribution decreases with the increase of virtual samples, which indicates that the model's prediction results are more stable. As shown in Fig. 12(b), the error on the rain weather test set also shows an overall decreasing trend with increasing virtual samples. However, unlike the previous results, the error distribution under rain weather conditions shows slightly increased volatility. While the overall trend is still decreasing, the higher errors for some of the data points suggest that the model may have encountered additional challenges, such as the handling of noisy data and the adaptation of complex environmental factors. The same trend can be observed from Fig. 12(c). The model's error decreases more slowly when the number of virtual samples increases. This suggests that the model faces more challenges under storm conditions, which are caused by the unpredictability of the influent flow during heavy weather.

The above analysis indicates that increasing the number of virtual samples does not always lead to a linear improvement in performance under extreme weather conditions, but may lead to unstable performance due to varying sample quality or insufficient model training. Upon expanding the sample size, the model exhibited diminishing

marginal returns in performance enhancement, necessitating more refined parameter adjustments to accommodate the heightened complexity of input data structures.

#### 4.5. Ablation experiment analysis

To systematically evaluate the contributions of individual modules in the MVSG-GS framework, we conducted comprehensive ablation analysis under identical experimental conditions, comparing six model variants including the baseline configuration. As shown in Table 14, five variant models are designed to comprehensively evaluate key modules.

Fig. 13 shows the statistical results of each model on the UCI wine quality dataset, while Fig. 14 visually demonstrates the effects of virtual samples generated by different models through corresponding feature distribution diagrams.

As shown in Fig. 13, when removing the cross-task knowledge transfer mechanism (Model A), the generation quality of virtual samples significantly deteriorates, manifested through increased distribution divergence from real data. The model's generalization capability becomes constrained, failing to effectively capture latent characteristics of authentic distributions. Meanwhile, elimination of the MBEA-GI module (Model B) causes unstable feature-space distributions in generated samples across multiple dimensions. Specifically in the silhouette score, virtual samples exhibit amplified discrepancies from real data, revealing the low-entropy guidance mechanism's critical role in maintaining structural consistency. Furthermore, simultaneous removal of both MBEA-GI and knowledge transfer modules (Model D) leads to substantial performance degradation, confirming the essentiality of knowledge-sharing mechanisms in multi-task modeling. While network structure reduction (Model C) causes moderate performance decline, the limited magnitude suggests network scale influences but doesn't dominantly determine generation efficacy. The combined elimination of knowledge transfer and generator input optimization (Model E) yields worst-case results, verifying these modules' synergistic importance in high-quality virtual sample generation through multi-task coordination. Finally, the adjusted knowledge transfer frequency (Model F) exerts pronounced effects on virtual sample quality. Excessively large transfer intervals would compromise guideline sharing efficiency, which consequently impedes the model's capacity to harness intra-domain information for accelerated training convergence.

Fig. 14 presents comparative distribution analyses of virtual versus

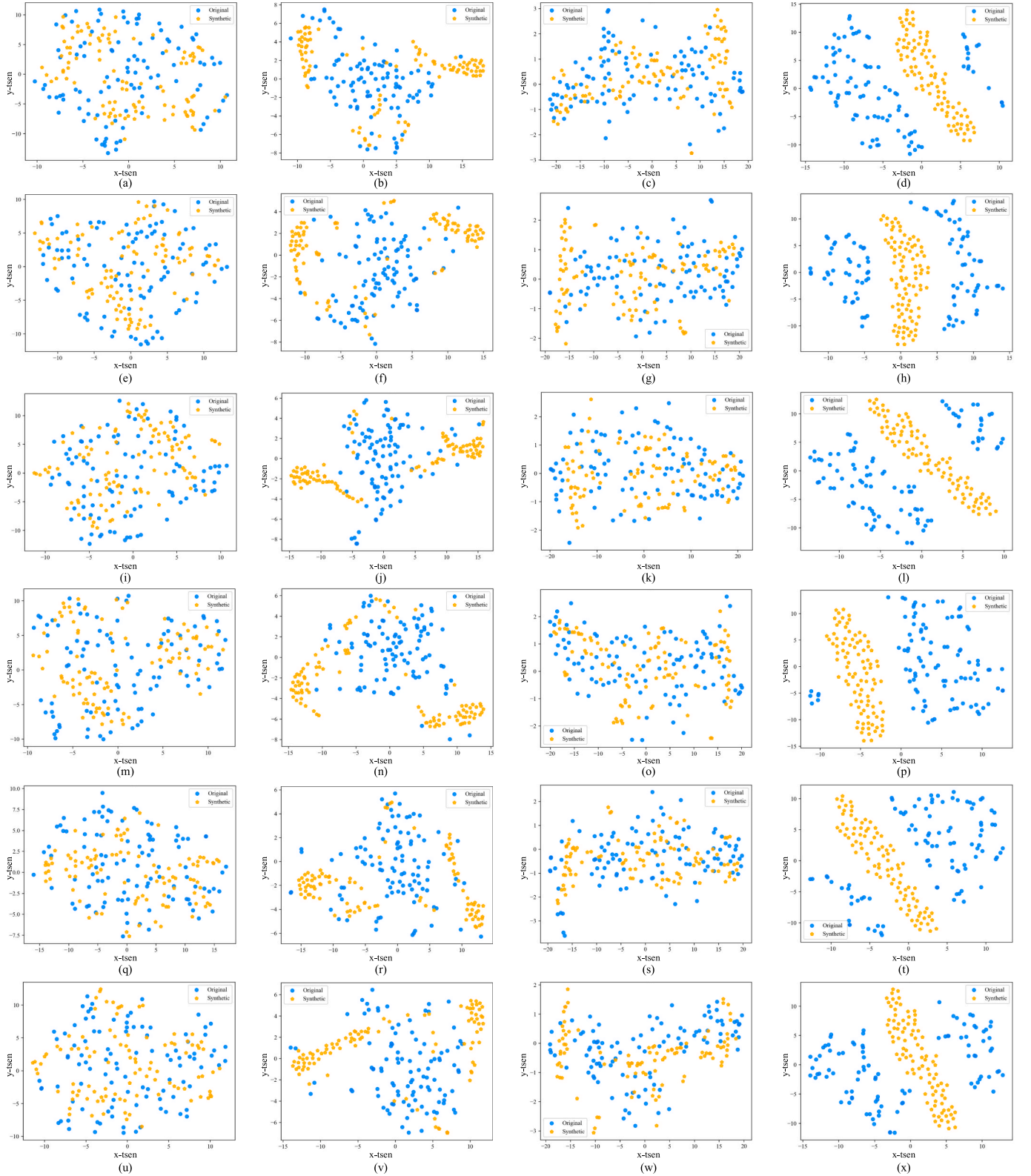


Fig. 9. t-SNE visualization results for original data and synthetic data based on MVSG-GS.

real samples across the first four feature dimensions. The MVSG-GS framework demonstrates superior alignment with authentic data distributions across all measured dimensions, particularly achieving near-perfect replication in the first and third feature spaces. This distributional congruence validates the combined efficacy of the MBEA-GI module and the sparse knowledge transfer mechanism in capturing

intrinsic data patterns. In contrast, Model A (without cross-task knowledge transfer) exhibits pronounced distributional discrepancies, particularly manifesting as dispersion anomalies in the second and fourth feature dimensions. This degradation highlights the mechanism's essential role in preserving inter-data correlations critical for high-fidelity generation. Similarly, Model B (MBEA-GI-ablated version)



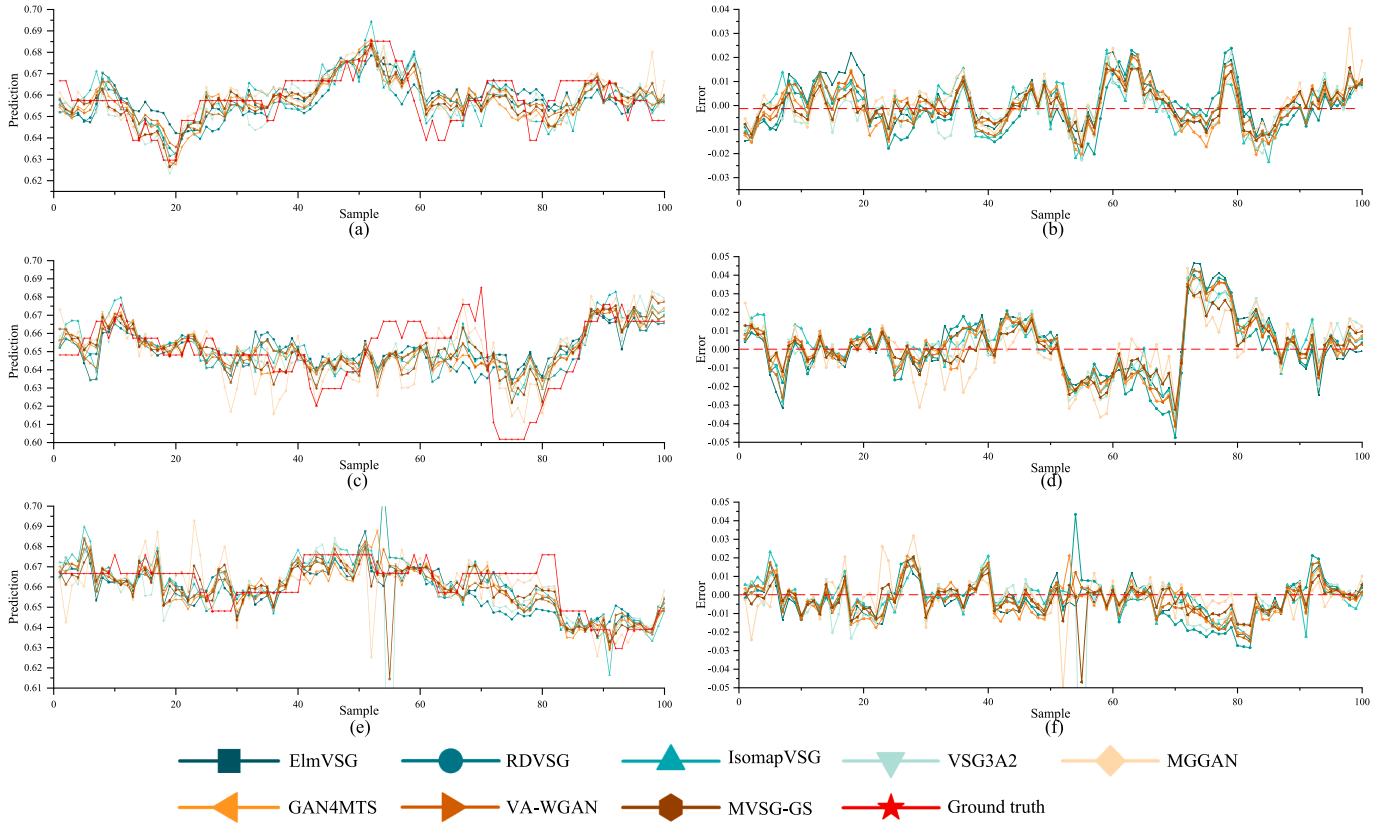


Fig. 10. Prediction results and errors of the COD soft sensor model trained with (a) dry, (c) rain and (e) storm weather data.

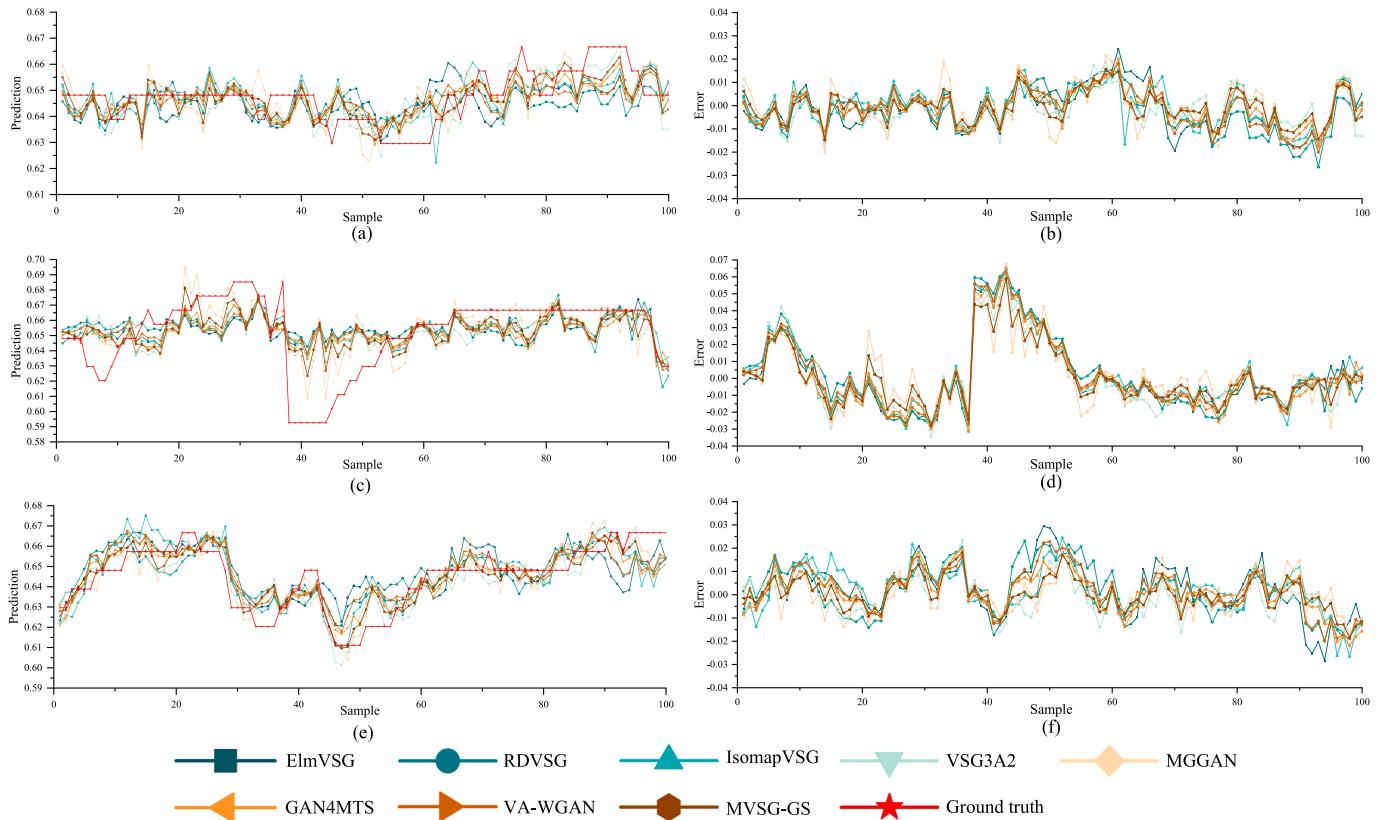


Fig. 11. Prediction results and errors of the BOD<sub>5</sub> soft sensor model trained with (a) dry, (c) rain and (e) storm weather data.

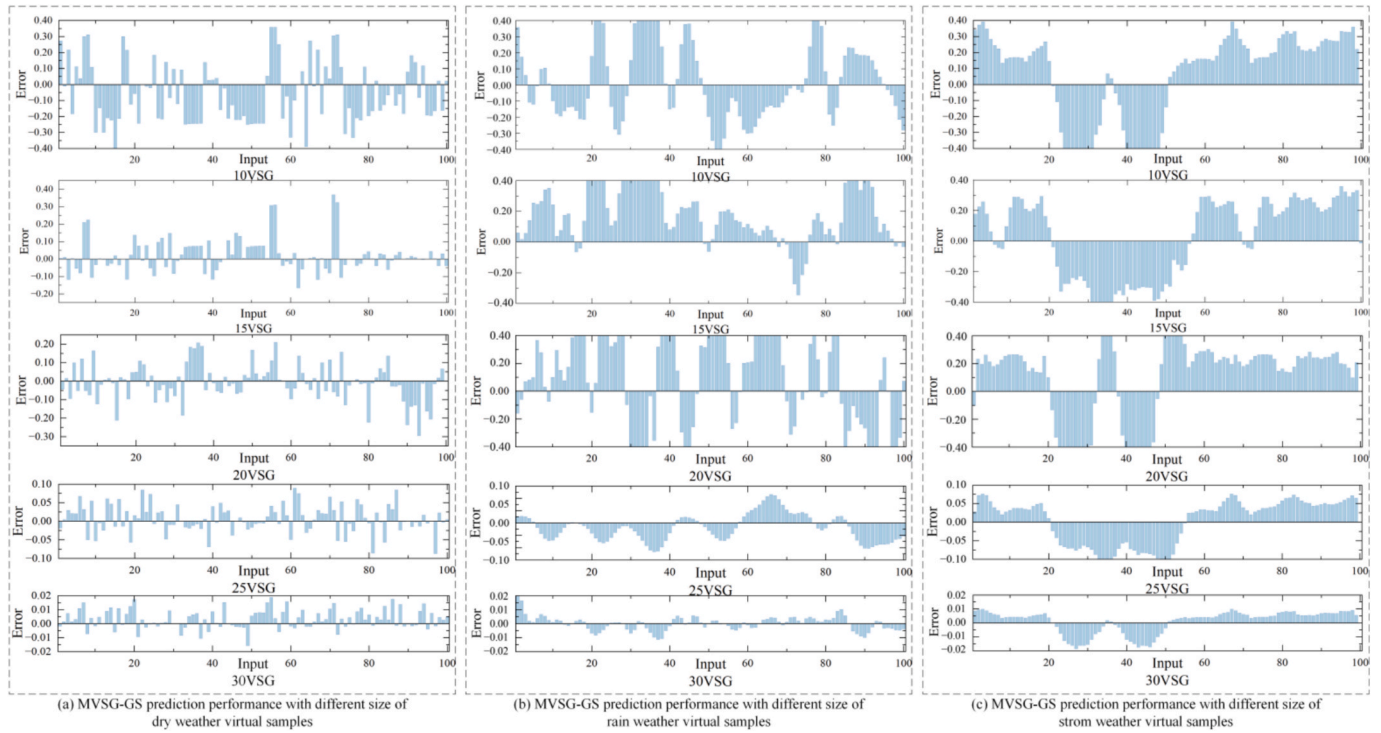


Fig. 12. MVSG-GS prediction performance with different size of (a) Dry weather, (b) Rain weather, (c) Storm weather virtual samples.

Table 14  
The variant models of MVSG-GS.

| No      | Description                                                                                                                                          |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Model A | Disabled the cross-task knowledge transfer mechanism by setting the knowledge transfer rate to 0, thereby isolating its impact on model performance. |
| Model B | Eliminated the MBEA-GI to investigate the role of low-entropy structural information in virtual sample quality enhancement.                          |
| Model C | Reduced network capacity by decreasing hidden layer dimensions from 128 to 64, quantifying architectural contributions to generation efficacy.       |
| Model D | Simultaneously removed both knowledge transfer and generator input optimization modules to assess their synergistic effects in multi-task learning.  |
| Model E | Extended knowledge transfer interval from 5 to 20 iterations, evaluating temporal sensitivity of cross-task knowledge integration.                   |

displays compromised structural consistency, most notably through irregular distribution patterns in the 4th feature dimension that deviate substantially from real data distributions, thereby underscoring MBEA-GI's indispensable function in stabilizing sample geometry. The frequency-reduced knowledge transfer variant (Model E) demonstrates relatively mitigated performance degradation compared to complete mechanism removal, suggesting temporal scheduling parameters influence – though do not fundamentally govern – distributional fidelity. Comparative analyses of remaining architectures (Models C&D) reveal dimension-specific performance variations that collectively reinforce the necessity of integrated module design for optimal virtual sample generation.

Further, we validated the practical performance of different models through experiments on soft-sensing measurements of key WWTP parameters under three weather conditions. Figs. 15–17 demonstrate the performance of MVSG-GS and its variants on different weather datasets. The main evaluation metrics include mean absolute error (MAE) and root mean squared error (RMSE). Experiment results reveal that MVSG-GS consistently achieves optimal performance across all training set configurations and weather conditions, significantly outperforming the other five model variants in both MAE and RMSE metrics. This indicates

that MVSG-GS possesses strong robustness and generalization capability to effectively handle complex and variable operational data. In comparison, Model A exhibits significant performance degradation after removing the cross-task knowledge transfer mechanism, particularly showing instability under limited data samples, verifying the critical role of knowledge transfer in enhancing learning efficiency and generalization. Model B suffers from quality degradation of generated virtual samples due to the removal of the MBEA-GI module, leading to compromised prediction accuracy that becomes more pronounced in small-sample scenarios (e.g., 10 + 90 configuration), demonstrating the importance of structural optimization in virtual sample enhancement. Model C slightly sacrifices expressive power through network capacity reduction for complexity control, showing performance decline when handling high-dimensional inputs or complex features, highlighting the need for balancing efficiency and representation capability in network architecture. Model D manifests substantial performance deterioration with increased MAE/RMSE fluctuations after simultaneous removal of knowledge transfer and evolutionary optimization mechanisms, further evidencing their synergistic enhancement effects in multi-task learning. Model E experiences minor performance degradation in certain tasks when extending knowledge transfer intervals, suggesting that timely and frequent knowledge updates benefit dynamic adaptation capability. Collectively, the integrated modules in MVSG-GS synergistically contribute to its superior performance in multi-task soft sensing applications.

In summary, the ablation experiments reveal distinct impacts of different modules on the overall performance of MVSG-GS. First, the cross-task knowledge transfer mechanism plays a pivotal role in enhancing the model's generalization capability, particularly under complex weather conditions, where it substantially improves prediction accuracy in small-sample scenarios. Second, the MBEA-GI module is critical for generating high-quality virtual samples—removal of this module leads to a marked decline in performance, underscoring the indispensable role of low-entropy structural information in virtual sample augmentation. Furthermore, adjustments to network capacity influence generative efficiency: while reducing hidden layer dimensions

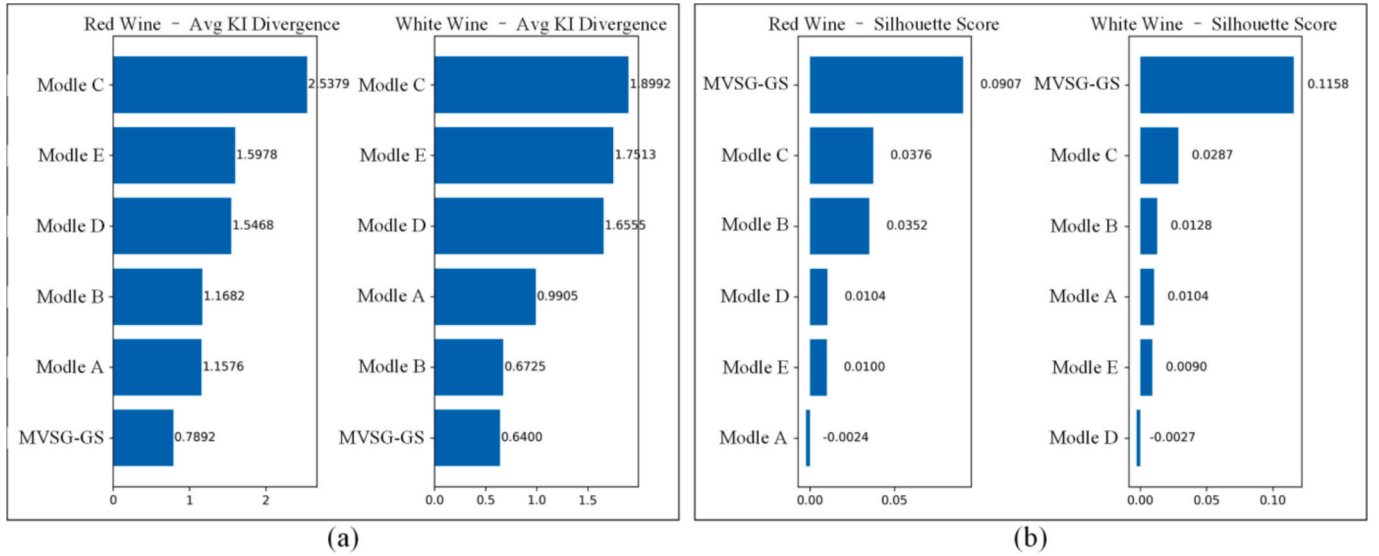


Fig. 13. The ablation comparisons of wine quality dataset with different variant models.

lowers model complexity, excessively constrained network capacity under limited data conditions may hinder the extraction of critical features. Notably, a clear synergistic effect exists between the knowledge transfer mechanism and the input optimization module, as simultaneous removal of both results in the most severe performance deterioration, demonstrating their collaborative enhancement of multi-task learning. Finally, the temporal interval of knowledge transfer affects the model's dynamic adaptability—prolonging migration intervals reduces update costs but may also diminish responsiveness to new tasks, thereby compromising.

#### 4.6. Computational efficiency analysis

To comprehensively evaluate model performance, we designed multi-dimensional comparative experiments from two aspects: computational efficiency and predictive accuracy. Regarding computational efficiency: (1) Time complexity was assessed by recording end-to-end time consumption from input loading to result output; (2) Space complexity was evaluated by statistically analyzing peak resource consumption during the training phase using memory monitoring tools; (3) Scalability was quantified by progressively expanding the training set and measuring variations in multiple performance metrics. Regarding performance evaluation: (4) Mean absolute error (MAE) and (5) root mean square error (RMSE) were employed to quantify deviations between model outputs and ground truth values. Experiments were conducted on WWTP datasets under three weather conditions. Each metric was averaged over five independent trials to eliminate random fluctuations, with horizontal comparisons against other competitors to validate the comprehensive advantages of the proposed method in efficiency-accuracy trade-offs.

Fig. 18 present the performance of MVSG-GS and other competitors (ElmVSG, RDVSG, IsomapVSG, VSG3A2, VA-WGAN, GAN4MTS, MGGAN) under (a) dry, (b) rain, and (c) storm weather. The radar charts of five metrics demonstrate the comprehensive performance of each method across multiple dimensions. Each axis represents an individual metric ordered from optimal to suboptimal performance, where higher scores indicate superior capability in the corresponding dimension. MVSG-GS exhibits higher time complexity when handling complex tasks compared to statistic-based approaches. However, its sparse transfer mechanism eliminates the need for fully independent training of dual networks in bi-objective tasks, demonstrating distinct advantages over methods such as VA-WGAN and GAN4MTS. Under stormy weather conditions, the time complexity of MVSG-GS increases, indicating

suboptimal computational efficiency of its multi-task training framework when processing extreme data, which necessitates further optimization. Regarding space complexity, MVSG-GS maintains stable memory consumption across all weather conditions with compact resource usage. In contrast, MGGAN and GAN4MTS require significantly higher memory resources when independently training dual networks, suggesting potential challenges in handling high-dimensional data due to substantial memory overhead. Scalability evaluations quantify performance improvements with increasing sample sizes. MVSG-GS demonstrates superior scalability across all weather conditions, sustaining high efficiency when processing datasets of varying scales. In terms of error metrics, MVSG-GS achieves lower root mean square error (RMSE) and mean absolute error (MAE) values under all three weather conditions, particularly excelling in rainy and stormy scenarios. Comprehensive analysis indicates that compared to data augmentation methods with simplistic architectures and strategies, the proposed MVSG-GS achieves higher-quality virtual samples at the expense of moderate computational efficiency. Meanwhile, methods such as ElmVSG and GAN4MTS exhibit larger errors across different weather conditions, revealing performance instability in multi-scenario applications. Fig. 19 illustrates computational efficiency comparisons between MVSG-GS and its five ablation models. The complete MVSG-GS model demonstrates relatively stable training time and memory consumption, with performance significantly outperforming other ablation variants. Notably, reducing the network scale (Model C) decreases both time and space complexity but causes substantial performance deterioration. Furthermore, reducing knowledge transfer frequency under extreme rainy and stormy conditions (Model E) substantially impacts scalability.

#### 4.7. Results discussion and limitation analysis

The experiment results demonstrate that the proposed MVSG-GS exhibits exceptional generalization capability and sample generation quality in multi-scenario dataset evaluations. Comparative analysis on the UCI wine quality dataset with varying sample scales ( $N = 100/500/1000$ ) reveals the model's effectiveness in capturing original data distribution characteristics under data scarcity. With merely 100 training samples, the generated samples achieve comparable distribution width and symmetry to real data in key parameters (pH value, density, alcohol content), further validated by boxplot analysis showing precise alignment of medians and interquartile ranges. For practical applications in WWTP key parameter soft sensor tasks, systematic verification was conducted under three weather conditions: dry, rainfall, and storm. The

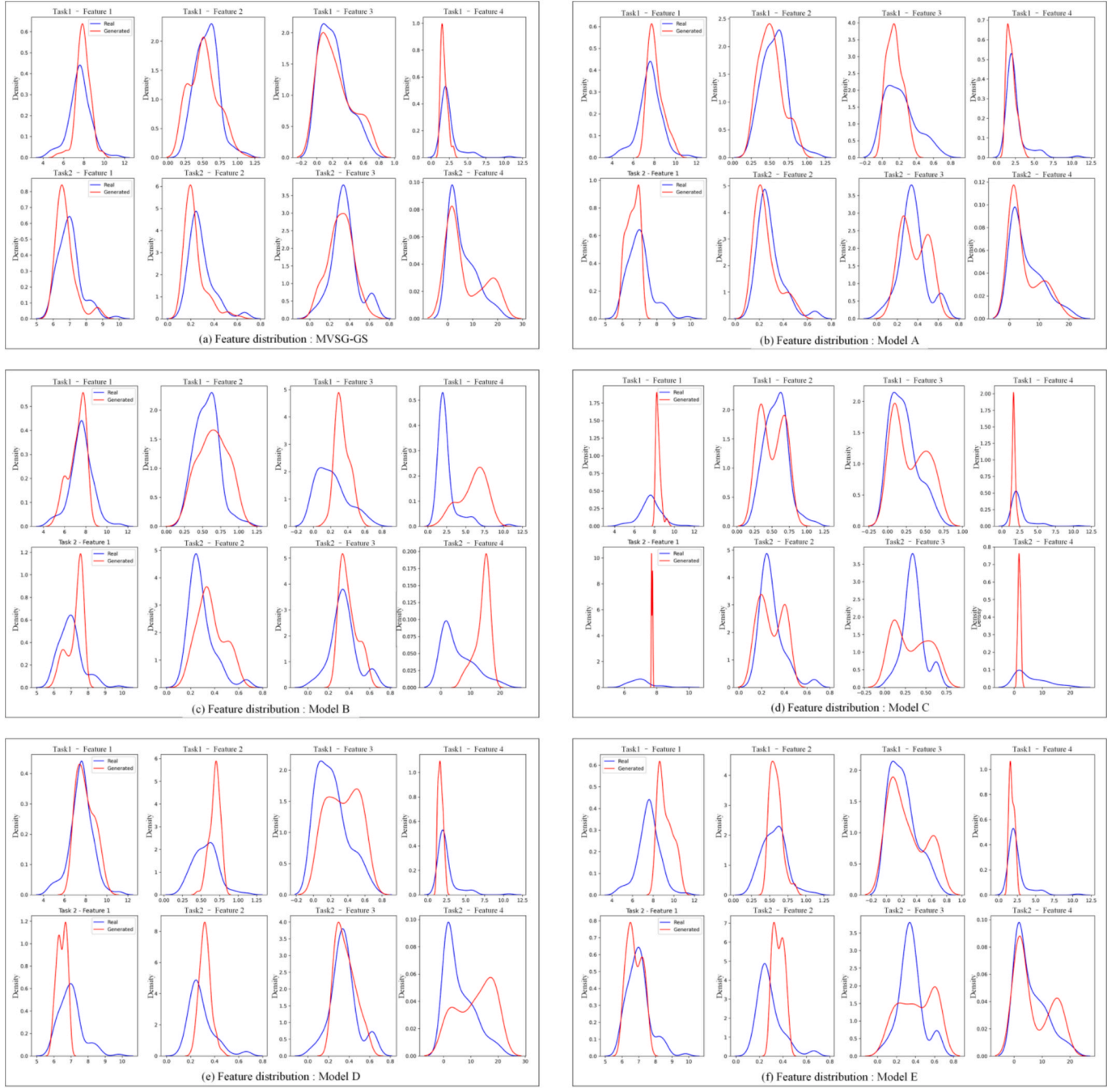


Fig. 14. Feature distribution of MVSG-GS and other 5 variant models.

method outperforms baseline approaches in 66.7 % of comparisons. t-SNE visualization confirms significant overlap between generated and original samples in 2D space, particularly maintaining structural integrity in BOD5 and COD datasets with notably reduced distribution deviation compared to baselines.

Ablation experiments systematically validate the synergistic effects among core framework components. Elimination of the cross-task knowledge transfer mechanism causes distribution anomalies in critical feature dimensions, severely impairing the model's ability to capture complex data structures. Removal of the MBEA-GI module induces feature space destabilization, manifested through systematic bias in silhouette score metrics and increased error dispersion, confirming the necessity of low-entropy guidance for maintaining sample structure consistency. The experiments also reveal strong inter-module coupling:

concurrent removal of knowledge transfer and structural optimization mechanisms causes performance degradation exceeding the cumulative effect of individual module failures. While adjustment of knowledge transfer frequency preserves core mechanisms, it significantly compromises the model's responsiveness to dynamic tasks.

However, the proposed method still presents several limitations requiring further investigation. Firstly, the relationship between virtual sample generation volume and model performance improvement exhibits nonlinear characteristics — structural discrepancies between synthetic and original real data distributions may emerge when synthetic samples exceed a critical threshold proportion. This phenomenon becomes particularly pronounced under extreme sparsity of original training data, potentially inducing model overfitting on synthetic samples. Secondly, while the method's effectiveness has been validated on



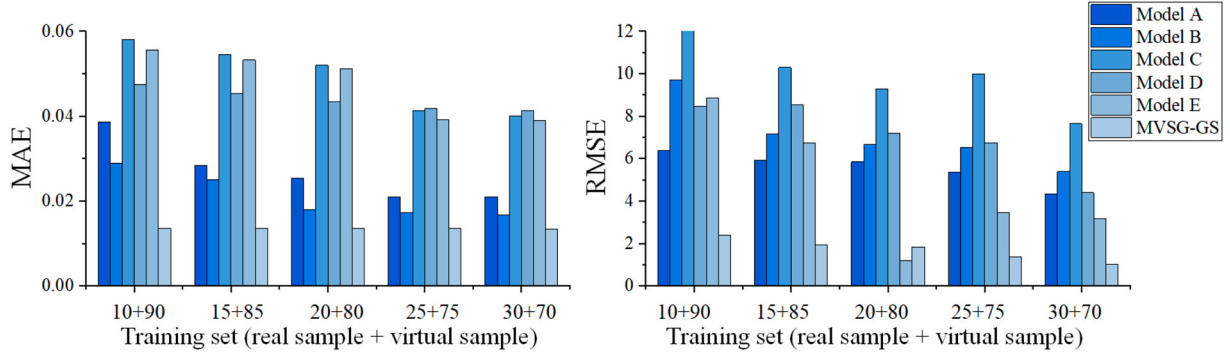


Fig. 15. The ablation comparisons of Dry weather dataset with adding different number of original samples.

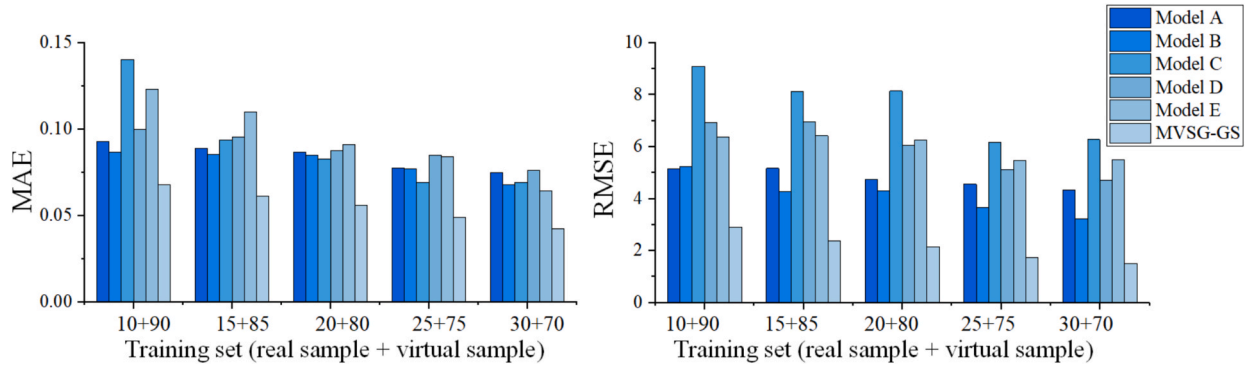


Fig. 16. The ablation comparisons of Rain weather dataset with adding different number of original samples.

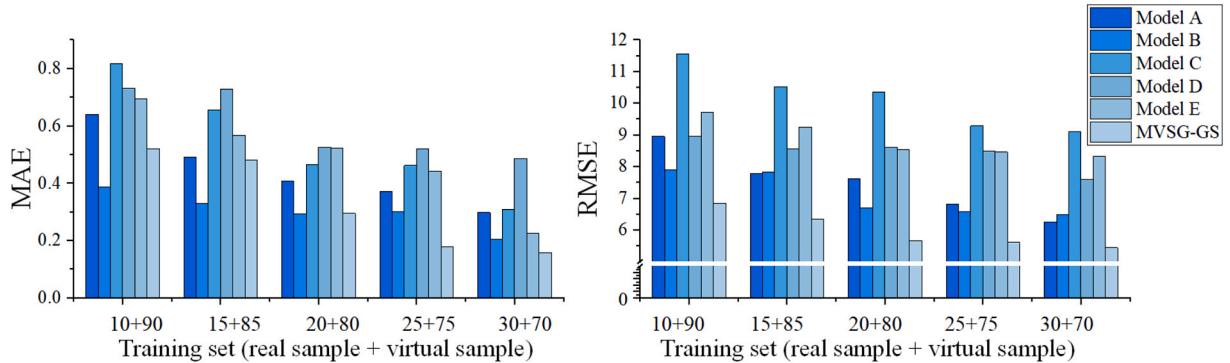
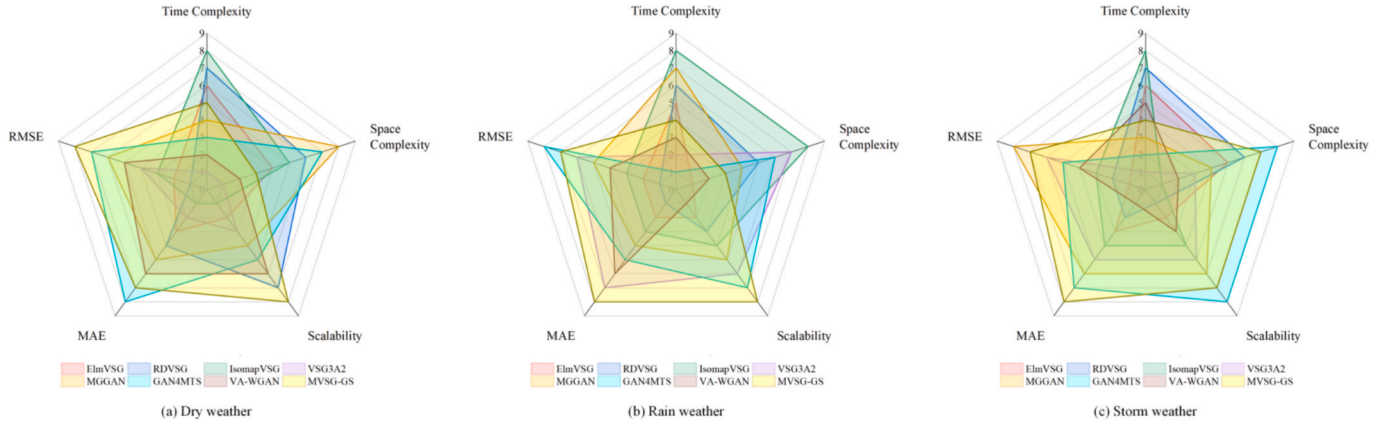


Fig. 17. The ablation comparisons of Storm weather dataset with adding different number of original samples.

