UltraModel: A Modeling Paradigm for Industrial Objects

Haoran Yang, **Yinan Zhang***, **Qunshan He**, **Yuqi Ye**, **Jing Zhao** and **Wenhai Wang***
College of Control Science and Engineering, Zhejiang University, Hangzhou, China
{22332111, zhangyinan, heqs, 0923144, zdzzlab}@zju.edu.cn, zhaojing9532@163.com,

Abstract

As Industrial 4.0 unfolds and digital twin technology rapidly advances, modeling techniques that can abstract real-world industrial objects into accurate and robust models, referred to modeling for industrial objects (MIO) tasks, have become increasingly crucial. However, existing works still face two major limitations. First, each of these works primarily focuses on modeling a specific industrial object. When the industrial objects change, the proposed methods often struggle to adapt. Second, they fail to fully consider latent relationships within industrial data, limiting the model's ability to leverage the data and resulting in suboptimal performance. To address these issues, we propose a novel modeling paradigm tailored for MIO tasks, named UltraModel. Specifically, a twin model graph module is designed to construct a customized graph based on the mechanisms of industrial objects and employ graph convolution to generate high-dimensional representations. Then, a multi-scale feature abstraction module and a spatial attention-based feature fusion module are proposed to complement each other in performing multi-scale feature abstraction and fusion on high-dimensional representations. Finally, the outputs are obtained by processing the fused representations through a feedforward network. Experiments on two different industrial objects demonstrate our UltraModel outperforms existing methods, offering a novel perspective for addressing industrial modeling challenges.

1 Introduction

In the industrial field, modeling techniques play an important role in optimizing the design, operation, and maintenance of industrial systems [Leng *et al.*, 2021; Zhang *et al.*, 2024]. The process of using various modeling techniques to abstract real-world industrial objects into models can be referred to modeling for industrial objects (MIO) tasks. Although industrial objects are diverse, they can be broadly categorized into two main types from modeling perspective. The first type is tangible industrial equipment from the real world, such as the distillation column shown in Fig. 1. This type of industrial objects represent the most important and common targets in the industrial field and the models developed for such objects are often called digital surrogates within the context of digital twins [Shao and Kibira, 2018]. The second type comprises certain abstract variables within industrial processes. These variables are difficult to measure directly through sensors but are crucial to the overall process. Therefore, it is necessary to develop a model to predict them as accurately as possible. Additionally, as Industry 4.0 unfolds, industries face increasing pressure to improve efficiency, reduce costs, and mitigate environmental harm [Borowski, 2021], making the development of accurate and robust models more critical than ever[1].

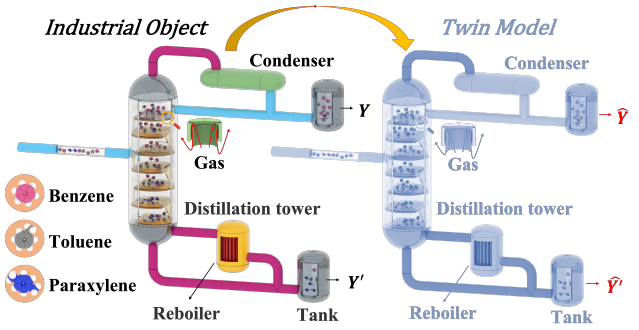

Figure 1: Schematic of modeling for distillation column. It also serves as the industrial background for the first dataset, DIS-COL, in the Experiment section of this paper.

Many previous works on modeling for industrial objects have focused on the development of mechanistic models. These models leverage domain knowledge and industry-specific mechanisms to rigorously represent industrial processes [Velten *et al.*, 2024]. However, mechanistic models typically come with high modeling costs and significant computational efforts [McBride *et al.*, 2020; Elsheikh *et al.*, 2023]. On the other hand, machine learning (ML) techniques, especially deep learning (DL), are advancing at an unprecedented pace [Sarker, 2021; Sharifani and Amini, 2023]. Data-driven models have the potential to break free from the constraints of mechanistic approaches, which are highly de-

*Corresponding author

[1]Appendix is available at https://github.com/SpriteAndMango/UltraModel/blob/main/Appendix.pdf

pendent on prior knowledge and unavoidable assumptions, allowing for more flexible and adaptable modeling for industrial objects. Nevertheless, current efforts to model industrial objects using machine learning or deep learning techniques still face two key limitations.

First, existing research focuses predominantly on one specific industrial object. For instance, Chen and Ierapetritou applied artificial neural networks (ANNs) and support vector regression (SVR) in a hybrid modeling framework to model the Continuous Stirred Tank Reactor (CSTR) [Chen and Ierapetritou, 2020]. Similarly, Li et al. used a surrogate model based on deep convolutional neural networks (CNNs) to model the degradation process of bandsaw blades [Li *et al.*, 2020]. Liu et al. designed a fault diagnosis model based on CNNs for electric machine [Liu *et al.*, 2016]. Su et al. developed a model for predicting the critical properties of chemicals using deep neural networks (DNNs) and long-short-term Memory (LSTM) [Su *et al.*, 2019]. All the aforementioned works share a common limitation: the proposed modeling approaches are restricted to the specific object described in their respective studies. When the modeling object changes, these models fail to adapt effectively, significantly undermining the generalizability of the proposed approaches.

