DNA: A DOUBLE-HELIX HIERARCHICAL DEEP INTERACTION NETWORK ARCHITECTURE FOR INDUSTRIAL DIGITAL TWIN

Anonymous authors
Paper under double-blind review

ABSTRACT

High-fidelity modeling is fundamental to industrial digital twin, yet a trade-off persists between the physical transparency of mechanistic models and the superior predictive power of data-driven approaches. Existing hybrid methods often fail to bridge this gap, suffering from shallow integration and rigid one-shot modeling strategies that struggle with complex industrial systems. To address these issues, we propose a **Double-Helix Hierarchical Deep Interaction Network Architecture (DNA)**. Inspired by the double-helix structure of biological *DNA*, our architecture intertwines mechanistic and data-driven modeling strands through two novel interaction strategies, forming a symbiotic architecture in which the two strands mutually enhance predictive capability. Furthermore, the hierarchical structure of DNA allows it to model complex systems in a multi-stage manner through two tailored deep learning frameworks. We evaluate the proposed approach on two distinct industrial systems, where DNA achieves reductions in MAE of 3.704% and 3.328%, respectively, compared to the strongest baseline. These results demonstrate the effectiveness of DNA and suggest its potential to provide new insights for the development of industrial digital twins.

1 INTRODUCTION

In industrial digital twin applications, developing high-fidelity models is essential for enabling real-time monitoring, prediction, and optimization Liu et al. (2021); Zhang et al. (2024). The fidelity of these models dictates the digital twin system's ability to mirror the intricate physical, chemical, and behavioral dynamics of real-world systems Peterson et al. (2025). In industrial practice, mechanistic models are often favored due to their reliability, stability, and physical transparency Schweidtmann et al. (2024), particularly in scenarios requiring stringent safety standards. Despite their theoretical rigor, mechanistic models suffer from several significant limitations. For instance, they often rely on simplifying assumptions during the modeling process, such as neglecting secondary reactions or assuming perfect mixing, which can introduce systematic biases into their predictions Himmelblau & Riggs (2012). Their predictive performance may also degrade markedly under unfamiliar operating conditions or previously unseen process states Bishnu et al. (2023). Morever, mechanistic models typically require substantial modeling effort and computational resources Elsheikh et al. (2023). These factors further limit the the flexibility of mechanistic approaches in practical applications.

Recently, many researchers have increasingly turned to develop data-driven models, aiming to mitigate the limitations associated with the mechanistic models Ameri (2025); Mansouri et al. (2025). Extensive studies demonstrate that data-driven models based on these methods often outperform traditional mechanistic models in terms of prediction accuracy, computational efficiency, and reduced development cost Mane et al. (2024); Pal et al. (2025); Yang et al. (2025). However, such data-driven models typically fail to consider the first principles governing the modeled system, making it difficult to generate physically grounded predictions Bradley et al. (2022). Subsequently, researchers have begun to explore hybrid modeling approaches that integrate mechanistic and data-driven models, aiming to combine the physical transparency of the former with the predictive power of the latter. One of the most representative approaches is structured hybrid modeling (SHM), which explicitly couples the two components through serial, parallel, or combined architectures, forming a clearly structured hybrid system Schweidtmann et al. (2024), as illustrated in Fig.4. However, such meth-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ods essentially amount to a simple stacking of mechanistic and data-driven components, following a unidirectional information flow, and thus fail to achieve effective bidirectional interaction between them. Moreover, most existing hybrid modeling approaches adopt a one-shot modeling strategy that attempts to directly predict the final output from the original input in a single step, regardless of the underlying complexity of the target system Su et al. (2019); Li et al. (2020), which, although enabling end-to-end prediction, their performance tends to degrade when applied to large-scale or highly complex industrial systems.