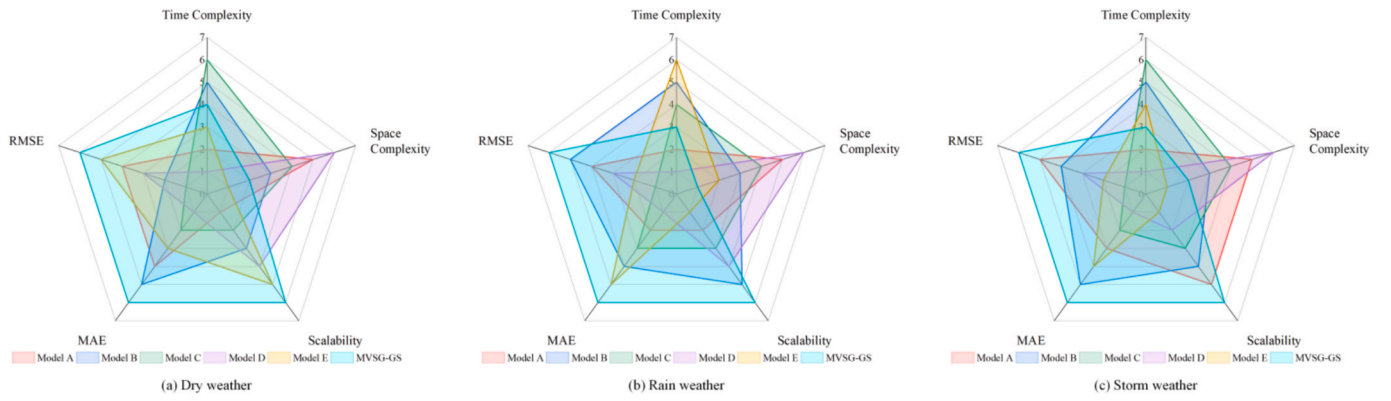
UCI benchmark datasets and practical applications, the current intra-domain transfer-based MVSG-GS framework still struggles to capture deep pattern correlations in domains with highly complex feature interactions (e.g., tasks involving multimodal dependencies or nonlinear coupled features). Specifically, the sparse knowledge transfer mechanism assumes shared latent constraints across tasks, which may not hold in heterogeneous feature spaces. The model's pattern recognition mechanism could be compromised in weakly correlated multi-task scenarios, thereby impacting its generalization reliability in open environments. Additionally, compared with standalone GAN architectures, the integration of metaheuristic optimization with adversarial training introduces higher computational overhead. Although evolutionary population initialization enhances sample quality, the iterative evaluation of dual-objective functions (KL divergence and cosine similarity) necessitates further exploration of scalability for high-dimensional industrial datasets.

## 5. Conclusions

In this study, a metaheuristic optimization method is used to transform the inputs of the generator into a low-entropy structure that is more in line with practical application scenarios, and a sparse knowledge transfer mechanism is employed to enhance the training efficiency. The proposed metaheuristic virtual sample generation model based on guidelines sharing (MVSG-GS) addresses the inherent limitations of conventional adversarial training by preserving optimal offspring solutions derived from diverse training objectives. MVSG-GS achieving competitive results on both general-purpose UCI benchmark datasets and domain-specific soft sensor modeling tasks for wastewater treatment processes. The results highlight the critical role of virtual samples in enhancing the generalization capability of soft sensor models under data scarcity conditions, particularly in chemical industry applications where operational data acquisition is inherently constrained. Although MVSG-GS requires slightly higher computational demands during



**Fig. 18.** The computational efficiency comparisons of MVSG-GS and other competitors under different weather WWTP dataset.



**Fig. 19.** The computational efficiency comparisons of ablation models under different weather WWTP dataset.

training, its superior generation quality compensates for this limitation, enabling efficient implementation even in industrial processes with restricted sample sizes. Overall, MVSG-GS achieves an optimal balance between inference speed and generation quality. Its combined efficiency and high-quality outputs surpass most baseline methods, making it more suitable for practical tasks in industrial application scenarios.

Future research should prioritize enhancing model robustness and generalization under varied operational conditions. These developments are essential for building intelligent systems that support decision-making in complex biochemical systems. Furthermore, this study presents a preliminary attempt to enhance the performance of VSG methods in real-world applications through knowledge transfer, which may provide new insights for subsequent investigations. VSG technology could potentially address the challenge of insufficient prior knowledge in industrial settings by enabling cross-domain model transfer.

#### CRediT authorship contribution statement

**Yu Peng:** Methodology, Visualization, Software, Writing – original draft. **Erchao Li:** Writing – review & editing, Conceptualization, Supervision.

#### Declaration of competing interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

**Ethics approval:** This article does not contain any studies with human participants or animals performed by any of the authors.

#### Funding

This work was supported by the National Natural Science Foundation of China (No. 62063019), Gansu Provincial Key R&D Program of Science and Technology (No. 25YFGA030), Key Project of Natural Science Foundation of Gansu Province (No. 24JRRA173) and Gansu Provincial Merit-based PhD Program (No. 24JRRA205).

#### Declaration of generative AI in scientific writing

During the preparation of this work, the authors used GPT-4 and Deepseek-R1 to assist with improving readability and language. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

## Appendix A. Exponential convergence via Lyapunov stability analysis

### Notation and Definitions

$z$ : Latent variable (input to the generator  $G$ )  
 $Z$ : Domain of latent variables  
 $G(z)$ : Generator network mapping  $z$  to data space  
 $D(x)$ : Discriminator network output (probability of  $x$  being real)  
 $F(z)$ : Fitness function driving AFC dynamics  
 $\lambda$ : Regularization parameter for gradient penalty  
 $\tau$ : Noise strength in the SDE (controls exploration)  
 $\rho$ : Strong concavity constant of  $F(z)$   
 $L_F$ : Lipschitz constant of  $\nabla_z F(z)$   
 $L_D, L_G$ : Lipschitz constants of  $D$  and  $G$   
 $J_G(z)$ : Jacobian matrix of generator  $G$   
 $W_t$ : Wiener process (Brownian motion) in the SDE  
 $p_t(z)$ : Probability density of  $z$  at time  $t$   
 $p^*(z)$ : Stationary distribution (target)  
 $V(t)$ : Lyapunov function (KL divergence from  $p_t$  to  $p^*$ )  
 $\nabla_z$ : Gradient operator with respect to  $z$   
 $\nabla_z^2$ : Hessian operator with respect to  $z$   
 $\Delta_z$ : Laplacian operator with respect to  $z$   
 $\|\bullet\|$ : Euclidean norm  
 $I$ : Identity matrix  
 $KL(p||q)$ : Kullback-Leibler divergence between distributions  $p$  and  $q$   
 $\partial_{G_i} D$ : Partial derivative of  $D$  with respect to the  $i$ -th component of  $G(z)$   
 $\nabla_{G(z)} D$ : Gradient of  $D$  with respect to  $G(z)$

#### Theorem Statement:

Consider the adaptive feedback coupling (AFC) dynamics for latent variable  $z \in Z \subseteq \mathbb{R}^d$ , governed by the fitness function:

$$F(z) = \log D(G(z)) + \lambda \|\nabla_z D(G(z))\|^2$$

where  $\lambda > 0$  is a regularization parameter.

#### Assume:

**Smoothness:** the discriminator  $D$  and generator  $G$  twice continuously differentiable, with  $\nabla_z D(G(z))$  and  $\nabla_z^2 D(G(z))$  being Lipschitz continuous.

**Strong concavity:**  $F(z)$  is  $\rho$ -strongly concave, i.e.,  $\nabla^2 F(z) \leq -\rho I$  for all  $z \in Z$ .

**Exploration Noise:** The noise strength  $\tau > 0$  and regularization parameter  $\lambda$  satisfy  $\tau \geq \frac{L_F^2}{2\rho^2}$ .

Then, there exists a Lyapunov function  $V(t) = KL(p_t || p^*)$ , where  $p^*(z) \propto \exp(F(z)/\tau)$ , satisfies

$$V(t) \leq V(0)e^{-\rho t} + \frac{L_F^2}{2\tau\rho}$$

implying exponential convergence to a neighborhood of  $p^*$ , with convergence rate  $\rho$  and steady-state error bounded by  $\frac{L_F^2}{2\tau\rho}$ .

#### Proof

##### Step 1: Define AFC Dynamics as a Stochastic Differential Equation (SDE):

The AFC update rule combines gradient ascent on  $F(z)$  and stochastic exploration:

$$dz_t = \nabla_z F(z_t) dt + \sqrt{2\tau} dW_t$$

where  $W_t$  is a Wiener process. The drift term  $\nabla_z F(z)$  is derived as follows:

##### First term:

$$\nabla_z \log D(G(z)) = J_G(z)^T \frac{\nabla_{G(z)} D(G(z))}{D(G(z))}$$

where  $J_G(z)$  is the Jacobian of  $G$ .

##### Second Term:

$$\nabla_z \|\nabla_z D(G(z))\|^2 = 2\lambda \nabla_z^2 D(G(z)) \nabla_z D(G(z))$$

where  $\nabla_z^2 D(G(z)) = J_G(z)^T \nabla_{G(z)}^2 D(G(z)) J_G(z) + \sum_i \partial_{G_i} D(G(z)) \nabla_z^2 G_i(z)$

By Assumption 1,  $\nabla_z F(z)$  is Lipschitz continuous with constant  $L_F$ , bounded as:

$$L_F \leq \frac{L_D L_G}{\min_z D(G(z))} + 2\lambda (L_D L_G^2 + L_D L_G \|\nabla_z^2 G(z)\|)$$

##### Step 2: Fokker-Planck equation for latent distribution

The probability density  $p_t(z)$  evolves according to the Fokker-Planck equation:

$$\frac{\partial p_t(z)}{\partial t} = -\nabla_z \bullet [p_t(z) \nabla_z F(z)] + \tau \Delta_z p_t(z)$$

Boundary terms vanish due to the rapid decay of  $p_t(z)$  at infinity (Assumption 3).

### Step 3: Lyapunov function and time derivative

Define the Lyapunov function as the KL divergence :

$$V(t) = KL(p_t \| p^*) = \int_Z p_t(z) \ln \frac{p_t(z)}{p^*(z)} dz$$

Differentiate  $V(t)$  with respect to time:

$$\frac{dV}{dt} = \int_Z \frac{\partial p_t(z)}{\partial t} \ln \frac{p_t(z)}{p^*(z)} dz + \int_Z p_t(z) \frac{\partial}{\partial t} \ln \frac{p_t(z)}{p^*(z)} dz$$

The second term vanishes because  $\partial_t \ln p^*$ , and  $p^*$  is stationary.

Substitute the Fokker-Planck equation into the first term:

$$\frac{dV}{dt} \int_Z (-\nabla_z \bullet [p_t(z) \nabla_z F(z)] + \tau \Delta_z p_t(z)) \ln \frac{p_t(z)}{p^*(z)} dz$$

Expand the divergence and Laplacian terms using integration by parts (boundary terms vanish due to decay at infinity):

Drift Term (Gradient Flow):

$$-\int_Z \nabla_z \bullet [p_t(z) \nabla_z F(z)] \ln \frac{p_t(z)}{p^*(z)} dz = \int_Z p_t(z) \nabla_z F(z) \bullet \nabla_z \ln \frac{p_t(z)}{p^*(z)} dz$$

Diffusion Term (Noise):

$$\tau \int_Z \Delta_z p_t \ln \frac{p_t(z)}{p^*(z)} dz = -\tau \int_Z \nabla_z p_t(z) \bullet \nabla_z \ln \frac{p_t(z)}{p^*(z)} dz = -\tau \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz$$

Combined Result:

$$\frac{dV}{dt} = \int_Z p_t(z) \nabla_z F(z) \bullet \nabla_z \ln \frac{p_t(z)}{p^*(z)} dz - \tau \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz$$

### Step 4: Bounding term 1 via strong concavity

From the definition of  $p^*(z) \propto \exp(F(z)/\tau)$ , we have  $\nabla_z \ln p^*(z) = \frac{\nabla_z F(z)}{\tau}$ . Substituting into Term 1:

$$\text{Term1} = \int_Z p_t(z) \nabla_z F(z) \bullet \nabla_z \ln \frac{p_t(z)}{p^*(z)} dz = \tau \int_Z p_t(z) \nabla_z \ln p^*(z) \bullet \nabla_z \ln \frac{p_t(z)}{p^*(z)} dz$$

Expanding  $\nabla_z \ln \frac{p_t}{p^*} = \nabla_z \ln p_t - \nabla_z \ln p^*$ , using the Cauchy-Schwarz inequality and Young's inequality

$$\text{Term1} \leq \frac{\tau}{2} \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz + \frac{\tau}{2} \int_Z p_t(z) \|\nabla_z \ln p^*(z)\|^2 dz$$

Since  $\|\nabla_z \ln p^*\|^2 = \|\nabla_z F(z)/\tau\|^2 \leq L_F^2/\tau^2$ , by Lipschitz continuity (Assumption 1):

$$\text{Term1} \leq \frac{\tau}{2} \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz + \frac{L_F^2}{2\tau}$$

### Step 5: Apply Logarithmic Sobolev Inequality (LSI)

Since  $F(z)$  is  $\rho$ -strongly concave, the stationary distribution  $p^*(z) \propto \exp(F(z)/\tau)$  satisfies the LSI with constant  $\rho/\tau$ :

$$\text{Term2} = \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz \geq \frac{2\rho}{\tau} V(t)$$

### Step 6: Exponential convergence rate synthesis

Substitute the bounds for Term 1 (from Step 4) and Term 2 (from Step 5) into the time derivative of the Lyapunov function:

$$\frac{dV}{dt} \leq \frac{\tau}{2} \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz + \frac{L_F^2}{2\tau} - \tau \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz$$

Simplify the inequality:

$$\frac{dV}{dt} \leq -\frac{\tau}{2} \int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz + \frac{L_F^2}{2\tau}$$

From Step 5, the LSI implies:  $\int_Z p_t(z) \|\nabla_z \ln \frac{p_t(z)}{p^*(z)}\|^2 dz \geq \frac{2\rho}{\tau} V(t)$ , substitute this into the inequality for  $\frac{dV}{dt}$ :

$$\frac{dV}{dt} \leq -\frac{\tau}{2} \bullet \frac{2\rho}{\tau} V(t) + \frac{L_F^2}{2\tau}$$

$$\frac{dV}{dt} \leq -\rho V(t) + \frac{L_F^2}{2\tau}$$



Multiply by the integrating factor  $e^{\rho t}$ :

$$\frac{d}{dt}[V(t)e^{\rho t}] \leq \frac{L_F^2}{2\tau}e^{\rho t}$$

Integrate over  $t \in [0, T]$ :

$$V(T)e^{\rho T} \leq V(0) + \frac{L_F^2}{4\rho\tau}(e^{-\rho T} - 1)$$

$$V(T) \leq V(0)e^{-\rho T} + \frac{L_F^2}{2\rho\tau}(1 - e^{-\rho T})$$

As  $T \rightarrow \infty$ , the steady-state error is bounded by:

$$\lim_{T \rightarrow \infty} V(T) \leq \frac{L_F^2}{2\rho\tau}$$

To emphasize exponential convergence, rewrite the inequality as:

$$V(t) \leq V(0)e^{-\rho t} + \frac{L_F^2}{2\rho\tau}$$

Under the noise strength condition  $\tau > \frac{L_F^2}{2\rho^2}$ , the steady-state error is suppressed to  $\frac{L_F^2}{2\rho\tau}$ .

**Q.E.D**

## Appendix B. Convergence speed improvement via entropy compression

### Notation and Definitions

- $\mathbf{z} \in \mathbb{R}^d$ : Latent variable in  $d$ -dimensional latent space.
- $d$ : Dimensionality of the latent space.
- $\theta \in \Theta$ : Parameters of the generator  $G_\theta(\mathbf{z})$ .
- $\Sigma_G$ : Covariance matrix of the original latent distribution  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \Sigma_G)$ .
- $\Sigma_S$ : Covariance matrix of the structured latent distribution  $q(\mathbf{z}) = \mathcal{N}(\mu_S, \Sigma_S)$ .
- $\lambda_G, \lambda_S$ : Eigenvalues of isotropic covariance matrices ( $\Sigma_G = \lambda_G I, \Sigma_S = \lambda_S I$ ).
- $H_G, H_S$ : Entropy of the original distribution  $p(\mathbf{z})$  and structured distribution  $q(\mathbf{z})$ .
- $\mu_S$ : Mean vector of the structured distribution  $q(\mathbf{z})$ .
- $\delta_z$ : Perturbation term for latent variables, where  $\mathbf{z} = \mu_S + \delta_z$ .
- $G_\theta(\mathbf{z})$ : Deterministic generator mapping parameterized by  $\theta$ .
- $p(\mathbf{z}), q(\mathbf{z})$ : Original and structured latent distributions (Gaussian).
- $\mathcal{L}_G(\theta)$ : Generator's loss function, assumed  $L$ -smooth and  $\mu$ -strongly convex.
- $\mathbf{x}_{\text{target}}$ : Target data (implicitly defined in the loss function).
- $\theta^*$ : Optimal parameters minimizing the loss function.
- $\Delta_0$ : Initial suboptimality gap:  $\mathcal{L}_G(\theta_0) - \mathcal{L}_G(\theta^*)$ .
- $J_\theta(\mathbf{z})$ : Jacobian matrix of the generator with respect to parameters  $\theta$ .
- $\Delta_J$ : Perturbation term of the Jacobian, bounded by  $\|\Delta_J\|_F \leq C_J \|\delta_z\|$ .
- $C_J$ : Lipschitz constant for local Jacobian stability.
- $I_q(\theta)$ : Fisher Information Matrix (FIM) under the structured distribution  $q(\mathbf{z})$ .
- $\sigma^2$ : Variance of additive Gaussian noise  $\in \mathcal{N}(\mathbf{0}, \sigma^2 I)$ .
- $L_G, L_S$ : Smoothness constants of the loss under  $p(\mathbf{z})$  and  $q(\mathbf{z})$ .
- $\mu_G^{\text{conv}}, \mu_S^{\text{conv}}$ : Strong convexity parameters of the loss under  $p(\mathbf{z})$  and  $q(\mathbf{z})$ .
- $t_G, t_S$ : Iteration complexity (number of steps to convergence) for  $p(\mathbf{z})$  and  $q(\mathbf{z})$ .
- $\kappa_G, \kappa_S$ : Condition numbers of the Jacobian matrix under  $p(\mathbf{z})$  and  $q(\mathbf{z})$ .
- $\lambda_{\min}(\bullet), \lambda_{\max}(\bullet)$ : Minimum and maximum eigenvalue of a matrix.
- $\det(\Sigma)$ : Determinant of a covariance matrix.
- $\text{Tr}(\Sigma)$ : Trace of a covariance matrix.
- $\|\bullet\|_F$ : Frobenius norm of a matrix.

### Theorem Statement

Let the latent variable space of the generator be  $\mathbf{z} \subseteq \mathbb{R}^d$ . Suppose the generator  $G_\theta(\mathbf{z})$  is a deterministic nonlinear mapping with parameters  $\theta \in \Theta$ , and the original latent distribution is  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \Sigma_G)$  with the entropy  $H_G = \frac{1}{2} \ln((2\pi e)^d \det(\Sigma_G))$ , where  $\Sigma_G$  is the covariance matrix. After evolutionary optimization, the structured latent distribution becomes  $q(\mathbf{z}) = \mathcal{N}(\mu_S, \Sigma_S)$  with the entropy  $H_S = \frac{1}{2} \ln((2\pi e)^d \det(\Sigma_S))$ , and  $\det(\Sigma_S) \ll \det(\Sigma_G)$ . Assume the generator's loss function  $\mathcal{L}_G(\theta)$  is  $L$ -smooth and  $\mu^{\text{conv}}$ -strongly convex in  $\theta$  and the Jacobian  $J_\theta(\mathbf{z}) = \frac{\partial G_\theta(\mathbf{z})}{\partial \theta}$  is Lipschitz-continuous in  $\mathbf{z}$ . Under the following additional assumptions:

**1. Local Jacobian Stability:** For  $\mathbf{z} \in \text{supp}(q(\mathbf{z}))$ ,  $\exists C_J > 0$  such that

$$\|J_\theta(\mathbf{z}) - J_\theta(\mu_S)\|_F \leq C_J \|\mathbf{z} - \mu_S\|$$

2. **Local Strong Convexity:**  $\mathcal{L}_G(\theta)$  is  $\mu$ -strongly convex in a neighborhood of the optimization trajectory.
3. **Isotropic Covariance:**  $\Sigma_G = \lambda_G I$  and  $\Sigma_S = \lambda_S I$ , where  $\lambda_G \gg \lambda_S$ .
4. **Concentration of  $q(\mathbf{z})$ :**  $\|\Sigma_S\|_F \rightarrow 0$ , ensuring  $\mathbf{z}$  concentrates near  $\mu_S$ .

Then, the convergence rate improves from  $O\left(\frac{L_G}{\mu_G} \ln\left(\frac{1}{\epsilon}\right)\right)$  to  $O\left(\frac{L_S}{\mu_S} \ln\left(\frac{1}{\epsilon}\right)\right)$ , with an acceleration ratio  $O\left(\frac{L_G/\mu_G}{L_S/\mu_S}\right)$ .

#### Proof

##### Setup 1 Fisher Information Matrix (FIM) Stabilization

Under the compressed latent distribution  $q(\mathbf{z})$ , samples satisfy  $\mathbf{z} = \mu_S + \delta_z$ , where  $\|\delta_z\|_F = O(\sqrt{\text{Tr}(\Sigma_S)}) \rightarrow 0$ . By the Local Jacobian Stability assumption:

$$J_\theta(\mathbf{z}) = J_\theta(\mu_S) + \Delta J, \quad \|\Delta J\|_F \leq C_J \|\delta_z\|$$

Under additive Gaussian noise  $\epsilon \sim N(0, \sigma^2 I)$ . The Fisher information matrix for  $\theta$  is:

$$I_q(\theta) = \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})]$$

Substituting  $J_\theta(\mathbf{z}) = J_\theta(\mu_S) + \Delta J$ :

$$I_q(\theta) = \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mu_S)^T J_\theta(\mu_S) + J_\theta(\mu_S)^T \Delta J + \Delta J^T J_\theta(\mu_S) + \Delta J^T \Delta J]$$

Due to symmetry of  $q(\mathbf{z})$ ,  $\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [\Delta J] = 0$ . Expanding and neglecting higher-order terms  $O(\|\delta_z\|^2)$ :

$$I_q(\theta) \approx \frac{1}{\sigma^2} ((J_\theta(\mu_S)^T J_\theta(\mu_S) + \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [\Delta J^T J_\theta(\mu_S) + J_\theta(\mu_S)^T \Delta J])$$

$$I_q(\theta) \approx \frac{1}{\sigma^2} J_\theta(\mu_S)^T J_\theta(\mu_S)$$

For the original distribution  $p(\mathbf{z})$ , the FIM  $I_p(\theta)$  involves averaging over a broader support, leading to smaller minimum eigenvalues due to Jacobian variability. Thus:

$$\lambda_{\min}(I_q) \geq \frac{1}{\sigma^2} \lambda_{\min}(J_\theta(\mu_S)^T J_\theta(\mu_S)) \gg \lambda_{\min}(I_p)$$

#### Step 2: Smoothness and strong convexity analysis

##### 2.1 Smoothness Constant $L$

The smoothness constant  $L$  is bounded by the spectral norm of the Hessian  $\nabla^2 L_G(\theta)$ . For mean squared error loss:

$$\mathcal{L}_G(\theta) = (\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [\|G_\theta(\mathbf{z}) - \mathbf{x}_{\text{target}}\|^2])$$

the Hessian is approximated as:

$$\nabla_{\theta}^2 \mathcal{L}_G(\theta) \approx 2 \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})]$$

Thus,  $L$  satisfies:

$$L = 2 \lambda_{\max}(\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})])$$

By the triangular inequality for spectral norms and the Local Jacobian Stability assumption:

$$\lambda_{\max}(\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})]) \leq \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [\lambda_{\max}(J_\theta(\mathbf{z})^T J_\theta(\mathbf{z}))]$$

By perturbation analysis:

$$\lambda_{\max}(J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})) \leq \lambda_{\max}(J_\theta(\mu_S)^T J_\theta(\mu_S)) + O(\sqrt{\text{Tr}(\Sigma_S)})$$

Since  $\sqrt{\text{Tr}(\Sigma_S)} \rightarrow 0$ , we have:

$$L_S = 2 \lambda_{\max}(J_\theta(\mu_S)^T J_\theta(\mu_S)) \leq L_G$$

##### 2.2 Strong convexity parameter $\mu$

The strong convexity parameter  $\mu$  is bounded by:

$$\mu = 2 \lambda_{\min}(\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})])$$

Using matrix concentration inequalities:

$$\lambda_{\min}(\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [J_\theta(\mathbf{z})^T J_\theta(\mathbf{z})]) \geq \lambda_{\min}(J_\theta(\mu_S)^T J_\theta(\mu_S)) - O(\sqrt{\text{Tr}(\Sigma_S)})$$

Since  $\sqrt{\text{Tr}(\Sigma_S)} \rightarrow 0$ , we have:

$$\mu_S \geq 2 \lambda_{\min}(J_\theta(\mu_S)^T J_\theta(\mu_S)) \geq \mu_G$$

#### Step 3: Convergence rate improvement

### 3.1 Iteration complexity

For  $L$ -smooth and  $\mu$ -strongly convex losses, gradient descent requires:

$$t \geq \frac{L}{\mu} \ln\left(\frac{\Delta_0}{\epsilon}\right)$$

where  $\Delta_0 = \mathcal{L}_G(\theta_0) - \mathcal{L}_G(\theta^*)$ . Substituting the parameters:

$$\text{Original distribution: } t_G = \frac{L_G}{\mu_G^{\text{conv}}} \ln\left(\frac{\Delta_0}{\epsilon}\right)$$

$$\text{Structured distribution: } t_S = \frac{L_S}{\mu_S^{\text{conv}}} \ln\left(\frac{\Delta_0}{\epsilon}\right)$$

The acceleration ratio is:

$$\frac{t_G}{t_S} = \frac{L_G / \mu_G^{\text{conv}}}{L_S / \mu_S^{\text{conv}}}$$

### 3.2 Entropy-acceleration relationship

From the entropy expressions:

$$H_G - H_S = \frac{1}{2} \ln\left(\frac{\det(\Sigma_G)}{\det(\Sigma_S)}\right) = \frac{1}{2} \ln\left(\frac{\prod_{i=1}^d \lambda(\Sigma_G)}{\prod_{i=1}^d \lambda(\Sigma_S)}\right)$$

Under isotropic covariance assumptions, we have:

$$H_G - H_S = \frac{d}{2} \ln\left(\frac{\lambda(\Sigma_G)}{\lambda(\Sigma_S)}\right)$$

To rigorously link entropy reduction to the acceleration ratio, we analyze the condition number  $\kappa = \frac{\lambda_{\max}(J^T J)}{\lambda_{\min}(J^T J)}$ .

Original distribution  $p(z)$ :

The Jacobian  $J_\theta(z)$  varies widely over  $z$   $p(z)$ , leading to large  $\lambda_{\max}$  and small  $\lambda_{\min}$ :

$$\kappa_G = \frac{\lambda_{\max}(\mathbb{E}_{z \sim p(z)} [J^T J])}{\lambda_{\min}(\mathbb{E}_{z \sim p(z)} [J^T J])} \gg 1$$

Structured distribution  $q(z)$ :

$J_\theta(z)$  concentrates near  $J_\theta(\mu_S)$ , so:

$$\kappa_S = \frac{\lambda_{\max}(J_\theta(\mu_S)^T J_\theta(\mu_S))}{\lambda_{\min}(J_\theta(\mu_S)^T J_\theta(\mu_S))}$$

If  $J_\theta(\mu_S)$  is well-conditioned ( $\kappa_S \approx 1$ ), then:

$$\frac{L_G / \mu_G^{\text{conv}}}{L_S / \mu_S^{\text{conv}}} \approx \kappa_G \gg 1$$

Using the relationship  $\det(\Sigma) = \prod_{i=1}^d \lambda_i(\Sigma)$  and isotropy  $\lambda_i(\Sigma) = \lambda(\Sigma)$ :

$$\det(\Sigma_G) = \lambda_G^d, \det(\Sigma_S) = \lambda_S^d$$

Thus:

$$H_G - H_S = \frac{d}{2} \ln\left(\frac{\lambda_G}{\lambda_S}\right)$$

Under the local linearity assumption ( $J_\theta(z) \approx J_\theta(\mu_S)$  for  $z \sim p(z)$ ), the condition number scales as:

$$\kappa_G \propto \frac{\lambda_G}{\lambda_S}$$

Therefore:

$$\frac{L_G / \mu_G^{\text{conv}}}{L_S / \mu_S^{\text{conv}}} = \kappa_G \propto \exp\left(\frac{2}{d} (H_G - H_S)\right)$$

This directly ties the acceleration ratio to the entropy reduction. The iteration complexity improves by a factor of  $O\left(\frac{L_G / \mu_G}{L_S / \mu_S}\right)$ , which scales exponentially with the entropy reduction  $H_S - H_G$ . A lower-entropy latent space reduces the effective search volume in parameter space, accelerating convergence.

Q.E.D.

## Appendix C. Theoretical proof of iteration complexity reduction under guideline sharing mechanism

### Notation and Definitions

1. Task setup

$T$ : number of tasks.

$D_{base}$ : Dimension of the over-parameterized base network  $\varepsilon$ .

$D_{task}$ : Dimension of a single task's network in non-shared training.

$S \in (0, 1)$ : Sharing rate, enforcing  $\|M_t\|_0 = S \bullet D_{task}$  for each task mask  $M_t$ .

$C \in (0, 1)$ : Overlap ratio between masks of two tasks, defined as  $C = \frac{\|M_i \cap M_j\|_0}{S \bullet D_{task}}$ .

2. Loss function

$L_t(\theta_t)$ : Loss function for task  $t$ , parameterized by  $\theta_t$ .

$\mu$ : Strong convexity constant of  $L_t$ .

$L$ : Smoothness constant of  $L_t$ .

3. Shared mechanism

$\theta_\varepsilon \in \mathbb{R}^D$ : Parameters of the base network  $\varepsilon$ .

$M_t \odot \theta_\varepsilon$ : Subnetwork for task  $t$ , where  $\odot$  denotes element-wise multiplication.

$\nabla_{shared} L$ : Gradient of the shared objective  $\frac{1}{T} \sum_{t=1}^T L_t(M_t \odot \theta_\varepsilon)$ .

4. Convergence metrics

$\kappa = L/\mu$ : Condition number in the non-shared scenario.

$\kappa' = L'/\mu'$ : Effective condition number in the shared scenario.

$\Delta_0^{non}$ : Initial distance to optimum in non-shared training.

$\Delta_0^{shared}$ : Initial distance to optimum in shared training.

**Theorem Statement:**

Consider  $T$  tasks with a shared base network under the following conditions:

with sharing rate  $S \in (0, 1)$ . Let  $K_{shared}$  and  $K_{nonshared}$  denote the iteration complexity to reach an  $\epsilon$ -optimal solution under shared and non-shared scenarios, respectively. Under the following assumptions:

**Task Loss Functions:** Each task loss  $L_t(\theta)$  is  $\mu$ -strongly convex and  $L$ -smooth.

**Shared Parameter Space:** Sharing rate  $S \in [0.5, 1)$ , where task masks  $\{M_t\}$  are generated via uniform random sampling. The expected overlap ratio satisfies  $E[C] \geq \frac{SD_{task}}{D_{base}} = C_0 > 0$ . The base network  $\varepsilon$  is over-parameterized, the base network dimension  $D_{base} \geq T \bullet SD_{task}$ .

**Task Gradients Alignment (Xu et al., 2025):**  $\exists \gamma > 0$  s.t.  $E[\nabla L_i \cdot \nabla L_j] \geq \gamma \forall i \neq j$ .

**Strong Convexity Preservation:** The shared objective  $L_{shared}(\theta_\varepsilon) = \frac{1}{T} \sum_{t=1}^T L_t(M_t \odot \theta_\varepsilon)$  is  $\mu'$ -strongly convex over the shared subspace  $\bigcup_{t=1}^T \text{supp}(M_t)$ , with  $\mu' \geq \mu \bullet C_0$ .

Then, there exists a constant  $\alpha \in (0, 1)$ , such that the iteration complexity satisfies:

$$K_{shared} \leq \alpha \cdot K_{nonshared}$$

**Proof**

**Step 1: Effective parameter space reduction**

**Lemma 1 (dimension reduction under masking)**

Under uniform random masking with sharing rate  $S$ , the effective parameter dimension of the shared mechanism satisfies:

$$E[Dim_{shared}] = SD_{task}(1 + (T-1)C_0) < TSD_{task} = Dim_{nonshared}$$

**Proof:**

Let  $M_t \subseteq \{1, 2, \dots, D_{base}\}$  denote the indices of non-zero entries in the mask for task  $t$ . By uniform random sampling, each index is selected independently with probability  $p = \frac{SD_{task}}{D_{base}}$ .

The expected size of the union of masks is:

$$E[|\bigcup_{t=1}^T M_t|] = D_{base}(1 - (1-p)^T)$$

Using the over-parameterization condition  $D_{base} \geq T \bullet SD_{task}$ , we have  $p \leq \frac{1}{T}$ . Expanding via the binomial approximation for small  $p$ :

$$E[|\bigcup_{t=1}^T M_t|] \approx D_{base} \left( Tp - \frac{T(T-1)}{2} p^2 \right) \leq TSD_{task} \left( 1 - \frac{(T-1)SD_{task}}{2D_{base}} \right)$$

Substituting  $C_0 = \frac{SD_{task}}{D_{base}} \leq \frac{1}{T}$ :

$$E[Dim_{shared}] \leq SD_{task}(TC_0 + 1 - C_0) < TSD_{task}$$

The inequality follows from  $\frac{SD_{task}}{D_{base}} \leq \frac{1}{T}$ . Thus, shared training reduces the effective dimension.

**Step 2: Condition number improvement**

**Lemma 2 (Effective smoothness under gradient alignment)**

Under gradient alignment  $E[\nabla L_i \cdot \nabla L_j] \geq \gamma$ , the effective smoothness constant  $L'$  of  $L_{shared}$  satisfies:

$$L' \leq \frac{L^2 + (T-1)\gamma}{T}$$

**Proof:**

The gradient of the shared objective is:

$$\nabla L_{shared} = \frac{1}{T} \sum_{t=1}^T L_t(M_t \odot \theta_\varepsilon)$$

Taking expectation over the gradient norm:

$$\mathbb{E}[\|\nabla_{\text{shared}} L\|^2] = \frac{1}{T^2} \left( \sum_{t=1}^T \mathbb{E}\|\nabla L_t\|^2 + \sum_{i \neq j} \mathbb{E}[\nabla L_i \bullet \nabla L_j] \right)$$

By L-smoothness of  $L_t$ ,  $\mathbb{E}\|\nabla L_t\|^2 \leq L^2$ , and gradient alignment  $\mathbb{E}[\nabla L_i \bullet \nabla L_j] \geq \gamma$ :

$$\mathbb{E}[\|\nabla_{\text{shared}} L\|^2] \leq \frac{1}{T^2} (TL^2 + T(T-1)\gamma) = \frac{L^2 + (T-1)\gamma}{T}$$

Corollary, the effective smoothness constant is (Zhu et al., 2021):

$$L' = \sqrt{\frac{L^2 + (T-1)\gamma}{T}}$$

**Lemma 3 (Strong convexity preservation)**

The shared objective  $L_{\text{shared}}$  satisfies  $\mu'$ -strongly convexity with  $\mu' \geq \mu \bullet C_0$

**Proof:**

For any  $\theta_1, \theta_2 \in \mathbb{R}^{D_{\text{base}}}$ , by the  $\mu$ -strongly convexity of  $L_t$ :

$$L_t(M_t \odot \theta_2) \geq L_t(M_t \odot \theta_1) + \nabla L_t(M_t \odot \theta_1)^\top (M_t \odot \theta_1) + \frac{\mu}{2} \|M_t \odot (\theta_2 - \theta_1)\|^2$$

Averaging over tasks and using  $\mathbb{E}\|M_t \odot (\theta_2 - \theta_1)\|^2 \geq C_0 \|\theta_2 - \theta_1\|^2$  (Xia et al., 2022):

$$L_{\text{shared}}(\theta_2) \geq L_{\text{shared}}(\theta_1) + \nabla L_{\text{shared}}(\theta_1)^\top (\theta_2 - \theta_1) + \frac{\mu}{2T} \|M_t \odot (\theta_2 - \theta_1)\|^2$$

Taking expectation over masks and using  $\mathbb{E}\|M_t \odot (\theta_2 - \theta_1)\|^2 \geq C_0 \|\theta_2 - \theta_1\|^2$ :

$$\mathbb{E}[L_{\text{shared}}(\theta_2)] \geq \mathbb{E}[L_{\text{shared}}(\theta_1)] + \mathbb{E}[\nabla L_{\text{shared}}(\theta_1)]^\top (\theta_2 - \theta_1) + \frac{\mu C_0}{2} \|\theta_2 - \theta_1\|^2$$

Thus,  $\mu' \geq \mu \bullet C_0$ .

**Step 3: Mask-guided pruning as a contraction mapping**

**Lemma 4 (Error contraction)**

When  $S \geq 0.5$ , iterative magnitude-based pruning satisfies (Ma et al., 2019):

$$\|\theta_\epsilon^{(k)} - \theta_\epsilon^*\| \leq \beta^k \|\theta_\epsilon^{(0)} - \theta_\epsilon^*\|$$

where  $\beta = \sqrt{S}$ .