Second, existing research fails to fully consider the latent relationships among variables in industrial data, which limits the model's ability to leverage the data and further results in suboptimal performance [Liu *et al.*, 2016; Chen and Ierapetritou, 2020; Li *et al.*, 2020]. Since many industrial objects can be modeled using the aforementioned mechanistic modeling approach, such mechanistic models can be viewed as a set of equations or functions that represent the behavior of industrial objects. For this type of industrial object, there are clear functional mapping relationships between different variables. This functional mapping is one type of the latent relationships among variables. In addition, there are variables that, while lacking a clear functional mapping relationship, are interdependent and coupled. Such coupled relationships can also be regarded as a form of latent relationships. Therefore, this characteristic distinguishes industrial data for MIO tasks from other datasets. To fully utilize such data, it is crucial to grasp their unique characteristics, thereby maximizing their potential.

To address these limitations, in this paper, we propose an effective modeling paradigm called UltraModel. UltraModel is dedicated to providing a generalizable modeling pipeline capable of solving the majority of MIO tasks. Specifically, UltraModel consists of three main components: twin model graph (TwG) module, multi-scale feature abstraction (MSFA) module and spatial attention-based feature fusion (SAFF) module. The TwG module defines the method for graph construction and, based on the constructed graph, employs graph convolution to generate high-dimensional representations for each node. Furthermore, to fully utilize the latent relationships among variables, the MSFA module ingeniously leverages the adjacency matrix of TwG to aggregate information for each node. It then employs a CNN-based network to perform multi-scale feature abstraction on these nodes. Subsequently, the SAFF module fuses the multi-scale features of each node using spatial attention mechanism, resulting in fused representations. Finally, UltraModel generates the model's output by processing the fused representations through a feedforward network. We highlight the contributions of this work as follows:

- To the best of our knowledge, we are the first to develop a modeling paradigm that can effectively address the challenge of modeling different types of industrial object.
- We propose the TwG module, which enables flexible graph construction that dynamically adapts to various industrial objects and task scenarios, thereby enhancing the universality of our approach.
- Building upon TwG, we propose the MSFA and SAFF modules, which complement each other in performing multi-scale feature abstraction and fusion, enabling a thorough exploration of the latent relationships between variables.
- We conduct extensive experiments on two distinct industrial objects, demonstrating that UltraModel outperforms baseline methods and validates its effectiveness for MIO tasks.

2 Preliminary

MIO tasks can be summarized as the design of models capable of predicting outputs for real-world industrial objects based on their inputs. However, for different industrial objects, the meaning of the inputs and what needs to be achieved in the end are different. For tangible industrial equipment, the inputs correspond to the operational variables in the physical world, such as temperature, pressure, the rate of material inflow and outflow, and other variables that can be directly controlled by operators [Zhu and Zhao, 2022]. The ultimate goal in this case is to develop a digital twin model that accurately simulates the behavior of the equipment under identical operational variables. For abstract variables in industrial processes, the input data typically consist of other variables that are closely related and easy to obtain. There may or may not be a clear functional mapping between these input variables and the abstract variables to be predicted. Ultimately, the goal is to develop a model capable of accurately predicting the values of the target abstract variables using these accessible variables. For simplicity, we collectively refer to the models developed for industrial objects as twin models.

The purpose of this study is to propose an effective modeling paradigm to develop twin models for different types of industrial objects. Thus, this work can be summarized as follows:

$$X = \begin{cases} OV, & \text{if } \mathcal{T} \text{ is a industrial equipment,} \\ EV, & \text{if } \mathcal{T} \text{ is an abstract variable,} \end{cases} \tag{1}$$

$$Y = TWIN(X), \tag{2}$$

where, $\mathcal{T}$ represents the industrial object to be modeled, OV denotes operational variables used for modeling industrial equipment, EV refers to easily accessible variables used for modeling abstract variables, and $TWIN(\cdot)$ is the twin model we aim to develop.

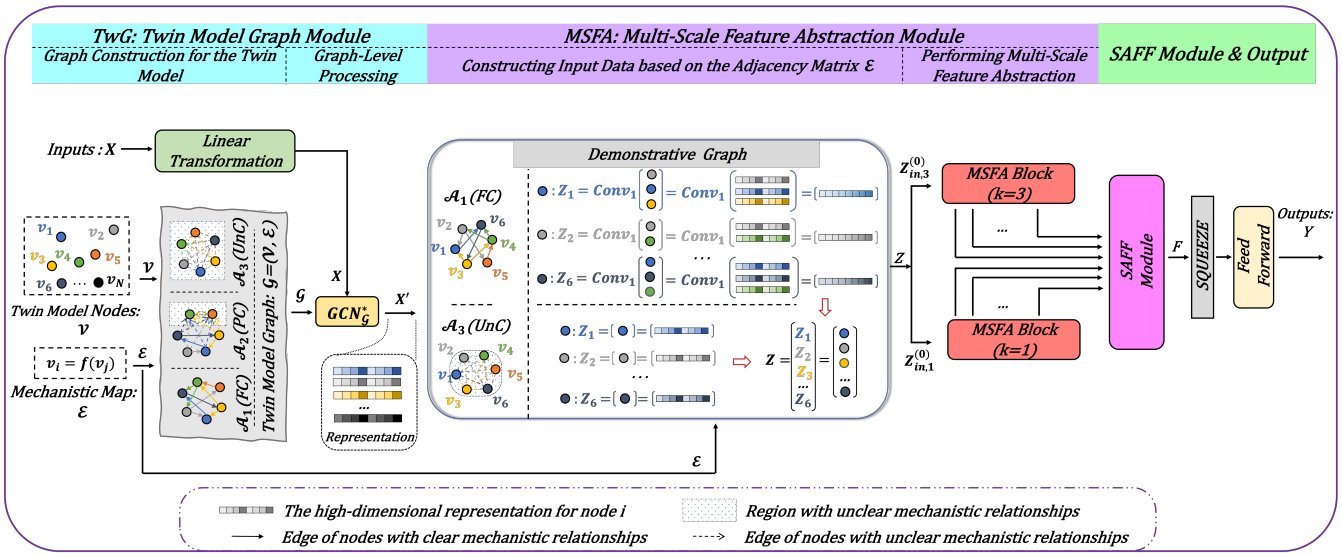

Figure 2: Overall framework of the UltraModel consists of three main modules: twin model graph (TwG) module, multi-scale feature abstraction (MSFA) module, spatial attention-based feature fusion (SAFF) module.