Most importantly, in real-world industrial applications, although data-driven models offer appealing advantages, their black-box nature and limited physical transparence have prevented their widespread adoption in safety-critical industrial settings. Consequently, data-driven modeling approaches remain largely confined to academic research. In contrast, mechanistic models, despite lacking these advantages, are fully interpretable and highly stable, and thus remain the preferred choice in industrial practice. Therefore, enhancing the performance of mechanistic models is of significant industrial value. This naturally leads to a key question: Can we design a modeling architecture that integrates mechanistic and data-driven models in a novel way, allowing it to leverage the superior predictive capability of data-driven models while providing effective feedback to improve the mechanistic models themselves? Such an architecture would allow mechanistic models to benefit from data-driven learning, providing a practical solution (the enhanced mechanistic model) for industrial scenarios where deploying data-driven models is impractical or undesirable.

The answer is yes. Inspired by the double-helix structure of biological *DNA* Klug (2004), we propose a novel Double-Helix Hierarchical **D**eep Interaction **N**etwork **A**rchitecture (DNA) for industrial digital twin. In the proposed architecture, the sequentially connected chains of mechanistic and data-driven models mirror the two long deoxyribonucleotide backbone strands of *DNA*; the interaction between mechanistic models and data-driven models resembles the base-pairing structure formed by nitrogenous bases in *DNA*; and the hierarchical structure also coincides with the spiraling progression of the *DNA*. Under the proposed architecture, we further introduce two distinct interaction strategies between mechanistic and data-driven models, enabling effective bidirectional interaction between the two components. In addition, we also propose two corresponding data-driven frameworks, which, together with the hierarchical structure of DNA, facilitate multi-stage modeling for complex industrial systems. The main contributions of our work are as follows:

- We propose a novel hybrid modeling architecture, DNA, in which data-driven models permanently embed their enhancements into the mechanistic models, enabling the resulting models to be decoupled from the data-driven model and independently deployed in industrial practice.
- We design two interaction strategies for different types of mechanistic models and propose two novel data-driven frameworks that are aligned with the above interaction strategies.
- Extensive experiments on two distinct industrial systems demonstrate the effectiveness and robustness of our DNA architecture. Another merit of DNA is that improvements in upstream stages can propagate downstream, leading to more accurate subsequent predictions.

2 Task Definition and Architecture Overview

2.1 Task Definition

A key step in building an industrial digital twin is to construct a high-fidelity model $\mathcal{M}$ of the target industrial system $\mathcal{O}$. Given the same input conditions X as the real-world system, the model should produce outputs that closely match the real outputs. In this way, the model can simulate the behavior of the physical system and serve as its digital twin. This task can be formulated as: $\min_{\mathcal{M}} \mathbb{E}_{X\sim\mathcal{X}}\left[|\mathcal{M}(X) - \mathcal{O}(X)|\right]$, where $\mathbb{E}[\cdot]$ denotes the mathematical expectation and $\mathcal{X}$ denotes the distribution of operating conditions. Three-phase pipelines are mainly used to transport mixtures of oil, gas, and water and play an important role in offshore oil and gas production Nwanwe & Duru (2022). Here, we use it as a representative example to illustrate the proposed approach.

We first construct an original mechanistic model $\mathcal{M}_{me}$ for the three-phase pipeline based on the Beggs-Brill theory Beggs & Brill (1973); Chaves et al. (2022). The model contains two sequential modeling stages. In Stage I, the mechanistic model $\mathcal{M}_{me}^{I}$ uses the original operating conditions X to calculate the physical properties of the different phases, denoted as Y_{me}^{I}. In Stage II,