**Proof:**

Let  $I_k \subseteq \{1, 2, \dots, D_{\text{task}}\}$  denote the indices of the retained parameters. The pruning error  $\delta^{(k+1)} = \theta_\epsilon^{(k)} - \theta_\epsilon^*$  satisfies:

$$\|\delta^{(k+1)}\|^2 = \sum_{i \in I_k} (\delta_i^{(k)})^2$$

By magnitude pruning, the removed parameters satisfy:

$$\sum_{i \notin I_k} (\delta_i^{(k)})^2 \geq (1-S) \|\delta^{(k)}\|^2$$

Substituting the pruned error bound into the decomposition:

$$\|\delta^{(k+1)}\|^2 = \sum_{i \in I_k} (\delta_i^{(k)})^2 = \|\delta^{(k)}\|^2 - \sum_{i \notin I_k} (\delta_i^{(k)})^2$$

Using  $\sum_{i \notin I_k} (\delta_i^{(k)})^2 \geq (1-S) \|\delta^{(k)}\|^2$ :

$$\|\delta^{(k+1)}\|^2 \leq \|\delta^{(k)}\|^2 - (1-S) \|\delta^{(k)}\|^2 = S \|\delta^{(k)}\|^2$$

Thus:

$$\beta = \sqrt{S}$$

**Step 4: Iteration complexity comparison**

**Lemma 5 (Initialization advantage)**

The shared initial distance satisfies:

$$\Delta_0^{\text{shared}} = \|\theta_\epsilon^{(0)} - \theta_\epsilon^*\| \leq \sqrt{T} \Delta_0^{\text{non}}$$

**Proof:**

By the optimal subnetwork approximation  $\|M_t \odot \theta_\epsilon^* - \theta_\epsilon^{\text{non}^*}\| \leq \epsilon_0$ :

$$\|\theta_\epsilon^{(0)} - \theta_\epsilon^*\| \leq \sqrt{T} \max_t \|\theta_t^{\text{non}(0)} - \theta_t^{\text{non}^*}\| = \sqrt{T} \Delta_0^{\text{non}}$$

**Final complexity analysis**



The iteration complexity for  $\epsilon$ -optimality in shared training is:

$$K_{shared} = O\left(\kappa' \log \frac{\sqrt{T} \Delta_0^{non}}{\epsilon}\right), \kappa' = \frac{L'}{\mu'} = \frac{\sqrt{\frac{L^2 + (T-1)\gamma}{T}}}{\mu C_0}$$

For non-shared training:

$$K_{nonshared} = T \cdot O\left(\kappa \log \frac{\Delta_0^{non}}{\epsilon}\right), \kappa = \frac{L}{\mu}$$

The improvement factor  $\alpha$  is:

$$\alpha = \frac{\kappa'}{\kappa} \cdot \frac{\log(\sqrt{T} \Delta_0^{non}/\epsilon)}{T \log(\Delta_0^{non}/\epsilon)} = \sqrt{\frac{1+(T-1)\gamma/L^2}{T}} \cdot \frac{1 + \frac{\log \sqrt{T}}{\log(\Delta_0^{non}/\epsilon)}}{T}$$

Under  $\gamma \leq L^2$ , we have  $\alpha < 1$ .

### Conclusion

Under the stated assumptions, the shared mechanism achieves strictly lower iteration complexity than independent training. The improvement factor  $\alpha$  explicitly depends on  $S$ ,  $C_0$ ,  $\gamma$  and  $T$  providing theoretical guidance for practical multi-task learning systems.

Q.E.D

### Appendix D. Impact of different key hyperparameter

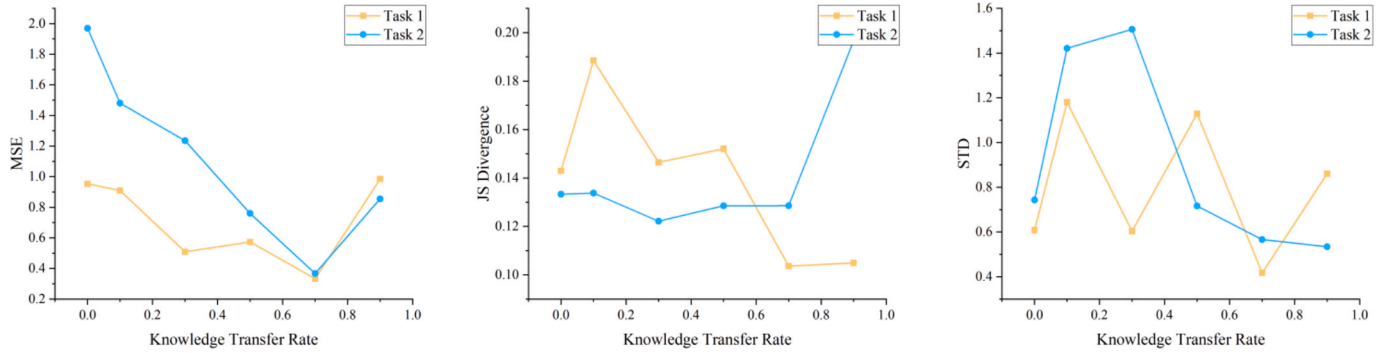


Fig. 1. Impact of different knowledge transfer rate

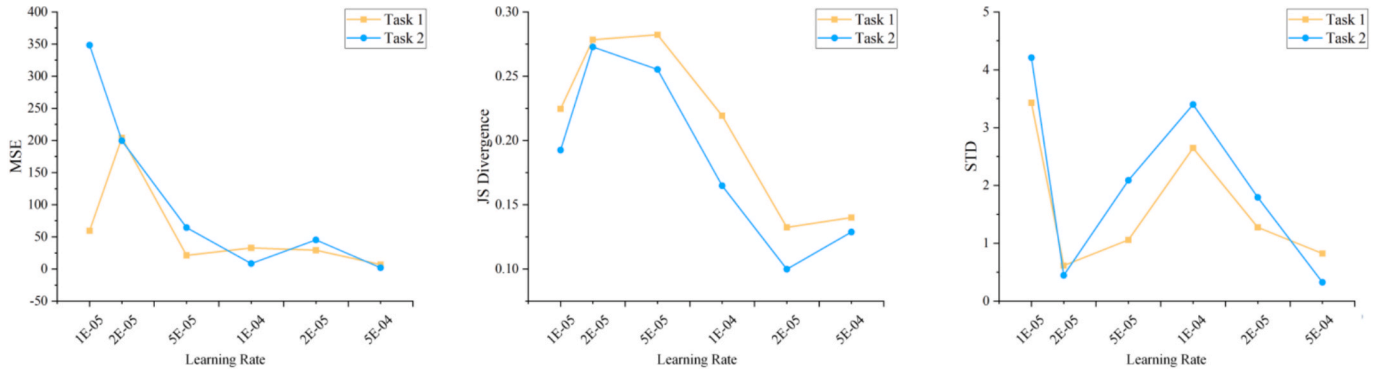


Fig. 2. Illustration of proposed guideline sharing between multi-task subnetworks

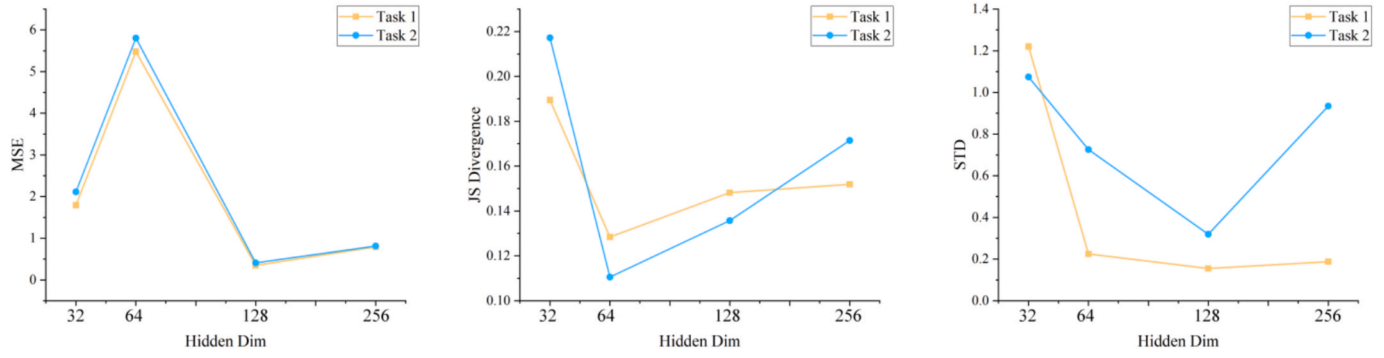


Fig. 3. Illustration of proposed guideline sharing between multi-task subnetworks

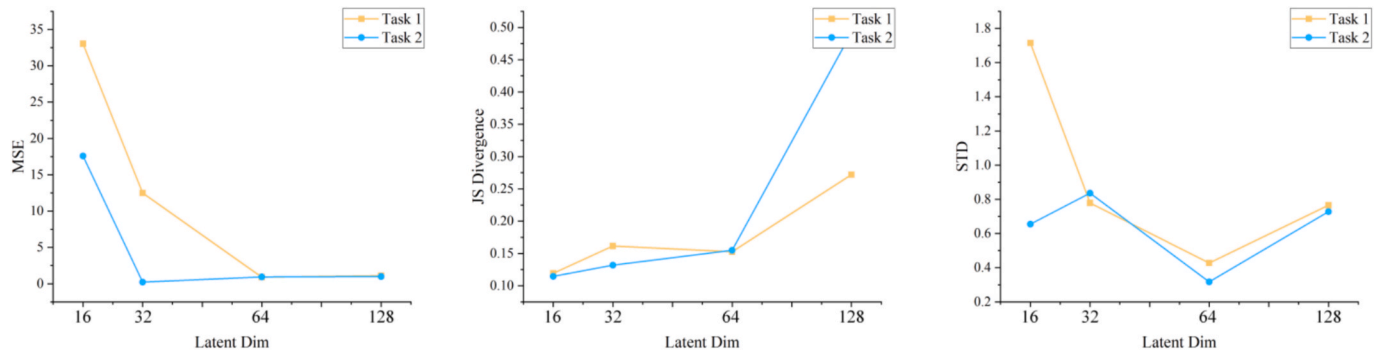


Fig. 4. Illustration of proposed guideline sharing between multi-task subnetworks

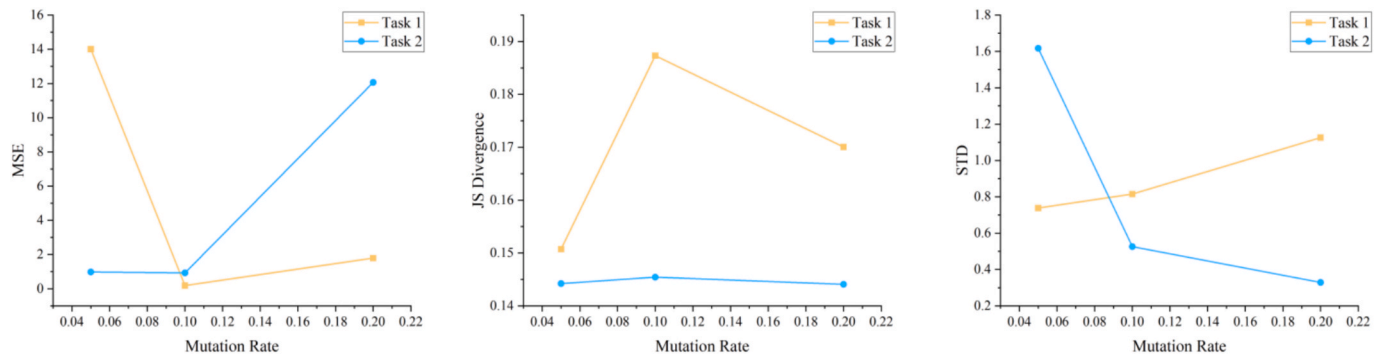


Fig. 5. Illustration of proposed guideline sharing between multi-task subnetworks

## Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## References

- Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, 28, 525–552.
- Asokan, S., & Spider, S. C. S. (2023). Gan: Leveraging friendly neighbors to accelerate GAN training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3883–3893).
- Chen, H., Huang, H., Zheng, Y., et al. (2024). A load forecasting approach for integrated energy systems based on aggregation hybrid modal decomposition and combined model. *Applied Energy*, 375, Article 124166.
- Chen, Z. W., Lv, Z. G., Di, R. H., et al. (2023). A novel virtual sample generation method to improve the quality of data and the accuracy of data-driven models. *Neurocomputing*, 548, Article 126380.
- Chen, Z. S., Zhu, M. Y., He, Y. L., et al. (2021). Quantile regression CGAN based virtual samples generation and its applications to process modeling. *CIESC Journal*, 72(3), 1529–1538.
- Cortez, P., Cerdeira, A., Almeida, F., et al. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4), 547–553.
- Cui, C., Tang, J., Xia, H., et al. (2023). Virtual sample generation method based on generative adversarial fuzzy neural network. *Neural Computing and Applications*, 35(9), 6979–7001.
- Fan, Z. (2022). Improving the accuracy of the neuroevolution machine learning potential for multi-component systems. *Journal of Physics: Condensed Matter*, 34(12), Article 125902.
- Fan, W., Huang, H., Liang, C., et al. (2023). Unsupervised meta-learning via spherical latent representations and dual VAE-GAN. *Applied Intelligence*, 53(19), 22775–22788.
- Frankle, J., Dziugaite, G. K., Roy, D., et al. (2020). Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning* (pp. 3259–3269). PMLR.

- Galván, E., & Mooney, P. (2021). Neuroevolution in deep neural networks: Current trends and future challenges. *IEEE Transactions on Artificial Intelligence*, 2(6), 476–493.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
- Gui, J., Sun, Z., Wen, Y., et al. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, 35(4), 3313–3332.
- Gutiérrez, S., & Tardaguila, J. (2023). Evolutionary conditional GANs for supervised data augmentation: The case of assessing berry number per cluster in grapevine. *Applied Soft Computing*, 147, Article 110805.
- Huang, W., Li, Y., & Huang, Y. (2020). Deep hybrid neural network and improved differential neuroevolution for chaotic time series prediction. *IEEE Access*, 8, 159552–159565.
- Karras, T., Aittala, M., Hellsten, J., et al. (2020). Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33, 12104–12114.
- Khan, S., Yin, P., Guo, Y., et al. (2024). Heterogeneous transfer learning: Recent developments, applications, and challenges. *Multimedia Tools and Applications*, 83(27), 69759–69795.
- Kim, D. H., & Song, B. C. (2021). Virtual sample-based deep metric learning using discriminant analysis. *Pattern Recognition*, 110, Article 107643.
- Li, D. C., Lin, W. K., Chen, C. C., et al. (2018). Rebuilding sample distributions for small dataset learning. *Decision Support Systems*, 105, 66–76.
- Li, D. C., & Wen, I. H. (2014). A genetic algorithm-based virtual sample generation technique to improve small data set learning. *Neurocomputing*, 143, 222–230.
- Liu, L., Liu, Z., Chang, J., et al. (2023). MGGAN: A multi-generator generative adversarial network for breast cancer immunohistochemical image generation. *Heliyon*, 9(10).
- Liu, Z., Wang, H., & Jin, Y. (2023). Performance indicator-based adaptive model selection for offline data-driven multiobjective evolutionary optimization. *IEEE Transactions on Cybernetics*, 53(10), 6263–6276.
- Liu, F., Wang, H., Zhang, J., et al. (2022). EvoGAN: An evolutionary computation assisted GAN. *Neurocomputing*, 469, 81–90.
- Ma, D. Z., Hu, X. G., Sun, Q. Y., et al. (2019). Cyber-physical abnormality diagnosis method using data feature fusion for pipeline network. *Acta Automatica Sinica*, 45(1), 163–173.
- Ming-Fei, Hu., Xin, Z., & Jian-Wei, L. (2022). Survey on deep generative model. *Acta Automatica Sinica*, 48(1), 40–74.
- Niyogi, P., Girosi, F., & Poggio, T. (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86(11), 2196–2209.
- Nouri, H. E., Ghandri, A., Driss, O. B., et al. (2023). Bi-EvoGAN: Bi-level Evolutionary approach for generative Adversarial Networks. *Applied Soft Computing*, 147, Article 110738.
- Perera, Y. S., Ratnaweera, D., Dasanayaka, C. H., et al. (2023). The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. *Engineering Applications of Artificial Intelligence*, 121, Article 105988.
- Poggio, T., & Vetter, T. (1992, 1347.). Recognition and structure from one 2D model view: Observations on prototypes, object classes and symmetries. *Laboratory Massachusetts Institute of Technology*.
- Qiao, J. F., Guo, Z. H., & Tang, J. (2020). Virtual sample generation method based on improved megatrend diffusion and hidden layer interpolation with its application. *CIESC Journal*, 71(12), 5681–5695.
- Revathi, S. A., & Sathish Babu, B. (2024). Synthesizing realistic knee MRI images: A VAE-GAN approach for enhanced medical data augmentation. *International Journal of Advanced Computer Science and Applications*, 15(11).
- Shen, L., & Qian, Q. (2022). A virtual sample generation algorithm supporting machine learning with a small-sample dataset: A case study for rubber materials. *Computational Materials Science*, 211, Article 111475.
- Shuvo, M. M. H., Islam, S. K., Cheng, J., et al. (2022). Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1), 42–91.
- Sun, Q., & Ge, Z. (2021). A survey on deep learning for data-driven soft sensors. *IEEE Transactions on Industrial Informatics*, 17(9), 5853–5866.
- Sun, T., Shao, Y., Li, X., et al. (2020). Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8936–8943).
- Wang, X., & Han, L. (2020). Data supplement for a soft sensor using a new generative model based on a variational autoencoder and Wasserstein GAN. *Journal of Process Control*, 85, 91–99.
- Wang, R., Li, T., Gao, Z., et al. (2024). A generative adversarial networks based methodology for imbalanced multidimensional time-series augmentation of complex electromechanical systems. *Applied Soft Computing*, 153, Article 111301.
- Wang, Z., Pei, Y., & Li, J. (2023). A survey on search strategy of evolutionary multi-objective optimization algorithms. *Applied Sciences*, 13(7), 4643.
- Wang, D. D., Tang, J., Xia, H., et al. (2022). Virtual sample generation method based on hybrid optimization with multi-objective PSO. *Acta Automatica Sinica*, 45(x), 1–22.
- Wang, C., Xu, C., Yao, X., et al. (2019). Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*, 23(6), 921–934.
- Wu, X., Feng, J., Shang, R., et al. (2024). Multi-task multi-objective evolutionary network for hyperspectral image classification and pansharpening. *Information Fusion*, 108, Article 102383.
- Xia, H., Tang, J., Qiao, J. F., et al. (2022). DF classification algorithm for constructing a small sample size of data-oriented DF regression model. *Neural Computing and Applications*, 34(4), 2785–2810.
- Xu, Y., Zhu, Q. X., Ke, W., et al. (2025). Virtual sample generation for soft-sensing in small sample scenarios using glow-embedded variational autoencoder. *Computers & Chemical Engineering*, 193, Article 108925.
- Yu, H., Fan, X., Wang, G., et al. (2023). VSG 3 a 2: A genetic algorithm-based virtual sample generation approach using information gain and acceptance-rejection sampling. *IEEE Transactions on Evolutionary Computation*.
- Yu, L., Zhang, X., & Yin, H. (2022). An extreme learning machine based virtual sample generation method with feature engineering for credit risk assessment with data scarcity. *Expert Systems with Applications*, 202, Article 117363.
- Zhang, H., Hao, K., Gao, L., et al. (2022). Optimizing deep neural networks through neuroevolution with stochastic gradient descent. *IEEE Transactions on Cognitive and Developmental Systems*, 15(1), 111–121.
- Zhang, X. H., Xu, Y., He, Y. L., et al. (2021). Novel manifold learning based virtual sample generation for optimizing soft sensor with small data. *ISA transactions*, 109, 229–241.
- Zhu, D., Cheng, X., Zhang, F., et al. (2020). Spatial interpolation using conditional generative adversarial neural networks. *International Journal of Geographical Information Science*, 34(4), 735–758.
- Zhu, Q. X., Liu, D. P., Xu, Y., et al. (2021). Novel space projection interpolation based virtual sample generation for solving the small data problem in development soft sensor. *Chemometrics and Intelligent Laboratory Systems*, 217, Article 104425.

# Competitive-Cooperative Multi-Task Optimization Algorithm with Historical Success Archive

ERCHAO LI, HONGXU LI, and YU PENG, College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China

Multi-task optimization plays a pivotal role in various scientific and engineering disciplines. Within this field, Competitive Multi-Task Optimization (CMTOP) has garnered significant attention as a specialized branch aimed at solving multiple optimization tasks concurrently under competitive settings. However, existing CMTOP algorithms suffer from inherent shortcomings, such as inefficient resource allocation and undirected randomization in auxiliary task selection, which considerably limit their overall performance. To overcome these challenges, this article proposes a novel Competitive Multi-Task Historical Success Archive Synergy Algorithm with a Success-Memory Framework (CMHSA-SMF). The algorithm incorporates three key innovations: (i) an adaptive task affinity weighting mechanism with dynamic thresholds that guides auxiliary task selection by capitalizing on distinct phase-dependent demands during early and late stages of optimization; (ii) a historical parameter-aware adaptive resource allocation strategy that dynamically distributes computational resources based on iterative performance feedback and success memory; and (iii) a collaborative co-evolutionary dynamic coupling strategy, which integrates adaptive random mating probability control and a dynamic cooperative evolutionary operator to enhance elite diversity, strengthen inter-task communication, and promote cross-task knowledge transfer for accelerated convergence. Extensive evaluations on widely-used benchmark suites, including C2TOP, C4TOP, and CCPLX, along with real-world applications in sensor coverage optimization and photovoltaic model parameter extraction, demonstrate that CMHSA-SMF significantly outperforms seven state-of-the-art peers (DE, SHADE, MFEA, MFDE, DEORA, MTSRA, and MTEA-HKTS) in terms of solution quality, convergence speed, and robustness.

CCS Concepts: • **Theory of computation** → **Algorithmic mechanism design**;

Additional Key Words and Phrases: competitive multi-task optimization, historical archive, adaptive similarity weighting factor, resource allocation, evolutionary operator

Associate Editor: Yu Yang

Li Erchao (Writing review and editing, Conceptualization, Supervision); Li Hongxu (Methodology, Writing-original draft); Yu Peng (Amendments and guidance).

This work was supported by the National Natural Science Foundation of China (No. 62063019), Key Project of Natural Science Foundation of Gansu Province (No. 24JRRA173), and Excellent Doctoral Program in Gansu Province (No. 24JRRA205).

Authors' Contact Information: Erchao Li, College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; e-mail: lecstarr@163.com; Hongxu Li (corresponding author), College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; e-mail: 2432611241@qq.com; Yu Peng, College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; e-mail: plik\_y@outlook.com. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1556-472X/2025/12-ART17

<https://doi.org/10.1145/3775058>

**ACM Reference format:**

Erchao Li, Hongxu Li, and Yu Peng. 2025. Competitive-Cooperative Multi-Task Optimization Algorithm with Historical Success Archive. *ACM Trans. Knowl. Discov. Data.* 20, 1, Article 17 (December 2025), 45 pages. <https://doi.org/10.1145/3775058>

## 1 Introduction

In contemporary complex and dynamic scientific and engineering domains, optimization problems permeate multidisciplinary fields. Their applications span parametric design in engineering systems and resource allocation in economic modeling [1], extending to model training in machine learning and route planning in logistics management [6]. The fundamental objective consistently remains identifying optimal solutions under prescribed constraints to achieve targeted outcomes. To address these challenges, metaheuristic optimization algorithms have emerged, demonstrating significant efficacy in resolving such problems [11]. However, as problem complexity escalates, traditional single-task optimization methods increasingly reveal limitations when handling correlated tasks. **Multi-Task Optimization (MTO)**, an emerging paradigm, has thus been proposed. This framework significantly enhances collective optimization efficiency by exploiting shared information across tasks and has been rigorously validated in cutting-edge applications [14]. Yuan et al. [15] leveraged the advantages of the **Multifactorial Evolutionary Algorithm (MFEA)** for evolutionary multi-tasking optimization. By incorporating domain-specific sequence encoding and rank-based selection mechanisms, they enhanced MFEA to substantially improve computational efficiency for permutation-based combinatorial optimization problems. Feng et al. [16] proposed an Explicit Evolutionary Multi-Tasking Algorithm, utilizing sparse mapping and distance-based user representation to enable efficient knowledge transfer in **Vehicle Routing Problems (VRP)**. To overcome representation deficiencies in the Capacitated VRP, Zhou et al. [17] developed a Permutation-Based MFEA. This algorithm employs sequence-based unified encoding and constraint-aware split decoding mechanisms, achieving accelerated convergence rates and superior solution quality when handling VRP with time windows in multi-task environments. Yang et al. [18] formulated mineral processing as a Multi-Objective Multifactorial Operational Indices Optimization problem. Resolving its modeling challenges through evolutionary multi-tasking optimization with cross-task knowledge transfer, they significantly accelerated convergence of the highest-precision process models.

In recent research advances, **Competitive Multi-Tasking Optimization (CMTO)** has emerged as a paradigm-level breakthrough beyond traditional multi-tasking frameworks, with its core characteristic being objective value comparability, which intrinsically induces inter-task competition mechanisms [19]. Compared to conventional **Multi-Task Optimization Problems (MTOPs)**, **Competitive Multi-Task Optimization Problems (CMTOPs)** exhibit significant uniqueness. Most notably, under the CMTO architecture, the objective values of all tasks are directly comparable, and this comparability further catalyzes inherent competition among tasks. Specifically, CMTO demonstrates two distinctive features relative to standard MTO models: First, during the initial phase of task processing, no prior knowledge about target tasks is required—no task is preset as the primary target, and all tasks are treated equally with identical exploratory priority from the outset; Second, all tasks within the CMTO framework share identical performance evaluation metrics for their objective functions, ensuring consistent assessment across tasks under a unified measurement scale, thereby avoiding resource allocation imbalances and optimization path deviations caused by divergent evaluation criteria. The intrinsic competitive nature of CMTO creates an ideal scenario for real-world applications requiring adaptive solution spaces, particularly **Variable-Length**



**Optimization Problems (VLPs)** [20]. Building upon this framework, Li et al. [19] proposed **Differential Evolution with Online Resource Allocation (DEORA)**, an online resource arbitration system embedding **Multi-Tasking Differential Evolution (MTDE)**. Although DEORA outperforms traditional evolutionary algorithms and **Multi-Tasking Evolutionary Algorithms (MTEAs)** on CMTOPs, persistent inefficiencies in MTDE operators compromise solution quality and convergence. To address this, Li et al. [23] developed **Modified Adaptive Differential Evolution Algorithm (MTSRA)**—an evolutionary algorithm integrating adaptive resource allocation, enhanced **Differential Evolution (DE)** mechanisms, and probabilistic knowledge transfer. While MTSRA advances CMTO through adaptive operators, its heavy reliance on stochastic auxiliary task selection introduces uncontrolled randomness escalation in later stages.

Notably, these stochastic deficiencies fundamentally stem from inadequate exploitation of historical knowledge transfer efficacy, whereas effectively leveraging prior knowledge for resource allocation constitutes the core challenge in building robust CMTOP solvers. Although existing studies explore knowledge transfer through strategies such as explicit autoencoding [24], affine transformations [25], knowledge-guided sampling [26], and archive mechanisms [27], these methods rely on single-generation populations or randomly updated archives, failing to ensure historical information validity in CMTO's dynamic optimization. Stanovov et al. [29] further revealed that systematic knowledge reuse based on successful historical archives (e.g., eliminating inefficient solutions) significantly enhances algorithmic performance. Inspired by this, we propose a historical success archive-based competitive multi-tasking cooperative optimization algorithm (CMHSA-SMF). By designing a lightweight **Similarity Metric Factor (SMF)**, it dynamically assesses similarity between primary and auxiliary tasks before each iteration to inform archive construction and task selection. Addressing computational complexity and adaptability limitations of traditional similarity discrimination methods (e.g., KLD divergence [30], maximum likelihood estimation [31], fitness landscape analysis [32]) in dynamic populations, SMF innovatively employs lightweight **Maximum Mean Discrepancy (MMD)** to construct historical archives. Our contributions are as follows:

- (1) An adaptive task affinity weighting mechanism (SMF) is devised to optimize inter-task knowledge transfer. Utilizing lightweight MMD metrics and phase-dependent weights  $\xi_t^x$ , it dynamically adjusts auxiliary task selection criteria for different similarity combinations, promoting more effective knowledge exchange while mitigating negative transfer risks.
- (2) A historical parameter-aware adaptive resource allocation mechanism is proposed to allocate computational resources toward promising tasks. This mechanism leverages dual-memory collaboration (**Success Memory (SM)** and **Failure Memory (FM)**) with dynamic coefficients ( $\alpha, \beta$ ) and exponentially decaying penalties ( $\gamma$ ), enabling more accurate quantification of recent task performance through historical data analytics, thereby reducing the likelihood of resource misallocation.
- (3) A collaborative co-evolutionary dynamic coupling strategy for multi-task scenarios is designed to accelerate convergence by integrating dynamic cooperative evolutionary operators and adaptive stochastic mating probability (RMP) control strategies, thereby promoting diverse task combinations. The accelerated convergence ensures accurate validity assessment of each task by CMHSA-SMF, enhancing the effectiveness of resource allocation mechanisms.

The remainder of this article is organized as follows: First, in Section 2, we introduce related work and motivations. Next, Section 3 details the proposed CMHSA-SMF algorithm. Second, experimental studies and analyses are presented in Section 4. Then, Section 5 demonstrates the performance of the proposed algorithm in solving real-world problems. Finally, conclusions and future research directions are summarized in Section 6.

## 2 Related Works

### 2.1 MTO

Inspired by the parallel task-processing capabilities of biological organisms in nature, MTO enhances convergence across tasks by exploiting shared knowledge among multiple optimization tasks. Consider a scenario where  $K$  tasks coexist in a multi-task environment. Without loss of generality, assume all tasks are minimization problems with equal processing priority. The  $j$ th task is denoted as  $T_j$ , with an objective function  $f_j: X_j \rightarrow \square$  defined over its search space  $X_j$ . Each task may be constrained by several equality or inequality conditions, and solutions satisfying all constraints are termed feasible solutions. Under this framework, the MTO problem is formally defined as:

$$\begin{cases} \arg \min x & \{f_1(x), f_2(x), \dots, f_{K-1}(x), f_K(x)\} \\ s.t. & g_j^m(x) \leq 0, m = 1, 2, \dots, p \\ & h_j^n(x) = 0, n = 1, 2, \dots, q, \end{cases} \quad (1)$$

where:  $x$  is a  $K$ -dimensional decision vector;  $x_j$  denotes the feasible solution for task  $T_j$  within its search space  $X_j$ ;  $g_j^m(x) \leq 0 (m = 1, 2, \dots, p)$  defines the  $p$  inequality constraints for task  $T_j$ ;  $h_j^n(x) = 0 (n = 1, 2, \dots, q)$  specifies the  $q$  equality constraints for task  $T_j$ .

Gupta et al. [14] proposed **Multifactorial Optimization (MFO)** and further developed the MFEA to solve evolutionary MTOPs. Since the introduction of MFEA, numerous variants of MTO algorithms based on MFEA have emerged. Bali et al. [33] replaced scalar parameters with symmetric matrices to regulate inter-task interaction intensity and constructed a data-driven online learning mechanism based on probabilistic models, effectively addressing negative transfer caused by unknown task similarities in conventional methods. Jiang et al. [34] introduced a block-level knowledge transfer framework, partitioning multi-task individuals into consecutive dimensional blocks to build modularized populations, combined with  $k$ -means clustering to dynamically identify cross-task and intra-task similar dimensional blocks, enabling knowledge transfer across non-aligned similar dimensions. Liang et al. [35] proposed a generative multi-objective evolutionary multi-task framework (EMT-GS) that integrates generative adversarial networks with inertial DE strategies to resolve inefficient knowledge transfer and slow convergence in multi-objective optimization scenarios. Ma et al. [36] developed a gene-similarity-based MTO framework (MFEA-GSMT), quantifying cross-task gene-level similarity via **Kullback–Leibler (KL)** divergence and incorporating mirror transformation strategies to suppress premature convergence, thereby avoiding early fusion and exploring other promising search domains. Liang et al. [37] designed a two-stage adaptive knowledge transfer framework based on population distributions (EMT-PD), modeling population search trends through probabilistic models and dynamically regulating multi-task collaboration via dual-phase adaptation. Zhao et al. [27] proposed a hybrid knowledge transfer strategy (MTEA-HKTS) that balances convergence speed and diversity by controlling migration frequencies across optimization phases. Additionally, other strategies integrated into MTO algorithms—such as affine transformations of decision spaces [25] and domain adaptation [38] have demonstrated exceptional performance.

### 2.2 CMTO

Inspired by the parallel task-processing capabilities of biological organisms in nature, evolutionary MTO facilitates convergence across tasks by leveraging shared knowledge among multiple optimization tasks. With the continuous advancement of evolutionary MTO, numerous branches have emerged, among which CMTO has demonstrated significant application potential [19]. A typical CMTO application scenario is the VLP [39], where the optimal solution vector length remains unknown prior to optimization due to potential variations in the number of constituent elements,

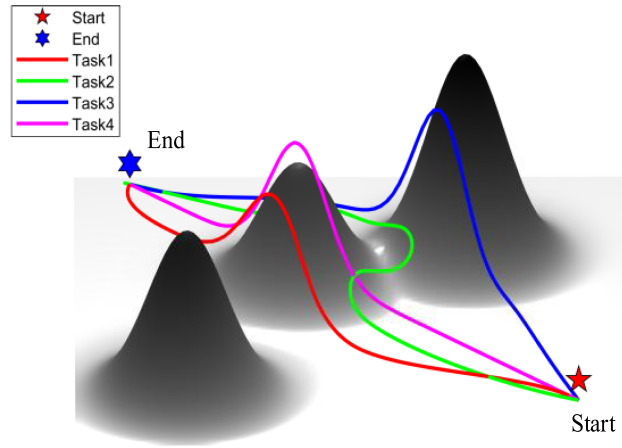


Fig. 1. Schematic illustration of the CMTOP problem.

rendering traditional evolutionary algorithms inefficient for such problems. By treating each variable length as an independent task component, VLP can be transformed into a MTO. However, the distinction between MTO and CMTOP lies in their objectives: MTO aims to identify an optimal solution set  $\{x_1^*, x_2^*, \dots, x_k^*\}$  for all tasks, while CMTOP seeks to determine the best solution  $x_b^*$  from this set  $\{x_1^*, x_2^*, \dots, x_k^*\}$ . To provide a more intuitive representation, we mathematically formulate this as:

$$x_i^* = \arg \min_{f_i(x_i)}, i = 1, 2, \dots, K, \quad (2)$$

$$f_b(x_b^*) \leq f_i(x_i^*), i = 1, 2, \dots, K \text{ and } i \neq b, \quad (3)$$

where: Equation (2) defines that for a CMTOP with  $K \geq 2$  tasks,  $X_i$  denotes the search space of the  $i$ th task,  $f_i: X_i \rightarrow \mathbb{R}$  represents the objective function, and the optimal solution for the  $i$ th task is  $x_i^*$ . Equation (3) specifies identifying the global optimal solution  $x_b^*$  from the  $K$  tasks as the objective value.

To intuitively illustrate the dynamic characteristics of CMTOP, Figure 1 demonstrates its core mechanisms through a path evolution example. In the initial phase (starting point), all tasks (Task 1–4) commence from the same initial solution set, sharing a global optimization objective (endpoint). Each task operates within an independent search space, yet their objective functions are comparable, enabling solutions from different tasks to compete for global optimality. The four distinctively colored paths in the figure represent candidate solution sequences generated by different tasks during evolution. During each iteration cycle, the algorithm dynamically allocates computational resources based on the current performance of task-specific solutions: if a task's candidate solution (e.g., Task 4 in Phase A, purple path) demonstrates closer proximity to the global optimum, resources are prioritized to accelerate its optimization; as iterations progress (Phase B), Task 1 (red path) may surpass others through information migration or local search; by the final phase (Phase C), Task 4 reestablishes dominance due to inherent search space properties or enhanced cross-task knowledge sharing. This process exemplifies the dynamic equilibrium between inter-task competition and collaboration in CMTOP, as well as the critical role of resource allocation strategies in achieving global convergence.

### 2.3 Motivation

To better address CMTOPs, Li et al. [19] proposed a unified framework for CMTOPs, as illustrated in Figure 2, and further developed the DEORA algorithm specifically for CMTOPs. DEORA leverages archive information generated in each iteration to manage competing tasks through online resource

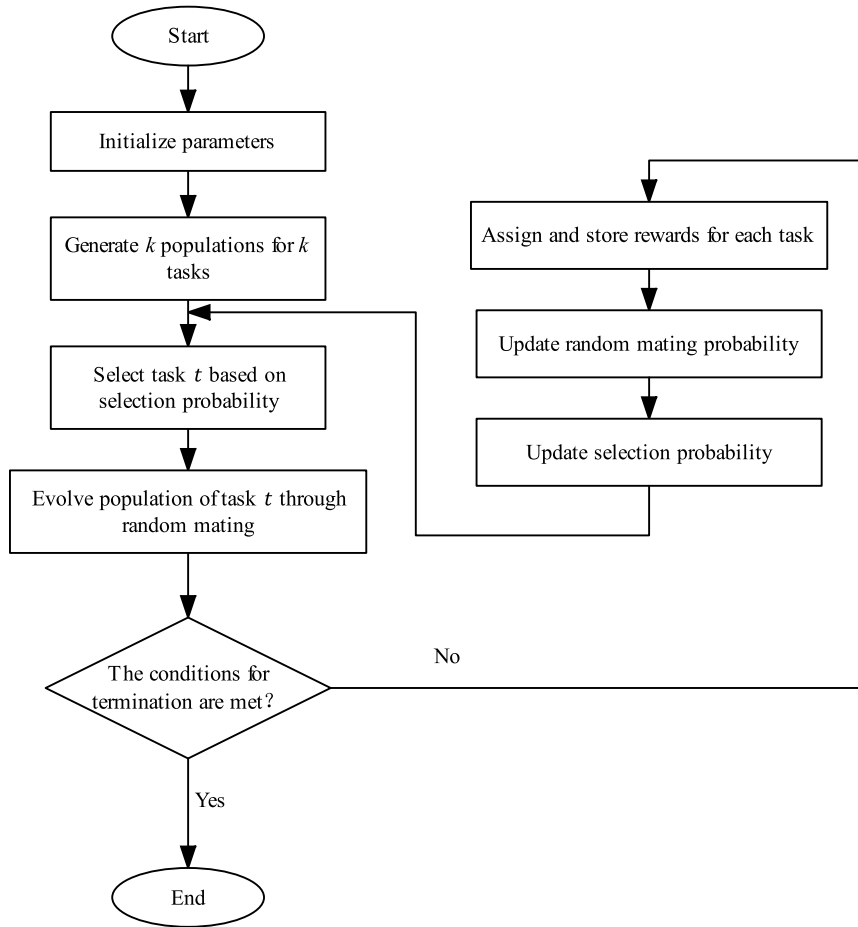


Fig. 2. Schematic diagram of the CMTOP framework.

allocation strategies, employing a progressive multi-tasking approach to accelerate problem-solving. Building upon the DEORA framework, Li et al. [23] introduced an MTSRA, which utilizes a success-history-based resource allocation strategy to manage and select competing tasks, coupled with an enhanced evolutionary operator MT-SHADE to update populations and expedite convergence. Compared to non-multitask EAs and MTEAs, DEORA, and MTSRA have achieved promising results in solving CMTOPs. However, DEORA's performance may deteriorate on certain CMTOPs due to suboptimal convergence properties of its operators. In MTSRA, excessive randomness in auxiliary task selection may impair convergence performance in later stages for some multi-task problems and compromise algorithmic stability. To overcome these limitations in DEORA and MTSRA, we propose CMHSA-SMF, an improved framework designed to handle CMTOPs more effectively.