3 Proposed Method

3.1 Overview

Fig. 2 illustrates the framework of the proposed UltraModel, which consists of three main components: TwG module, MSFA module, and SAFF module. The TwG module involves two steps. First, it constructs a graph $\mathcal{G}$ for the twin model based on functional mapping or coupled relationships among the input variables. Then, it performs graph convolution on the constructed graph $\mathcal{G}$ to obtain high-dimensional representations X' for each node. The MSFA module also consists of two steps. First, it aggregates the high-dimensional representations obtained through the graph convolution operation using the adjacency matrix $\mathcal{E}$ derived from the TwG module, producing the input Z for the MSFA block. Subsequently, the MSFA block performs multi-scale feature abstraction on Z. The SAFF module is designed to fuse the abstracted multi-scale features into fused representations F. Finally, the fused representations F are processed through a feedforward network to produce the model's output Y.

3.2 TwG: Twin Model Graph

The input data used to model industrial objects are typically sparse and low-dimensional. For example, operational variables such as temperature, pressure, material inflow and outflow are all one-dimensional real numbers. Building an accurate digital twin model of a highly complex industrial object based solely on such sparse and low-dimensional raw input data is extremely difficult. Therefore, appropriate preprocessing of these raw data is a critical step. Here, we adopt a dimensional enhancement strategy.

As mentioned previously, the variables in industrial data used for MIO tasks often exhibit latent relationships, which can be either clear functional mappings or coupled relationships. If we can enhance the dimensionality of the variables while simultaneously incorporating information from other variables that have a latent relationship with the current one, the resulting high-dimensional representations will be more meaningful compared to those obtained solely through dimensional enhancement methods. Motivated by this, we propose the TwG module based on graph neural networks (GNNs).

Graph Construction for the Twin Model

A key characteristic of GNNs is their ability to aggregate information from neighboring nodes while processing the current node [Xu *et al.*, 2018; Du *et al.*, 2021]. This aligns with the aforementioned motivation, prompting us to adopt graph neural network methods to more effectively address MIO tasks. Suppose our input data used to model industrial objects consists of N variables. The values of these variables are one-dimensional real numbers, denoted as $x_i \in \mathbb{R}$, where $i = 1, 2, \ldots, N$. Thus, for each sample, we have $X = [x_1, x_2, \ldots, x_N]$ and we treat different variables as nodes, represented as $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. The construction of graphs for the twin model is determined based on the varying mechanisms of the corresponding industrial objects and can be categorized into three cases: **Fully clear mechanisms of industrial object** ($\mathcal{A}_1$); **Partially clear mechanisms of industrial object** ($\mathcal{A}_2$); **Unclear mechanisms of industrial object** ($\mathcal{A}_3$). More details can be found in Appendix A.1. We denote the graph construction process as $\mathcal{G} = G(X)$ in the following. It is worth noting that, in all the aforementioned cases, the weight of any edge connecting nodes is a learnable parameter. Subsequently, we further explore how to leverage GNNs to obtain high-dimensional representations.

Graph-Level Processing

To be general, we begin with the input $X \in \mathbb{R}^{N\times 1}$. Initially, the input undergoes a linear transformation without bias to map them into a higher-dimensional space, yielding $X \in \mathbb{R}^{N\times D}$, where D is the dimensionality of the trans-

formed space. Subsequently, we construct the twin model graph $\mathcal{G}$ though $\mathcal{G} = G(X)$. Then, we apply graph convolution to X based on $\mathcal{G}$. The graph convolution operates as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right), \tag{3}$$

here, $\tilde{A} = A_w + I_N$ is the adjacency matrix of the graph $\mathcal{G}$ with added self-connections. I_N is the identity matrix, A_w is the weighted adjacency matrix reflected by $\mathcal{E}$ in the graph. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $W^{(l)} \in \mathbb{R}^{D\times D}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as the $ReLU(\cdot)$. $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the l^{th} layer, $H^{(0)} = X$. The above graph-level processing can be denoted as $X' = GraphConv(X)$. The resulting $X' = [x'_1, x'_2, \ldots, x'_N]$ denotes the high-dimensional representation obtained from the TwG Module, where $x'_i \in \mathbb{R}^D, i = 1, 2, \ldots, N$.

3.3 MSFA: Multi-Scale Feature Abstraction

After obtaining the high-dimensional representation X' from the TwG module, it is essential to design a network architecture capable of thoroughly exploring and abstracting the high-level information embedded within these representations.

Constructing Input Data based on the Adjacency Matrix

The ultimate goal of the MSFA module is to use MSFA blocks to perform deeper feature abstraction on the high-dimensional representation X' generated by the TwG module. However, before this step, X' needs to be processed to ensure compatibility with the subsequent MSFA blocks. Following the design philosophy of the TwG module, we propose an input data construction method based on the adjacency matrix. The detailed process is as follows.