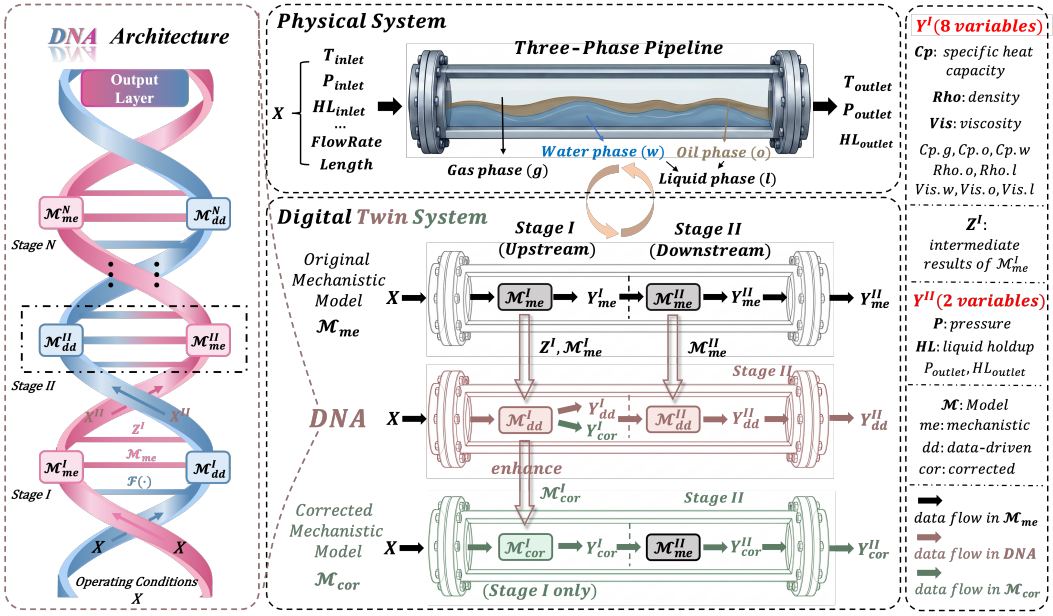


Figure 1: Task demonstration and DNA architecture. Left: DNA architecture; middle: the three-phase pipeline digital twin system; right: definitions of the variables.

$\mathcal{M}^{II}_{me}$ takes X, Y^I_{me} as input and performs further calculations to obtain the outputs Y^{II}_{me} for the two downstream variables. The detailed data flow is provided in Appendix C.4. Following the Beggs-Brill theory, $\mathcal{M}^I_{me}$ mainly consists of explicit functions: $y = f(x_1, x_2, \ldots, x_n)$ and can be written as $\mathcal{M}^I_{me} = \{f_1, f_2, \ldots, f_n\}$. In contrast, $\mathcal{M}^{II}_{me}$ mainly consists of implicit equations: $F(x_1, x_2, \ldots, x_n, y, \dot{y}, \ldots) = 0$ and can be written as $\mathcal{M}^{II}_{me} = \{F_1, F_2, \ldots, F_n\}$, where $\dot{y} = \frac{dy}{dt}$ denotes the time derivative of y.

2.2 Architecture Overview

The hierarchical structure of DNA is designed for complex industrial systems that require multi-stage modeling. As shown in Fig.1, $\mathcal{M}^I_{me}$ and $\mathcal{M}^{II}_{me}$ are sequentially connected along the mechanistic-model chain (red), while the corresponding data-driven models, $\mathcal{M}^I_{dd}$ and $\mathcal{M}^{II}_{dd}$, form the data-driven-model chain (blue). Together, they form the DNA architecture. DNA provides two interaction strategies between mechanistic and data-driven models.

In Stage I, $\mathcal{M}^I_{dd}$ takes the operating conditions X, intermediate results Z^I of $\mathcal{M}^I_{me}$, and mechanistic model $\mathcal{M}^I_{me}$ as inputs, where $\mathcal{M}^I_{me}$ provides physical prior knowledge. $\mathcal{M}^I_{dd}$ simultaneously generates predictions Y^I_{dd} and coefficients $coefs$, which are used to construct a correction function $\mathcal{F}(\cdot)$. The correction function is fed back to $\mathcal{M}^I_{me}$ to refine its predictions, yielding the enhanced model $\mathcal{M}^I_{cor}$ and its outputs Y^I_{cor}. We call this interaction a bidirectional asynchronous compensation strategy. In Stage II, $\mathcal{M}^{II}_{dd}$ incorporates prior knowledge of $\mathcal{M}^{II}_{me}$ through graph neural networks (GNNs), allowing the mechanistic and data-driven components to function as a unified model and jointly produce the predictions Y^{II}_{dd}. We call this interaction a mechanism-embedding strategy. The green pipeline in Fig.1 is a control configuration for the original mechanistic model (black). Only $\mathcal{M}^I_{me}$ is replaced by the enhanced model $\mathcal{M}^I_{cor}$, while the remaining components are unchanged. This configuration evaluates the gain propagation capability of DNA: whether the improvement obtained at the upstream stages can propagate to downstream stages such that subsequent modeling stages also benefit from the upstream enhancement.