### 3 Proposed Method

#### 3.1 Structure of CMHSA-SMF

CMHSA-SMF consists of three components: an adaptive task affinity weighting with dynamic thresholds, a historical parameter-aware adaptive resource allocation mechanism, and a collaborative co-evolutionary dynamic coupling strategy for multi-task scenarios. The historical parameter-aware adaptive resource allocation mechanism adaptively allocates limited computational resources to more promising tasks, thereby improving the resolution of CMTOPs within constrained time-frames. The similarity weighting factor (SMF) dynamically selects auxiliary tasks by calculating inter-task similarities under varying phase-specific requirements, accelerating convergence. The

**Algorithm 1:** Framework of the CMHSA-SMF

Input:

$N$  (the population size of all tasks),  $K$  (the number of tasks),  $AN$  (the archive size),  $H$  (Success memory size),  $RH$  (Historical memory repository size),  $F$  (Fail memory size),  $RH$  (Record fail memory size),.

Output:

$x_b^*$  (The best solution found for  $K$  tasks).

```

1: Initialize the populations  $\{P_1, P_2, \dots, P_k\}$ , each with size  $N/K$ ;
2: Initialize the reward memories  $\{R_1, R_2, \dots, R_k\}$  and selection probabilities  $\{p_1, p_2, \dots, p_k\}$ ;
3: Initialize the parameter memories  $\{R_1, R_2, \dots, R_k\}$  and archives  $\{A_1, A_2, \dots, A_k\}$ ;
4: Initialize the random mating probability matrix  $RMP$ ;
5: while Termination condition is not met do
6:   If  $FE \leq Bate * Prob.maxFE$  then
7:     Stage 1: Evolution  $\rightarrow$  Intra-individual crossover mutation.
8:   else
9:     Stage 2: Competition  $\rightarrow$  Intra-tasks crossover mutation.
10:  end If
11:   $k \leftarrow$  Select the main task use the roulette wheel method based on the selection probability  $\{p_1, p_2, \dots, p_k\}$ ;
12:   $\xi \leftarrow$  Calculate Adaptive Similarity Weight Factor  $\xi$  with task  $t$  as main task, Algorithm 2;
13:   $j \leftarrow$  History-Aware Parameter Self-Adaptive Mechanism for auxiliary task selection, Algorithm 3;
14:   $F$  and  $CR \leftarrow$  Allocate parameters  $F$  and  $CR$  for  $P_t$  based on  $M_t$ ;
15:   $P_t \leftarrow$  Generating populations  $P_t$  refer to Eq.(13), (14) and (15);
16:   $R_t$  and  $R_j \leftarrow$  Calculate the success rewards  $r_t^{total}, r_j^{total}$  and update  $R_t$  and  $R_j$  refer to Eq.(8), (9) and (10);
17:   $M_t \leftarrow$  Calculate the success and fail parameters and update  $M_t$ ;
18:   $A_t \leftarrow$  Update the historical archives  $A_t$ , Algorithm 4;
19:   $RMP(t, j) \leftarrow$  Update  $RMP$  based on  $r_t^{total}, r_j^{total}$  refer to Eq.(16) and (17);
20:   $\{p_1, p_2, \dots, p_k\} \leftarrow$  Update  $\{p_1, p_2, \dots, p_k\}$  based on  $\{R_1, R_2, \dots, R_k\}$  refer to Eq.(11) and (12);
21: end while
22: return  $x_b^*$ 

```

collaborative co-evolutionary dynamic coupling strategy for multi-task scenarios enhances efficiency when processing different task groups.

The pseudo-code of the proposed CMHSA-SMF is outlined in Algorithm 1. During initialization, the algorithm generates  $K$  populations  $P_1, P_2, \dots, P_k$ , each corresponding to one task, with a population size of  $N/K$ . It initializes historical rewards  $R_1, R_2, \dots, R_k$ , selection probabilities  $p_1, p_2, \dots, p_k$ , parameter memory  $M_1, M_2, \dots, M_k$ , historical archives  $A_1, A_2, \dots, A_k$ , and the **Random Mating Probability (RMP)** matrix. After initialization, the main loop begins. First, a main task  $t$  is selected using roulette wheel selection based on the task selection probabilities  $P_1, P_2, \dots, P_k$ . The MMD calculates the similarity between each task and the main task  $t$ . The historical parameter-aware adaptive resource allocation mechanism then selects an auxiliary task  $j$  via roulette wheel selection to assist the evolution of the main task. Once the main task  $t$  and auxiliary task  $j$  are determined, the algorithm assigns parameters  $F$  and  $CR$  to population  $P_t$  using the parameter allocation strategy from the SHADE algorithm. The evolutionary operator **Dynamic Multi-Task SHADE (DMT-SHADE)** generates offspring for  $P_t$ , followed by selection to update the population. This operator employs information from the auxiliary task population  $P_j$ , guided by the  $RMP(t, j)$ , to enhance the evolution of the main task population  $P_t$ . After evolving the main task population  $P_t$ , the total rewards  $r_t^{total}$  and  $r_j^{total}$  for  $P_t$  and  $P_j$  are computed. These rewards update the reward memory matrices  $R_i, R_j$  and the  $RMP$  matrix. Additionally, successful individuals are identified and



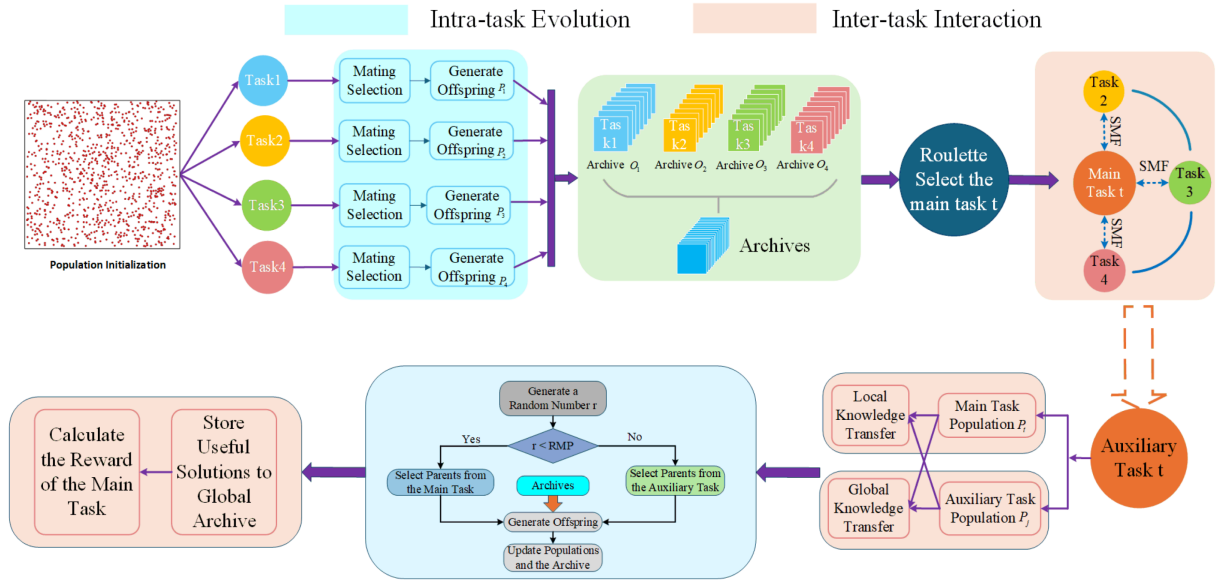


Fig. 3. Overall structure of CMHSA-SMF.

analyzed to update the parameter memory  $M_k$  and historical archives  $A_k$  using their parameters. Finally, CMHSA-SMF updates the selection probabilities  $p_1, p_2, \dots, p_k$  based on the reward memory of all tasks. Upon meeting termination criteria, the algorithm outputs the optimal solution  $X_b^*$  for all tasks. The algorithm workflow is illustrated in Figure 3. The following sections detail each component of CMHSA-SMF.

### 3.2 Adaptive Task Affinity Weighting with Dynamic Thresholds

From the logical architecture of the proposed CMHSA-SMF algorithm, the evolution of the main task population  $P_t$  stems from two components: intra-population iterations of task  $t$  and knowledge-assisted evolution from the auxiliary task population  $P_j$ . Therefore, the selection of auxiliary tasks critically impacts the evolutionary trajectory of  $P_t$ . In MTSRA, auxiliary tasks are selected stochastically, introducing significant uncertainty regarding the utility of transferred solutions from auxiliary tasks. The absence of explicit criteria for auxiliary task selection amplifies instability in solution transfers, thereby significantly increasing the risk of negative transfer and computational resource wastage. To resolve this limitation, we design an adaptive task affinity weighting with dynamic thresholds (SMF), which dynamically quantifies inter-task similarities through a phase-dependent adaptive weighting mechanism, thereby establishing explicit criteria for auxiliary task selection.

**3.2.1 Similarity Metric.** To begin with, after selecting the primary task, we proceed to assess the similarity between this task and other related tasks by examining their population relationships. Typically, two prominent methods are utilized for quantifying these relationships: KL divergence and MMD. However, due to the computational complexity associated with KL divergence, it is deemed unsuitable for our algorithm. Consequently, we employ the simplified version of MMD as proposed in EMaTO-MKT [40] to evaluate the similarity between two tasks, which we approximate as multivariate Gaussian distributions. The MMD is computed in each iteration as follows:

$$MMD^2[\Gamma, X, Y] = \frac{1}{u(u-1)} \sum_{i \neq j}^u k(x_i, x_j) + \frac{1}{v(v-1)} \sum_{i \neq j}^v k(y_i, y_j) - \frac{2}{uv} k(x_i, y_j). \quad (4)$$

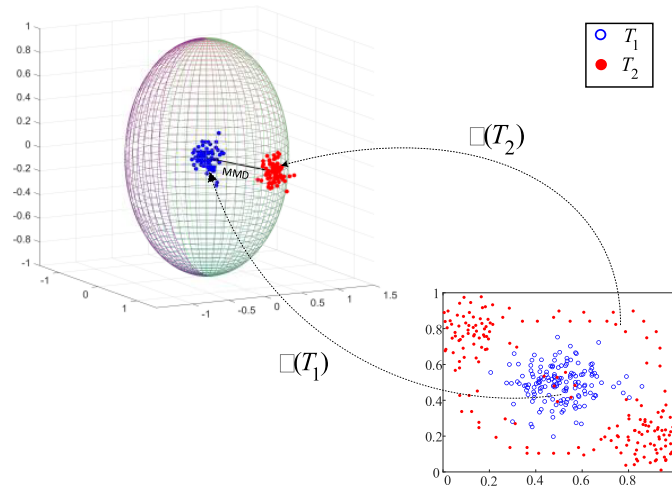


Fig. 4. Schematic of MMD in CMHSA-SMF.

In this context, the kernel function  $k(x, x')$  is utilized to compute the MMD between the two distributions. The Gaussian kernel function transforms data from the original space into an infinite-dimensional space, thereby effectively capturing the underlying information regardless of the dimensionality of the decision space, as illustrated in Figure 4. We apply the proposed Gaussian kernel function as follows:

$$k(x, x') = \exp(-\|x - x'\| / (2\sigma^2)), \quad (5)$$

where  $\sigma$  is the width parameter of the Gaussian kernel function, used to compute the MMD value.

**3.2.2 Adaptive Similarity Weighting Factor.** In the early evolutionary stages, our objective is to enhance inter-population differences to boost diversity and improve the algorithm's exploration capacity, thereby avoiding premature convergence to local optima. In later stages, reduced inter-population differences are necessary to facilitate the transfer of similar solutions and accelerate convergence. To address these phase-specific requirements, we propose adaptive weighting factors that allow the algorithm to maintain sufficient diversity for exploration in the initial phases while achieving rapid convergence in the later phases. First, the similarity degree  $D_u^v$  between the main task and other tasks is computed using Equations (4) and (5), where  $u$  denotes the selected main task,  $v$  represents candidate auxiliary tasks, and  $D_u^v$  quantifies the similarity between tasks  $u$  and  $v$ . The constructed similarity factor is formulated as:

$$\Psi_t^i = \frac{1}{(|D_t^1 - D_t^\tau|) + 1}, \quad (6)$$

where:  $D_t^i$  denotes the similarity value between the  $i$ th task and task  $t$ . During the initial algorithm phase,  $D_t^\tau$  is calculated as the similarity value corresponding to the minimum and average values between the randomly selected  $i$ th task and task  $t$ ; in the final phase,  $D_t^\tau$  is randomly selected from the interval between the average and maximum similarity values, as detailed in Algorithm 2.

To facilitate the application of the similarity factor in roulette wheel selection of auxiliary tasks while mitigating the risk of probability model collapse, we introduce a similarity weighting factor. This factor is designed to alleviate potential issues, resulting in the following formulation of the adaptive similarity weighting factor:

$$\xi_t^x = \Psi_t^x / \sum_{i=1}^k \Psi_t^i \quad (7)$$

---

**Algorithm 2:** Adaptive Task Affinity Weighting with Dynamic Thresholds
 

---

Input:

 $P$ : the populations  $\{P_1, P_2, \dots, P_k\}$  $\sigma$ : Gaussian kernel bandwidth

Output:

 $\xi$ : Similarity weighting factor (SMF).1: Initialize MMD measurement vector  $D \leftarrow 0_{1 \times K}$ ;2: **For** each task populations  $\{P_1, P_2, \dots, P_k\}$  **do**3:  $D_t^i \leftarrow$  Calculate the similarity values between the main task  $t$  and other tasks refer to Eq. (4) and (5).4: **end For**5: Sort the similarity values  $D$  in ascending order;6:  $D_t^r \leftarrow \text{Select } rand(min)D_t^i, midD_t^i$ ;7:  $\Psi \leftarrow$  Calculate the similarity factor  $\Psi$  refer to Eq.(6);8:  $\xi \leftarrow$  Calculate the adaptive similarity weight factor  $\xi$  refer to Eq.(7);9: return  $\xi$ 

### 3.3 Historical Parameter-Aware Adaptive Resource Allocation Mechanism

The historical parameter-aware adaptive resource allocation mechanism achieves precise quantification and historical backtracking of performance for both primary and auxiliary tasks through a dynamic coupling evaluation framework between the SM and FM. It is executed as a subroutine within the main evolutionary loop (Algorithm 1). The inputs to this subroutine are the current population, the success and failure memories, and the reward histories from the previous generation. Its outputs are the updated task selection probabilities  $\{P_1, P_2, \dots, P_k\}$  for the next generation. The SM enhances exploration capabilities by recording reward allocations from positive improvements and balancing weights between global and task-specific local optima. Conversely, the FM suppresses invalid search paths through penalty terms for degeneracy behaviors and an exponentially decaying mechanism for historical failure patterns, thereby reducing the likelihood of selecting ineffective tasks. Simultaneously, the task selection mechanism incorporates adaptive similarity weight factors and dynamic balance coefficients to self-adjust auxiliary task selection criteria during different algorithm phases (exploration and convergence), effectively avoiding excessive randomness while mitigating risks of local optima entrapment. This mechanism is invoked at every generation after the offspring population is generated and evaluated, but before the next generation's task selection probabilities are updated. Full implementation details are provided in Algorithm 3.

**3.3.1 Dual-Memory Collaborative Reward Allocation.** The proposed algorithm introduces a dual-memory collaborative composite reward allocation strategy. By establishing a coupled evaluation framework between the SM and FM, it achieves dynamic quantitative evaluation of task performance.

#### (1) Construction of SM

The constructed SM records positive improvements generated during the evolutionary process. When task  $t$  at generation  $g$  satisfies  $f_{t,g}^* < f_{t,g-1}^*$ , the generation-specific reward  $r_t(g)$  is computed and stored in SM. The reward  $r_t(g)$  calculation for the main task  $t$  at generation  $g$  references the DEORA algorithm as follows:

$$r_t(g) = \eta r_t^b(g) + (1 - \eta) r_t^p(g), \quad (8)$$

**Algorithm 3:** History-Aware Parameter Self-Adaptive Mechanism

Input:

 $P$ : the populations  $\{P_1, P_2, \dots, P_k\}$  $RH$ : Historical memory repository size $\eta$ : Reward control parameters $\alpha, \beta$ : Dynamic balance coefficient $\gamma$ : Failure penalty decay factor

Output:

 $j$ : auxiliary task $R$ : Task-integrated reward vector1: **while** each task populations  $\{P_1, P_2, \dots, P_k\}$  do2:   **For** each task populations  $\{P_1, P_2, \dots, P_k\}$  do2:      $j \leftarrow$  Select auxiliary task  $j$  via roulette wheel selection based on selection probability  $p$  refer to Eq.(20).3:   **end For**4:    $r_t(g) \leftarrow$  Calculate the rewards of the main task in generation  $g$  refer to Eq.(8),(9)and(10);5:    $r_j(g) \leftarrow$  Calculate the rewards of the auxiliary task in generation  $g$  refer to Eq.(11), (12) and (13);6:    $c_t(g) \leftarrow$  Calculate the failure penalty value of the main task  $t$  refer to Eq.(14);7:    $c_j(g) \leftarrow$  Calculate the failure penalty value of the auxiliary task  $j$  refer to Eq.(15);8:    $r_t^{total}(g) \leftarrow$  Calculate the composite reward of the main task  $t$  refer to Eq.(16);9:    $r_j^{total}(g) \leftarrow$  Calculate the composite reward of the auxiliary task  $j$  refer to Eq.(17);10:   Calculate the probability  $P_t$  of selecting  $t$  for the main task refer  $t$  to Eq.(18);11:   Calculate the probability  $P_j$  of selecting  $t$  for the auxiliary task  $j$  refer to Eq.(19);12: **end while**13: **return**  $r_t^{total}(g), r_j^{total}(g)$  and the auxiliary task  $j$ 

where:

$$r_t^b(g) = \max \left\{ \frac{f_{g-1}^{**} - f_{t,g}^*}{f_{g-1}^{**}}, 0 \right\}, \quad (9)$$

$$r_t^p(g) = \frac{1}{N} \sum_{i=1}^N \max \left\{ \frac{f_{t,g-1,i} - f_{t,g,i}}{f_{t,g-1,i}}, 0 \right\}, \quad (10)$$

where:  $t$  represents the main task, and  $g$  indicates the current algorithm iteration generation. In Equation (9),  $\eta$  controls the ratio between  $r_t^b(g)$  and  $r_t^p(g)$ ,  $r_t^b(g)$  represents global optimal improvement in the current generation,  $r_t^p(g)$  represents optimal improvement of task  $t$  in the current generation,  $r_t(g)$  represents reward of the main task  $t$  in the current generation;  $f_{g-1}^{**}$  is optimal solution of all tasks in the previous generation,  $f_{t,g}^*$  is optimal solution of task  $t$  in the current generation.

The reward mechanism for auxiliary tasks is equally critical to algorithmic performance since its selection strategy directly governs the evolutionary trajectory of the main task. In CMHSA-SMF, the auxiliary task reward calculation incorporates adaptive weighting factors, which dynamically measure similarity between main and auxiliary tasks while integrating phase-specific demands (exploration/convergence phases) for auxiliary tasks, thereby achieving precisely calibrated adaptation of reward computation. The reward  $r_j(g)$  for auxiliary tasks in CMHSA-SMF is calculated as:

$$r_j(g) = \eta r_j^b(g) + (1 - \eta) r_j^p(g), \quad (11)$$

where:

$$r_j^b(g) = I(x_t^* \in P_t^j) \max \left\{ \frac{f_{g-1}^{**} - f_{t,g}^*}{f_{g-1}^{**}}, 0 \right\}, \quad (12)$$

$$r_j^p(g) = \begin{cases} \frac{\sum_{i=1}^N I_i^{t,j} \max \left\{ \frac{f_{t,g-1,i} - f_{t,g,i}}{f_{t,g-1,i}} (1 + \xi_t^j), 0 \right\}}{\sum_{i=1}^N I_i^{t,j}} & \text{if } \sum_{i=1}^N I_i^{t,j} > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

where:  $t$  denotes the main task and  $j$  represents the auxiliary task;  $r_j^b(g)$  denotes the global optimal improvement in the current generation;  $r_j^p(g)$  indicates the optimal improvement of task  $j$  in the current generation;  $r_j(g)$  represents the reward when task  $j$  is selected as the auxiliary task in the current generation; in Equation (9),  $I(x_t^* \in P_t^j)$  denotes the indicator function that outputs 1 when condition  $x_t^* \in P_t^j$  is satisfied and 0 otherwise;  $P_t^j$  signifies that the parent individual in the current offspring is from the main task population  $x_t^* \in P_t^j$  and the maternal individual originates from the auxiliary task population  $P_j$ ;  $f_{g-1}^{**}$  represents the optimal solution across all tasks in the previous generation;  $f_{t,g}^*$  denotes the optimal solution of task  $t$  in the current generation; in Equation (10), the solution information from the previous and current generations is leveraged, with the reward memory incorporating the similarity weighting factor  $1 + \xi_t^j$  to enhance the utility evaluation of solutions while strengthening inter-task correlations, thereby maximizing algorithm performance across different optimization phases with dynamically shifting objectives.

## (2) Construction of FM

The constructed FM archives historical records of task performance degradation. By documenting unsuccessful evolutionary attempts across generations, the algorithm precisely selects auxiliary tasks required by the main task, thereby achieving accelerated convergence. When the main task  $t$  satisfies  $f_{t,g}^* \geq f_{t,g-1}^*$ , the failure penalty value is computed as:

$$c_t(g) = \frac{|f_{t,g}^* - f_{t,g-1}^*|}{f_{t,g-1}^*}. \quad (14)$$

When auxiliary task  $j$  causes fitness degradation in over 30% of individuals within the main task population, compute the failure penalty:

$$c_j(g) = \sum_{i=1}^N I(f_t(u_i) > f_t(x_i)) \cdot \frac{f_t(u_i) - f_t(x_i)}{f_t(x_i)}, \quad (15)$$

where:  $c_t(g)$  and  $c_j(g)$  represent the failure penalties for tasks  $t$  and  $j$ , respectively;  $u_i$  denotes the newly generated solution,  $x_i$  is the original solution, and  $f_t(u_i) > f_t(x_i)$  indicates that the fitness of the new solution  $u_i$  is inferior to that of the original solution  $x_i$ .

From the aforementioned two components, the composite reward for the main task  $t$  can be expressed as:

$$r_t^{total}(g) = \alpha \cdot r_t^{SM}(g) - \beta \cdot c_t^{FM}(g), \quad (16)$$

where  $r_t^{SM}(g)$  is the success reward defined in Equation (11),  $c_t^{FM}(g)$  represents the mean failure penalty extracted from the most recent  $RH/2$  generations in FM,  $\alpha$  and  $\beta$  are dynamic balancing coefficients.



Table 1. Structure of the Parameterized Success–Failure Memory in CMHSA-SMF

| Generation | Reward for Task 1 | Reward for Task 2 | ... | Reward for Task $k$ |
|------------|-------------------|-------------------|-----|---------------------|
| 1          | $r_1^{total}(1)$  | $r_2^{total}(1)$  | ... | $r_j^{total}(g)$    |
| 2          | $r_1^{total}(1)$  | $r_2^{total}(2)$  | ... | $r_k^{total}(2)$    |
| ...        | ...               | ...               | ... | ...                 |
| RH         | $r_1^{total}(RH)$ | $r_2^{total}(RH)$ | ... | $r_k^{total}(RH)$   |

The composite reward for auxiliary task  $j$  is expressed as:

$$r_j^{total}(g) = r_j^b(g) + r_j^p(g) - \gamma \cdot \frac{\sum_{i=1}^{RH} c_j^{FM}(i)}{RH}, \quad (17)$$

where  $\gamma = e^{-(g-g_i)/\tau}$  is the penalty decay factor, which mitigates the long-term impact of historical failure records through exponential decay.  $\tau = RH/2$  controls the decay rate of memory influence.

During the evolutionary process, the rewards of the main task and auxiliary tasks in each generation are recorded in the reward memory, which reflect the performance of both successful and unsuccessful individuals and populations. The structure of the reward memory is illustrated in Table 1, with a fixed length of  $RH$ . In each generation, only the task-specific rewards  $r_t^{total}(g)$  and  $r_j^{total}(g)$  are computed, and the reward memories  $R_t$  and  $R_j$  are updated accordingly. When the reward memory reaches its maximum capacity  $RH$ , a FIFO strategy is applied: the oldest historical records are removed to free space for new reward data, thereby ensuring the latest reward information is consistently preserved while retaining a controlled amount of historical data to support long-term trend analysis and optimization. Through this mechanism, the reward memory not only archives task-specific rewards across generations but also maintains recency and validity via continuous updates, facilitating persistent refinement and enhancement of the evolutionary process.

**3.3.2 Task Selection Mechanism.** The differential resource allocation across tasks manifests in their distinct task selection probabilities. In DEORA, the selection probability leverages all reward records, utilizing parameter  $\gamma$  to control algorithmic preference for recent records. When  $\gamma$  is large, DEORA prioritizes newer records, leading to a stronger bias toward recently updated main tasks. Conversely, when  $\gamma$  is small, DEORA becomes more susceptible to early-stage record influences, which may induce misjudgment of task rewards. In contrast, MTSRA employs success history memory for task selection. While this approach avoids bias toward recent improvements and insulates against early historical impacts, its auxiliary task selection proves excessively stochastic for evolutionary purposes. This underutilizing of historical memory fails to fully exploit the algorithm's inherent characteristics during initial and later phases, thereby constraining convergence stability and limiting overall algorithmic robustness.

In the selection of the main task  $t$ , the method proposed in MTSRA is adopted to calculate its selection probability. The selection probability  $p_t$  for the main task  $t$  is expressed as:

$$p_t = \begin{cases} \frac{1}{K} & \text{if } g \leq \lambda G_{max} \\ \frac{p_{min}}{K} + \frac{(1-p_{min}) \sum_{i=1}^{RH} r_t^{total}(i)}{\sum_{j=1}^K r_j^{total}(i)} & \text{otherwise,} \end{cases} \quad (18)$$

where  $\frac{p_{min}}{K}$  is the minimum selection probability for tasks,  $g$  denotes the current generation number,  $G_{max}$  represents the maximum generation threshold  $G_{max} = \frac{E}{A}$ . When  $g \leq \lambda G_{max}$  (initial phase threshold), all tasks are selected with uniform probability to ensure uniform exploration in early

generations. If this condition is not met, tasks transition to the competitive phase, where selection probabilities are recalculated based on reward memory data.

To ensure the evolutionary advantage of task  $t$ , the selection of auxiliary tasks is critically pivotal. During early algorithmic phases, the significant disparity between main and auxiliary tasks is leveraged to enhance search capability, whereas in later stages, inter-task similarity is exploited to accelerate convergence. This dual-phase strategy prevents entrapment in local optima while amplifying convergence efficiency. Thus, based on the reward memory with adaptive similarity weighting factors, the selection probability  $p_j$  for auxiliary task  $j$  is formulated as:

where:  $\omega_j = \tanh\left(\sum_{i=1}^{RH} c_j^{RH}(i)\right)$  denotes the FM-based penalty term, a design that ensures exponential decay in the selection probability of auxiliary tasks that frequently induce task degradation; the adaptive similarity weighting factor  $\xi_t^j$  is introduced to address phase-specific requirements for auxiliary tasks, where selection criteria dynamically adapt across different optimization phases. This mechanism enhances algorithmic robustness and ensures convergence superiority.

### 3.4 Co-Evolutionary Dynamic Coupling Strategy for Multi-Task Scenarios

**3.4.1 Adaptive RMP Control.** To achieve finer-grained regulation of RMPs across tasks, we adopt an RMP matrix structure analogous to that used in MFEA-II [33]. Unlike MFEA-II's similarity-driven dynamic RMP adjustment, our method implements an iterative progressive update strategy. The Adaptive RMP Control mechanism is invoked at the end of each generation  $g$ , immediately after the historical resource allocation mechanism has calculated the composite rewards  $r_t^{total}(g)$  and  $r_j^{total}(g)$  for all task pairs. At generation  $g$ , let  $t$  denote the main task and  $j$  represent the auxiliary task. Upon completion of generation  $g$ , the RMP matrix is updated according to the following rules:

$$\begin{aligned} & \text{if } R_j(g) \leq R_i(g), (j = 1, \dots, K, j \neq k) \\ & \quad \begin{cases} RMP(t, j) = \min\{RMP(t, j) + \delta, 1\} \\ RMP(t, t) = \max\{RMP(t, t) - \delta, 0\} \end{cases} \\ & \text{otherwise} \\ & \quad \begin{cases} RMP(t, j) = \min\{RMP(t, j) - \delta, 0\} \\ RMP(t, t) = \max\{RMP(t, t) + \delta, 1\}, \end{cases} \end{aligned} \quad (19)$$

where  $g$  is a small increment, defined as:

$$\delta = \frac{N \cdot K \cdot (K - 1)}{E} = \frac{K \cdot (K - 1)}{G_{max}}, \quad (20)$$

where  $K$  denotes the number of tasks, and  $r_t^{total}(g)$  and  $r_j^{total}(g)$  are computed via Equation (8). It should be emphasized that the incremental RMP update method employed here bears partial resemblance to those in MFMP and MTSRA, yet with distinct purposes: In MFMP, RMP is a scalar designed to regulate global mating probability rather than per-task-pair mating probability; In MTSRA, auxiliary tasks are randomly selected, emphasizing uniform mating probability across all task pairs. Inspired by these studies, CMHSA-SMF adopts a roulette wheel selection mechanism based on task-specific probabilities, where the sum of RMP values for all tasks is constrained to 1. This design dynamically regulates the generation of transferred solutions to produce higher-quality solutions. Crucially, if an auxiliary task provides no beneficial contributions to the main task, their inter-task RMP is reduced; conversely, it is amplified. It is invoked as an integral part of the main algorithm loop (Algorithm 1, Line 19). The mechanism takes the current RMP matrix and the newly computed composite rewards for the main and auxiliary tasks from the current generation as its input. It then applies the update rules outlined in Equation (19) to each element  $RMP(t, j)$  in the

matrix, producing the updated *RMP* matrix that will be used in the next generation  $g+1$ . This dual mechanism ensures dynamic self-adaptation to auxiliary task utility while enhancing algorithmic robustness and enabling accelerated convergence.

### 3.4.2 Dynamic Co-Evolutionary Operator: DMT-SHADE.

#### (1) Mutual Pheromone Mutation Strategy

Leveraging knowledge transfer is pivotal for generating high-quality solutions. The DE algorithm has found extensive applications in multi-objective optimization and constrained optimization due to its robust optimization capabilities [41]. The SHADE algorithm, employing the current-to-pbest/1 mutation strategy, significantly enhances convergence performance in single-objective optimization [42]. To further improve DE's efficacy in MTO, MTSRA incorporates the dual-population mutation strategy proposed by Li et al. [43], which was further innovated to develop the more powerful MT-SHADE strategy. However, MT-SHADE's update mechanism over-relies on *RMP*, being entirely constrained by historical population data of the main task  $t$  and auxiliary task  $j$ . To address this limitation, we propose the advanced DMT-SHADE strategy to balance inter-task disparities. The refined mutation strategy is detailed as follows:

$$v_i = \begin{cases} x_i + F_i \cdot (x_p^t - x_i) + F_i \cdot (x_b^t - x_{r1}^t) & \text{if } rand > rmp \\ x_i + F_i \cdot (x_p^j - x_i) + F_i \cdot (x_b^j - x_{r1}^j) & \text{otherwise,} \end{cases} \quad (21)$$

where:  $rmp = RMP(t, j)$ ,  $x_i$  is the  $i$ th individual in the population  $P_t$  of task  $t$ ,  $F_i$  denotes the mutation parameters of  $x_i$ ,  $x_p^t$  is an individual randomly selected from the top  $(100 \cdot p)\%$  in population  $P_t$  of task  $t$ ,  $x_b^t$  and  $x_b^j$  represent the historical optimal solutions of tasks  $t$  and  $j$ , respectively,  $x_{r1}^t$  is an individual randomly selected from the union of population  $P_t$  and archive  $A_t$  (i.e., chosen uniformly from  $P_t \cup A_t$ ),  $x_{r1}^j$  is an individual randomly selected from  $P_j \cup A_j$ ,  $rand$  denotes a uniformly distributed random number in  $[0,1]$ . When the *RMP* is not satisfied, MT-SHADE performs mutation operations solely using individuals from the main task population  $P_t$  and archive  $A_t$ . This approach risks losing high-quality genetic information. To mitigate this, DMT-SHADE incorporates a global optimality factor to prevent useful information loss and guide individuals toward superior solutions. Conversely, when *RMP* is satisfied, the optimality factor similarly directs individual mutations to enhance solution quality.

After completing the mutation operation, a binary crossover strategy is adopted to generate offspring  $A_t$ . The binary crossover strategy is described as follows:

$$u_{i,d} = \begin{cases} v_{i,d} & \text{if } rand \leq CR_i \text{ or } d = d_{rand} \\ x_{i,d} & \text{otherwise,} \end{cases} \quad (22)$$

where  $v_{i,d}$  and  $x_{i,d}$  denote the values of  $v_i$  and  $x_i$  in the  $d$ th dimension,  $d_{rand}$  is an integer randomly selected from the index set  $[1, D^i]$ ,  $D^i$  represents the dimensionality of  $x_i$ .

At this stage, the offspring individuals  $u_1, u_2, \dots, u_N$  of population  $P_t$  have been generated. Subsequently, a 1v1 tournament selection strategy is employed to select the next-generation population  $P_t^{new}$  from both the original population  $P_t$  and its offspring  $u_1, u_2, \dots, u_N$ , as follows:

$$\mathbf{x}_i^{new} = \begin{cases} u_i & \text{if } f_t(u_i) \leq f_t(x_i) \\ x_i & \text{otherwise.} \end{cases} \quad (23)$$

#### (2) Archive Update strategy

The effectiveness of archive-based knowledge reuse relies on strategic archive updates, which are crucial for enhancing algorithm performance. We implement a tournament selection-based

**Algorithm 4:** Archive Update Strategy

Input:

 $A_t$ : Archive of the current main task  $t$  $P$ : the current population $N$ : population size $G$ : generation number $AN$ : archive size

Output:

 $A_t$ : The updated archive.1:  $tempP \leftarrow A_t \left( \text{mod} \left( G, \frac{AN}{N} \right) * N + 1 : \left( \text{mod} \left( G, \frac{AN}{N} \right) + 1 \right) * N \right)$ ;2:  $tempP \leftarrow$  Applying a tournament selection strategy to  $tempP$  and  $P$ ;3:  $A_t \left( \text{mod} \left( G, \frac{AN}{N} \right) * N + 1 : \left( \text{mod} \left( G, \frac{AN}{N} \right) + 1 \right) * N \right) \leftarrow tempP$ 4: return the updated archive  $A_t$ .

update strategy from MTEA-HKTS [27], as outlined in Algorithm 4. The archive size is set as an integer multiple of the population size  $N$  to store multi-generational data.

During each archive update strategy, first, based on the current iteration count  $G$ ,  $N$  individuals are extracted from the main task  $t$ 's archive to form a temporary population  $tempPOP$ . Subsequently, the tournament selection mechanism is applied between  $tempPOP$  and the current population  $Pop$ , with the winners are retained in  $tempPOP$ . Finally, the individuals in  $tempPOP$  replace those at corresponding positions in the archive.

Assuming the total archive size is  $3N$ , which means the archive can accommodate three complete populations. During the  $G$ th iteration, tournament selection is utilized to update  $\text{mod}(G,3)$  populations within the archive, where "mod" represents the modulo operation. Compared to traditional random update schemes, this strategy not only strengthens the retention of cross-generational population information and reduces uncertainty caused by stochastic factors, but also provides a rich high-quality dataset beneficial for subsequent model training processes. Through this mechanism, the archive effectively accumulates and leverages historical population data, thereby strengthening the algorithm's adaptability and stability in complex optimization problems.

### 3.5 Computational Complexity Analysis

CMHSA-SMF consists of three core components: an adaptive task affinity weighting mechanism with dynamic thresholds, a parameter-aware adaptive resource allocation mechanism, and a collaborative co-evolutionary dynamic coupling strategy for multi-task scenarios. Under the assumption of  $k$  tasks, a total population size of  $N$ , and a decision space dimensionality of  $D$ , the per-iteration computational complexity of CMHSA-SMF is analyzed as follows.

The evolutionary operator does not introduce additional complexity compared to MTSRA, and its time complexity can be expressed as  $N * \log(N)$ . The majority of computational resources are consumed by the adaptive task affinity weighting component. As detailed in Section 3.3, the most computationally expensive operation within this component is the MMD calculation, which has a time complexity of  $O(N^2D)$ . Task selection and reward allocation contribute complexities of  $O(K)$  and  $O(N)$ , respectively.

The Adaptive RMP Control mechanism within the co-evolutionary strategy incurs negligible overhead. Each  $RMP(t, j)$  element update requires only a simple comparison and a scalar addition operation, as specified in Equation (19). Given that the RMP matrix size is  $K^2$ , the total cost per generation is  $O(K^2)$ . Since  $K$  is typically small (ranging from 2 or 4 in practice), this cost becomes

Table 2. Algorithmic Complexity and Empirical Runtime Analysis

| Algorithm | Complexity                              | Time (s) |
|-----------|-----------------------------------------|----------|
| DEORA     | $O((KN^* \max(D, 1)))$                  | 9.83     |
| MTSRA     | $O(N^* \log(N))$                        | 18.67    |
| MTEA-HKTS | $O(N^* \log(N) + TN^2D + TN^* \log(N))$ | 23.17    |
| CMHSA-SMF | $O(N^* \log(N) + N^2D + N)$             | 26.83    |

insignificant compared to the task affinity computation cost  $O(N^2D)$ . Thus, the RMP control does not materially contribute to the overall computational complexity.

The overall complexity of the historical parameter-aware adaptive resource allocation mechanism is therefore  $O(N^2D) + O(K) + O(N)$ . Combining all components, the total time complexity of CMHSA-SMF per iteration is  $O(N^* \log(N)) + O(N^2D) + O(K) + O(N)$ . Given that  $K$  is generally small, the  $O(K)$  term is often negligible, simplifying the overall complexity to  $O(N^* \log(N) + N^2D + N)$ . This indicates that the complexity is primarily determined by the population size, decision variable dimensionality, and number of objectives.