The inputs to the MSFA module are X' and the binary adjacency matrix A_b, derived from $\mathcal{E}$. Unlike the weighted adjacency matrix A_w used in graph convolution, A_b is a binary matrix that only indicates connectivity between nodes. Based on A_b, we construct input data for each node according to the following rules. For any node v_i, the high-dimensional representations $x'_j \in \mathbb{R}^D$ of all nodes $v_j \in \mathcal{N}(v_i)$ are stacked along the 0^{th} dimension to form Z_i, where $i = 1, 2, \ldots, N$:

$$Z_i = STACK(x'_j, dim = 0), x'_j \in \mathcal{N}(v_i), \tag{4}$$

if $\mathcal{N}(v_i)$ contains K nodes, then $Z_i \in \mathbb{R}^{K\times D}$. Subsequently, an additional dimension is added to Z_i to facilitate subsequent convolution operations, resulting in $Z_i \in \mathbb{R}^{K\times D\times 1}$. A 2D convolutional layer , $Conv_1(\cdot)$, with a kernel size of 1 and a channel size of 1 is applied to Z_i to obtain the aggregated representation for node i, $Z_i \in \mathbb{R}^{1\times D\times 1}$,

$$Z_i = Conv_1(UNSQUEEZE(Z_i, dim = -1)). \tag{5}$$

Finally, the aggregated representations of all nodes are concatenated along the channel dimension to form the input Z for the MSFA block, $Z \in \mathbb{R}^{N\times D\times 1}$.

$$Z = STACK(Z_i, dim = 0), i = 1, 2, \ldots, N. \tag{6}$$

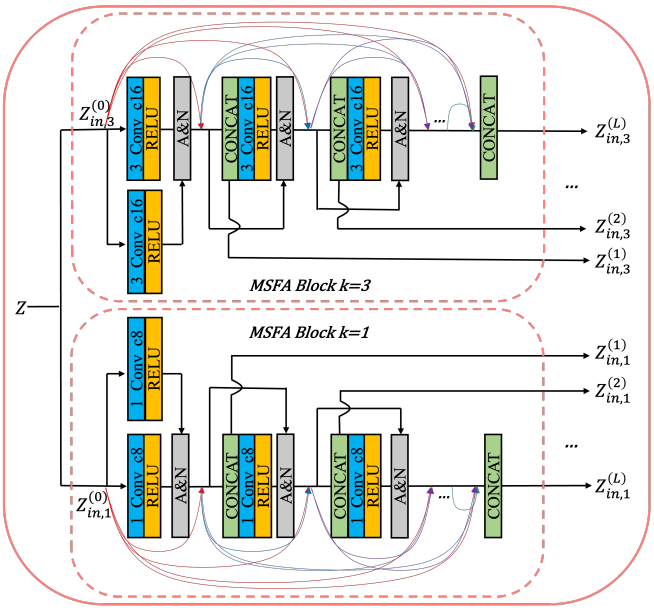

Figure 3: The structure of the multi-scale feature abstraction (MSFA) block. In this work, we design two MSFA blocks with convolutional kernel sizes of (3,1) and (1,1), respectively, to abstract features from the input. $Z^{(l)}_{in,k}$ denotes the input to the l^{th} convolutional layer ($l = 0, 1, 2, \ldots, L$) in the block ($k = 1, 3$). Regarding the details of each layer, k represents the size of the convolutional kernel, and c represents the number of channels.

For nodes with unclear mechanisms, as illustrated in Fig. 2, we adopt the strategy of directly using the high-dimensional representations obtained from the TwG module as their final aggregated representations without any additional processing.

Performing Multi-Scale Feature Abstraction

Motivated by advances in computer vision where CNN-based networks are utilized to abstract high-level information from images [Alpay *et al.*, 2024], we designed the MSFA block. The structure of the MSFA block is illustrated in Fig.3. To perform deeper feature abstraction at various receptive fields, we design a parallel, cascaded CNN network with different convolutional kernel sizes and we adopt residual dense block [He *et al.*, 2016; Zhang *et al.*, 2018] into the MSFA. To facilitate deep feature aggregation, we design a recursive feature abstraction mechanism where the input of each layer is constructed by concatenating the outputs of all preceding layers and the original input feature. Formally, the input $Z^{(l)}_{in,k}$ and output $Z^{(l)}_{out,k}$ at layer l are defined as follows, where $l = 0, 1, \ldots, L$:

$$Z^{(l)}_{in,k} = \begin{cases} Z, & \text{if } l = 0, \\ CAT(Z^{(j)}_{(out,k)} | j = 0, 1, \ldots, l-1, Z), & \text{if } l \geq 1, \end{cases} \tag{7}$$

$$Z^{(l)}_{out,k} = \sigma(Conv(Z^{(l)}_{in,k}), \tag{8}$$

where $CAT(\cdot)$ denotes the concatenation operation, $Conv(\cdot)$ represents a convolutional transformation, and $\sigma(\cdot)$ is $ReLU(\cdot)$ activation function. This mechanism allows the model to progressively incorporate multi-level features while preserving the original input information.

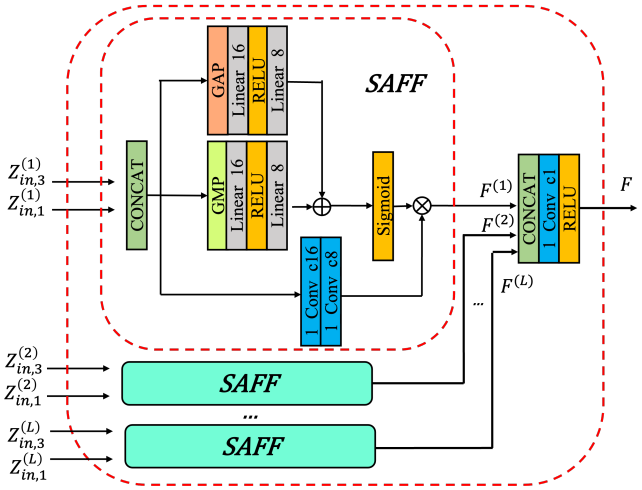

Figure 4: The structure of spatial attention-based feature fusion (SAFF) module. $F^{(l)}$ represents the fused hierarchical features at layer l, while F is the final output of the SAFF module. GAP and GMP refer to Global Average Pooling and Global Max Pooling, respectively. Linear_num denotes a fully connected layer, where num specifies the number of neurons in the layer (num=8,16).