In particular, DNA is an industrial modeling architecture rather than merely a specific model. Within the DNA architecture, the data-driven model $\mathcal{M}^i_{dd}$ at each stage $i \in \{1, 2, \ldots, N\}$ can be replaced by any suitable existing deep learning method. Thus, DNA provides a general architecture for integrating mechanistic and data-driven models across multiple modeling stages.

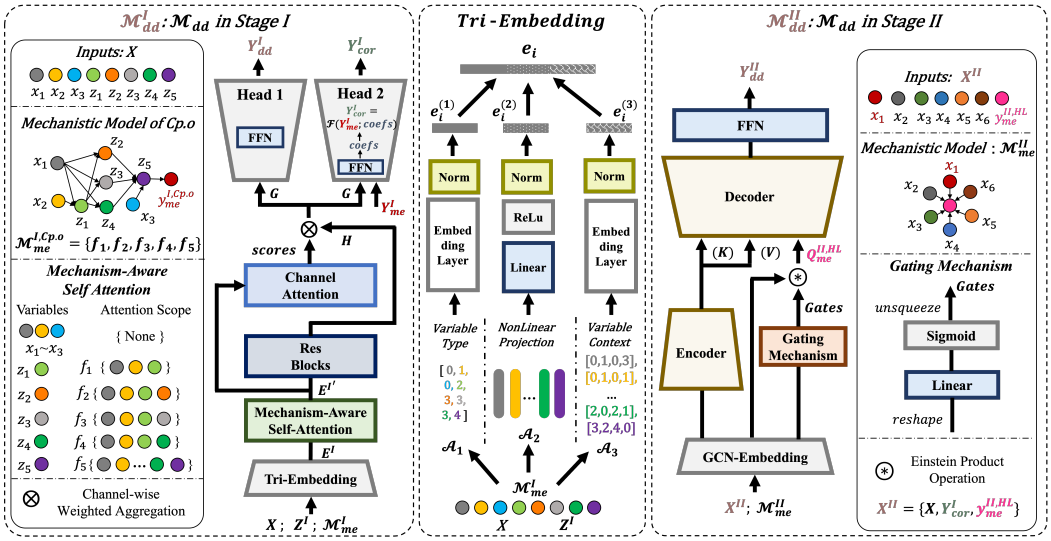


Figure 2: Framework of the proposed data-driven modeling methods and Tripartite Embedding (Tri-Embedding). Left: a concrete $Cp.o$ case is used to illustrate the topology of the mechanistic model in Stage I, the logic of the proposed Mechanism-Aware Self-Attention (MASA), and the overall framework of $\mathcal{M}^{I}_{dd}$. Middle: the workflow of the proposed Tri-Embedding, illustrated using the same $Cp.o$ case. Right: HL is used as an case to illustrate the topology of the mechanistic model in Stage II, the proposed gating mechanism, and the overall framework of $\mathcal{M}^{II}_{dd}$.

3 METHODOLOGY

In addition to the DNA architecture and two interaction strategies, this work also develops two data-driven frameworks tailored to these strategies, as shown in Fig.2.