As summarized in Table 2, which presents the theoretical computational complexity and average runtime over 30 independent runs on the C2TOP-Case1-CI-HS benchmark, CMHSA-SMF shows marginally higher theoretical complexity than baseline algorithms. However, the actual computation time difference is statistically insignificant, demonstrating that the framework maintains high solution quality while remaining computationally feasible.

## 4 Experiments and Discussions

### 4.1 Benchmark Problems

To validate the optimization performance of the proposed CMHSA-SMF algorithm, we selected two classical test suites **Competitive Two-Task Optimization Problems (C2TOP)** and **Competitive Four-Task Optimization Problems (C4TOP)** [33], along with the **Competitive Complex Multi-Tasking Optimization Problems (CCPLX)** [23] test suite for more complex problems, to further evaluate its efficacy. The detailed configurations of these test suites are described below.

**4.1.1 C2TOP.** The design of the C2TOP problem draws on challenges from MTO benchmarks. This suite comprises nine bi-task optimization problems, categorized into three classes based on the intersection degree of global optima across tasks: **Complete Intersection (CI)**, **Partial Intersection (PI)**, **Non-Intersection (NI)**. Each intersection class is further subdivided into three types, differentiated by fitness landscape similarity: **High Similarity (HS)**, **Medium Similarity (MS)**, **Low Similarity (LS)**. The classification framework is summarized in Table 3.

To enable C2TOP to simulate the characteristics of MTO benchmarks while introducing inter-task diversity, C2TOP applies specific offsets to the objective functions of each task. In the bi-task optimization scenario, the theoretical optimal solutions of the original objective functions  $f_1$  and  $f_2$  are both zero. However, to ensure distinct optimal solutions for the two tasks, constant offsets  $c_1$  and  $c_2$  are added to  $f_1$  and  $f_2$ , respectively, thereby forming the adjusted objective functions  $f_c^1(x)$  and  $f_c^2(x)$ . The specific expressions are as follows:

$$f_c^1(x) = f_1(x) + c_1, \quad (24)$$

$$f_c^2(x) = f_2(x) + c_2. \quad (25)$$

Based on the variance of  $c_1$  and  $c_2$ , C2TOP constructs the following four scenarios:



Table 3. Properties of C2TOP

| Category | Function        | Distribution Characteristics | Intersection Degree | Inter-Task Similarity |
|----------|-----------------|------------------------------|---------------------|-----------------------|
| CI+HS    | Griewank(T1)    | Multimodal, Non-Separable    | CI                  | 1.000                 |
|          | Rastrigin(T2)   | Multimodal, Non-Separable    |                     |                       |
| CI+MS    | Ackley(T1)      | Multimodal, Non-Separable    | CI                  | 0.2261                |
|          | Schwefel(T2)    | Multimodal, Non-Separable    |                     |                       |
| CI+LS    | Ackley(T1)      | Multimodal, Non-Separable    | CI                  | 0.002                 |
|          | Rastrigin(T2)   | Multimodal, Separable        |                     |                       |
| PI+HS    | Rastrigin(T1)   | Multimodal, Non-Separable    | PI                  | 0.8670                |
|          | Sphere(T2)      | Unimodal, Separable          |                     |                       |
| PI+MS    | Ackley(T1)      | Multimodal, Non-Separable    | PI                  | 0.2154                |
|          | Rosenbrock(T2)  | Multimodal, Non-Separable    |                     |                       |
| PI+LS    | Ackley(T1)      | Multimodal, Non-Separable    | PI                  | 0.0725                |
|          | Weierstrass(T2) | Multimodal, Non-Separable    |                     |                       |
| NI+HS    | Rosenbrock(T1)  | Multimodal, Non-Separable    | NI                  | 0.9434                |
|          | Rastrigin(T2)   | Multimodal, Non-Separable    |                     |                       |
| NI+MS    | Griewank(T1)    | Multimodal, Non-Separable    | NI                  | 0.3669                |
|          | Weierstrass(T2) | Multimodal, Non-Separable    |                     |                       |
| NI+LS    | Rastrigin(T1)   | Multimodal, Non-Separable    | NI                  | 0.0016                |
|          | Schwefel(T2)    | Multimodal, Separable        |                     |                       |

- (1) *C2TOP* – Case1:  $c_1 = 10, c_2 = 0$ ;
- (2) *C2TOP* – Case2:  $c_1 = 0, c_2 = 10$ ;
- (3) *C2TOP* – Case3:  $c_1 = 100, c_2 = 0$ ;
- (4) *C2TOP* – Case4:  $c_1 = 0, c_2 = 1,000$ .

Through this approach, C2TOP not only retains the core characteristics of MTO benchmarks but also enhances inter-task heterogeneity, thereby better simulating the complexity and diversity inherent in real-world multi-task environments.

**4.1.2 C4TOP.** To evaluate the performance of algorithms in CMTOP, particularly for scenarios involving more than two tasks, Li et al. [33] proposed the C4TOP test suite. This suite builds upon the MTO benchmark framework [44] and selects six classical single-objective benchmark functions: Rastrigin, Griewank, Sphere, Rosenbrock, Ackley, and Weierstrass. By leveraging these base functions, C4TOP constructs 15 complex MTO test problems, as summarized in Table 4. Each problem undergoes varying degrees of transformations in both domain and codomain to enhance the diversity and challenge levels of the problems.

The design of C4TOP not only preserves the characteristics of the original single-objective functions but also ensures heterogeneity among tasks through the introduction of inter-task offsets and transformations. Each test problem  $P_j$  contains multiple subtasks  $T_{i,j}$ , and the objective function  $f_c^i(x)$  of each subtask can be expressed as:

$$f_c^j(x) = f_j(x) + c_j, \quad (26)$$

where  $f_j(x)$  represents the original single-objective benchmark function, and  $c_j$  is a constant offset added to each task to ensure distinct optimal solutions across tasks.

**4.1.3 Competitive Complex Multi-Tasking Optimization Problems (CCPLX).** To validate the performance of the proposed algorithm on more complex problems, LI et al. [23] proposed the CCPLX test suite. This suite is based on the challenging problem set from the WCCI2020 Complex Multi-Task Test Suite and incorporates the competitive multi-task construction methodology of C2TOP

Table 4. C4TOP Component Tasks and Optimal Shifts

| Problem | $f_1$     | $c_1$ | $f_2$      | $c_1$ | $f_3$      | $c_3$ | $f_4$       | $c_4$ |
|---------|-----------|-------|------------|-------|------------|-------|-------------|-------|
| C4TOP1  | Rastrigin | 0     | Griewank   | 150   | Sphere     | 100   | Rosenbrock  | 50    |
| C4TOP2  | Rastrigin | 0     | Griewank   | 150   | Sphere     | 100   | Ackley      | 25    |
| C4TOP3  | Rastrigin | 0     | Griewank   | 150   | Sphere     | 100   | Weierstrass | 10    |
| C4TOP4  | Rastrigin | 0     | Griewank   | 150   | Rosenbrock | 50    | Ackley      | 25    |
| C4TOP5  | Rastrigin | 0     | Griewank   | 150   | Rosenbrock | 50    | Weierstrass | 10    |
| C4TOP6  | Rastrigin | 0     | Griewank   | 150   | Ackley     | 25    | Weierstrass | 10    |
| C4TOP7  | Rastrigin | 0     | Sphere     | 100   | Rosenbrock | 50    | Ackley      | 25    |
| C4TOP8  | Rastrigin | 0     | Sphere     | 100   | Rosenbrock | 50    | Weierstrass | 10    |
| C4TOP9  | Rastrigin | 0     | Sphere     | 100   | Ackley     | 25    | Weierstrass | 10    |
| C4TOP10 | Rastrigin | 0     | Rosenbrock | 50    | Ackley     | 25    | Weierstrass | 10    |
| C4TOP11 | Griewank  | 125   | Sphere     | 75    | Rosenbrock | 25    | Ackley      | 0     |
| C4TOP12 | Griewank  | 140   | Sphere     | 90    | Rosenbrock | 40    | Weierstrass | 0     |
| C4TOP13 | Griewank  | 140   | Sphere     | 90    | Ackley     | 15    | Weierstrass | 0     |
| C4TOP14 | Griewank  | 140   | Rosenbrock | 40    | Ackley     | 15    | Weierstrass | 0     |
| C4TOP15 | Sphere    | 90    | Rosenbrock | 40    | Ackley     | 15    | Weierstrass | 0     |

[44]. The CCPLX test suite aims to provide a more comprehensive and complex evaluation platform for testing and comparing the performance of diverse MTO algorithms. CCPLX can be expressed as:

- (1) *CCPLX – Case1*:  $c_1 = 100, c_2 = 0$ ;
- (2) *CCPLX – Case2*:  $c_1 = 0, c_2 = 1,000$ .

## 4.2 Experiment Settings

Two experiments are conducted to evaluate the performance of the proposed CMHSA-SMF. First, the Competitive Bi-Task Optimization Test Suite (C2TOP) was employed to validate the efficacy of the proposed historical parameter-aware adaptive resource allocation mechanism and the collaborative co-evolutionary dynamic coupling strategy for multi-task scenarios. Subsequently, the Competitive Quad-Task Optimization Test Suite (C4TOP) was utilized to further verify the effectiveness of these strategies. Finally, experiments on the Competitive Complex Multi-Task Suite (CCPLX) were conducted to demonstrate the algorithm's capability in addressing CMTOPs. All experiments were performed on a computer equipped with an Intel i7-9750H CPU running at 2.60 GHz and an NVIDIA GeForce RTX 1660 Ti GPU with 6 GB of graphics memory, equipped with 16 GB DDR4 RAM and a 1 TB SSD. This study compared seven representative evolutionary algorithms, including both state-of-the-art CMTO algorithms (e.g., MFEA [14], MFDE [19], MTEA-HKTS [27]) and classical MTO algorithms, as well as classic DE. The experimental validation comprised three phases: Comparative analysis with state-of-the-art MTO algorithms, including MFEA, MFDE, and MTEA-HKTS, was conducted to evaluate the competitiveness of CMHSA-SMF; Superiority verification of the crossover operator was performed by benchmarking CMHSA-SMF against classical DE and SHADE; Effectiveness in CMTOPs was assessed through comparisons with DEORA and MTSRA, which are specifically designed for CMTOP.

The number of tasks  $K$  in C2TOP, C4TOP, and CCPLX is set to 2, 4, and 2, respectively. The total population size per task is 100, resulting in a total problem size of  $K * 100$  per problem. For C2TOP and CCPLX, the maximum number of function evaluations  $E$  per problem is  $K * N * 1,000$ , while for C4TOP, it is  $K * N * 500$ , as summarized in Table 5.

Table 5. Global Parameter Settings

| Parameter                            | C2TOP           | C4TOP         | CCPLX           |
|--------------------------------------|-----------------|---------------|-----------------|
| Number of Tasks ( $K$ )              | 2               | 4             | 2               |
| Population Size ( $N$ )              | $K * 100$       | $K * 100$     | $K * 100$       |
| Maximum Function Evaluations ( $E$ ) | $K * N * 1,000$ | $K * N * 500$ | $K * N * 1,000$ |

Table 6. Parameter Settings of the Employed Multi-Objective Optimization Algorithms

| Algorithm | Population Size | Parameter Setting                                                                                        |
|-----------|-----------------|----------------------------------------------------------------------------------------------------------|
| DE        | 100             | $CR = 0.9, F = 0.5$                                                                                      |
| SHADE     | 100             | $p = 0.1, H = 100$                                                                                       |
| MFEA      | 100             | $rmf = 0.3, SBX = 2, mum = 5$                                                                            |
| MFDE      | 100             | $rmf = 0.3, F = 0.5, CR = 0.9$                                                                           |
| DEORA     | 100             | $CR = 0.9, F = 0.5, \alpha = 0.5, \beta = 0.2, \gamma = 0.85, pmin = 1.0, P0 = 0.3$                      |
| MTSRA     | 100             | $p = 0.1, H = 100, \alpha = 0.5, \beta = 0.2, pmin = 1.0, P0 = 0.3, RH = 100$                            |
| MTEA-HKTS | 100             | $mu = 2, mum = 5, F = 0.5, CR = 0.5, minx = 0.1, Lb = 0.1, Ub = 0.7, w = 0.3$                            |
| CMHSA-SMF | 100             | $p = 0.1, H = 100, \alpha = 0.5, \beta = 0.5, \gamma = 0, \lambda = 0.2, pmin = 1.0, P0 = 0.3, RH = 100$ |

The population size for all compared algorithms was set to 100, and the maximum number of fitness evaluations per problem was set to  $10^5$ . To ensure fair comparisons, this study adopted the recommended parameter configurations from the literature that yielded the best performance for each algorithm, with specific parameters listed in Table 6. Additionally, statistical analysis was performed using the Wilcoxon rank-sum test at a significance level of 0.05 to validate the performance of the proposed method. All experiments were executed over 30 independent runs for each algorithm.

### 4.3 Comparative Analysis of Classical CMTO Benchmark Suites

The mean objective values and their standard deviations across 30 independent runs for all compared algorithms on the C2TOP and C4TOP test suites are presented in Table 7 and Table 8, respectively. Each column in the tables corresponds to the performance of an algorithm across different test problems. For each test case in C2TOP and C4TOP, the symbols “+”, “-”, and “=” appended to the results indicate whether the algorithm’s performance is significantly superior to, significantly inferior to, or statistically comparable to the baseline algorithm (CMHSA-SMF), respectively, based on the Wilcoxon rank-sum test ( $\alpha = 0.05$ ). The last row labeled “Total +/–/=” summarizes the aggregated statistical outcomes across all test sets. Figure 5 provides a more intuitive visualization of the convergence behaviors of all compared algorithms on the C2TOP and C4TOP test suites. Based on the experimental data in Tables 7 and 8 and Figure 5, the following conclusions can be drawn:

- (1) CMHSA-SMF demonstrates universal superiority over non-competitive algorithms. The algorithm exhibits statistically superior performance over DE, MFEA, and MFDE across all test problems. This is attributed to the fundamental fact that these algorithms are not designed for competitive environments. Their operators are either oblivious to other tasks or employ a fixed cooperative transfer strategy, which is detrimental when tasks are in conflict.

Table 7. Average Objective Value and Standard Deviation Obtained by DE, SHADE, MFEA, MFDE, DEORA, MTSRA, MTEA-HKTS, and CMHSA-SMF on C2TOP for 30 Independent Runs

| Problems | Task | DE                  | SHADE                       | MFEA                | MFDE                | DEORA                      | MTSRA                       | MTEA-HKTS                  | CMHSA-SMF                  |
|----------|------|---------------------|-----------------------------|---------------------|---------------------|----------------------------|-----------------------------|----------------------------|----------------------------|
| CI-HS    | T1   | 1.00e+01(6.96e-04)- | <b>1.00e+01(1.68e-15)=</b>  | 1.00e+01(4.83e-03)- | 1.00e+01(2.04e-04)- | 1.00e+01(7.03e-04)-        | <b>1.00e+01(5.76e-16)=</b>  | <b>1.00e+01(1.63e-15)=</b> | <b>1.00e+01(0.00e+00)</b>  |
|          | T2   | 4.02e+02(1.62e+01)- | 1.33e+02(1.47e+01)-         | 8.45e+01(3.76e+01)- | 5.57e-01(2.98e-01)- | 4.63e+01(1.13e+02)         | <b>0.0e+00(0.00e+00)=</b>   | 1.41e-13(6.32e-13)=        | <b>0.00e+00(0.00e+00)</b>  |
| CI-MS    | T1   | 1.03e+01(5.96e-01)- | 1.00e+01(4.89e-15)-         | 1.11e+01(7.52e-01)- | 1.00e+01(3.98e-03)- | 1.00e+01(7.01e-03)-        | <b>1.00e+01(4.16e-15)=</b>  | 1.00e+01(8.69e-11)-        | <b>1.00e+01(1.63e-15)</b>  |
|          | T2   | 4.00e+02(1.45e+01)- | 1.37e+02(1.52e+01)-         | 8.46e+01(3.96e+01)- | 1.28e-01(6.25e-02)- | 3.05e+01(7.36e+01)-        | <b>0.00e+00(0.00e+00)=</b>  | <b>0.00e+00(0.00e+00)=</b> | <b>0.00e+00(0.00e+00)</b>  |
| CI-LS    | T1   | 3.11e+01(4.30e-02)- | 3.07e+01(7.56e-02)-         | 3.01e+01(7.16e-02)+ | 3.11e+01(2.95e-02)- | 3.11e+01(3.92e-02)-        | 3.00e+01(3.83e+00)+         | <b>1.20e+01(6.17e+00)+</b> | 3.05e+01(5.48e-02)         |
|          | T2   | 1.35e+04(7.37e+02)- | 2.21e+02(7.44e+01)-         | 2.15e+03(2.65e+02)- | 1.39e+04(4.83e+02)- | 1.36e+04(1.54e+03)-        | 1.81e+02(1.15e+02)-         | 2.01e+02(4.55e+02)=        | <b>6.39e-04(7.69e-06)</b>  |
| PI-HS    | T1   | 4.05e+02(1.90e+01)- | 1.43e+02(1.17e+01)-         | 2.41e+02(5.37e+01)- | 7.64e+01(2.64e+01)= | 5.89e+02(9.08e+01)-        | 3.17e+02(2.06e+01)-         | <b>1.00e+01(0.00e+00)+</b> | 8.46e+01(9.51e+00)         |
|          | T2   | 1.37e-03(6.36e-04)- | 1.56e-27(3.20e-28)-         | 1.00e-04(2.97e-04)- | 1.92e-03(8.85e-04)- | 1.25e-07(1.80e-07)-        | <b>0.00e+00(0.00e+00)=</b>  | 4.73e-24(2.02e-23)-        | <b>0.00e+00(0.00e+00)</b>  |
| PI-MS    | T1   | 1.07e+01(1.15e+00)- | 1.00e+01(5.47e-15)=         | 1.03e+01(6.05e-01)- | 1.00e+01(4.77e-03)- | 1.00e+01(6.22e-02)-        | 1.00e+01(5.25e-04)-         | 1.02e+01(5.11e-01)-        | 1.00e+01(4.54e-15)         |
|          | T2   | 6.10e+01(3.18e+01)- | 2.97e+01(4.39e+00)-         | 9.49e+01(1.54e+01)- | 4.74e+01(5.29e-01)- | 6.96e+01(2.02e+01)-        | 1.03e+00(4.60e+00)=         | 3.95e+01(1.70e+01)-        | 2.30e-03(2.37e-03)         |
| PI-LS    | T1   | 1.10e+01(1.24e+00)- | 1.00e+01(4.31e-15)-         | 2.63e+01(7.41e+00)- | 1.04e+01(2.72e-01)- | 1.52e+01(3.26e+00)-        | 1.00e+01(5.00e-12)-         | 1.03e+01(6.73e-01)-        | <b>1.00e+01(8.15e-16)</b>  |
|          | T2   | 3.64e+00(2.26e+00)- | <b>-1.98e-18(0.00e+00)=</b> | 1.57e+01(7.06e+00)- | 4.75e-01(3.32e-01)- | 8.00e-03(8.15e-03)-        | <b>-1.98e-18(0.00e+00)=</b> | 2.05e-08(2.32e-08)-        | <b>-1.98e-18(0.00e+00)</b> |
| NI-HS    | T1   | 7.05e+01(2.81e+01)- | 3.77e+01(1.76e+00)-         | 1.43e+02(5.91e+01)- | 5.77e+01(8.96e-01)- | 5.62e+01(1.31e+00)         | 4.58e+01(1.88e+00)-         | 5.26e+01(1.45e+01)-        | <b>2.76e+01(1.83e+00)</b>  |
|          | T2   | 4.00e+02(2.09e+01)- | 1.35e+02(1.46e+01)-         | 1.07e+02(4.34e+01)- | 9.61e-01(9.32e-01)- | 2.85e+02(1.14e+02)-        | <b>0.00e+00(0.00e+00)=</b>  | 1.77e+01(7.91e+01)=        | <b>0.00e+00(0.00e+00)</b>  |
| NI-MS    | T1   | 1.00e+01(4.61e-04)- | <b>1.00e+01(1.58e-15)=</b>  | 1.00e+01(5.00e-03)- | 1.00e+01(5.65e-04)- | 1.10e+01(2.62e-02)-        | 1.00e+01(3.62e-03)-         | 1.00e+01(6.74e-07)-        | 1.00e+01(2.08e-15)         |
|          | T2   | 1.41e+01(7.74e+00)- | 8.40e-01(4.86e-01)=         | 2.10e+01(3.84e+00)- | 9.49e-01(1.74e-01)= | <b>4.96e-01(3.08e-01)+</b> | 2.52e+00(9.00e-01)-         | <b>2.59e-04(9.14e-04)+</b> | 1.05e+00(6.12e-01)         |
| NI-LS    | T1   | 4.15e+02(1.12e+01)- | 1.45e+02(1.31e+01)-         | 2.62e+02(6.38e+01)- | 4.06e+02(1.82e+01)- | 4.08e+02(1.77e+01)-        | 1.20e+02(1.62e+01)-         | <b>6.56e+01(2.95e+01)+</b> | 8.01e+01(8.57e+00)         |
|          | T2   | 1.30e+04(7.37e+02)- | 2.58e+02(5.90e+01)-         | 2.01e+03(3.73e+02)- | 2.53e+03(6.26e+02)- | 1.41e+04(7.63e+02)-        | 5.80e+03(1.73e+03)-         | 1.28e+03(1.93e+03)-        | <b>2.36e+01(6.20e+01)</b>  |
| +/-/=    |      | 0/18/0              | 0/13/5                      | 1/17/0              | 0/16/2              | 1/17/0                     | 1/9/8                       | 4/9/4                      | Base                       |

(Continued)

Table 7. Continued

| Problems | Task | DE                  | SHADE                      | MFEA                | MFDE                | DEORA               | MTSRA                      | MTEA-HKTS                  | CMHSA-SMF                 |
|----------|------|---------------------|----------------------------|---------------------|---------------------|---------------------|----------------------------|----------------------------|---------------------------|
| CI-HS    | T1   | 1.32e-03(6.59e-04)- | 3.69e-04(1.65e-03)=        | 2.76e-03(4.04e-03)- | 3.92e-04(1.44e-04)- | 8.38e-06(1.11e-05)- | <b>0.00e+00(0.00e+00)=</b> | <b>0.00e+00(0.00e+00)=</b> | <b>0.00e+00(0.00e+00)</b> |
|          | T2   | 4.06e+02(1.55e+01)- | 1.41e+02(1.49e+01)-        | 8.75e+01(2.58e+01)- | 1.05e+01(2.08e-01)- | 1.98e+02(1.67e+02)- | <b>1.00e+01(4.08e-16)=</b> | <b>1.00e+01(0.00e+00)=</b> | <b>1.00e+01(0.00e+00)</b> |
| CI-MS    | T1   | 6.70e-01(7.81e-01)- | 1.44e-14(2.44e-15)-        | 1.07e+00(6.99e-01)- | 1.64e-02(3.01e-03)- | 4.45e-04(3.08e-04)- | <b>4.44e-16(0.00e+00)=</b> | 9.49e-12(3.67e-11)-        | 1.50e-15(3.48e-15)        |
|          | T2   | 4.13e+02(1.59e+01)- | 1.44e+02(1.24e+01)-        | 8.54e+01(4.33e+01)- | 1.01e+01(4.98e-02)- | 2.80e+02(1.15e+02)- | <b>1.00e+01(0.00e+00)=</b> | <b>1.00e+01(0.00e+00)=</b> | <b>1.00e+01(0.00e+00)</b> |
| CI-LS    | T1   | 2.11e+01(4.07e-02)- | 2.08e+01(5.12e-02)-        | 2.01e+01(8.85e-02)= | 2.11e+01(3.74e-02)- | 2.12e+01(3.10e-02)- | 2.01e+01(3.79e+00)-        | <b>5.13-06(2.27e-05)+</b>  | 1.53e+01(9.12e+00)        |
|          | T2   | 1.34e+04(7.23e+02)- | 2.63e+02(6.73e+01)-        | 2.17e+03(4.15e+02)- | 1.35e+04(8.24e+02)- | 1.41e+04(4.54e+02)- | 2.07e+02(1.60e+02)-        | 2.18e+01(5.30e+01)-        | <b>1.00e+01(3.21e-06)</b> |
| PI-HS    | T1   | 3.99e+02(1.50e+01)- | 1.36e+02(1.05e+01)-        | 2.62e+02(6.17e+01)- | 7.09e+01(4.24e+01)+ | 4.07e+02(1.80e+01)- | 1.23e+02(1.31e+01)-        | <b>0.00e+00(0.00e+00)+</b> | 7.90e+01(7.99e+00)        |
|          | T2   | 1.00e+01(4.54e-04)- | 1.00e+01(4.08e-16)=        | 1.00e+01(2.10e-03)- | 1.00e+01(9.11e-04)- | 1.00e+01(4.72e-03)- | 1.00e+01(5.00e-09)-        | <b>1.00e+01(9.98e-16)=</b> | 1.00e+01(0.00e+00)        |
| PI-MS    | T1   | 8.84+01(6.70e-01)-  | 1.48e-14(7.94e-16)-        | 3.63e-01(6.28e-01)- | 2.38e-02(6.18e-03)- | 2.20e-02(3.04e-02)- | <b>5.41e-15(4.67e-15)+</b> | 1.60e-01(3.93e-01)-        | 1.27e-14(2.15e-15)        |
|          | T2   | 6.71e+01(2.36e+01)- | 4.46e+01(1.78e+01)-        | 1.02e+02(1.15e+01)- | 6.04e+01(1.22e+01)- | 8.69e+01(2.55e+01)- | 9.82e+01(2.39e+00)-        | 5.19e+01(1.43e+01)-        | <b>1.00e+01(5.10e-03)</b> |
| PI-LS    | T1   | 8.82-01(1.12e+00)-  | 1.46e-14(1.15e-15)-        | 1.35e+01(8.87e+00)- | 6.41e-01(6.83e-01)- | 1.08e-04(3.34e-05)- | <b>4.32e-16(0.00e+00)=</b> | 5.77e-02(2.58e-01)-        | 2.04e-15(3.15e-15)        |
|          | T2   | 1.29e+01(2.83e+00)- | <b>1.00e+01(0.00e+00)=</b> | 2.26e+01(8.15e+00)- | 1.05e+01(4.14e-01)- | 1.00e+01(2.57e-02)- | <b>1.00e+01(0.00e+00)=</b> | 1.00e+01(1.20e-08)-        | <b>1.00e+01(0.00e+00)</b> |
| NI-HS    | T1   | 5.91e+01(2.43e+01)- | 2.99e+01(3.08e+00)-        | 1.18e+02(3.58e+01)- | 4.79e+01(9.52e-01)- | 4.86e+01(1.28e+01)- | 2.48e+01(1.44e+00)-        | 4.02e+01(1.73e+01)-        | <b>1.81e+01(1.51e+00)</b> |
|          | T2   | 4.10e+02(1.77e+01)- | 1.43e+02(1.33e+01)-        | 1.29e+02(4.74e+01)- | 1.32e+01(9.91e+00)- | 3.09e+02(1.05e+02)- | 1.00e+01(2.27e-05)-        | 2.78e+01(7.97e+01)=        | <b>1.00e+01(0.00e+00)</b> |
| NI-MS    | T1   | 1.45e-03(8.26e-04)- | <b>0.00e+00(0.00e+00)+</b> | 5.40e-03(5.08e-03)- | 1.26e-03(4.44e-04)- | 4.14e-06(1.30e-05)- | 3.88e-17(6.52e-17)+        | 3.70e-04(1.66e-03)-        | 1.11e-16(1.40e-16)        |
|          | T2   | 2.24e+01(5.95e+00)- | 1.09e+01(6.51e-01)=        | 2.89e+01(4.07e+00)- | 1.10e+01(3.39e-01)= | 3.22e+01(3.58e+00)- | 1.23e+01(6.97e-01)-        | <b>1.00e+01(6.89e-05)+</b> | 1.12e+01(4.42e-01)        |
| NI-LS    | T1   | 4.07e+02(1.17e+01)- | 1.31e+02(1.23e+01)-        | 2.54e+02(6.07e+01)- | 3.99e+02(1.84e+01)- | 3.93e+02(1.66e+01)- | 1.09e+02(1.66e+01)-        | <b>4.84e+01(2.00e+01)+</b> | 6.84e+01(7.23e+00)        |
|          | T2   | 1.37e+04(4.57e+02)- | 2.04e+02(5.73e+01)-        | 2.26e+03(5.28e+02)- | 2.51e+03(6.11e+02)- | 1.44e+04(7.62e+02)- | 6.95e+03(1.53e+03)-        | 5.42e+02(1.39e+03)=        | <b>1.59e+01(2.65e+01)</b> |
| +/-/=    |      | 0/18/0              | 1/13/4                     | 0/17/1              | 1/16/1              | 0/18/0              | 2/10/6                     | 4/8/6                      | Base                      |

(Continued)



Table 7. Continued

| Problems | Task | DE                  | SHADE                      | MFEA                | MFDE                | DEORA                      | MTSRA                       | MTEA-HKTS                  | CMHSA-SMF                  |
|----------|------|---------------------|----------------------------|---------------------|---------------------|----------------------------|-----------------------------|----------------------------|----------------------------|
| CI-HS    | T1   | 1.00e+03(3.88e-04)- | 1.00e+03(1.11e-13)=        | 1.00e+03(5.21e-03)- | 1.00e+03(1.46e-04)- | 1.00e+03(1.35e-02)-        | <b>1.00e+03(0.00e+00)=</b>  | <b>1.00e+03(0.00e+00)=</b> | <b>1.00e+03(0.00e+00)</b>  |
|          | T2   | 3.99e+02(1.91e+01)- | 1.37e+02(1.64e+01)-        | 8.90e+01(2.91e+01)- | 4.69e-01(1.85e-01)- | 1.44e+02(1.65e+02)-        | <b>0.00e+00(0.00e+00)=</b>  | 1.50e-15(6.75e-15)=        | <b>0.00e+00(0.00e+00)</b>  |
|          | T1   | 1.00e+03(8.74e-01)- | <b>1.00e+03(2.61e-14)=</b> | 1.00e+03(6.72e-01)- | 1.00e+03(3.60e-03)- | 1.00e+03(5.12e-02)-        | <b>1.00e+03(5.83e-14)=</b>  | 1.00e+03(2.33e-10)-        | <b>1.00e+03(1.11e-13)</b>  |
|          | T2   | 4.01+02(1.67e+01)-  | 1.31e+02(1.72e+01)-        | 6.85e+01(2.84e+01)- | 1.40e-01(7.04e-02)- | 2.46e+02(1.67e+02)-        | <b>0.00e+00(0.0e+00)=</b>   | <b>0.00e+00(0.0e+00)=</b>  | <b>0.00e+00(0.0e+00)</b>   |
| CI-LS    | T1   | 1.02e+03(5.75e-02)- | 1.02e+03(4.63e-02)-        | 1.02e+03(5.17e-02)- | 1.02e+03(3.82e-02)- | 1.02e+03(2.72e-02)-        | 1.02e+03(7.19e-02)-         | <b>1.00e+03(7.11e-05)+</b> | 1.01e+03(4.59e+00)         |
|          | T2   | 1.31e+04(7.16e+02)- | 2.14e+02(6.51e+01)-        | 2.19e+03(2.96e+02)- | 1.33e+04(6.91e+02)- | 1.37e+04(1.23e+03)-        | 1.46e+02(8.15e+01)-         | 1.77e+01(7.95e+01)-        | <b>5.92e+00(2.65e+01)</b>  |
|          | T1   | 1.40e+03(1.60e+01)- | 1.13e+03(1.27e+01)-        | 1.22e+03(6.14e+01)- | 1.05e+03(1.94+01)+  | 1.70e+03(8.90e+01)-        | 1.31e+03(2.53e+01)-         | <b>1.00e+03(0.0e+00)+</b>  | 1.07e+03(1.01e+01)         |
|          | T2   | 1.26e-03(6.44e-04)- | 1.58e-27(2.30e-28)-        | 2.30e-03(7.21e-03)- | 1.99e-03(9.91e-04)- | 2.42e-08(1.47e-08)-        | <b>0.00e+00(0.0e+00)=</b>   | 1.04e-24(2.86e-24)-        | 3.02e-29(9.88e-29)         |
| PI-HS    | T1   | 1.00e+03(9.30e-01)- | <b>1.00e+03(2.61e-14)=</b> | 1.00e+03(6.05e-01)- | 1.00e+03(6.43e-03)- | 1.70e+03(8.90e+01)-        | 1.00e+03(1.59e-02)-         | 1.00e+03(4.23e-01)-        | <b>1.00e+03(1.63e-13)</b>  |
|          | T2   | 5.55e+01(2.78e+01)- | 2.93e+01(2.67e+00)-        | 9.49e+01(2.40e+01)- | 4.94e+01(9.00e+00)- | 2.42e-08(1.47e-08)-        | <b>2.94e-04(8.34e-04)+</b>  | 4.41e+01(1.04e+01)-        | 3.97e-03(5.10e-03)         |
|          | T1   | 1.00e+03(4.83e-01)- | <b>1.00e+03(2.61e-14)=</b> | 1.01e+03(6.69e+00)- | 1.00e+03(9.22e-01)- | 1.00e+03(2.02e+00)-        | 1.00e+03(6.59e-12)-         | 1.00e+03(8.42e-01)-        | <b>1.00e+03(1.04e-13)</b>  |
|          | T2   | 3.68e+00(2.78e+00)- | <b>-1.99-18(0.00e+00)=</b> | 1.59e+01(6.55e+00)- | 6.07e-01(6.97e-01)- | 2.39e-03(1.13e-03)-        | <b>-1.98e-18(0.00e+00)=</b> | 2.41e-08(3.64e-08)-        | <b>-1.98e-18(0.00e+00)</b> |
| NI-HS    | T1   | 1.06e+03(2.79e+01)- | 1.03+03(1.81e+00)-         | 1.15e+03(1.09e+02)- | 1.04e+03(9.53e-01)- | 1.04e+03(9.21e-01)-        | 1.03e+03(1.49e+00)-         | 1.04e+03(1.06e+01)-        | <b>1.01e+03(2.26e+00)</b>  |
|          | T2   | 4.03e+02(1.28e+01)- | 1.35e+02(1.15e+01)-        | 1.26e+02(3.02e+01)- | 1.63e+00(1.43e+00)- | 1.47e+02(1.34e+02)-        | <b>0.00e+00(0.00e+00)=</b>  | 1.69e+01(7.55e+01)=        | <b>0.00e+00(0.00e+00)</b>  |
|          | T1   | 1.00e+03(3.69e-04)- | <b>1.00e+03(1.17e-13)=</b> | 1.00e+03(7.19e-03)- | 1.00e+03(7.15e-04)- | 1.00e+03(6.77e-02)-        | 1.00e+03(5.41e-10)-         | 1.00e+03(5.57e-07)-        | <b>1.00e+03(1.28e-13)</b>  |
|          | T2   | 1.02e+01(4.89e+00)- | 8.94-01(5.56e-01)=         | 2.02e+01(3.49e+00)- | 9.63e-01(1.99e-01)+ | <b>3.47e-01(2.24e-01)+</b> | 2.10e+00(7.61e-01)=         | <b>5.78e-04(1.55e-03)+</b> | 1.43e+00(9.64e-01)         |
| NI-LS    | T1   | 1.39e+03(1.40e+01)- | 1.13e+03(1.25e+01)-        | 1.26e+03(5.96e+01)- | 1.40e+03(1.50e+01)- | 1.39e+03(1.87e+01)-        | 1.29e+03(2.89e+01)-         | <b>1.05e+03(2.23e+01)+</b> | 1.07e+03(7.49e+00)         |
|          | T2   | 1.35e+04(7.86e+02)- | 2.32e+02(9.71e+01)-        | 2.17e+03(4.79e+02)- | 2.64e+03(6.87e+02)- | 1.41e+04(6.45e+02)-        | <b>1.07e+00(1.36e+00)+</b>  | 3.84e+02(7.57e+02)=        | 5.92e+00(2.65e+01)         |
| +/-/=    |      | 0/18/0              | 0/11/7                     | 0/18/0              | 2/16/0              | 1/17/0                     | 2/8/8                       | 4/9/5                      | Base                       |

(Continued)

Table 7. Continued

| Problems   | Task | DE                  | SHADE                      | MFEA                | MFDE                 | DEORA               | MTSRA                      | MTEA-HKTS                  | CMHSA-SMF                 |
|------------|------|---------------------|----------------------------|---------------------|----------------------|---------------------|----------------------------|----------------------------|---------------------------|
| CI-HS      | T1   | 1.30e-03(5.02e-04)- | <b>0.00e+00(0.00e+00)=</b> | 3.18e-03(4.57e-03)- | 3.99e-04(2.62e-04)-  | 2.06e-06(1.45e-06)- | <b>0.00e+00(0.00e+00)=</b> | 3.70e-04(1.66e-03)=        | <b>0.00e+00(0.00e+00)</b> |
|            | T2   | 1.40e+03(1.84e+01)- | 1.13e+03(1.47e+01)-        | 1.09e+03(4.17e+01)- | 1.00e+03(3.74e-01)-  | 1.12e+03(1.28e+02)- | <b>1.00e+03(2.61e-14)=</b> | 1.00e+03(2.22e-01)=        | <b>1.00e+03(0.00e+00)</b> |
| CI-MS      | T1   | 3.89e-01(3.45e-01)- | 1.44e-14(1.40e-15)-        | 9.08e-01(8.84e-01)- | 1.49e-02(3.93e-03)-  | 3.38e-04(1.68e-04)- | <b>4.44e-16(0.00e+00)=</b> | 1.49e-10(5.58e-10)-        | <b>4.44e-16(0.00e+00)</b> |
|            | T2   | 1.40e+03(1.68e+01)- | 1.13e+03(1.39e+01)-        | 1.08e+03(4.23e+01)- | 1.00e+03(7.59e-02)-  | 1.30e+03(9.36e+01)- | <b>1.00e+03(0.00e+00)=</b> | <b>1.00e+03(0.00e+00)=</b> | <b>1.00e+03(0.00e+00)</b> |
| CI-LS      | T1   | 2.11e+01(4.45e-02)- | 2.07e+01(5.83e-02)-        | 2.01e+01(6.08e-02)+ | 2.11e+01(5.60e-02)-  | 2.11e+01(3.94e-02)- | 2.00e+01(3.99e+00)+        | <b>9.98e-06(3.14e-05)+</b> | 2.05e+01(5.72e-02)        |
|            | T2   | 1.43e+04(5.82e+02)- | 1.20e+03(6.64e+01)-        | 3.16e+03(4.44e+02)- | 1.45e+04(9.56e+02)-  | 1.50e+04(6.18e+02)- | 1.14e+03(1.17e+02)-        | 1.02e+03(1.06e+02)-        | <b>1.00e+03(4.67e-06)</b> |
| PI-HS      | T1   | 4.03e+02(1.59e+01)- | 1.31e+02(1.28e+01)-        | 2.42e+02(5.23e+01)- | 6.98e+01(1.89e+01)+  | 4.01e+02(1.54e+01)- | 1.04e+02(1.02e+01)-        | <b>0.00e+00(0.00e+00)+</b> | 7.70e+01(8.70e+00)        |
|            | T2   | 1.00e+03(5.46e-04)- | 1.00e+03(1.17e-13)=        | 1.00e+03(2.05e-03)- | 1.00e+03(8.93e-04)-  | 1.00e+03(3.77e-01)- | 1.00e+03(7.31e-04)-        | <b>1.00e+03(4.52e-14)=</b> | 1.00e+03(1.17e-13)        |
| PI-MS      | T1   | 1.15e+00(1.94e+00)- | 1.48e-14(7.94e-16)-        | 4.88e-01(6.71e-01)- | 2.08e-02(6.45e-03)-  | 2.37e-03(3.31e-03)- | <b>5.41e-15(4.52e-15)+</b> | 1.26e-01(3.91e-01)-        | 1.32e-14(2.68e-15)        |
|            | T2   | 1.06e+03(2.81e+01)- | 1.03e+03(1.42e+01)-        | 1.10e+03(3.71e+01)- | 1.04e+03(3.86e-01)-  | 1.10e+03(1.13e+01)- | 1.08e+03(8.99e+00)-        | 1.04e+03(1.04e+01)-        | <b>1.00e+03(3.80e+00)</b> |
| PI-LS      | T1   | 1.67e+00(3.99e+00)- | 1.46e-14(1.63e-15)-        | 1.72e+01(6.67e+00)- | 5.10e-01(5.85e-01)-  | 6.46e-05(2.47e-05)- | <b>4.44e-16(0.00e+00)=</b> | 3.90e-01(8.56e-01)-        | 2.93e-15(4.62e-15)        |
|            | T2   | 1.00e+03(2.78e+00)- | <b>1.00e+03(0.00e+00)=</b> | 1.01e+03(7.25e+00)- | 1.00e+03(3.40e-01)-  | 1.00e+03(5.69e-02)- | <b>1.00e+03(0.00e+00)=</b> | 1.00e+03(1.40e-08)-        | <b>1.00e+03(0.00e+00)</b> |
| NI-HS      | T1   | 5.51e+01(2.03e+01)- | 3.70e+01(2.00e+01)-        | 1.43e+02(1.02e+02)- | 4.79e+01(9.08e-01)-  | 4.50e+01(7.45e-01)- | 1.96e+01(1.32e+00)=        | 4.23e+01(1.45e+01)-        | <b>1.88e+01(1.78e+00)</b> |
|            | T2   | 1.40e+03(1.06e+01)- | 1.13e+03(1.68e+01)-        | 1.11e+03(3.92e+01)- | 1.00e+03(1.90e+00)-  | 1.34e+03(1.95e+01)- | 1.00e+03(1.72e+00)-        | 1.05e+03(1.25e+02)=        | <b>1.00e+03(0.00e+00)</b> |
| NI-MS      | T1   | 1.23e-03(5.61e-04)- | <b>5.55e-18(2.48e-17)+</b> | 6.63e-03(1.10e-02)- | 1.359e-03(5.30e-04)- | 5.25e-07(4.25e-07)- | 2.22e-17(4.56e-17)+        | 2.31e-07(2.99e-07)-        | 1.11e-16(1.30e-16)        |
|            | T2   | 1.01e+03(6.66e+00)- | 1.00e+03(4.34e-01)=        | 1.01e+03(3.98e+00)- | 1.00e+03(2.94e-01)=  | 1.02e+03(1.13e+00)- | 1.00e+03(8.48e-01)-        | <b>1.00e+03(8.81e-03)+</b> | 1.00e+03(8.00e-01)        |
| NI-LS      | T1   | 4.03e+02(1.88e+01)- | 1.28e+02(1.34e+01)-        | 2.72e+02(5.15e+01)- | 3.99e+02(1.05e+01)-  | 3.95e+02(1.80e+01)- | 1.05e+02(1.33e+01)-        | <b>6.06e+01(2.20e+01)+</b> | 7.53e+01(9.35e+00)        |
|            | T2   | 1.43e+04(7.71e+02)- | 1.21e+03(6.47e+01)-        | 3.25e+03(4.16e+02)- | 3.37e+03(7.17e+02)-  | 1.54e+04(4.61e+02)- | 8.25e+03(1.57e+03)-        | 1.57e+03(1.39e+03)=        | <b>1.00e+03(2.65e+01)</b> |
| +/-/=      |      | 0/18/0              | 1/13/4                     | 1/17/0              | 1/16/1               | 0/18/0              | 2/8/8                      | 4/8/6                      | Base                      |
| Total+/-/= |      | 8/35/29             | 2/50/20                    | 2/69/1              | 4/64/4               | 2/70/0              | 8/35/29                    | 16/34/22                   | Base                      |

Best result in each row is highlighted. “+/-/=” denotes the algorithm in each column is significantly better, worse, and similar to the base algorithm on the test set or case above it.