3.4 SAFF: Spatial Attention-based Feature Fusion

After processed by MSFA module, we abstract multi-scale, high-level features of the data at different receptive fields. To fully leverage these high-level features, an effective feature fusion network is essential. To this end, we further propose the SAFF module. The structure of the SAFF module is illustrated in Fig.4. Inspired by the channel attention mechanism [Woo *et al.*, 2018], this module takes the high-level features $Z^{(l)}_{in,3}$ and $Z^{(l)}_{in,1}$ from the same level but different receptive fields as inputs, and performs feature fusion through a 1×1 convolution. Subsequently, based on the logic of channel attention, it generates channel attention scores for the fused features and applies weighting. Finally, the corresponding hierarchical feature $F^{(l)}$ is obtained, where $l = 1, 2, \ldots, L$, as follows:

$$Z^{(l)} = CAT(Z^{(l)}_{in,3}, Z^{(l)}_{in,1}), \tag{9}$$

$$score = \sigma(\mathcal{L}(GAP(Z^{(l)})) + \mathcal{L}(GMP(Z^{(l)}))), \tag{10}$$

$$F^{(l)} = \mathcal{P}(Z^{(l)}) \odot score, \tag{11}$$

where $GAP(\cdot)$, $GMP(\cdot)$ denote global average pooling[ref] and global max pooling, respectively; $\mathcal{L}(\cdot)$ refers to a series of linear transformation with activation function $ReLU$ [Nair and Hinton, 2010]; $\sigma(\cdot)$ denotes the Sigmoid activation function [Kyurkchiev and Markov, 2015]; $\mathcal{P}(\cdot)$ refers to a series of 1×1 convolutional operations for feature fusion; $\odot$ is the elementwise multiplication operation; and $F^{(l)}$ is the fused hierarchical feature at the l^{th} layer. Finally, the SAFF module performs further fusion of the hierarchical features $F^{(l)}$ from all layers to obtain the final output, F, as follows:

$$F = Conv_1(CAT(F^{(l)})), \tag{12}$$

Finally, F undergoes dimensionality reduction and is then fed into a feedforward network to obtain the output Y of the industrial object's twin model, as shown below:

$$Y = FFN(SQUEEZE(F)), \tag{13}$$

where $FFN(\cdot)$ represents the feedforward network.

4 Experiment

4.1 Experimental Setting

Datasets. We conducted experiments on two fundamentally different types of industrial objects to validate the effectiveness and generalization capabilities of our UltraModel. The first type involves tangible industrial equipment, specifically the distillation column, which is widely used in the chemical industry [Choi *et al.*, 2023]. The second type focuses on abstract variables that are difficult to measure directly yet are critical to industrial processes. Here, we choose the acentric factor, a key parameter that describes the extent to which real gases deviate from ideal gas behavior, playing a significant role in thermodynamics and fluid mechanics [Biswas *et al.*, 2023]. Furthermore, these two industrial objects exhibit a key distinction: the former belongs to industrial objects with unclear mechanisms, while the latter pertains to those with fully clear mechanisms. Consequently, we employ two entirely different types of industrial object to validate the generalizability of the proposed method and leverage these two distinct mechanism scenarios to demonstrate its effectiveness. We refer to the distillation column dataset as DIS-COL and the acentric factor dataset as ACE-FAC. A detailed introduction to these two industrial objects and their datasets can be found in Appendix B.1.

Baselines. To ensure a comprehensive comparison, we selected three different types of models as baselines. These include decision tree-based machine learning models: XGBoost [Chen and Guestrin, 2016] and LightGBM [Ke *et al.*, 2017]; sequence-based deep learning models: Transformer [Vaswani, 2017]; and graph-based deep learning models: GCN [Kipf and Welling, 2016], GAT [Veličković *et al.*, 2017], RADA [Chen *et al.*, 2024], DGDL [Zhu and Zhao, 2022], and TGCN-S [Kong *et al.*, 2023]. Additionally, for the ACE-FAC dataset, where the mechanism is fully clear, we include not only the aforementioned baselines but also its mechanistic model, the Lee-Kesler method [Lee and Kesler, 1975; Boublia *et al.*, 2023]. The results derived from this method serve as the benchmark for model comparison.

Metrics. We use three metrics to evaluate the performance of all models: MAE, RMSE and R^2 [Chicco *et al.*, 2021]. Specifically, on the ACE-FAC dataset, we use the results of its mechanistic model as benchmark. To more intuitively illustrate the performance improvements of other models over the benchmark, we further introduce a metric called Model Improvement Rate (MIR), which is calculated as follows:

$$\text{MIR} = \frac{1}{2} \times \left(\frac{\text{MAE}_b - \text{MAE}}{\text{MAE}_b} + \frac{\text{RMSE}_b - \text{RMSE}}{\text{RMSE}_b}\right) \times 100\%, \tag{14}$$

where, MAE_b and RMSE_b represent the MAE and RMSE of the benchmark, respectively.

The **experimental setup** and **implementation details** can be found in Appendix B.