3.1 DATA-DRIVEN FRAMEWORK IN STAGE I

Tri-Embedding. Industrial datasets often comprise both observable variables (e.g., temperature, pressure, flow rate) and latent ones (e.g., specific heat capacity, density, viscosity) Koretsky (2012), all represented as scalar real values. Our datasets follow the same convention. Relying solely on such low-dimensional inputs to model complex industrial systems is clearly insufficient. Therefore, effective methods for dimensional enhancement are crucial. When the mechanistic model of the industrial system consists of a large number of functions with strongly nested and coupled dependencies, such as $\mathcal{M}^{I,Cp.o}_{me}$ in Fig.2, it can result in extremely complex topological graphs. Although graph-based methods, for example graph convolutional networks (GCNs) Kipf & Welling (2016), are effective in generating high-dimensional representations, their direct application to highly complex topologies can substantially increase the computational burden. To address this issue, we propose the tripartite embedding (Tri-Embedding), which involves only embedding lookups and a simple nonlinear projection, resulting in minimal computational overhead. The Tri-Embedding process consists of three components: **Variable Type Encoding** ($\mathcal{A}_1$); **Nonlinear Projection** ($\mathcal{A}_2$); **Variable Context Encoding** ($\mathcal{A}_3$). Each component will generate their own representation $(e_i^{(1)}, e_i^{(2)}, e_i^{(3)})$, then we concatenate all three parts to obtain the full high dimensional representation of variable i, $e_i = CAT(e_i^{(1)}, e_i^{(2)}, e_i^{(3)})$. Further implementation detail of Tri-Embedding can be found in Appendix B.1.1. In the mechanistic model, the output variable Y^I_{me} is the final computed result and does not participate in the internal computation process. To maintain consistency with the mechanistic model, we exclude Y^I_{me} from the input X^I of $\mathcal{M}^I_{dd}$ as well. The process of generating a high-dimensional representation E^I for each variable is formulated as follows:$E^I = T(X^I; \mathcal{M}^I_{me}), E^I \in \mathbb{R}^{n \times d_m}$, where $T(\cdot)$ denotes the Tri-Embedding operation, $X^I = \{X, Z^I\} = \{x_1, \ldots, x_{n_1}, z_1, \ldots, z_{n_2}\} \in \mathbb{R}^{n \times 1}$, $E^I = \{e_1, e_2, \ldots, e_n\} \in \mathbb{R}^{n \times d_m}$, $n = n_1 + n_2$ is the total number of variables in the input, and $d_m = 32$ is the embedding dimension, as specified in the Dim column of Tab.6.

Data-Driven Framework. The self-attention mechanism is well suited to our setting, as it can capture dependencies among variables regardless of their relative topological distance, which aligns with the complex topology of mechanistic models where relevant variables may be far apart. However, as illustrated by $\mathcal{M}_{me}^{I,Cp.o}$ in Fig.2, the computation of intermediate physical quantities $\{z_i\}_{i=1}^{5}$ is inherently directional. For example, z_1 depends only on x_1 and x_2, and is independent of all other variables. In other words, according to the underlying mechanism of the industrial system, variables beyond x_1 and x_2 should not influence z_1. Therefore, vanilla self-attention, which computes attention weights across all input tokens indiscriminately, contradicts the real-world mechanistic constraints in our digital twin setting. To address this issue, we propose Mechanism-Aware Self-Attention (MASA). MASA assigns a dedicated attention scope to each z_i based on $\mathcal{M}_{me}^{I}$, restricting its attention to variables that are mechanistically relevant to itself. This design offers two key benefits: (i) it enforces mechanistic consistency in $\mathcal{M}_{dd}^{I}$; (ii) it eliminates unnecessary attention operations, reducing computational cost, as shown in Fig.3d. Given the high-dimensional input representation E^I, MASA produces updated representation $E^{I'}$ that aggregate more informative features, as formulated below:

$$E^{I'} = MASA(E^I, E^I, E^I), E^{I'} \in \mathbb{R}^{n \times d_m}. \tag{1}$$

Moreover, in mechanistic model, different input variables may contribute unequally to the final output Y_{me}^{I}. Treating all variables identically during prediction can therefore be suboptimal. Inspired by the channel attention mechanism Woo et al. (2018), we introduce a channel attention module that assigns each variable an importance score, reflecting its contribution to the prediction. Specifically, we first add an extra dimension to $E^{I'}$ to form $E_{exp}^{I'}$, which is then fed into three ResBlocks He et al. (2016) to extract deeper features H. In parallel, $E_{exp}^{I'}$ is processed by the channel attention module to produce a $scores$ vector whose length matches the number of input variables. $scores$ are then used to compute a weighted aggregation of H along the channel dimension, resulting in the fused representation G. More details about ResBlocks and channel attention module can be found in Appendix B.1.2 and B.1.3. These processes are formulated as:

$$E_{exp}^{I'} = unsqueeze(E^{I'}, dim = -1), E_{exp}^{I'} \in \mathbb{R}^{n \times d_m \times 1}, H = \mathcal{R}(E_{exp}^{I'}), H \in \mathbb{R}^{n \times d_m \times 1}, \tag{2}$$