Table 8. Average Objective Value and Standard Deviation Obtained by DE, SHADE, MFEA, MFDE, DEORA, MTSRA, MTEA-HKTS, and CMHSA-SMF on C4TOP for 30 Independent Runs

| Problems | Task | DE                  | SHADE                      | MFEA                | MFDE                | DEORA               | MTSRA                      | MTEA-HKTS                  | CMHSA-SMF                 |
|----------|------|---------------------|----------------------------|---------------------|---------------------|---------------------|----------------------------|----------------------------|---------------------------|
| C4TOP1   | T1   | 4.30e+02(2.06e+01)- | 1.41e+02(7.96e+00)-        | 2.24e+01(2.80e+01)- | 3.64e+02(1.82e+01)- | 2.19e+02(1.62e+02)- | <b>0.00e+00(0.00e+00)=</b> | 8.69e-01(3.89e+00)-        | 1.42e-15(6.36e-15)        |
|          | T2   | 1.50e+02(1.30e-01)- | 1.50e+02(1.13e-14)-        | 1.50e+02(1.21e-02)- | 1.50e+02(1.26e-01)- | 1.50e+02(4.36e-01)- | 1.50e+02(7.89e-05)-        | 1.50e+02(8.69e-13)-        | <b>1.50e+02(0.00e+00)</b> |
|          | T3   | 1.10e+02(2.57e+00)- | 1.00e+02(2.56e-11)-        | 1.03e+02(2.16e+00)- | 1.12e+02(3.63e+00)- | 3.09e+02(5.85e+02)- | 1.00e+02(3.52e-04)-        | 1.00e+02(1.02e-06)-        | <b>1.00e+02(2.11e-14)</b> |
|          | T4   | 3.15e+03(1.34e+03)- | 1.01e+02(2.12e+01)-        | 1.12e+02(3.30e+01)- | 9.65e+02(3.71e+02)- | 1.07e+02(3.31e+01)- | 9.21e+01(9.33e-01)-        | 9.69e+01(2.32e-01)-        | <b>8.52e+01(1.20e+00)</b> |
| C4TOP2   | T1   | 4.34e+02(1.64e+01)- | 1.42e+02(9.29e+00)-        | 6.27e+01(1.31e+01)- | 3.91e+02(1.97e+01)- | 1.99e+02(1.86e+02)- | <b>0.00e+00(0.00e+00)+</b> | 8.21e-04(3.64e-03)-        | 1.26e-11(1.92e-11)        |
|          | T2   | 1.50e+02(4.73e-02)- | 1.50e+02(9.55e-11)-        | 1.50e+02(4.41e-02)- | 1.50e+02(1.13e-01)- | 1.50e+02(5.58e-01)- | 1.50e+02(3.10e-10)-        | 1.50e+02(2.71e-13)-        | <b>1.50e+02(1.84e-14)</b> |
|          | T3   | 1.10e+02(3.03e+00)- | 1.00e+02(1.01e-11)-        | 1.05e+02(2.14e+00)- | 1.11e+02(4.23e+00)- | 1.01e+02(2.80e+00)- | 1.00e+02(9.28e-03)-        | 1.00e+02(2.48e-05)-        | <b>1.00e+02(1.22e-14)</b> |
|          | T4   | 4.49e+01(8.20e-03)- | 4.52e+01(3.13e-02)-        | 4.50e+01(3.92e-03)- | 4.50e+01(5.41e-04)- | 4.62e+01(5.65e-02)- | 4.35e+01(1.45e+00)-        | <b>2.69e+01(5.17e+00)+</b> | 3.36e+01(7.31e+00)        |
| C4TOP3   | T1   | 4.32e+02(1.75e+01)- | 1.42e+02(1.09e+01)-        | 6.07e+01(6.66e+00)- | 3.84e+02(1.30e+01)- | 4.05e+02(6.26e+01)- | <b>2.04e-11(4.85e-11)+</b> | 1.50e-02(6.36e-02)-        | 5.94e-10(2.23e-09)        |
|          | T2   | 1.50e+02(1.16e-01)- | 1.50e+02(1.30e-14)-        | 1.50e+02(2.54e-02)- | 1.50e+02(1.06e-01)- | 1.50e+02(5.78e-01)- | 1.50e+02(1.12e-07)-        | 1.50e+02(4.96e-11)-        | <b>1.50e+02(1.84e-14)</b> |
|          | T3   | 1.12e+02(3.89e+00)- | 1.00e+02(1.52e-11)-        | 1.04e+02(3.72e+00)- | 1.12e+02(4.32e+00)- | 1.03e+02(2.29e+00)- | 1.00e+02(3.99e-02)-        | 1.00e+02(7.47e-06)-        | <b>1.00e+02(1.63e-14)</b> |
|          | T4   | 1.61e+01(1.08e+00)- | <b>1.00e+01(2.50e-04)+</b> | 1.09e+01(6.47e-01)- | 1.51e+01(5.36e-01)- | 1.24e+01(1.99e+00)- | 1.00e+01(2.39e-02)-        | 1.01e+01(1.41e-02)-        | 1.00e+01(6.24e-02)        |
| C4TOP4   | T1   | 4.28e+02(2.35e+01)- | 1.44e+02(1.15e+01)-        | 2.05e+01(2.42e+01)- | 3.72e+02(1.49e+01)- | 2.33e+02(1.97e+02)- | <b>0.00e+00(0.00e+00)=</b> | 1.02e+00(4.40e+00)-        | <b>0.00e+00(0.00e+00)</b> |
|          | T2   | 1.50e+02(1.05e-01)- | 1.50e+02(9.22e-15)-        | 1.50e+02(9.41e-03)- | 1.50e+02(1.15e-01)- | 1.50e+02(4.82e-01)- | 1.50e+02(3.82e-13)-        | 1.50e+02(3.82e-13)-        | <b>1.50e+02(0.00e+00)</b> |
|          | T3   | 3.03e+03(1.45e+03)- | 9.52e+01(1.35e+01)-        | 1.07e+02(1.29e+01)- | 9.50e+02(3.66e+02)- | 9.77e+01(1.14e+00)- | 9.17e+01(1.85e+00)-        | 9.29e+01(1.46e+01)-        | <b>8.48e+01(1.04e+00)</b> |
|          | T4   | 4.50e+01(1.14e-02)- | 4.52e+01(2.72e-02)-        | 4.50e+01(5.83e-03)- | 4.50e+01(1.71e-03)- | 4.61e+01(5.66e-02)- | 4.07e+01(6.83e+00)-        | <b>2.79e+01(7.24e+00)=</b> | 3.23e+01(9.29e+00)        |
| C4TOP5   | T1   | 4.36e+02(1.91e+01)- | 1.41e+02(1.00e+01)-        | 2.85e+01(2.60e+01)- | 3.71e+02(1.72e+01)- | 2.91e+02(1.50e+02)- | <b>0.00e+00(0.00e+00)=</b> | 1.08e+00(3.23e+00)-        | <b>0.00e+00(0.00e+00)</b> |
|          | T2   | 1.50e+02(9.99e-02)- | 1.50e+02(0.00e+00)-        | 1.50e+02(2.07e-02)- | 1.50e+02(1.47e-01)- | 1.50e+02(4.47e-01)- | 1.50e+02(1.80e-05)-        | 1.50e+02(2.82e-12)-        | <b>1.50e+02(0.00e+00)</b> |
|          | T3   | 2.72e+03(1.40e+03)- | 9.60e+01(1.54e+01)-        | 1.29e+02(6.15e+01)- | 8.85e+02(2.54e+02)- | 1.07e+02(2.72e+00)- | 9.55e+01(5.63e-01)-        | 9.76e+01(2.08e-01)-        | <b>8.55e+01(1.06e+00)</b> |
|          | T4   | 1.65e+01(9.13e-01)- | <b>1.00e+01(2.65e-04)+</b> | 1.07e+01(4.54e-01)- | 1.52e+01(4.98e-01)- | 1.88e+01(6.37e+00)- | 1.00e+01(1.80e-03)-        | 1.01e+01(1.47e-02)-        | 1.00e+01(6.94e-03)        |
| C4TOP6   | T1   | 4.22e+02(2.21e+01)- | 1.41e+02(1.30e+01)-        | 6.32e+01(9.57e+00)- | 3.93e+02(1.93e+01)- | 3.30e+02(1.35e+02)- | <b>1.53e-12(4.89e-12)+</b> | 8.97e-01(3.21e+00)-        | 1.77e-08(5.98e-08)        |
|          | T2   | 1.50e+02(8.06e-02)- | 1.50e+02(9.22e-15)-        | 1.50e+02(3.69e-02)- | 1.50e+02(1.29e-01)- | 1.50e+02(5.46e-01)- | 1.50e+02(3.22e-09)-        | 1.50e+02(4.31e-12)-        | <b>1.50e+02(1.60e-14)</b> |
|          | T3   | 4.49e+01(5.50e-03)- | 4.52e+01(1.92e-02)-        | 4.50e+01(3.96e-03)- | 4.49e+01(3.23e-03)- | 4.62e+01(4.70e-02)- | 4.40e+01(1.64e-01)-        | 3.30e+01(1.01e+01)=        | <b>2.83e+01(3.30e+00)</b> |
|          | T4   | 1.57e+01(6.95e-01)- | <b>1.00e+01(3.55e-04)+</b> | 1.06e+01(1.95e-01)- | 1.54e+01(5.15e-01)- | 1.06e+01(6.44e-01)- | 1.00e+01(2.33e-02)-        | 1.01e+01(1.70e-02)-        | 1.00e+01(5.96e-03)        |
| C4TOP7   | T1   | 4.34e+02(2.04e+01)- | 1.44e+02(9.05e+00)-        | 7.93e+01(8.48e+00)- | 4.18e+02(2.03e+01)- | 3.25e+02(1.29e+02)- | 3.48e-01(6.67e-01)-        | 3.03e+00(6.29e+00)=        | <b>6.66e-02(2.68e-01)</b> |
|          | T2   | 1.11e+02(4.05e+00)- | 1.00e+02(1.82e-11)-        | 1.06e+02(3.10e+00)- | 1.14e+02(5.07e+00)- | 1.42e+02(1.62e+02)- | 1.00e+02(3.26e-02)-        | 1.00e+02(2.52e-09)-        | <b>1.00e+02(1.60e-14)</b> |
|          | T3   | 3.08e+03(1.48e+03)- | 9.30e+01(2.33e+00)-        | 7.15e+02(3.50e+02)- | 1.70e+03(5.42e+02)- | 1.04e+02(1.37e+01)- | 9.48e+01(1.37e+01)-        | 9.53e+01(1.06e+01)-        | <b>8.88e+01(1.24e+01)</b> |
|          | T4   | 4.50e+01(1.77e-03)- | 4.52e+01(2.83e-02)-        | 4.50e+01(6.39e-03)- | 4.50e+01(3.13e-03)- | 4.62e+01(3.98e-02)- | 4.38e+01(1.06e-01)-        | <b>2.60e+01(4.66e+00)+</b> | 2.86e+01(4.63e+00)        |
| C4TOP8   | T1   | 4.23e+02(2.41e+01)- | 1.37e+02(9.89e+00)-        | 7.36e+01(7.76e+00)- | 4.20e+02(1.89e+01)- | 4.02e+02(5.91e+01)- | 2.10e+00(4.74e+00)-        | 1.84e+00(4.57e+00)=        | <b>3.63e-01(1.29e+00)</b> |
|          | T2   | 1.11e+02(3.54e+00)- | 1.00e+02(2.06e-11)-        | 1.06e+02(2.42e+00)- | 1.14e+02(5.47e+00)- | 1.15e+02(1.05e+01)- | 1.00e+02(1.57e-01)-        | 1.00e+02(1.15e-08)-        | <b>1.00e+02(2.37e-14)</b> |
|          | T3   | 3.14e+03(1.32e+03)- | 9.62e+01(1.31e+01)-        | 5.73e+02(1.28e+02)- | 1.64e+03(7.61e+02)- | 1.25e+02(3.41e+01)- | 1.06e+02(2.33e+01)-        | 9.63e+01(5.78e+00)-        | <b>9.20e+01(1.79e+01)</b> |
|          | T4   | 1.59e+01(8.04e-01)- | <b>1.00e+01(2.43e-04)+</b> | 1.09e+01(7.07e-01)- | 1.55e+01(6.63e-01)- | 3.22e+01(8.76e+00)- | 1.01e+01(4.18e-01)-        | 1.01e+01(1.56e-02)-        | 1.00e+01(1.46e-02)        |

(Continued)

Table 8. Continued

| Problems | Task | DE                  | SHADE                      | MFEA                | MFDE                | DEORA               | MTSRA                       | MTEA-HKTS                  | CMHSA-SMF                 |
|----------|------|---------------------|----------------------------|---------------------|---------------------|---------------------|-----------------------------|----------------------------|---------------------------|
| C4TOP9   | T1   | 4.38e+02(1.97e+01)- | 1.44e+02(1.03e+01)-        | 8.19e+01(1.27e+01)- | 4.24e+02(2.05e+01)- | 4.53e+02(1.88e+01)- | 3.29e+01(6.89e+00)-         | <b>3.52e+00(8.65e+00)+</b> | 2.14e+01(4.76e+00)        |
|          | T2   | 1.11e+02(4.18e+00)- | 1.00e+02(1.24e-11)-        | 1.08e+02(3.51e+00)- | 1.14e+02(5.07e+00)- | 1.02e+02(1.41e+00)- | 1.00e+02(4.25e+03)-         | 1.00e+02(5.87e-10)-        | <b>1.00e+02(1.34e-14)</b> |
|          | T3   | 4.50e+01(1.69e-04)- | 4.52e+01(2.58e-02)-        | 4.50e+01(4.83e-03)- | 4.49e+01(1.49e-02)- | 4.62e+01(4.69e-02)- | 4.49e+01(4.28e-02)-         | <b>3.50e+01(1.03e+01)=</b> | 4.02e+01(1.95e+00)        |
|          | T4   | 1.65e+01(8.45e-01)- | <b>1.00e+01(3.12e-04)=</b> | 1.06e+01(4.62e-01)- | 1.57e+01(7.55e-01)- | 1.10e+01(2.58e-01)- | 1.00e+01(2.24e+03)+         | 1.01e+01(1.82e-02)-        | 1.00e+01(5.91e-02)        |
| C4TOP10  | T1   | 4.37e+02(1.72e+01)- | 1.47e+02(9.68e+00)-        | 7.84e+01(1.31e+01)- | 4.27e+02(2.07e+01)- | 4.00e+02(6.63e+01)- | 2.48e-01(9.06e-01)-         | 1.07e+01(1.26e+01)-        | <b>1.90e-02(7.55e-02)</b> |
|          | T2   | 2.95e+03(1.83e+03)- | 9.90e+01(1.90e+01)-        | 7.14e+02(3.59e+02)- | 2.05e+03(6.20e+02)- | 1.04e+02(1.24e+01)- | 9.73e+01(1.20e+01)-         | 9.69e+01(3.08e-01)-        | <b>8.63e+01(1.09e+00)</b> |
|          | T3   | 4.49e+01(2.74e-03)- | 4.52e+01(2.30e-02)-        | 4.50e+01(5.34e-03)- | 4.50e+01(4.03e-03)- | 4.62e+01(4.77e-02)- | 4.40e+01(7.52e-02)-         | <b>3.10e+01(9.40e+00)+</b> | 3.27e+01(4.05e+00)        |
|          | T4   | 1.60e+01(1.22e+00)- | <b>1.00e+01(3.02e-04)+</b> | 1.02e+01(3.50e-01)- | 1.56e+01(6.38e-01)- | 1.64e+01(3.12e+00)- | 1.00e+01(4.96e-02)-         | 1.01e+01(1.43e-02)-        | 1.00e+01(1.31e-02)        |
| C4TOP11  | T1   | 1.25e+02(1.06e-01)- | 1.25e+02(6.18e-12)-        | 1.25e+02(1.63e-02)- | 1.25e+02(1.31e-01)- | 1.25e+02(4.49e-01)- | 1.25e+02(7.00e-04)-         | 1.25e+02(7.56e-11)-        | <b>1.25e+02(1.46e-14)</b> |
|          | T2   | 8.57e+01(3.67e+00)- | 7.50e+01(9.07e-12)-        | 7.90e+01(1.48e+00)- | 8.86e+01(3.81e+00)- | 7.71e+01(1.66e+00)- | 7.50e+01(3.64e-03)-         | 7.50e+01(9.47e-10)-        | <b>7.50e+01(1.42e-14)</b> |
|          | T3   | 2.94e+03(1.05e+03)- | 7.23e+01(1.64e+01)-        | 9.58e+01(4.39e+01)- | 9.41e+02(3.24e+02)- | 8.09e+01(2.03e+01)- | <b>5.75e+01(4.93e+00)+</b>  | 6.78e+01(1.38e+01)-        | 6.08e+01(1.16e+00)        |
|          | T4   | 1.99e+01(9.69e-03)- | 2.02e+01(2.77e-02)-        | 2.00e+01(6.02e-03)- | 1.99e+01(1.83e-02)- | 2.12e+01(4.94e-02)- | 1.69e+01(5.80e+00)-         | <b>9.16e-01(4.06e+00)+</b> | 1.00e+01(7.57e+00)        |
| C4TOP12  | T1   | 1.40e+02(7.44e-02)- | 1.40e+02(1.65e-03)=        | 1.40e+02(2.07e-02)- | 1.40e+02(1.45e-01)- | 1.40e+02(5.18e-01)- | 1.40e+02(4.04e-03)-         | 1.40e+02(2.29e-10)-        | <b>1.40e+02(0.00e+00)</b> |
|          | T2   | 1.00e+02(2.87e+00)- | 9.00e+01(1.56e-11)-        | 9.34e+01(1.55e+00)- | 1.02e+02(4.04e+00)- | 1.17e+02(5.47e+01)- | 9.00e+01(7.80e-02)-         | 9.00e+01(1.97e-08)-        | <b>9.00e+01(1.34e-14)</b> |
|          | T3   | 3.04e+03(1.25e+03)- | 8.56e+01(1.49e+01)-        | 1.08e+02(3.62e+01)- | 1.04e+03(3.34e+02)- | 1.03e+02(3.34e+01)- | 9.10e+01(1.75e+01)-         | 8.76e+01(1.37e-01)-        | <b>7.54e+01(9.19e-01)</b> |
|          | T4   | 6.03e+00(7.72e-01)- | 1.22e-03(2.33e-04)+        | 1.29e+00(1.13e+00)- | 5.46e+00(5.11e-01)- | 1.13e+01(4.85e+00)- | <b>-3.18e-18(0.00e+00)+</b> | 1.11e-01(1.05e-02)-        | 5.37e-03(1.57e-02)        |
| C4TOP13  | T1   | 1.40e+02(9.20e-02)- | 1.40e+02(0.00e+00)=        | 1.40e+02(5.79e-02)- | 1.40e+02(1.23e-01)- | 1.42e+02(2.94e-01)- | 1.40e+02(9.44e-03)-         | 1.40e+02(1.65e-03)-        | <b>1.40e+02(0.00e+00)</b> |
|          | T2   | 1.01e+02(3.39e+00)- | 9.00e+01(8.68e-12)-        | 9.69e+01(2.63e+00)- | 1.02e+02(4.46e+00)- | 9.32e+01(2.02e+00)- | 9.00e+01(4.29e-04)-         | 9.00e+01(3.08e-09)-        | <b>9.00e+01(1.60e-14)</b> |
|          | T3   | 3.49e+01(1.15e-02)- | 3.52e+01(2.65e-02)-        | 3.50e+01(4.98e-03)- | 3.50e+01(1.57e-03)- | 3.62e+01(3.67e-02)- | 3.42e+01(3.32e-01)-         | 2.50e+01(1.03e+01)=        | <b>2.39e+01(5.68e+00)</b> |
|          | T4   | 6.03e+00(9.18e-01)- | <b>1.37e-03(3.02e-04)+</b> | 1.13e+00(8.75e-01)- | 5.41e+00(4.77e-01)- | 2.00e-01(8.04e-02)- | <b>-3.18e-18(0.00e+00)+</b> | 1.16e-01(1.51e-02)-        | 4.19e-03(1.61e-02)        |
| C4TOP14  | T1   | 1.40e+02(9.55e-02)- | 1.40e+02(1.65e-03)=        | 1.40e+02(1.44e-02)- | 1.40e+02(1.23e-01)- | 1.40e+02(4.99e-01)- | 1.40e+02(8.48e-06)-         | 1.40e+02(1.65e-03)-        | <b>1.40e+02(1.13e-14)</b> |
|          | T2   | 2.92e+03(1.31e+03)- | 8.59e+01(1.21e+01)-        | 1.12e+02(2.72e+01)- | 1.06e+03(4.02e+02)- | 9.66e+01(1.83e+01)- | 8.37e+01(1.52e+00)-         | 8.69e+01(1.57e-01)-        | <b>7.53e+01(1.04e+00)</b> |
|          | T3   | 3.49e+01(1.65e-03)- | 3.52e+01(1.99e-02)-        | 3.50e+01(5.61e-03)- | 3.50e+01(1.09e-03)- | 3.62e+01(2.72e-02)- | 3.40e+01(5.11e-02)-         | <b>2.20e+01(9.74e+00)=</b> | 2.28e+01(6.81e+00)        |
|          | T4   | 6.42e+00(8.14e-01)- | <b>1.17e-03(1.64e-04)+</b> | 7.26e-01(3.43e-01)- | 5.58e+00(5.76e-01)- | 2.01e+00(1.12e+00)- | 1.44e-03(6.47e-03)+         | 1.16e-01(1.78e-02)-        | 7.64e-03(2.67e-02)        |
| C4TOP15  | T1   | 1.00e+02(3.35e+00)- | 9.00e+01(1.07e-11)-        | 9.82e+01(2.93e+00)- | 1.04e+02(4.74e+00)- | 9.76e+01(4.05e+00)- | 9.00e+01(3.80e-02)-         | 9.00e+01(4.40e-05)-        | <b>9.00e+01(1.26e-14)</b> |
|          | T2   | 3.64e+03(1.42e+03)- | 9.42e+01(2.44e+01)=        | 1.71e+03(1.17e+03)- | 3.42e+03(8.61e+02)- | 1.25e+02(5.43e+01)- | 1.32e+02(3.46e+01)-         | <b>8.69e+01(2.93e-01)=</b> | 9.73e+01(2.77e+01)        |
|          | T3   | 3.50e+01(2.16e-03)- | 3.52e+01(1.97e-02)-        | 3.50e+01(6.64e-03)- | 3.50e+01(2.77e-03)- | 3.62e+01(5.42e-02)- | 3.48e+01(3.32e-01)-         | <b>2.59e+01(1.02e+01)=</b> | 2.78e+01(3.54e+00)        |
|          | T4   | 6.36e+00(1.17e+00)- | 1.18e-03(2.27e-04)=        | 9.57e-01(6.52e-01)- | 6.01e+00(7.36e-01)- | 9.99e+00(4.63e+00)- | -3.18e-18(0.00e+00)+        | 1.12e-02(5.22e-02)-        | 1.27e-02(3.43e-02)        |
| +/-/=    |      | 0/60/0              | 8/46/6                     | 0/60/0              | 0/60/0              | 0/60/0              | 9/48/3                      | 5/46/9                     | Base                      |

Best result in each row is highlighted. “+/-/=” denotes the algorithm in each column is significantly better, worse, and similar to the base algorithm on the test set or case above it.

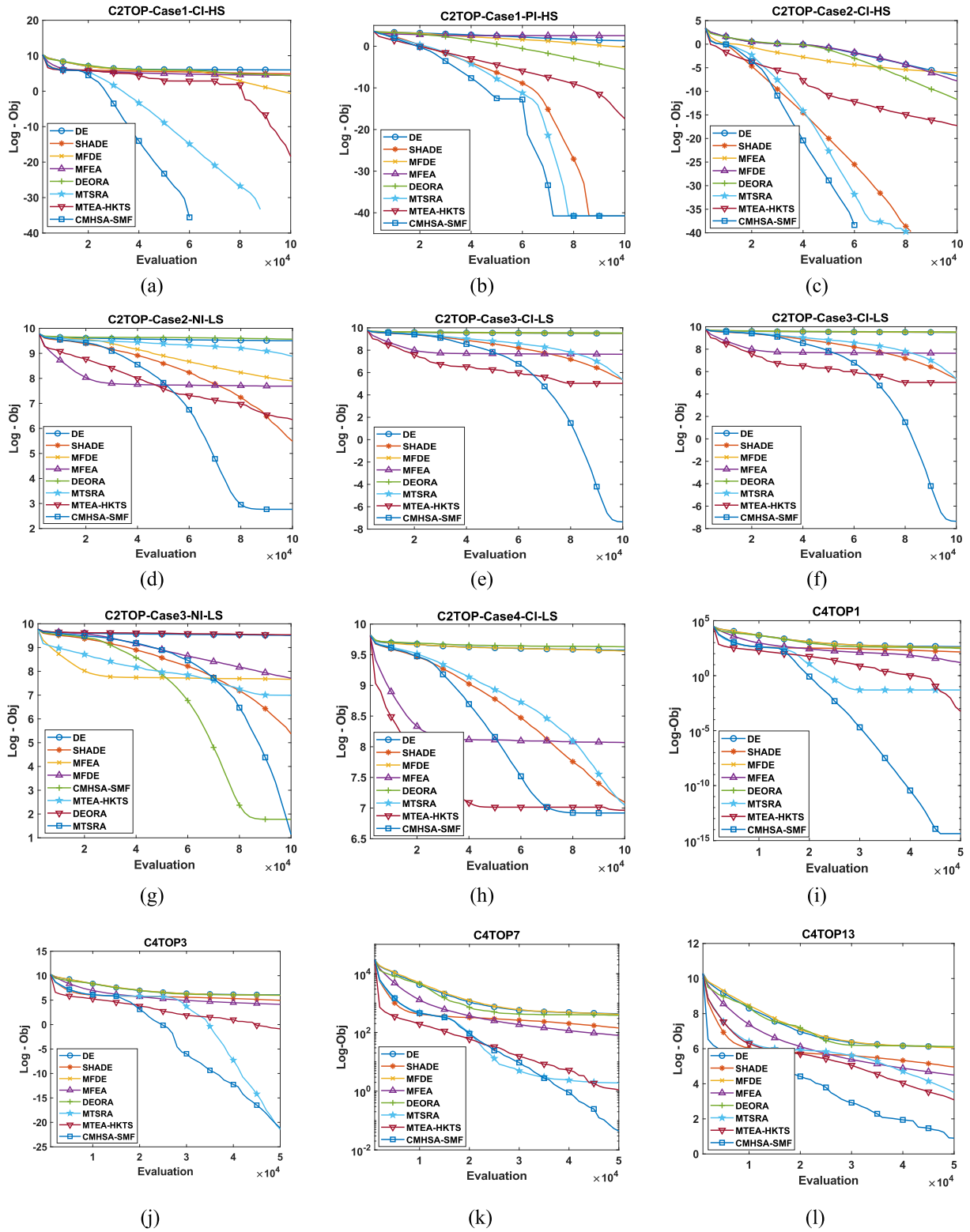


Fig. 5. Panels (a)–(l) are average objective value convergence (log style) for 30 independent runs of DE, SHADE, MFDE, MFEA, MTDE, MFEA, DEORA, MTSRA, and CMHSA-SMF on partial test functions for C2TOP and C4TOP.



The resounding success of CMHSA-SMF, evidenced by the overwhelming win rates such as the 8 wins versus 35 losses for DE on C2TOP, validates the necessity of its core design for CMTOP.

- (2) The algorithm outperforms general-purpose powerful optimizers through its specialized CMTOP mechanisms. CMHSA-SMF further demonstrates clear superiority over both the non-multitask SHADE and the hyper-multitask MTEA-HKTS. While these algorithms employ powerful variants of the SHADE operator, they lack the specific mechanisms for arbitration and controlled transfer. The superior performance of CMHSA-SMF highlights the decisive advantage of integrating the DMT-SHADE operator with a dedicated resource allocation strategy, a synergy that is highly effective for CMTOPs.
- (3) CMHSA-SMF establishes a new state-of-the-art against specialized CMTOP algorithms. DEORA and MTSRA are algorithms specifically designed for CMTOPs. The statistical results clearly show that CMHSA-SMF significantly outperforms them. DEORA demonstrates superiority in only 2 out of 72 C2TOP cases, while MTSRA, though an improvement over DEORA, is still consistently outperformed. This establishes that CMHSA-SMF's historical parameter-aware resource allocation provides a more dynamic and accurate arbitration strategy than DEORA, and its adaptive task affinity weighting offers a more principled method for auxiliary task selection than the more stochastic approach in MTSRA. This controlled selection is critical for mitigating negative transfer, a fact supported by CMHSA-SMF's sustained late-stage improvement seen in Figure 5(e)–(h) while others stagnate.
- (4) The adaptive task affinity weighting mechanism is pivotal for scaling complexity. The performance superiority of CMHSA-SMF on the C4TOP test suite is significantly greater than that observed on C2TOP. This heightened advantage is primarily attributed to the Adaptive Task Affinity Weighting mechanism. As the number of competing tasks increases from two to four, the risk of negative transfer and resource misallocation escalates. The mechanism's ability to dynamically identify the most suitable auxiliary task from a larger pool of competitors becomes critically important. The results in Table 8 and the enhanced convergence in Figure 5(i)–(l) demonstrates that this mechanism is exceptionally effective at navigating more complex competitive landscapes, ensuring efficient and constructive knowledge transfer.

In summary, the comprehensive evaluation across the C2TOP and C4TOP benchmark suites conclusively demonstrates that CMHSA-SMF establishes a new state-of-the-art for CMTOP. Its superior performance stems from a fundamentally sound architectural design that directly addresses the core challenges of the paradigm: intelligent resource arbitration through a historical parameter-aware mechanism and the mitigation of negative transfer via adaptive task affinity weighting. The algorithm's scalability and consistency, evidenced by its enhanced performance on the more complex four-task problems, confirm its robustness and practical efficacy in handling the inherent conflicts of CMTOPs.