Dataset	Model	MAE	RMSE	R^2	MIR(%)
DIS-COL	XGBoost [Chen and Guestrin, 2016]	0.0096±0.0001	0.0255±0.0003	0.9766±0.0007	——
	LightGBM [Ke *et al.*, 2017]	0.0108±0.0001	0.0279±0.0002	0.9651±0.0005	——
	Transformer [Vaswani, 2017]	<u>0.0064±0.0014</u>	<u>0.0116±0.0033</u>	<u>0.9948±0.0025</u>	——
	GCN [Kipf and Welling, 2016]	0.0152±0.0004	0.0464±0.0007	0.9059±0.0033	——
	GAT [Veličković *et al.*, 2017]	0.0152±0.0001	0.0456±0.0003	0.9085±0.0015	——
	RADA [Chen *et al.*, 2024]	0.0156±0.0006	0.0442±0.0028	0.9144±0.0099	——
	DGDL [Zhu and Zhao, 2022]	0.0111±0.0014	0.0238±0.0031	0.9698±0.0077	——
	TGCN-S [Kong *et al.*, 2023]	0.0167±0.0012	0.0440±0.0013	0.9149±0.0021	——
	UltraModel(Ours)	**0.0044±0.0013**	**0.0064±0.0015**	**0.9984±0.0006**	——
ACE-FAC	Lee-Kesler*[Lee and Kesler, 1975]	0.0951	0.2044	0.8924	0.00
	XGBoost [Chen and Guestrin, 2016]	0.0415±0.0097	0.0577±0.0145	0.9906±0.0028	64.07
	LightGBM [Ke *et al.*, 2017]	0.0552±0.0162	0.0869±0.0248	0.9786±0.0080	49.72
	Transformer [Vaswani, 2017]	0.0322±0.0142	0.0414±0.0181	0.9947±0.0035	72.94
	GCN [Kipf and Welling, 2016]	0.0158±0.0034	0.0207±0.0037	0.9988±0.0007	86.63
	GAT [Veličković *et al.*, 2017]	0.0150±0.0014	0.0218±0.0014	0.9986±0.0006	86.78
	RADA [Chen *et al.*, 2024]	<u>0.0143±0.0007</u>	<u>0.0194±0.0008</u>	<u>0.9989±0.0003</u>	<u>87.74</u>
	DGDL [Zhu and Zhao, 2022]	0.0261±0.0113	0.0323±0.0112	0.9969±0.0025	78.38
	TGCN-S [Kong *et al.*, 2023]	0.0192±0.0056	0.0250±0.0054	0.9982±0.0007	83.79
	UltraModel(Ours)	**0.0052±0.0011**	**0.0089±0.0023**	**0.9998±0**	**95.09**

Table 1: Overall prediction performance on DIS-COL and ACE-FAC, where the best results are highlighted in **bold**, the second best scores are <u>underlined</u>, and the benchmark model is denoted as benchmark*.

4.2 Experimental Results

Performance Comparison on DIS-COL Dataset

This dataset is for modeling real-world industrial equipment, distillation tower. Due to the complexity of the internal mechanisms and the severe coupling of various operational variables affecting the equipment, we regard it as the $\mathcal{A}_3$ case and conduct relevant experiments. The experimental results are shown in Table 1. Overall, among the three different types of baselines, the sequence-based deep learning models perform best on this dataset. Compared with the other two types of baselines, it shows a more obvious advantage in all metrics. The second is the decision tree-based machine learning model, and the third is the graph-based deep learning model. Although, Transformer performs best among all baselines, our UltraModel has a 38.04% performance improvement over Transformer. In this dataset, a fully connected strategy is adopted for the connections between nodes, causing graph-based models to aggregate information from all other nodes indiscriminately when generating high-dimensional representations for each node. This prevents them from fully leveraging their advantages, resulting in the performance of the graph-based models not surpassing that of other baselines.

As shown in Fig. 1, the twin model of distillation column constructed in this study is designed to separate a mixture of benzene, toluene, and xylene. To provide a clearer presentation of the experimental results, we calculate the MAE and RMSE between the predicted percentage compositions (y_1, y_2, y_3) of the three substances and their respective ground truths. The results of all baselines are summarized in Fig. 5, with additional detailed experimental data provided in Appendix C.1.

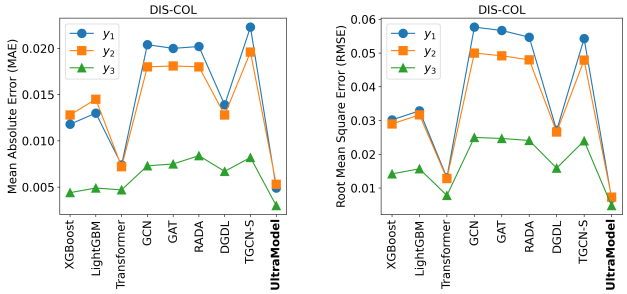

Figure 5: Comparison of prediction errors for the three substances across all models on the DIS-COL dataset.

Performance Comparison on ACE-FAC Dataset

As mentioned earlier, unlike DIS-COL, the mechanisms here are fully clear, enabling us to construct the graph neural network using the $\mathcal{A}_1$ method. For graph-based baselines, the graph is no longer fully connected across all nodes; instead, it is constructed based on the functional mapping relationships between different variables. The advantage of this approach lies in the subsequent graph convolution process, where the network aggregates information only from nodes directly related to the current node, rather than adopting an indiscriminate aggregation strategy. Consequently, this type of baseline performs better than other baseline models, with 80% of such models achieving MIR exceeding 80%. This observation further validates the significance of our proposed TwG module from baselines perspective. Among the baselines, RADA performed the best, achieving an MIR of 87.74%. However, our UltraModel still shows a remarkable 58.88% performance improvement over RADA.