$$scores = \sigma(L(GAP(E_{exp}^{I'}))), scores \in \mathbb{R}^{n \times 1 \times 1}, G = H \otimes scores, G \in \mathbb{R}^{1 \times d_m \times 1}, \tag{3}$$

where $unsqueeze(\cdot)$ denotes the dimension expansion operation; $\mathcal{R}(\cdot)$ denotes ResBlocks; $GAP(\cdot)$ represent global average pooling; $L(\cdot)$ denotes a series of linear transformations with $ReLU$ activations Nair & Hinton (2010); $\sigma(\cdot)$ is the $Sigmoid$ function that generates channel-wise weights; and $\otimes$ is the channel-wise weighted aggregation. Finally, the fused feature G is fed into two prediction heads to generate the data-driven prediction Y_{dd}^{I} and the correction function $\mathcal{F}(\cdot)$. This process is formulated as follows:

$$Y_{dd}^{I} = Head_1(squeeze(G)), coefs = Head_2(squeeze(G)), Y_{cor}^{I} = \mathcal{F}(Y_{me}^{I}; coefs), \tag{4}$$

here, $squeeze(\cdot)$ is used to remove redundant dimensions; $Head_1(\cdot)$ is a two-layer feedforward network, while $Head_2(\cdot)$ is a three-layer feedforward network that produces a set of coefficients $coefs$ to construct $\mathcal{F}(\cdot)$ to refine the original mechanistic model's outputs Y_{me}^{I} .

3.2 Data-Driven Framework in Stage II

GCN-Embedding. When the mechanistic model of an industrial system is formulated as algebraic or differential equations rather than explicit functions, the modeling task becomes one of predicting the unknown variables within these equations. Stage II is designed to handle such cases. In this case, since variables other than the unknowns are typically either computed in earlier stages or inherently known, the topological graphs of the mechanistic model $\mathcal{M}_{me}^{II}$ in this stage are typically simpler than those of $\mathcal{M}_{me}^{I}$ in Stage I, as shown in Fig.2. This relatively simple topology alleviates concerns about the computational overhead associated with GCNs. Moreover, we aim to provide a more diverse data-driven strategy for digital twin modeling. These considerations motivate us to adopt GCNs as an alternative approach for generating high-dimensional representations of the input X^{II}. In addition, a key difference here is that since the target variable y_{me}^{II} itself participates in the equation-solving process, we additionally include y_{me}^{II} as part of the input to $\mathcal{M}_{dd}^{II}$.

$$E^{II} = GCN(X^{II}; \mathcal{G}), E^{II} \in \mathbb{R}^{(n+1) \times d_m}, \tag{5}$$

where $X^{II} = \{x_1, \ldots, x_n, y_{me}^{II}\} \in \mathbb{R}^{(n+1) \times 1}$, $E^{II} = \{e_1, \ldots, e_n, e_{y_{me}^{II}}\} \in \mathbb{R}^{(n+1) \times d_m}$, and $\mathcal{G}$ denotes the topological graph of $\mathcal{M}_{me}^{II}$.

Data-Driven Framework. Encoder–Decoder architectures have demonstrated remarkable performance across a wide range of tasks due to their strong ability to effectively capture complex relationships between inputs and outputs. We adopt the Transformer backbone in our design, with the key distinction lying in the decoder. Detailed architectures of the encoder $Enc(\cdot)$ and decoder $Dec(\cdot)$ are provided in the Appendix B.2.1. The encoder component largely follows the standard Transformer encoder architecture, and the encoding process is formulated as $O_{en} = Enc(E^{II}, E^{II}, E^{II})$, where $O_{en} \in \mathbb{R}^{(n+1)\times d_m}$. The decoder component only includes the cross-attention module of a standard Transformer and all subsequent layers. Since y_{me}^{II} originates from the mechanistic model prediction, it inherently carries physical or domain-specific information and remains highly correlated with the ground truth y_{true}^{II}. Therefore, we use y_{me}^{II} as the query to guide the attention mechanism. However, this design relies on the quality of y_{me}^{II}. When the quality of mechanistic model's output is poor, y_{me}^{II} may guide attention toward misleading patterns, thus exacerbating the propagation of errors. To mitigate this risk, we further introduce a gating mechanism $GM(\cdot)$ that generates $Gates$ for input variables and uses the Einstein product $\circledast$ to compute their weighted sum as the final query Q_{me}, as illustrated in Stage II of Fig.2. This process can be formulated as follows:

$$Gates = GM(E^{II}), Gates \in \mathbb{R}^{(n+1)\times 1}, Q_{me} = E^{II} \circledast Gates, Q_{me} \in \mathbb{R}^{1\times d_m}. \tag{6}$$