#### 4.4 Comparison on More Complex Problems

Table 9 details the mean objective values and standard deviations of all algorithms after 30 independent runs on the two CCPLX test suites. Each column in the table corresponds to the performance of an algorithm across different test problems. For each test case in CCPLX, the symbols “+”, “-”, and “=” appended to the results indicate whether the algorithm's performance is significantly superior to, significantly inferior to, or statistically comparable to the baseline algorithm (CMHSA-SMF), respectively, based on the Wilcoxon rank-sum test ( $\alpha = 0.05$ ). The last row labeled “Total +/-=” summarizes the aggregated statistical outcomes across all test sets. Based on the experimental data in Table 9, the following conclusions can be drawn:

Table 9. Average Objective Value and Standard Deviation Obtained by DE, SHADE, MFEA, MFDE, DEORA, MTSRA, MTEA-HKTS, and CMHSA-SMF on CCPLX for 30 Independent Runs

| Problems | Task | DE                   | SHADE                      | MFEA                | MFDE                | DEORA               | MTSRA                      | MTEA-HKTS                  | CMHSA-SMF                 |
|----------|------|----------------------|----------------------------|---------------------|---------------------|---------------------|----------------------------|----------------------------|---------------------------|
| P1       | T1   | 1.61e+03(8.92e+00)-  | 1.60e+03(7.19e-01)=        | 1.64e+03(5.93e+00)- | 1.68e+03(4.48e+00)- | 1.61e+03(3.54e+00)- | 1.60e+03(6.75e-01)-        | 1.60e+03(1.25e+00)-        | <b>1.60e+03(4.07e-01)</b> |
|          | T2   | 6.14e+02(6.84e+00)-  | 6.00e+02(4.75e-01)=        | 6.44e+02(8.01e+00)- | 6.12e+02(4.22e+00)- | 6.01e+02(6.30e-01)- | 6.01e+02(6.54e-01)-        | <b>6.00e+02(5.38e-03)+</b> | 6.01e+02(4.16e-01)        |
| P2       | T1   | 11.70e+03(4.72e-03)- | 1.70e+03(2.69e-03)-        | 1.70e+03(1.67e-02)- | 1.70e+03(2.87e-03)- | 1.70e+03(1.57e-01)- | 1.70e+03(3.44e-03)=        | 1.70e+03(2.97e-03)-        | <b>1.70e+03(4.58e-13)</b> |
|          | T2   | 7.01e+02(4.57e-03)-  | 7.00e+02(1.65e-03)=        | 7.00e+02(1.29e-02)- | 7.00e+02(3.25e-03)- | 7.00e+02(2.28e-03)- | <b>7.00e+02(1.20e-13)=</b> | 7.00e+02(7.83e-08)+        | 7.00e+02(1.65e-03)        |
| P3       | T1   | 1.08e+07(3.19e+06)-  | 7.62e+03(2.15e+03)=        | 2.16e+06(1.40e+06)- | 1.28e+07(2.56e+06)- | 1.59e+07(4.21e+06)- | 7.55e+03(1.27e+03)=        | 6.44e+05(7.23e+05)-        | <b>6.80e+03(1.13e+03)</b> |
|          | T2   | 1.22e+07(2.82e+06)-  | 6.37e+03(1.33e+03)=        | 2.50e+06(1.55e+06)- | 1.24e+07(3.26e+06)- | 1.69e+07(5.09e+06)- | 7.13e+03(1.73e+03)-        | 2.84e+06(4.85e+06)-        | 5.63e+03(1.53e+03)        |
| P4       | T1   | 2.30e+03(7.06e-02)-  | <b>2.30e+03(3.66e-02)=</b> | 2.30e+03(1.12e-01)- | 2.30e+03(7.99e-02)- | 2.30e+03(7.16e-02)- | 2.30e+03(6.24e-02)-        | 2.30e+03(7.48e-02)=        | 2.30e+03(4.14e-02)        |
|          | T2   | 1.30e+03(5.04e-02)-  | 1.30e+03(3.90e-02)=        | 1.30e+03(5.93e-02)- | 1.30e+03(5.69e-02)- | 1.30e+03(5.62e-02)- | 1.30e+03(4.03e-02)=        | 1.30e+03(3.78e-02)-        | <b>1.30e+03(3.65e-02)</b> |
| P5       | T1   | 2.53e+03(1.23e+00)-  | 2.51e+03(1.83e+00)-        | 2.53e+03(8.88e+00)- | 2.53e+03(1.12e+00)- | 2.53e+03(1.57e+00)- | 2.52e+03(1.99e+00)-        | 2.51e+03(2.71e+00)=        | <b>2.51e+03(1.55e+00)</b> |
|          | T2   | 1.53e+03(2.19e+00)-  | 1.51e+03(1.61e+00)-        | 1.53e+03(7.98e+00)- | 1.53e+03(1.37e+00)- | 1.53e+03(1.80e+00)- | 1.51e+03(1.56e+00)-        | 1.53e+03(1.80e+00)-        | <b>1.51e+03(1.23e+00)</b> |
| P6       | T1   | 5.75e+06(1.31e+06)-  | 1.05e+04(2.92e+03)=        | 1.19e+06(9.09e+05)- | 5.48e+06(1.31e+06)- | 5.83e+06(1.80e+06)- | 9.58e+03(2.44e+03)=        | 6.57e+05(4.12e+05)-        | <b>8.92e+03(1.90e+03)</b> |
|          | T2   | 4.93e+06(1.21e+06)-  | 6.85e+03(8.26e+02)=        | 1.05e+06(7.17e+05)- | 5.27e+06(1.36e+06)- | 6.07e+06(1.85e+06)- | 6.55e+03(1.91e+03)=        | 5.97e+06(6.12e+06)-        | <b>6.32e+03(1.18e+03)</b> |
| P7       | T1   | 4.97e+03(1.05e+02)-  | 3.39e+03(1.02e+02)=        | 4.31e+03(3.34e+02)- | 4.95e+03(1.62e+02)- | 4.85e+03(1.67e+02)- | 3.44e+03(1.19e+02)=        | 3.77e+03(2.74e+02)-        | <b>3.37e+03(9.77e+01)</b> |
|          | T2   | 3.98e+03(1.60e+02)-  | 2.50e+03(1.34e+02)=        | 3.44e+03(2.62e+02)- | 3.98e+03(2.17e+02)- | 3.94e+03(1.35e+02)- | 2.48e+03(1.09e+02)=        | 3.67e+03(3.47e+02)-        | <b>2.47e+03(1.27e+02)</b> |
| P8       | T1   | 1.52e+03(2.55e-02)-  | 1.52e+03(7.77e-02)-        | 1.52e+03(9.45e-02)= | 1.52e+03(3.92e-02)- | 1.52e+03(2.75e-02)- | 1.52e+03(6.79e-02)-        | <b>1.52e+03(3.89e-02)+</b> | 1.52e+03(1.13e-01)        |
|          | T2   | 5.21e+02(4.27e-02)-  | 5.24e+02(8.32e-02)-        | 5.20e+02(8.89e-02)+ | 5.21e+02(3.31e-02)- | 5.21e+02(3.24e-02)- | 5.20e+02(1.03e-01)-        | 5.21e+02(3.16e-02)-        | <b>5.20e+02(1.00e-01)</b> |
| P9       | T1   | 1.57e+04(3.20e+02)-  | 1.00e+04(3.97e+02)-        | 9.24e+03(7.68e+02)- | 1.57e+04(3.85e+02)- | 1.55e+04(3.73e+02)- | 1.01e+04(3.60e+02)-        | 8.69e+03(7.75e+02)-        | <b>7.94e+03(5.65e+02)</b> |
|          | T2   | 1.62e+03(2.46e-01)-  | 1.62e+03(4.00e-01)-        | 1.62e+03(5.44e-01)- | 1.66e+03(2.83e-01)- | 1.62e+03(2.38e-01)- | 1.62e+03(3.89e-01)-        | 1.62e+03(2.58e-01)-        | <b>1.61e+03(1.25e+00)</b> |
| P10      | T1   | 3.46e+04(7.70e+03)-  | <b>3.21e+03(5.76e+01)=</b> | 4.11e+04(1.53e+04)- | 3.49e+04(8.84e+03)- | 2.88e+04(8.31e+03)- | 3.27e+03(7.07e+01)=        | 1.52e+04(7.00e+03)-        | 3.21e+03(5.76e+01)        |
|          | T2   | 5.87e+06(1.69e+06)-  | 7.32e+03(2.20e+03)+        | 1.64e+06(1.34e+06)- | 5.43e+06(1.53e+06)- | 6.41e+06(1.42e+06)- | <b>6.99e+03(1.27e+03)+</b> | 7.32e+06(6.51e+06)-        | 7.32e+03(2.20e+03)        |
| +/-/     |      | 0/20/0               | 1/7/12                     | 1/18/1              | 0/20/0              | 0/20/0              | 1/10/9                     | 3/15/4                     | Base                      |

(Continued)

Table 9. Continued

| Problems   | Task | DE                  | SHADE                      | MFEA                | MFDE                | DEORA                | MTSRA                      | MTEA-HKTS                  | CMHSA-SMF                 |
|------------|------|---------------------|----------------------------|---------------------|---------------------|----------------------|----------------------------|----------------------------|---------------------------|
| P1         | T1   | 6.15e+02(9.47e+00)- | 6.00e+02(5.92e-01)-        | 6.45e+02(5.63e+00)- | 6.12e+02(5.59e+00)- | 6.01e+02(5.44e-01)-  | 6.00e+02(7.72e-01)-        | 6.04e+02(1.38e+00)-        | <b>6.00e+02(3.21e-01)</b> |
|            | T2   | 1.61e+03(7.34e+00)- | 1.60e+03(5.50e-01)=        | 1.64e+03(6.47e+00)- | 1.61e+03(6.38e+00)- | 1.60e+03(2.34e+00)-  | 1.69e+03(6.94e-01)-        | <b>1.60e+03(3.20e-01)+</b> | <b>1.08e+03(5.70e-01)</b> |
| P2         | T1   | 7.02e+02(6.44e-03)- | 17.0e+02(1.86e-13)-        | 7.00e+02(2.23e-02)- | 7.00e+02(5.11e-03)- | 7.00e+02(4.02e-03)-  | 7.00e+02(1.65e-03)-        | 7.00e+02(4.59e-03)-        | <b>7.00e+02(3.69e-14)</b> |
|            | T2   | 1.70e+03(5.90e-03)- | 1.70e+03(1.65e-03)=        | 1.70e+03(1.73e-02)- | 1.70e+03(3.75e-03)- | 1.72e+03(1.34e-01)-  | <b>1.70e+03(2.02e-13)=</b> | 1.70e+03(5.87e-08)=        | <b>1.70e+03(1.38e-13)</b> |
| P3         | T1   | 1.23e+07(3.57e+06)- | 5.96e+03(1.09e+03)=        | 2.31e+06(1.13e+06)- | 1.31e+07(3.65e+06)- | 1.49e+07(4.68e+06)-  | <b>5.69e+03(1.13e+03)+</b> | 4.69e+05(4.62e+05)-        | 6.59e+03(1.50e+03)        |
|            | T2   | 1.16e+07(2.27e+06)- | <b>6.99e+03(1.27e+03)=</b> | 2.47e+06(1.40e+06)- | 1.33e+07(3.25e+06)- | 1.48e+07(3.88e+06)-  | 7.33e+03(1.74e+03)=        | 5.16e+06(8.79e+06)-        | 7.47e+03(1.42e+03)        |
| P4         | T1   | 1.30e+03(6.28e-02)- | <b>1.30e+03(2.96e-02)+</b> | 1.35e+03(1.20e-01)- | 1.30e+03(5.58e-02)- | 1.37e+03(6.61e-02)-  | 1.30e+03(5.08e-02)-        | 1.30e+03(5.99e-02)=        | 1.34e+03(4.97e-02)        |
|            | T2   | 2.30e+03(4.99e-02)- | 2.30e+03(3.68e-02)=        | 2.35e+03(5.66e-02)- | 2.35e+03(6.25e-02)- | 2.30e+03(4.74e-02)-  | 2.33e+03(3.16e-02)=        | 2.35e+03(3.04e-02)-        | <b>2.30e+03(3.26e-02)</b> |
| P5         | T1   | 1.53e+03(1.60e+00)- | 1.51e+03(1.10e+00)-        | 1.52e+03(7.04e+00)- | 1.53e+03(1.20e+00)- | 1.53e+03(1.55e+00)-  | 1.51e+03(1.36e+00)-        | 1.51e+03(2.56e+00)-        | <b>1.58e+03(1.24e+00)</b> |
|            | T2   | 2.56e+03(1.14e+00)- | 2.51e+03(8.76e-01)-        | 2.53e+03(6.85e+00)- | 2.53e+03(1.30e+00)- | 2.52e+03(1.59e+00)-  | 2.52e+03(2.10e+00)-        | 2.53e+03(1.64e+00)-        | <b>2.52e+03(1.31e+00)</b> |
| P6         | T1   | 4.79e+06(1.81e+06)- | <b>7.02e+03(2.16e+03)+</b> | 1.62e+06(9.98e+05)- | 4.79e+06(1.31e+06)- | 6.22e+06(1.90e+06)-  | 9.66e+03(2.80e+03)=        | 5.79e+05(3.69e+05)-        | 9.23e+03(2.48e+03)        |
|            | T2   | 5.37e+06(1.16e+06)- | <b>7.36e+03(1.55e+03)+</b> | 9.78e+05(5.52e+05)- | 5.31e+06(1.51e+06)- | 5.79e+06(1.84e+06)-  | 7.40e+03(1.05e+03)=        | 7.15e+06(6.06e+06)-        | 8.24e+03(1.45e+03)        |
| P7         | T1   | 3.90e+03(1.30e+02)- | 2.35e+03(9.67e+01)=        | 3.13e+03(2.48e+02)- | 3.94e+03(2.18e+02)- | 3.81e+03(1.92e+02)-  | 2.39e+03(9.44e+01)=        | 2.70e+03(2.15e+02)-        | <b>2.35e+03(1.07e+02)</b> |
|            | T2   | 4.86e+03(2.29e+02)- | <b>3.45e+03(1.29e+02)=</b> | 4.23e+03(4.00e+02)- | 4.84e+03(2.16e+02)- | 4.96e+03(2.06e+02)-  | 3.56e+03(1.49e+02)=        | 4.57e+03(3.67e+02)-        | 3.47e+03(1.33e+02)        |
| P8         | T1   | 5.21e+02(3.78e-02)- | 5.20e+02(1.03e-01)-        | 5.20e+02(1.02e-01)+ | 5.21e+02(4.23e-02)- | 5.21e+02(3.08e-02)-  | 5.20e+02(9.24e-02)-        | <b>5.20e+02(3.54e-02)+</b> | 5.20e+02(1.17e-01)        |
|            | T2   | 1.52e+03(3.50e-02)- | 1.52e+03(8.57e-02)-        | 1.52e+03(9.79e-02)+ | 1.52e+03(2.89e-02)- | 1.52e+03(4.92e-02)-  | 1.52e+03(8.03e-02)-        | 1.52e+03(3.56e-02)-        | 1.52e+03(2.21e-01)        |
| P9         | T1   | 1.46e+04(3.05e+02)- | 9.01e+03(4.12e+02)-        | 8.23e+03(9.73e+02)- | 1.46e+04(4.52e+02)- | 1.44e+04(4.05e+02)-  | 8.99e+03(4.79e+02)-        | 8.22e+03(9.66e+02)-        | <b>7.49e+03(5.08e+02)</b> |
|            | T2   | 2.62e+03(1.91e-01)- | 2.62e+03(3.97e-01)-        | 2.62e+03(8.78e-01)- | 2.62e+03(1.69e-01)- | 2.622e+03(2.53e-01)- | 2.62e+03(4.13e-01)-        | 2.62e+03(2.20e-01)-        | 2.61e+03(7.18e-01)        |
| P10        | T1   | 3.40e+04(7.68e+03)- | <b>2.21e+03(6.93e+01)+</b> | 3.18e+04(1.91e+04)- | 3.12e+04(7.96e+03)- | 2.87e+04(6.70e+03)-  | 2.90e+03(2.95e+03)-        | 1.48e+04(6.87e+03)-        | 2.29e+03(8.07e+01)        |
|            | T2   | 6.15e+06(1.61e+06)- | <b>7.60e+03(1.36e+03)+</b> | 2.15e+06(1.32e+06)- | 5.97e+06(1.73e+06)- | 6.76e+06(1.39e+06)-  | 7.78e+03(1.43e+03)+        | 1.02e+07(6.54e+06)-        | 9.89e+03(3.11e+03)        |
| +/-/=      |      | 0/20/0              | 5/8/7                      | 2/18/0              | 0/20/0              | 0/20/0               | 2/11/7                     | 2/16/2                     | Base                      |
| Total+/-/= |      | 0/40/0              | 6/15/19                    | 3/36/1              | 0/40/0              | 0/40/0               | 3/21/16                    | 5/31/6                     | Base                      |

Best result in each row is highlighted. “+/-/=” denotes the algorithm in each column is significantly better, worse, and similar to the base algorithm on the test set or case above it.

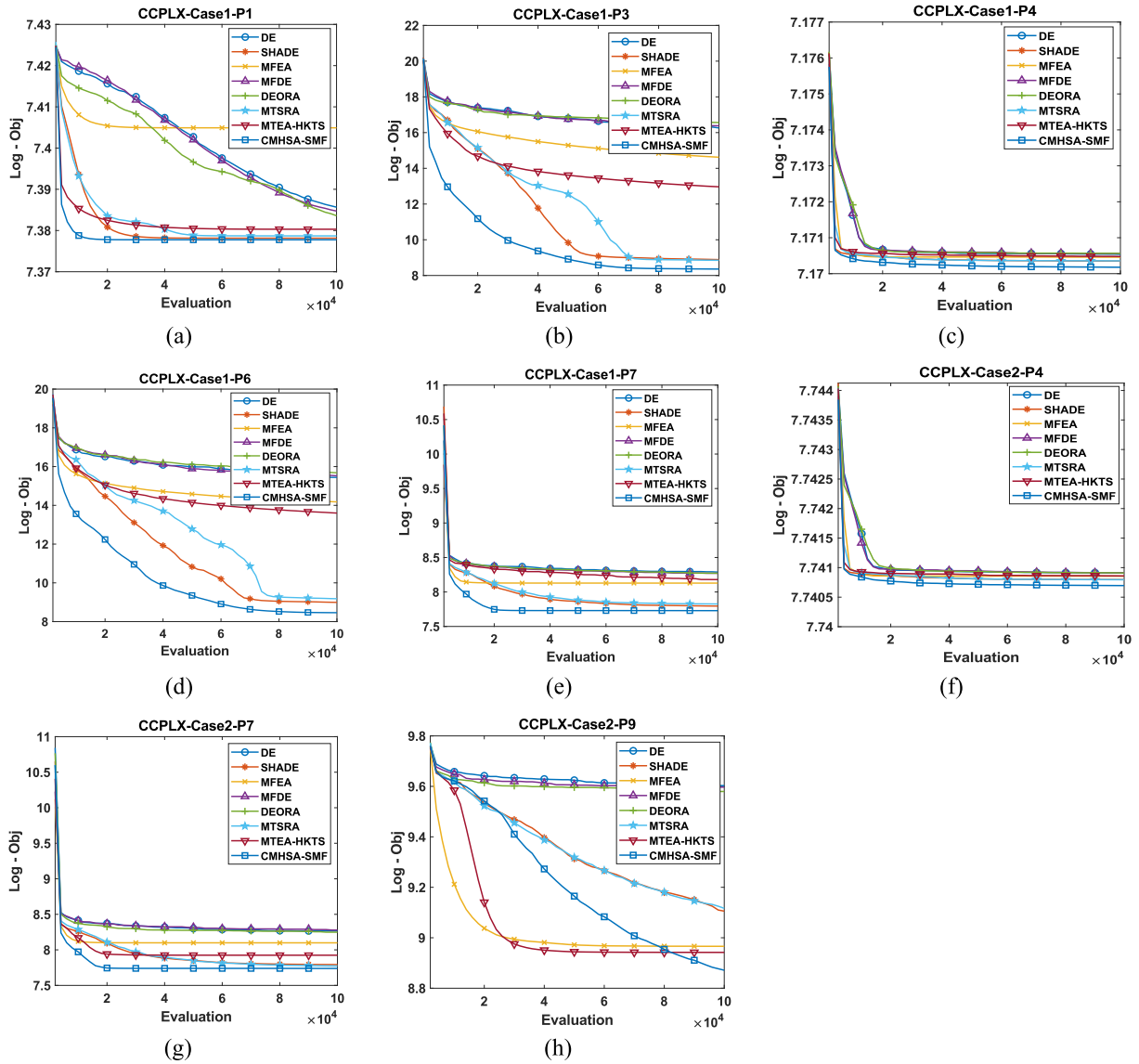


Fig. 6. Panels (a)–(h) are average objective value convergence (log style) for 30 independent runs of DE, SHADE, MFEA, MTDE, MFEA, DEORA, MTSRA, and CMHSA-SMF on partial test functions for CCPLX.

- (1) The core CMTOP mechanisms generalize to more complex problems. CMHSA-SMF exhibits statistically superior performance over DE, MFEA, and MFDE across all test problems in the CCPLX suite. This reaffirms that the fundamental design of the proposed algorithm, specifically its DMT-SHADE operator and historical parameter-aware adaptive resource allocation mechanism that is robust and effective even when problems involve more complex variable interactions and constraints, solidifying its value for competitive optimization.
- (2) The algorithm maintains robust competitiveness against advanced counterparts. The proposed CMHSA-SMF maintains highly competitive performance compared to SHADE, MTSRA, and MTEA-HKTS. This robustness stems from its adaptive resource allocation strategy and accurate search preservation mechanism. These components enable CMHSA-SMF to demonstrate consistent efficacy when confronted with the complex competitive dynamics inherent in problems involving multivariable interactions and high-dimensional search

spaces. The rapid early convergence dominance shown in Figure 6(a), (b), and (d) is a direct result of this optimized allocation mechanism.

- (3) The design exhibits learning and recovery capabilities in adversarial conditions. The algorithm demonstrates a key strength in its ability to learn from and adapt to initially unfavorable conditions. As seen in Figure 6(h), CMHSA-SMF can experience a transient performance deficit against other algorithms in early-to-mid stages. However, through the sustained integration of its historical success-guided adaptation and dynamic resource reallocation, it enables progressive performance recovery, ultimately yielding competitive convergence efficacy. This underscores the long-term robustness and adaptability of the proposed approach.
- (4) Specific limitations highlight directions for future work. The proposed algorithm exhibits limitations on a small number of specific test functions. These cases may indicate unresolved challenges in addressing certain high-dimensional complex problems with extreme deceptive competitive dynamics. Future investigations may therefore explore potential improvements for these specific scenarios, such as enhancing the sensitivity of the arbitration mechanisms. Nonetheless, CMHSA-SMF maintains robust and competitive performance across the entire CCPLX test suite, demonstrating its overall capability in handling diverse and complex CMTOP scenarios.
- (5) Deep Dive into Specific Underperformance and Mitigation Strategies. The article briefly mentions underperformance on specific CCPLX tasks (e.g., Case1-P2-T2, Case1-P8-T1). A deeper analysis suggests that these cases likely represent pathological instances of extreme competition or high-dimensional deception. The primary cause may be that the current MMD-based similarity measure, while effective for general distribution comparison, fails to fully capture the intricate and potentially non-linear adversarial relationships between tasks in such complex scenarios. This can lead the SMF mechanism to permit knowledge transfer that is ultimately detrimental. Additionally, in very high-dimensional problems within CCPLX, the algorithm's dependence on historical archives might initially struggle to provide diverse and high-quality building blocks for effective mutation. To address these limitations, potential fixes could be investigated in future work. These include: (1) Integrating alternative similarity measures that are more robust to high-dimensional spaces or that can directly quantify the impact of transfer on task performance; (2) Implementing an additional layer of filtering or validation for cross-task transfers in early generations to reduce the risk of catastrophic negative transfer; (3) Enhancing the archive management strategy to better preserve diversity and promote quality in exponentially large search spaces.

In conclusion, the experimental results on the challenging CCPLX test suite validate the generalizability and robustness of the proposed CMHSA-SMF framework. The algorithm not only maintains its competitive edge against a range of advanced competitors but also demonstrates an impressive capacity for recovery and adaptation in complex, adversarial optimization scenarios. While specific edge cases reveal potential avenues for further refinement, the overarching evidence confirms that CMHSA-SMF provides a powerful and reliable solution for navigating the intricate and competitive landscapes of real-world MTOPs.

#### 4.5 Component Analysis

To evaluate the contribution of each component in the proposed CMHSA-SMF framework, an ablation study was conducted. The CMHSA-SMF model comprises three key components: a cooperative co-evolutionary dynamic coupling strategy for multi-task scenarios, a dynamic threshold-based adaptive task affinity weighting method, and a historical parameter-aware resource adaptive allocation mechanism. Within the cooperative co-evolutionary dynamic coupling strategy, two



primary elements are proposed: an adaptive RMP control mechanism and a dynamic cooperative evolutionary operator (DMT-SHADE). Accordingly, this section focuses on analyzing these two components, while the archive update strategy is excluded from the discussion. The adaptive task affinity weighting based on dynamic thresholds serves as a foundation for subsequent resource allocation, and its effect is primarily reflected in the historical parameter-aware resource allocation mechanism. Therefore, no separate experimental validation for this component is provided in this subsection. To systematically evaluate the contributions of the individual components, the following four variants of CMHSA-SMF were constructed:

- (1) RA-no: This variant removes the resource allocation strategy from CMHSA-SMF.
- (2) RA-online: Replacing the historical parameter-aware resource allocation mechanism of CMHSA-SMF with the online resource allocation strategy in DEORA constructs this variant.
- (3) RMP-fix: This variant employs a fixed RMP (0.3, 0.5) instead of the adaptive RMP control strategy.
- (4) SHADE: This variant was constructed by replacing the dynamic cooperative evolutionary operator DMT-SHADE in CMHSA-SMF with SHADE.

To rigorously evaluate the contributions of individual components within the proposed CMHSA-SMF framework, a comprehensive ablation study was conducted. The statistical test results comparing CMHSA-SMF with its four variants on the C2TOP and C4TOP benchmark suites are summarized in Table 10. Overall, CMHSA-SMF demonstrates superior performance over all ablated versions across the majority of test cases, confirming the architectural coherence and synergistic integration of its core mechanisms. Specifically, the exclusion of the resource allocation strategy (RA-no) or its substitution with an online resource allocation approach (RA-online) results in significant performance degradation, highlighting the essential role of the success history-based adaptive resource allocation module. Similarly, employing a fixed random mating probability (RMP-fix) markedly diminishes optimization efficacy, underscoring the importance of adaptive RMP control in facilitating efficient cross-task knowledge transfer. Furthermore, while the replacement of the dynamic cooperative evolutionary operator DMT-SHADE with the standard SHADE operator retains partial competitiveness, its consistent underperformance relative to the complete CMHSA-SMF framework substantiates the critical role of DMT-SHADE in enhancing cross-task optimization efficiency. In conclusion, each constituent component of CMHSA-SMF is empirically validated as indispensable, and their coordinated interaction underpins the algorithm's robustness and superior performance in MTO environments.

#### 4.6 Parameter Sensitivity Analysis

In the CMHSA-SMF algorithm, the parameters  $p$  and  $H$  of the DMT-SHADE operator are inherent parameters of the SHADE operator and are therefore not included in the scope of this section's analysis. Similarly,  $\lambda$ ,  $pmin$ ,  $P0$ , and  $RH$  represent the actual action point parameters of the resource allocation strategy, the minimum probability threshold for task selection, the initial value of the RMP, and the memory capacity of the history-based resource allocation strategy, respectively. As their operational mechanisms are adopted from the MTSRA and DEORA algorithms without modification, they are not analyzed in detail here.

This section focuses on analyzing the three core parameters of CMHSA-SMF:  $\alpha$ ,  $\beta$  and  $\gamma$ . To verify the sensitivity of CMHSA-SMF to different parameter settings, we conducted an experimental analysis. We evaluated their sensitivity based on convergence curves (Figures 7 and 8) by testing different parameter configurations on the C2TOP problem. The better the performance, the lower the position of the convergence curve and the less it shifts to the right. The specific conclusions are as follows:

Table 10. Comparison Result between CMHSA-SMF with RA-No, RA-Online, RMP-Fix, SHADE, and CMHSA-SMF Based on Wilcoxon Rank-Sum Test on C2TOP and C4TOP

| VS.<br>CMHS<br>A-SMF | RA-No |     |    | RA-Online |    |    | RMP-Fix |    |    | SHADE |     |    |
|----------------------|-------|-----|----|-----------|----|----|---------|----|----|-------|-----|----|
|                      |       |     |    |           |    |    | 0.3     |    |    | 0.5   |     |    |
|                      | +     | -   | =  | +         | -  | =  | +       | -  | =  | +     | -   | =  |
| C2TOP                |       |     |    |           |    |    |         |    |    |       |     |    |
| -Case1               | 1     | 16  | 1  | 3         | 10 | 5  | 0       | 12 | 6  | 0     | 13  | 5  |
| C2TOP                |       |     |    |           |    |    |         |    |    |       |     |    |
| -Case2               | 0     | 17  | 1  | 2         | 9  | 7  | 2       | 13 | 3  | 0     | 14  | 4  |
| C2TOP                |       |     |    |           |    |    |         |    |    |       |     |    |
| -Case3               | 1     | 16  | 1  | 3         | 10 | 5  | 2       | 11 | 5  | 0     | 12  | 6  |
| C2TOP                |       |     |    |           |    |    |         |    |    |       |     |    |
| -Case4               | 0     | 15  | 3  | 4         | 8  | 6  | 1       | 12 | 5  | 1     | 15  | 2  |
| C4TOP                |       |     |    |           |    |    |         |    |    |       |     |    |
| C4TOP                | 4     | 51  | 5  | 10        | 38 | 12 | 10      | 45 | 5  | 5     | 51  | 4  |
| Total                | 6     | 115 | 11 | 22        | 75 | 35 | 15      | 93 | 24 | 6     | 105 | 21 |

“+/-/=” denotes the algorithm in each column is significantly better, worse, and similar to the original CMHSA-SMF.

- (1) The dynamic balance coefficient of the dual memory bank is a critical component of the resource allocation strategy. It is evident that the values of  $\alpha$  and  $\beta$  have a negligible impact on performance. However, in certain instances, specifically when  $\alpha = 0.3$  and  $\beta = 0.7$ , a decline in performance is observed. When  $\alpha < 0.5$ , the strategy demonstrates a preference for the FM bank, and when  $\beta < 0.5$ , the strategy demonstrates a preference for the SM bank. It is noteworthy that the most robust and optimal performance is consistently achieved when both dynamic balance coefficients,  $\alpha$  and  $\beta$ , are both set to 0.5.
- (2)  $\gamma$ : The penalty decay factor is a critical component in the resource allocation strategy. The penalty decay factor,  $\gamma$ , has been shown to have a greater impact on the selection of auxiliary tasks than the factors  $\alpha$  and  $\beta$ . This suggests that it is more sensitive to the performance of the algorithm. A fixed value for the penalty decay factor  $\gamma$  results in less stable performance compared to the proposed dynamic adaptation strategy. When  $\gamma$  is dynamically set to  $e^{-(g-g_i)/\tau}$ , which facilitates the selection of auxiliary tasks from different evolutionary periods, the dynamic adjustment of the value of  $\gamma$  results in relatively stable and superior performance.

## 5 Real-World Applications

In real-world scenarios, CMTD demonstrates substantial practical applicability, with prominent use cases including hyperparameter optimization for machine learning models and logistics distribution route planning, showcasing its adaptability to complex engineering challenges [47]. This section presents the proposed CMHSA-SMF framework to address practical optimization challenges, followed by a rigorous exposition of problem formulations and empirical result analysis.

### 5.1 Sensor Coverage Optimization Problem

To validate the practical efficacy of the proposed MTO algorithm within a benchmark scenario, this study selects the sensor coverage problem as a representative test case to evaluate the performance advantages of CMHSA-SMF over state-of-the-art alternatives. As an NP-hard combinatorial optimization problem, its core challenge lies in constructing sensor node deployment schemes

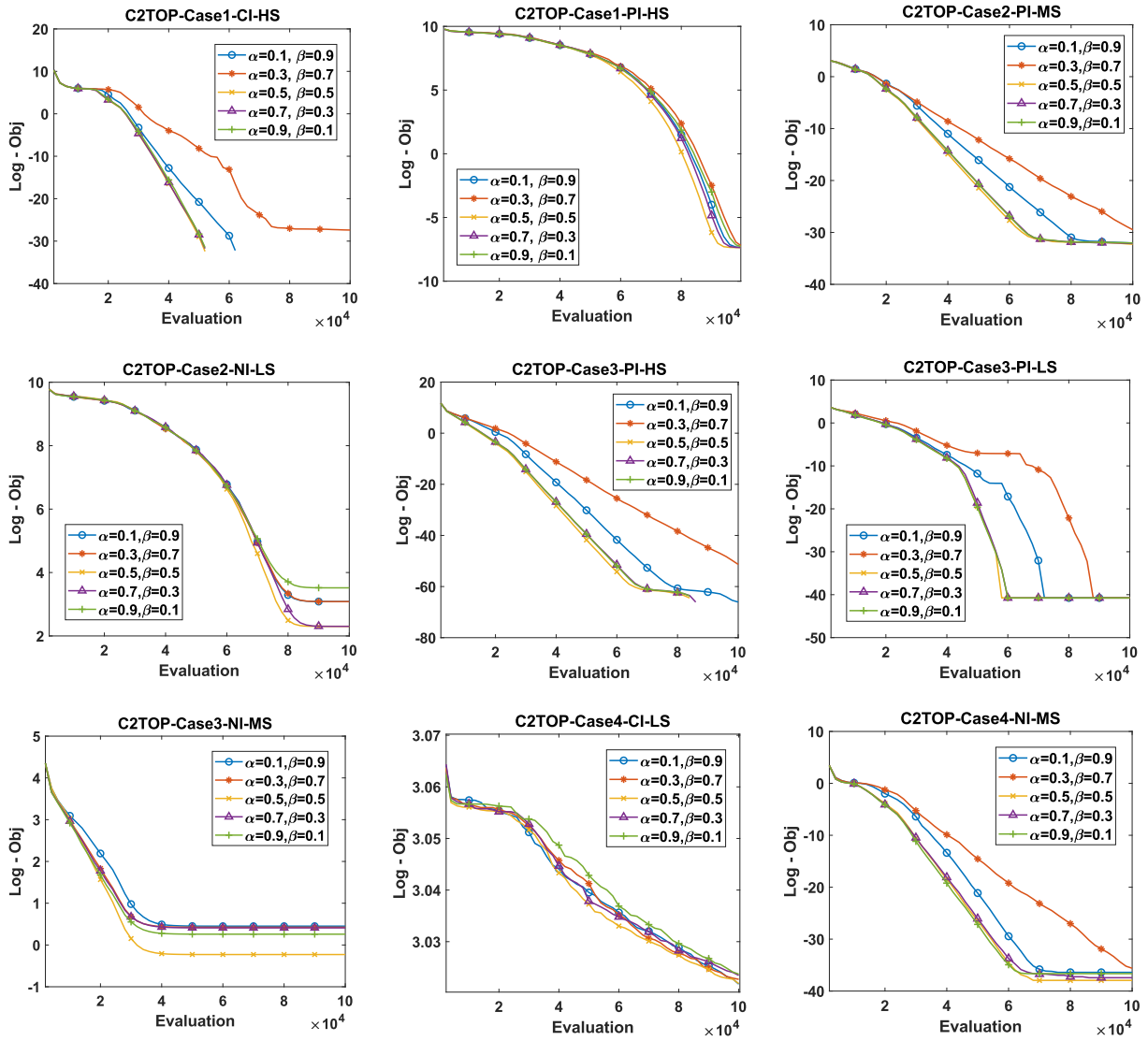


Fig. 7. Sensitivity analysis of dynamic balance coefficients  $\alpha$  and  $\beta$  in the dual memory resource allocation strategy in the C2TOP benchmark test.

that globally maximize user coverage in monitoring areas while minimizing hardware costs [49]. Existing studies have developed diverse models addressing directional sensing, probabilistic detection, and obstacle-prone regions [50], focusing on balancing coverage redundancy and blind zone mitigation caused by stochastic deployments. Although the sensor coverage problem investigated herein represents a simplified variant, it retains substantial complexity [49]. The problem requires deploying multiple sensors in a designated area to serve distributed users, with the dual objectives of maximizing user coverage rate and minimizing sensor deployment costs. The corresponding objective functions and boundary constraints are formulated as follows:

$$\text{Minimize : } f(x) = 1,000 \cdot \left( 1 - \frac{A \cup_{i=1}^S \pi r_i^2(x_i, y_i)}{A} \right) + \sum_{i=1}^S (1 + 10 \cdot r_i^2),$$

$$\mathbf{x} = (x_1, y_1, r_1, \dots, x_S, y_S, r_S),$$

$$\text{Subject to : } 0 < I_{ph} < 1, 0 < I_o < 10^{-6}, 0 < R_s < 0.5,$$

$$-1 < x_i < 1, -1 < y_i < 1, 0.1 < r_i < 0.25,$$

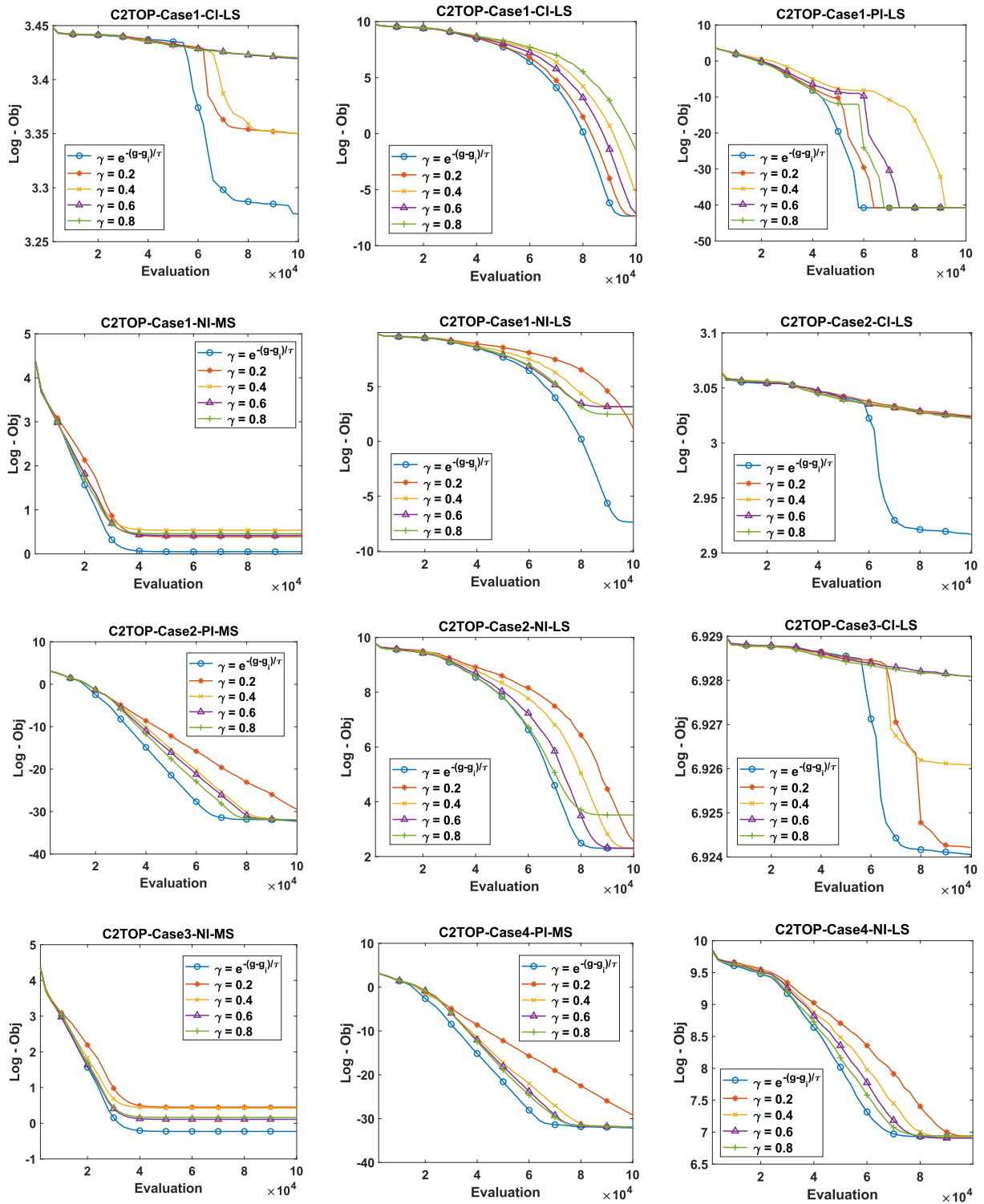


Fig. 8. Sensitivity analysis of penalty decay factor  $\gamma$  in resource allocation strategies in the C2TOP benchmark test.

where  $A$  denote the set of users to be covered, and  $x$  represent the sensor parameter vector, where each component  $(x_i, y_i)$  corresponds to the planar coordinates of the  $i$ th sensor node. The coverage radius of each sensor is denoted as  $r_i$ , and  $S$  represents the total number of deployed sensors. A schematic illustration of the sensor coverage problem is provided in Figure 9, where red dots

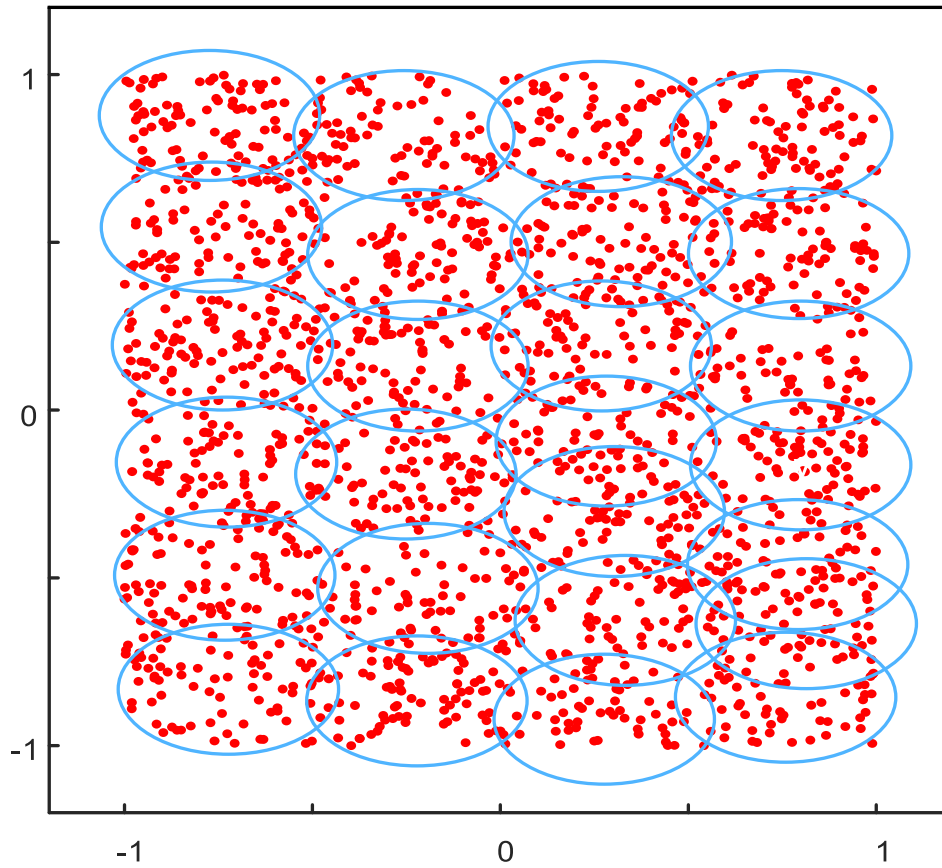


Fig. 9. An example of a sensor coverage problem. Red dots indicate the points to be served, blue circles indicate sensor signal coverage.

indicate user locations requiring service, and blue circular regions depict sensor coverage areas. Due to the inherent uncertainty in determining the optimal  $S$ , multiple scenarios with varying  $S$  values must be considered. This study employs the CMTOP paradigm proposed in [19] to address this variable-length parameter problem by constructing parallel optimization tasks for distinct  $S$  configurations. All experimental settings were consistent with those in Section 4.2, and each algorithm was independently executed 30 times.