Dataset	TwG	MSFA	SAFF	MAE	RMSE	R^2	MIR(%)
DIS-COL	✗	✗	✗	0.0141±0.0007	0.0397±0.0032	0.9431±0.0094	0.00
	✓	✗	✗	0.0096±0.0021	0.0201±0.0046	0.9848±0.0070	40.64
	✗	✓	✓	0.0048±0.0011	0.0098±0.0009	0.9965±0.0007	70.64
	✓	✓	✓	**0.0044±0.0013**	**0.0064±0.0015**	**0.9984±0.0006**	**76.34**
ACE-FAC	✗	✗	✗	0.0112±0.0017	0.0206±0.0066	0.9988±0.0007	0.00
	✓	✗	✗	0.0073±0.0019	0.0114±0.0008	0.9996±0.0001	39.74
	✗	✓	✓	0.0064±0.0016	0.0105±0.0025	0.9997±0.0001	45.94
	✓	✓	✓	**0.0052±0.0011**	**0.0089±0.0023**	**0.9998±0**	**55.18**

Table 2: Evaluation of the significance of different components of the proposed framework.

4.3 Ablation Studies

We conducted a comprehensive ablation study to assess the individual contributions of each component in our model. The results are summarized in Table 2. To facilitate a more effective comparison between models, we introduced the MIR metric in the ablation study as well. Unlike the previously described setup, the benchmark here refers to the model without the TwG, MSFA, and SAFF modules. Additionally, the MSFA and SAFF modules in our work are designed to complement each other. Therefore, in the ablation studies, we always include or exclude them together. For clarity, we refer to them collectively as the MSFA-SAFF module.

Overall, compared to the case where no modules are included, UltraModel with all modules achieves a 76% performance improvement on the DIS-COL dataset and a 55% improvement on the ACE-FAC dataset. This demonstrates the significant contribution of the TwG and MSFA-SAFF modules to the overall model architecture, as well as their effectiveness. Specifically, when the MSFA-SAFF module is missing, the model is unable to abstract and fuse features in greater depth. Even with the TwG module present, the overall performance of the model remains suboptimal, with improvements of 41% and 40% on the two datasets, respectively. On the other hand, when the MSFA-SAFF module is present but the TwG module is missing, the model can perform in-depth feature abstraction and fusion, resulting in a significant performance improvement. However, without the TwG module, the model cannot generate more meaningful high-dimensional representations prior to feature abstraction, which limits its ability to achieve further breakthroughs. Thus, although the model shows considerable improvement, it does not reach its full potential. In conclusion, the ablation study reveals that the MSFA-SAFF module provides a coarse-grained, substantial performance enhancement, while the TwG module operates at a finer-grained level, complementing the MSFA-SAFF module and enabling further advancements in model performance.

4.4 Visualization

We take the benzene prediction results (y_1) on the DIS-COL dataset as an example to visualize its predicted values against true values, as shown in Fig. 6. More visualization results can be found in the Appendix C.2. In the figure, the x-axis represents the model's predicted values, and the y-axis represents the real values. The closer the distribution of the model's predictions is to the line $y = x$, the better the model's performance. As shown in the figure, our UltraModel (represented by red circles) demonstrates a clear advantage over other baselines.

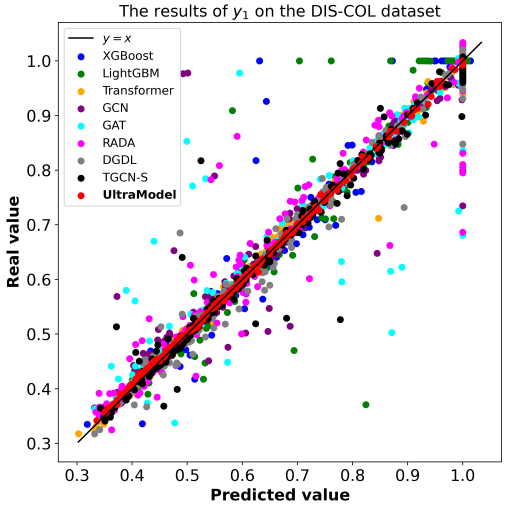

Figure 6: Visualization of benzene prediction results y_1 on the DIS-COL dataset.

5 Conclusion

In this work, we propose a novel modeling paradigm for various industrial objects (named UltraModel), which is the first attempt to address the challenge of modeling different types of industrial objects. Equipped with the TwG module, our UltraModel can flexibly adapt to various industrial objects with different underlying mechanisms. Furthermore, the collaboration between the MSFA module and the SAFF module enhances the model's feature abstraction and fusion capabilities, significantly improving its data mining capability. We validate UltraModel on two datasets representing distinct industrial objects: the distillation column and acentric factor, both of which are representative but fundamentally different and the results demonstrate UltraModel's superior performance in the MIO task. Hope our research can provide inspiration and assistance to relevant researchers and contribute to filling the gap in the field of industrial modeling.

References

[Alpay *et al.*, 2024] Kadir Cenk Alpay, Ahmet Oğuz Akyüz, Nicola Brandonisio, Joseph Meehan, and Alan Chalmers. Deepduohdr: A low complexity two exposure algorithm for hdr deghosting on mobile devices. *IEEE Transactions on Image Processing*, 2024.

[Biswas *et al.*, 2023] Sayandeep Biswas, Yunsie Chung, Josephine Ramirez, Haoyang Wu, and William H Green. Predicting critical properties and acentric factors of fluids using multitask machine learning. *Journal of Chemical Information and Modeling*, 63(15):4574–4588, 2023.