Finally, decoder performs cross-attention using Q_{me}, O_{en}, and O_{en} as the query(Q), key(K), and value(V), respectively. Its output is then passed through a three-layer feedforward network $FFN(\cdot)$ to predict the final output Y_{dd}^{II}, as formulated below:

$$Y_{dd}^{II} = FFN(Dec(Q_{me}, O_{en}, O_{en})). \tag{7}$$

Details of the model optimization procedure are provided in the Appendix B.3.

4 Experiment

4.1 Industrial Systems

Three-phase pipeline. As described in Section 2.1, we use the three-phase pipeline as the primary industrial system in our experiments. We first collect real-world data under 759 distinct operating conditions across 20 locations, including the operating conditions and measurements of ten variables, comprising eight physical properties in Stage I and two downstream variables in Stage II, as shown in Fig.1. The operating conditions serve as the inputs (X), while the measurements of the ten variables serve as the ground-truth outputs ($Y_{true}^{I}, Y_{true}^{II}$). We then simulate the mechanistic model under the same conditions to obtain mechanistic predictions (Y_{me}^{I}, Y_{me}^{II}) and intermediate variables Z^I. These data yield eight datasets for Stage I and two for Stage II. In total, the mechanistic model generates 15,180 samples. More details about three-phase pipeline are provided in Appendix C.

Acentric factor. Acentric factor is a key parameter that characterizes the extent to which real gases deviate from ideal gas behavior, playing an important role in thermodynamics and fluid mechanics Biswas et al. (2023). Unlike tangible industrial equipment(three-phase pipeline), the acentric factor represents an abstract variable that is difficult to measure directly yet remains critical to industrial processes. Therefore, the three-phase pipeline and the acentric factor correspond to fundamentally different types of industrial object in real-world scenarios. Accordingly, we select the acentric factor as the second experimental target to further evaluate the generalization capability of DNA. More details about acentric factor are provided in the Appendix D.

4.2 Baselines

Our tasks can generally be categorized as regression problems. Accordingly, we consider four groups of baselines that have shown strong performance in regression tasks, including (i) Tree-based method: LightGBM Ke et al. (2017); (ii) Transformer-based methods: Transformer Vaswani (2017), FlowformerMH Wu et al. (2022b), iTransformer Liu et al. (2024), and MultiPatchFormer (MPFormer)Naghashi et al. (2025); (iii) Graph-based methods: GATv2 Veličković et al. (2017), DGDL Zhu & Zhao (2022), TGCN-S Kong et al. (2023), and RaDA Chen et al. (2024); (iv) Other types of models: TimesNet Wu et al. (2022a), TimeXer Wang et al. (2024) and UMixer Ma et al. (2024). Detailed descriptions of the baselines and the rationale for their selection are provided in the Appendix E.1.

Table 1: Overall performance on the two modeling task. **Red**: the best, Blue: the 2nd best.