Figure 10 illustrates the mean objective function values of various algorithms across distinct problem instances. The convergence profiles of these values reveal that CMHSA-SMF achieves optimal performance in both convergence speed and precision, particularly in high-dimensional task combinations. Compared to state-of-the-art CMTOP algorithms (DEORA and MTSRA), CMHSA-SMF exhibits competitive coverage rates, thereby validating the efficacy of its collaborative MTO mechanism. These results corroborate that CMHSA-SMF retains its algorithmic superiority even in complex coverage scenarios characterized by dynamic variations in sensor quantities.

## 5.2 Parameter Extraction of Photovoltaic (PV) Models

The extraction of parameters for PV models plays a pivotal role in the control, simulation, and optimization of PV system applications [45]. Existing studies have proposed various PV models, with the most widely adopted being the single-diode model, double-diode model, and the PV-module-based single-diode model, whose equivalent circuits are illustrated in Figure 11. Analytical derivations of these models yield distinct objective function formulations, which are inherently tied to the current-voltage characteristics and output power profiles of PV modules.



During parameter extraction, the judicious selection of models and precise identification of parameters are critical for enhancing PV system performance, enabling efficient system-level optimization, and ensuring accurate simulation predictions. The objective functions for each model are formulated as constrained optimization problems, where the constraints reflect physically meaningful parameter ranges to ensure solutions are both feasible and representative of real-world PV behavior. For instance:

- $I_{ph}$  (photocurrent) is constrained to typical operating ranges under standard illumination conditions.
- $I_o$  (reverse saturation current) is bounded to reflect its extremely small magnitude in practical diodes.
- $R_s$  (series resistance) and  $R_{sh}$  (shunt resistance) are limited to values consistent with manufacturing standards and empirical data.
- $a$  (ideality factor) is constrained based on semiconductor physics, typically between 1 and 2 for silicon-based cells.

These constraints not only ensure physical plausibility but also improve the efficiency and stability of the optimization process. The objective functions for each model are expressed as follows:

— Single-diode model:

$$\begin{aligned} \text{Minimize : } f(x) &= I_{ph} - I_o \left[ \exp \left( \frac{V + IR_s}{aV_t} \right) - 1 \right] - \frac{V + IR_s}{R_{sh}} - I, \\ x &= (I_{ph}, I_o, R_s, R_{sh}, a), \\ \text{Subject to : } &0 < I_{ph} < 1, \quad 0 < I_o < 10^{-6}, \quad 0 < R_s < 0.5, \\ &0 < R_{sh} < 100, \quad 1 < a < 2. \end{aligned}$$

— Double-diode model:

$$\begin{aligned} \text{Minimize : } f(x) &= I_{ph} - I_{o1} \left[ \exp \left( \frac{V + IR_s}{\alpha_1 V_t} \right) - 1 \right] - I_{o2} \left[ \exp \left( \frac{V + IR_s}{\alpha_2 V_t} \right) - 1 \right] - \frac{V + IR_s}{R_{sh}} - I, \\ x &= (I_{ph}, I_{o1}, R_s, R_{sh}, \alpha_1, I_{o2}, \alpha_2), \\ \text{Subject to : } &0 < I_{ph} < 1, \quad 0 < I_{o1} < 10^{-6}, \quad 0 < R_s < 0.5, \\ &0 < R_{sh} < 100, \quad 1 < \alpha_1 < 2, \quad 0 < I_{o2} < 10^{-6}, \quad 1 < \alpha_2 < 2. \end{aligned}$$

— PV-module-based single-diode model:

$$\begin{aligned} \text{Minimize : } f(x) &= I_{ph} - I_o \left[ \exp \left( \frac{V + IR_s N_s}{aN_s V_t} \right) - 1 \right] - \frac{V + IR_s N_s}{R_{sh} N_s} - I, \\ x &= (I_{ph}, I_o, R_s, R_{sh}, a), \\ \text{Subject to : } &0 < I_{ph} < 2, \quad 0 < I_o < 10^{-5}, \quad 0 < R_s < 2, \\ &0 < R_{sh} < 2,000, \quad 1 < a < 50. \end{aligned}$$

The parameter extraction for PV models can be formulated as an MTOP [46]. Given that the objective function values derived from each model exhibit comparable scales, this problem is further structured as a CMTOP. Under the CMTOP framework, the solution corresponds to the optimal model output that simultaneously satisfies all predefined constraints. The maximum number of function evaluations was set to 60,000 for this problem. All algorithmic parameters were maintained

Table 11. The Best, Mean, Median, Worst, and Standard Deviation of the Objective Value Obtained by DE, SHADE, MFEA, MFDE, DEORA, MTSRA, MTEA-HKTS, and CMHSA-SMF on Competitive 3-Task Parameter Extraction of PV Models for 30 Independent Runs

| Problem | Classify           | Task | DE                | SHADE             | MFEA       | MFDE       | DEORA             | MTSRA      | MTEA-HKTS  | CMHSA-SMF         |
|---------|--------------------|------|-------------------|-------------------|------------|------------|-------------------|------------|------------|-------------------|
| PEPVM   | Mean               | T1   | 9.8602e-04        | 9.8602e-04        | 1.7538e-03 | 1.0473e-03 | 9.8602e-04        | 1.9388e-03 | 1.7273e-03 | <b>9.8602e-04</b> |
|         |                    | T2   | 2.5705e-03        | 1.0638e-03        | 1.5904e-03 | 1.0292e-03 | <b>9.8250e-04</b> | 1.9935e-03 | 1.3981e-03 | 9.9664e-04        |
|         |                    | T3   | 2.5624e-03        | 2.5864e-03        | 2.9797e-03 | 2.4275e-03 | 2.4251e-03        | 1.2322e-02 | 2.7087e-03 | <b>2.4251e-03</b> |
|         | Median             | T1   | 9.8602e-04        | 9.8602e-04        | 1.6518e-03 | 1.0168e-03 | 9.8602e-04        | 2.4480e-03 | 1.8431e-03 | <b>9.8602e-04</b> |
|         |                    | T2   | 9.8418e-04        | 1.0299e-03        | 1.5142e-03 | 1.0145e-03 | 9.8248e-04        | 1.9709e-03 | 1.3770e-03 | <b>9.8248e-04</b> |
|         |                    | T3   | <b>2.4251e-03</b> | 2.6293e-03        | 2.6106e-03 | 2.4251e-03 | 2.4251e-03        | 3.3685e-03 | 2.6016e-03 | 2.6081e-03        |
|         | Best               | T1   | 9.8602e-04        | 9.8602e-04        | 1.0136e-03 | 9.8602e-04 | 9.8602e-04        | 9.8602e-04 | 1.0525e-03 | <b>9.8602e-04</b> |
|         |                    | T2   | 9.8283e-04        | 9.8248e-04        | 1.0134e-03 | 9.8557e-04 | 9.8248e-04        | 9.8535e-04 | 9.9101e-04 | <b>9.8248e-04</b> |
|         |                    | T3   | 2.4251e-03        | 2.4251e-03        | 2.4671e-03 | 2.4251e-03 | 2.4251e-03        | 2.4251e-03 | 2.4251e-03 | <b>2.4251e-03</b> |
|         | Worst              | T1   | 9.8602e-04        | <b>9.8602e-04</b> | 2.4427e-03 | 1.2623e-03 | 9.8602e-04        | 2.4480e-03 | 2.3942e-03 | 9.8602e-04        |
|         |                    | T2   | 9.8586e-04        | 1.2761e-03        | 2.4290e-03 | 1.1362e-03 | <b>9.8263e-04</b> | 3.1264e-03 | 2.1176e-03 | 9.8602e-04        |
|         |                    | T3   | 2.4822e-03        | 2.7862e-03        | 6.3660e-03 | 2.4400e-03 | 2.4251e-03        | 6.0711e-02 | 4.7702e-03 | <b>2.4251e-03</b> |
|         | Standard Deviation | T1   | 6.19e-10          | 1.71e-17          | 4.59e-04   | 7.30e-05   | 5.61e-10          | 6.61e-04   | 4.33e-04   | <b>1.71e-17</b>   |
|         |                    | T2   | 1.23e-04          | 8.47e-05          | 4.00e-04   | 4.55e-05   | <b>3.36e-08</b>   | 8.56e-04   | 2.93e-04   | 3.73e-05          |
|         |                    | T3   | 8.13e-05          | 1.02e-04          | 8.84e-04   | 4.36e-06   | 8.70e-13          | 1.80e-02   | 5.09e-04   | <b>1.45e-17</b>   |

Best result in each row is highlighted.

consistent with those defined in Section 4.2, and each algorithm underwent 30 independent runs to ensure statistical reliability.

The optimal values, mean, median, worst values, and standard deviation of the objective function for DE, SHADE, MFEA, MTDE, DEORA, MTSRA, MTEA-HKTS, and CMHSA-SMF are summarized in Table 11. The table reveals that CMHSA-SMF outperforms all competing algorithms in the majority of metrics, with exceptions observed in the mean value for Task 1, median value for Task 3, worst values for Tasks 1 and 2, and standard deviation for Task 2, where it marginally underperforms specific baselines. A comprehensive analysis of the table demonstrates that CMHSA-SMF retains a statistically significant advantage in PV model parameter extraction. Furthermore, Figure 12 presents the convergence profiles of the mean objective values across different problems, where CMHSA-SMF consistently exhibits superior convergence speed and solution accuracy compared to other algorithms.

### 5.3 Performance Analysis and Discussion

The exceptional performance of CMHSA-SMF in both the sensor coverage optimization and PV parameter extraction problems is a direct validation of its core design principles. The adaptive task affinity weighting mechanism dynamically leverages task relationships, fostering exploration by transferring knowledge from disparate tasks in early phases and exploiting similarities for refined convergence in later phases. This intelligent knowledge transfer is a key reason for the algorithm's superior Best and Mean values evident in the results presented Table 11.

Furthermore, the algorithm's robustness and stability, evidenced by consistently low Standard Deviation values across both applications, are attributable to the historical parameter-aware resource allocation mechanism. By rewarding successful searches and penalizing negative transfers through its dual-memory framework, the algorithm efficiently manages the competitive inter-task relationships. This is complemented by the co-evolutionary coupling strategy, which fine-tunes inter-task interactions. The synergy of these components ensures not only high-quality solutions but also rapid and reliable convergence, as clearly demonstrated in the convergence profiles illustrated in Figures 10 and 12.

In conclusion, the exceptional performance of CMHSA-SMF in solving real-world CMTOPs is not accidental but is a direct consequence of its well-designed architectural components. The

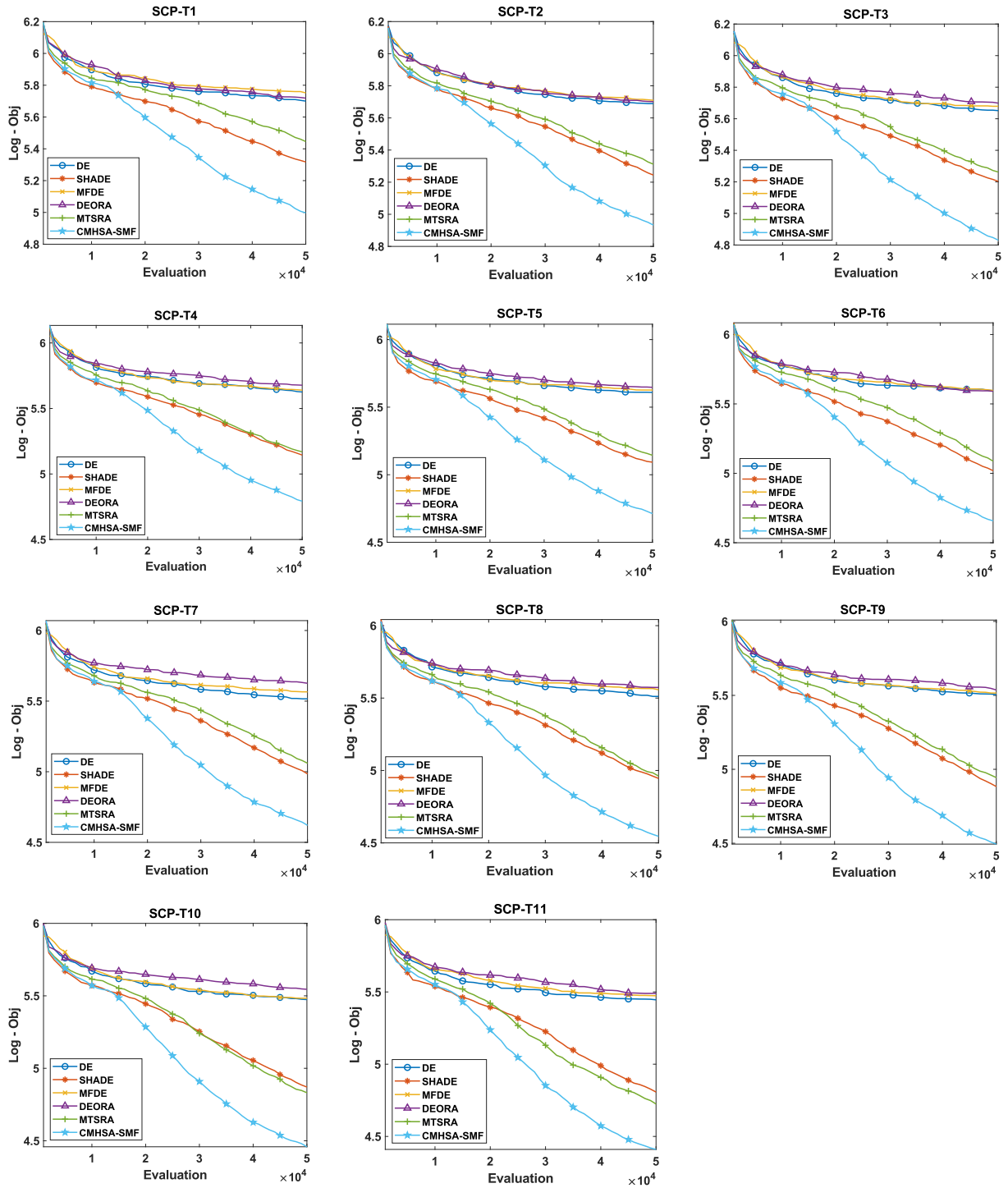


Fig. 10. Average objective value convergence (log style) for 30 independent runs of DE, SHADE, MFDE, DEORA, MTSRA, and CMHSA-SMF on sensor coverage problems.

SMF mechanism enables intelligent task pairing, the historical parameter-aware allocation ensures efficient and robust resource utilization, and the co-evolutionary coupling strategy dynamically optimizes the search process. Together, they empower CMHSA-SMF to effectively address the challenges of accuracy, efficiency, and robustness in complex practical applications like sensor coverage optimization and PV model parameter extraction.

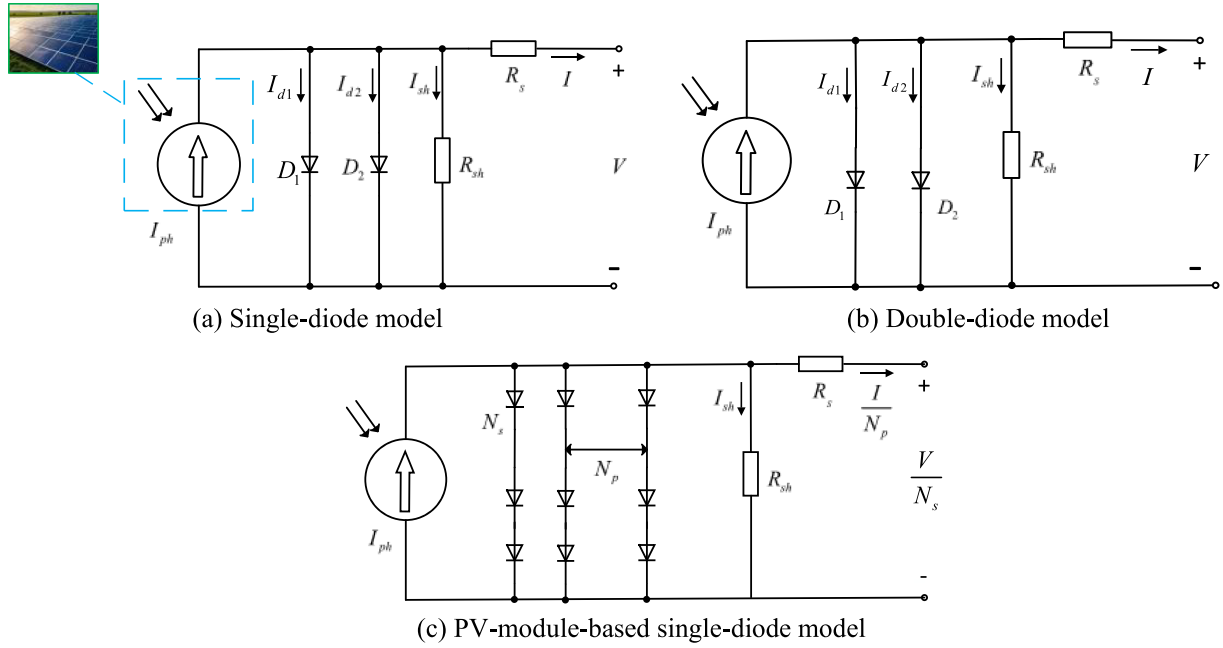


Fig. 11. Equivalent circuit of three competitive PV models.

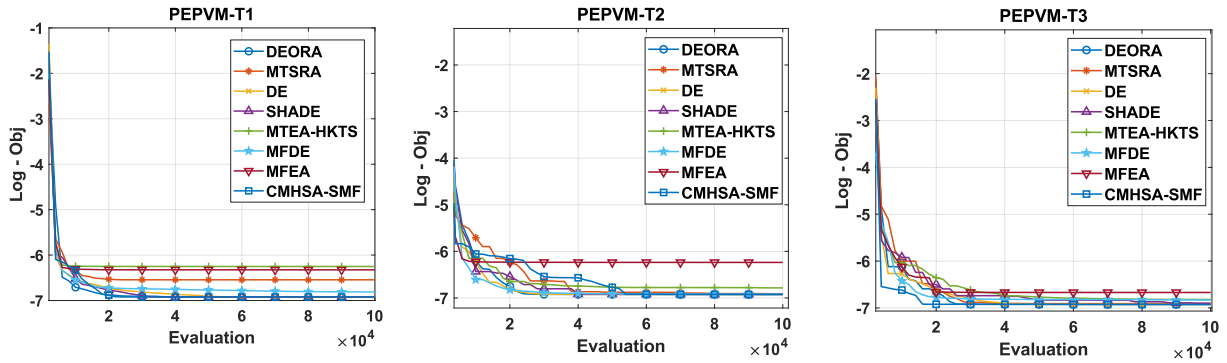


Fig. 12. Average objective value convergence (log style) for 30 independent runs of DE, SHADE, MFEA, MTDE, MFEA, DEORA, MTSRA, and CMHSA-SMF on partial test functions for PV.

## 6 Conclusions and Future Work

As an emerging branch within the field of MTO, CMTO represents a research direction with significant practical relevance. This study addresses the limitations of existing solutions for CMTO problems by proposing CMHSA-SMF, a novel competitive multi-task adaptive cooperative optimization algorithm designed for CMTOPs. The algorithm introduces three key innovations: an adaptive task affinity weighting with dynamic thresholds that optimizes auxiliary task selection by exploiting phase-dependent requirements during early and late optimization stages; a historical parameter-aware adaptive resource allocation mechanism that dynamically distributes computational resources based on iterative task performance metrics and success records; and a collaborative co-evolutionary dynamic coupling strategy integrating adaptive RMP control and dynamic cooperative evolutionary operators to enhance elite population diversity, intensify inter-population communication, and facilitate cross-task knowledge transfer for accelerated convergence. CMHSA-SMF demonstrates superior performance over existing algorithms across multiple benchmark

problems. Its practical efficacy is further validated through real-world applications in sensor coverage optimization and PV model parameter extraction, achieving significant cost reductions and RMSE improvements.

Future research will explore applications of CMHSA-SMF in broader practical domains, such as hyperparameter optimization for machine learning models and logistics distribution route planning, to validate its adaptability and efficacy in diverse complex scenarios. Furthermore, integrating machine learning methods could further enhance the algorithm's performance.

*Data Availability Statement.* The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

*Ethics Approval.* This article does not contain any studies with human participants or animals performed by any of the authors.

*Conflict of Interests.* All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

## References

- [1] Zhihua Cui, Tianhao Zhao, Linjie Wu, A Kai Qin, and Jianwei Li. 2023. Multi-objective cloud task scheduling optimization based on evolutionary multi-factor algorithm. *IEEE Transactions on Cloud Computing* 11, 4 (September 2023), 3685–3699. DOI: <https://doi.org/10.1109/TCC.2023.3315014>
- [2] Ting Wu, Siqi Bu, Xiang Wei, Guibin Wang, and Bin Zhou. 2021. Multitasking multi-objective operation optimization of integrated energy system considering biogas-solar-wind renewables. *Energy Conversion and Management* 229 (February 2021), 113736. DOI: <https://doi.org/10.1016/j.enconman.2020.113736>
- [3] Penghong Wang, Xingtao Wang, Wenrui Li, Xiaopeng Fan, and Debin Zhao. 2024. DV-Hop localization based on distance estimation using multi-node and hop loss in IoT. *IEEE Internet of Things Journal* 11, 16 (May 2024), 27803–27815. DOI: <https://doi.org/10.1109/JIOT.2024.3404492>
- [4] Travis Miranda Lakshmi, Aruldoss Martin, and Venkatasamy Prasanna Venkatesan. 2015. A genetic bankrupt ratio analysis tool using a genetic algorithm to identify influencing financial ratios. *IEEE Transactions on Evolutionary Computation* 20, 1 (April 2015), 38–51. DOI: <https://doi.org/10.1109/TEVC.2015.2424313>
- [5] Yang Cui, Peng Zeng, Zheng Wang, M. Wang, J. Zhang, and Y. Zhao. 2021. Low-carbon economic dispatch of electricity-gas-heat integrated energy system with carbon capture equipment considering price-based demand response. *Power System Technology* 45, 2 (March 2021), 447–461.
- [6] Zixiao Pan, Deming Lei, and Ling Wang. 2020. A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling. *IEEE Transactions on Cybernetics* 52, 6 (October 2020), 5051–5063. DOI: <https://doi.org/10.1109/TCYB.2020.3026571>
- [7] Haoxiang Wang, Bhaba R Sarker, Jing Li, and Jian Li. 2021. Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning. *International Journal of Production Research* 59, 19 (October 2021), 5867–5883. DOI: <https://doi.org/10.1080/00207543.2020.1794075>
- [8] Fuqing Zhao, Fumin Yin, Ling Wang, and Yang Yu. 2024. A co-evolution algorithm with dueling reinforcement learning mechanism for the energy-aware distributed heterogeneous flexible flow-shop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 55, 3 (December 2024), 1794–1809. DOI: <https://doi.org/10.1109/TSMC.2024.3510384>
- [9] Reem Khir, Alan Erera, and Alejandro Toriello. 2023. Robust planning of sorting operations in express delivery systems. *European Journal of Operational Research* 306, 2 (April 2023), 615–631. DOI: <https://doi.org/10.1016/j.ejor.2022.07.024>
- [10] Wei Wu, Yang Liu, Wei Liu, G. Wu, and W. Ma. 2020. A novel autonomous vehicle trajectory planning and control model for connected-and-autonomous intersections. *Acta Automatica Sinica* 46, 9 (September 2020), 1971–1985. DOI: <https://doi.org/10.16383/j.aas.c190002>
- [11] Wen-Chuan Wang, Lei Xu, Kwok-Wing Chau, Yong Zhao, and Dong-Mei Xu. 2022. An orthogonal opposition-based-learning Yin-Yang-pair optimization algorithm for engineering optimization. *Engineering with Computers* 38, 2 (June 2022), 1149–1183. DOI: <https://doi.org/10.1007/s00366-020-01248-9>
- [12] Wen-Chuan Wang, Wei-Can Tian, Dong-Mei Xu, and Hong-Fei Zang. 2024. Arctic puffin optimization: A bio-inspired metaheuristic algorithm for solving engineering design optimization. *Advanced Engineering Software* 195 (September 2024), 103694. DOI: <https://doi.org/10.1016/j.advengsoft.2024.103694>



- [13] Jun Wang, Wen-Chuan Wang, Xiao-Xue Hu, Lin Qiu, and Hong-Fei Zang. 2024. Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artificial Intelligence Review* 57, 4 (March 2024), 98. DOI: <https://doi.org/10.1007/s10462-024-10723-4>
- [14] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. 2015. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation* 20, 3 (July 2015), 343–357. DOI: <https://doi.org/10.1109/TEVC.2015.2458037>
- [15] Yuan Yuan, Yew-Soon Ong, Abhishek Gupta, Puay Siew Tan, Hua Xu. 2016. Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP. In *Proceedings of the 2016 IEEE Region 10 Conference (TENCON)*. IEEE, 3157–3164. DOI: <https://doi.org/10.1109/TENCON.2016.7848632>
- [16] Liang Feng, Yuxiao Huang, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Ke Tang, and Kay Chen Tan. 2020. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. *IEEE Transactions on Cybernetics* 51, 6 (March 2020), 3143–3156. DOI: <https://doi.org/10.1109/TCYB.2019.2962865>
- [17] Lei Zhou, Liang Feng, Jinghui Zhong, Yew-Soon Ong, Zexuan Zhu, and Edwin Sha. 2016. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. In *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–8. DOI: <https://doi.org/10.1109/SSCI.2016.7850039>
- [18] Cuie Yang, Jinliang Ding, Yaochu Jin, Chengzhi Wang, and Tianyou Chai. 2018. Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes. *IEEE Transactions on Automation Science and Engineering*. 16, 3 (September 2018), 1046–1057. DOI: <https://doi.org/10.1109/TASE.2018.2865593>
- [19] Genghui Li, Qingfu Zhang, and Zhenkun Wang. 2022. Evolutionary competitive multitasking optimization. *IEEE Transactions on Evolutionary Computation* 26, 2 (January 2022), 278–289. DOI: <https://doi.org/10.1109/TEVC.2022.3141819>
- [20] Hui Li, Qingfu Zhang, Jingda Deng, and Zong-Ben Xu. 2017. A preference-based multiobjective evolutionary approach for sparse optimization. *IEEE Transactions on Neural Networks and Learning Systems* 29, 5 (March 2017), 1716–1731. DOI: <https://doi.org/10.1109/TNNLS.2017.2677973>
- [21] Matthew L. Ryerkerk, Ronald C. Averill, Kalyanmoy Deb, and Erik D. Goodman. 2017. Solving metameric variable-length optimization problems using genetic algorithms. *Genetic Programming and Evolvable Machines* 18, 2 (June 2017), 247–277. DOI: <https://doi.org/10.1007/s10710-016-9282-8>
- [22] Chuan-Kang Ting, Chung-Nan Lee, Hui-Chun Chang, and Jain-Shing Wu. 2009. Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Part B (Cybernetics)* 39, 4 (August 2009), 945–958. DOI: <https://doi.org/10.1109/TSMCB.2008.2010951>
- [23] Yanchi Li, Wenying Gong, and Shuijia Li. 2023. Evolutionary competitive multitasking optimization via improved adaptive differential evolution. *Expert Systems with Applications* 217 (May 2023), 119550. DOI: <https://doi.org/10.1016/j.eswa.2023.119550>
- [24] Liang Feng, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Yew Soon Ong, Kay Chen Tan, and Alex Kai Qin. 2018. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics* 49, 9 (July 2018), 3457–3470. DOI: <https://doi.org/10.1109/TCYB.2018.2845361>
- [25] Xiaoming Xue, Kai Zhang, Kay Chen Tan, Liang Feng, Jian Wang, Guodong Chen, Xinggang Zhao, Liming Zhang, and Jun Yao. 2020. Affine transformation-enhanced multifactorial optimization for heterogeneous problems. *IEEE Transactions on Cybernetics* 52, 7 (December 2020), 6217–6231. DOI: <https://doi.org/10.1109/TCYB.2020.3036393>
- [26] Yanchi Li, Wenying Gong, and Shuijia Li. 2023. Multitask evolution strategy with knowledge-guided external sampling. *IEEE Transactions on Evolutionary Computation* 28, 6 (November 2023), 1733–1745. DOI: <https://doi.org/10.1109/TEVC.2023.3330265>
- [27] Ben Zhao, Zhihua Cui, Jinqian Yang, Xingjuan Cai, Jianghui Cai, and Jinjun Chen. 2024. A multi-task evolutionary algorithm for solving the problem of transfer targets. *Information Sciences* 681 (October 2024), 121214. DOI: <https://doi.org/10.1016/j.ins.2024.121214>
- [28] Zhihua Cui, Ben Zhao, Tianhao Zhao, Xingjuan Cai, and Jinjun Chen. 2024. An adaptive differential evolution algorithm based on archive reuse. *Information Sciences* 668 (May 2024), 120524. DOI: <https://doi.org/10.1016/j.ins.2024.120524>
- [29] Vladimir Stanovov, Shakhnaz Akhmedova, and Eugene Semkin. 2020. Archive update strategy influences differential evolution performance. In *Proceedings of the 11th International Conference on Advances in Swarm Intelligence (ICSI '20)*. Springer, Cham, 397–404. DOI: [https://doi.org/10.1007/978-3-030-53956-6\\_35](https://doi.org/10.1007/978-3-030-53956-6_35)
- [30] Sridhar Mahadevan, Bamdev Mishra, and Shalini Ghosh. 2018. A unified framework for domain adaptation using metric learning on manifolds. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Cham, 843–860. DOI: [https://doi.org/10.1007/978-3-030-10928-8\\_50](https://doi.org/10.1007/978-3-030-10928-8_50)
- [31] Zhe Liu, Yang Liu, and Rong Gao. 2025. Maximum likelihood estimation for multi-factor uncertain differential equations. *Journal of Computational and Applied Mathematics* 454 (January 2025), 116205. DOI: <https://doi.org/10.1016/j.cam.2024.116205>

- [32] Erik Pitzer, and Michael Affenzeller. 2012. A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent Engineering Systems*. János Fodor, Ryszard Klempous, and Carmen Paz Suárez Araujo (Eds), Springer, Berlin, 161–191. DOI : [https://doi.org/10.1007/978-3-642-23229-9\\_8](https://doi.org/10.1007/978-3-642-23229-9_8)
- [33] Kavitesh Kumar Bali, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. 2019. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation* 24, 1 (March 2019), 69–83. DOI : <https://doi.org/10.1109/TEVC.2019.2906927>
- [34] Yi Jiang, Zhi-Hui Zhan, Kay Chen Tan, and Jun Zhang. 2023. Block-level knowledge transfer for evolutionary multitask optimization. *IEEE Transactions on Cybernetics* 54, 1 (May 2023), 558–571. DOI : <https://doi.org/10.1109/TCYB.2023.3273625>
- [35] Zhengping Liang, Yingmiao Zhu, Xiyu Wang, Zhi Li, and Zexuan Zhu. 2022. Evolutionary multitasking for optimization based on generative strategies. *IEEE Transactions on Evolutionary Computation* 27, 4 (July 2022), 1042–1056. DOI : <https://doi.org/10.1109/TEVC.2022.3189029>
- [36] Xiaoliang Ma, Yongjin Zheng, Zexuan Zhu, Xiaodong Li, Lei Wang, Yutao Qi, and Junshan Yang. 2021. Improving evolutionary multitasking optimization by leveraging inter-task gene similarity and mirror transformation. *IEEE Computational Intelligence Magazine* 16, 4 (October 2021), 38–53. DOI : <https://doi.org/10.1109/MCI.2021.3108311>
- [37] Zhengping Liang, Weiqi Liang, Zhiqiang Wang, Xiaoliang Ma, Ling Liu, and Zexuan Zhu. 2021. Multiobjective evolutionary multitasking with two-stage adaptive knowledge transfer based on population distribution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 7 (July 2021), 4457–4469. DOI : <https://doi.org/10.1109/TSMC.2021.3096220>
- [38] Wu Lin, Qiuzhen Lin, Liang Feng, and Kay Chen Tan. 2023. Ensemble of domain adaptation-based knowledge transfer for evolutionary multitasking. *IEEE Transactions on Evolutionary Computation* 28, 2 (March 2023), 388–402. DOI : <https://doi.org/10.1109/TEVC.2023.3259067>
- [39] Mohammed Alswaiti, Mohanad Albughdadi, and Nor Ashidi Mat Isa. 2019. Variance-based differential evolution algorithm with an optional crossover for data clustering. *Applied Soft Computing* 80 (July 2019), 1–17. DOI : <https://doi.org/10.1016/j.asoc.2019.03.013>
- [40] Yanchi Li, Wenyin Gong, and Shuijia Li. 2023. Evolutionary competitive multitasking optimization via improved adaptive differential evolution. *Expert Systems with Applications* 217 (May 2023), 119550. DOI : <https://doi.org/10.1016/j.eswa.2023.119550>
- [41] Zhaobo Liu, Guo Li, Haili Zhang, Zhengping Liang, and Zexuan Zhu. 2023. Multifactorial evolutionary algorithm based on diffusion gradient descent. *IEEE Transactions on Cybernetics* 54, 7 (May 2023), 4267–4279. DOI : <https://doi.org/10.1109/TCYB.2023.3270904>
- [42] Caitong Yue, Ponnuthurai N. Suganthan, Jing Liang, Boyang Qu, Kunjie Yu, Yongsheng Zhu, and Li Yan. 2021. Differential evolution using improved crowding distance for multimodal multiobjective optimization. *Swarm and Evolutionary Computation* 62 (April 2021), 100849. DOI : <https://doi.org/10.1016/j.swevo.2021.100849>
- [43] Radka Poláková. 2017. L-SHADE with competing strategies applied to constrained optimization. In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Los Alamitos, CA, 1683–1689. DOI : <https://doi.org/10.1109/CEC.2017.7969504>
- [44] Genghui Li, Qiuzhen Lin, and Weifeng Gao. 2020. Multifactorial optimization via explicit multipopulation evolutionary framework. *Information Sciences* 512 (February 2020), 1555–1570. DOI : <https://doi.org/10.1016/j.ins.2019.10.066>
- [45] Bingshui Da, Yew-Soon Ong, Liang Feng, A. Kai Qin, Abhishek Gupta, Zexuan Zhu, Chuan-Kang Ting, Ke Tang, and Xin Yao. 2017. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results. arXiv:1706.03470. Retrieved from <https://arxiv.org/abs/1706.03470>
- [46] Shuijia Li, Wenyin Gong, and Qiong Gu. 2021. A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models. *Renewable and Sustainable Energy Reviews* 141 (May 2021), 110828. DOI : <https://doi.org/10.1016/j.rser.2021.110828>
- [47] Yanchi Li, Wenyin Gong, and Shuijia Li. 2023. Multitasking optimization via an adaptive solver multitasking evolutionary framework. *Information Sciences* 630 (June 2023), 688–712. DOI : <https://doi.org/10.1016/j.ins.2022.10.099>
- [48] Zhongwang Zhang, and Zhi-Qin John Xu. 2024. Implicit regularization of dropout. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 6 (June 2024), 4206–4217. DOI : <https://doi.org/10.1109/TPAMI.2024.3357172>
- [49] Jian Xu, Fei Huang, Yunfei Cui, and Xue Du. 2022. Multi-objective path planning based on deep reinforcement learning. In *Proceedings of the 2022 41st Chinese Control Conference (CCC)*. IEEE, Los Alamitos, CA, 3273–3279. DOI : <https://doi.org/10.23919/CCC55666.2022.9902302>
- [50] Hanxiao Li, Kaizhou Gao, Pei-Yong Duan, Jun-Qing Li, and Le Zhang. 2022. An improved artificial bee colony algorithm with Q-learning for solving permutation flow-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 5 (November 2022), 2684–2693. DOI : <https://doi.org/10.1109/TSMC.2022.3219380>
- [51] Xiaofeng Han, Xiang Cao, Errol L. Lloyd, and Chien-Chung Shen. 2008. Deploying directional sensor networks with guaranteed connectivity and coverage. In *Proceedings of the 5th Annual IEEE Communications Society Conference on*

- Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, Los Alamitos, CA, 153–160. DOI: <https://doi.org/10.1109/SAHCN.2008.28>
- [52] Santpal Singh Dhillon, and Krishnendu Chakrabarty. 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proceedings of the 2003 IEEE Wireless Communications and Networking*. IEEE, Los Alamitos, CA, 1609–1614. DOI: <https://doi.org/10.1109/WCNC.2003.1200627>
- [53] Azzedine Boukerche, and Xin Fei. 2007. A coverage-preserving scheme for wireless sensor network with irregular sensing range. *Ad Hoc Networks* 5, 8 (November 2007), 1303–1316. DOI: <https://doi.org/10.1016/j.adhoc.2007.02.020>

Received 16 April 2025; revised 28 August 2025; accepted 28 October 2025

证书号第8291053号



专利公告信息

# 发明专利证书

发明名称：污水处理软测量虚拟样本生成方法、装置、设备和介质

专利权人：兰州理工大学

地址：730050 甘肃省兰州市七里河区兰工坪路287号

发明人：彭钰;李二超;包寅寅;沈逸镭;黄琛学;郑旭;赫玉涛;王泽  
罗耀国

专利号：ZL 2025 1 0788607.9

授权公告号：CN 120296431 B

专利申请日：2025年06月13日

授权公告日：2025年09月23日

申请日时申请人：兰州理工大学

申请日时发明人：彭钰;李二超;包寅寅;沈逸镭;黄琛学;郑旭;赫玉涛;王泽  
罗耀国

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。  
专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长  
申长雨

申长雨