[Borowski, 2021] Piotr F Borowski. Digitization, digital twins, blockchain, and industry 4.0 as elements of management process in enterprises in the energy sector. *Energies*, 14(7):1885, 2021.

[Boublia *et al.*, 2023] Abir Boublia, Tarek Lemaoui, Ghaiath Almustafa, Ahmad S Darwish, Yacine Benguerba, Fawzi Banat, and Inas M AlNashef. Critical properties of ternary deep eutectic solvents using group contribution with extended lee–kesler mixing rules. *ACS omega*, 8(14):13177–13191, 2023.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[Chen and Ierapetritou, 2020] Yingjie Chen and Marianthi Ierapetritou. A framework of hybrid model development with identification of plant-model mismatch. *AIChE Journal*, 66(10):e16996, 2020.

[Chen *et al.*, 2024] Yitao Chen, Yalin Wang, Qingkai Sui, Xiaofeng Yuan, Kai Wang, and Chenliang Liu. Residual-aware deep attention graph convolutional network via unveiling data latent interactions for product quality prediction in industrial processes. *Expert Systems with Applications*, 245:123078, 2024.

[Chicco *et al.*, 2021] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.

[Choi *et al.*, 2023] Yeongryeol Choi, Bhavana Bhadriaju, Hyungtae Cho, Jongkoo Lim, In-Su Han, Il Moon, Joseph Sang-Il Kwon, and Junghwan Kim. Data-driven modeling of multimode chemical process: Validation with a real-world distillation column. *Chemical Engineering Journal*, 457:141025, 2023.

[Du *et al.*, 2021] Jinlong Du, Senzhang Wang, Hao Miao, and Jiaqiang Zhang. Multi-channel pooling graph neural networks. In *IJCAI*, pages 1442–1448, 2021.

[Elsheikh *et al.*, 2023] Mohamed Elsheikh, Yak Ortmanns, Felix Hecht, Volker Roßmann, Stefan Krämer, and Sebastian Engell. Control of an industrial distillation column using a hybrid model with adaptation of the range of validity and an ann-based soft sensor. *Chemie Ingenieur Technik*, 95(7):1114–1124, 2023.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Kong *et al.*, 2023] Liyuan Kong, Chunjie Yang, Siwei Lou, Yu Cai, Xiaoke Huang, and Mingyang Sun. Collaborative extraction of intervariable coupling relationships and dynamics for prediction of silicon content in blast furnaces. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2023.

[Kyurkchiev and Markov, 2015] Nikolay Kyurkchiev and Svetoslav Markov. Sigmoid functions: some approximation and modelling aspects. *LAP LAMBERT Academic Publishing, Saarbrucken*, 4, 2015.

[Lee and Kesler, 1975] Byung Ik Lee and Michael G Kesler. A generalized thermodynamic correlation based on three-parameter corresponding states. *AIChE Journal*, 21(3):510–527, 1975.

[Leng *et al.*, 2021] Jiewu Leng, Dewen Wang, Weiming Shen, Xinyu Li, Qiang Liu, and Xin Chen. Digital twins-based smart manufacturing system design in industry 4.0: A review. *Journal of manufacturing systems*, 60:119–137, 2021.

[Li *et al.*, 2020] Pin Li, Xiaodong Jia, Jianshe Feng, Feng Zhu, Marcella Miller, Liang-Yu Chen, and Jay Lee. A novel scalable method for machine degradation assessment using deep convolutional neural network. *Measurement*, 151:107106, 2020.

[Liu *et al.*, 2016] Ruonan Liu, Guotao Meng, Boyuan Yang, Chuang Sun, and Xuefeng Chen. Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine. *IEEE Transactions on Industrial Informatics*, 13(3):1310–1320, 2016.

[McBride *et al.*, 2020] Kevin McBride, Edgar Ivan Sanchez Medina, and Kai Sundmacher. Hybrid semi-parametric modeling in separation processes: a review. *Chemie Ingenieur Technik*, 92(7):842–855, 2020.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[Sarker, 2021] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160, 2021.

[Shao and Kibira, 2018] Guodong Shao and Deogratias Kibira. Digital manufacturing: Requirements and challenges for implementing digital surrogates. In *2018 Winter Simulation Conference (WSC)*, pages 1226–1237. IEEE, 2018.

[Sharifani and Amini, 2023] Koosha Sharifani and Mahyar Amini. Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, 10(07):3897–3904, 2023.

[Su *et al.*, 2019] Yang Su, Zihao Wang, Saimeng Jin, Weifeng Shen, Jingzheng Ren, and Mario R Eden. An architecture of deep learning in qspr modeling for the prediction of critical properties using molecular signatures. *AIChE Journal*, 65(9):e16678, 2019.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Velten *et al.*, 2024] Kai Velten, Dominik M Schmidt, and Katrin Kahlen. *Mathematical modeling and simulation: introduction for scientists and engineers*. John Wiley & Sons, 2024.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Zhang *et al.*, 2018] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[Zhang *et al.*, 2024] Yinan Zhang, Wenhai Wang, Qiang Yang, Xiaoyu Tang, Wei Ruan, Yingze Li, Surong Daoerji, Xiang Zhang, Yuqi Ye, Jiawei Huang, et al. Promoting digital twin technology application for process industry: A novel generation modelling platform and its implementations. *Digital Twins and Applications*, 1(1):51–74, 2024.

[Zhu and Zhao, 2022] Kun Zhu and Chunhui Zhao. Dynamic graph-based adaptive learning for online industrial soft sensor with mutable spatial coupling relations. *IEEE Transactions on Industrial Electronics*, 70(9):9614–9622, 2022.