Category		DNA		Tree-based		Transformer-based								Graph-based								Others					
Model		Ours		LightGBM		Transformer		FlowformerMH		iTransformer		MPFormer		GATv2		DGDL		TGCNS		RaDA		TimesNet		TimeXer		Umixer	
Metric		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Stage I Cp.g	MM	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72	49.58	50.72
	DD	**0.763**	1.073	0.913	1.205	1.754	2.013	1.467	1.965	1.037	1.312	0.795	1.073	0.851	1.094	1.387	1.736	0.832	1.065	1.300	1.619	0.794	**1.060**	0.807	1.068	0.843	1.100
	Cor	0.831	1.067	—	—	1.253	1.943	0.780	1.111	1.130	1.747	0.812	1.120	0.770	**1.051**	0.799	1.143	0.823	1.101	**0.767**	1.073	0.792	1.138	0.859	1.130	0.792	1.069
	Avg	0.797	**1.070**	—	—	1.503	1.978	1.123	1.538	1.083	1.530	0.804	1.096	0.811	1.073	1.093	1.440	0.827	1.083	1.034	1.346	**0.793**	1.099	0.833	1.099	0.817	1.084
Cp.o	MM	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03	15.01	16.03
	DD	1.704	**2.335**	2.114	2.789	1.716	2.402	2.003	2.578	1.803	2.438	**1.673**	2.416	1.730	2.394	2.139	2.695	2.225	2.742	2.419	3.044	1.728	2.368	1.837	2.476	1.733	2.420
	Cor	1.895	2.474	—	—	1.958	2.517	1.893	2.509	1.871	2.515	1.936	2.491	1.894	2.511	1.892	2.488	1.910	2.500	1.893	2.473	1.906	2.510	**1.880**	2.516	1.895	**2.457**
	Avg	**1.799**	**2.404**	—	—	1.837	2.460	1.948	2.543	1.837	2.477	1.805	2.454	1.812	2.453	2.016	2.592	2.068	2.621	2.156	2.759	1.817	2.439	1.859	2.496	1.814	2.438
Cp.w	MM	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898	2.354	2.898
	DD	0.724	1.017	1.929	2.390	2.692	2.864	1.799	1.985	1.084	1.437	0.677	1.027	1.674	1.851	1.151	1.519	0.874	1.128	0.850	1.225	**0.553**	**0.835**	0.668	0.936	0.774	1.054
	Cor	**1.219**	**1.611**	—	—	1.761	2.183	1.675	2.141	1.685	2.143	1.723	2.201	1.630	2.068	1.754	2.185	1.716	2.222	1.732	2.198	1.701	2.150	1.664	2.098	1.685	2.129
	Avg	**0.972**	**1.314**	—	—	2.226	2.523	1.737	2.063	1.384	1.790	1.200	1.614	1.652	1.960	1.452	1.852	1.295	1.675	1.291	1.711	1.127	1.492	1.166	1.517	1.229	1.592
Rho.o	MM	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15	13.93	14.15
	DD	**0.793**	**1.367**	1.245	1.915	0.956	1.473	1.537	1.852	0.824	1.401	1.021	1.676	1.469	1.793	1.332	1.765	1.644	2.038	0.838	1.403	1.045	1.707	1.097	1.636	0.845	1.392
	Cor	1.504	2.397	—	—	1.532	2.385	1.557	2.394	1.595	**2.381**	1.571	2.382	1.625	2.417	1.576	2.390	1.591	2.390	1.552	2.388	1.779	2.430	**1.486**	2.398	1.601	2.384
	Avg	**1.148**	**1.882**	—	—	1.244	1.929	1.547	2.123	1.210	1.891	1.296	2.029	1.547	2.105	1.454	2.077	1.617	2.214	1.195	1.896	1.412	2.068	1.291	2.017	1.223	1.888
Rho.l	MM	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64	23.07	26.64
	DD	2.434	**3.437**	3.204	4.019	2.696	3.576	3.998	4.556	2.523	3.681	3.178	4.249	4.345	4.859	3.078	4.118	3.202	4.098	3.483	4.190	2.748	3.846	3.137	4.534	**2.259**	3.519
	Cor	6.583	7.946	—	—	6.643	7.990	**6.564**	7.941	6.754	8.153	6.722	8.120	**6.564**	**7.938**	6.683	8.036	6.646	8.000	6.590	7.953	6.957	8.445	6.771	8.203	6.776	8.135
	Avg	**4.509**	**5.692**	—	—	4.670	5.783	5.281	6.248	4.638	5.917	4.950	6.185	5.454	6.399	4.880	6.077	4.924	6.049	5.036	6.072	4.852	6.146	4.954	6.368	4.517	5.827
Vis.w	MM	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45	9.239	15.45
	DD	2.191	**3.125**	2.738	3.832	2.436	3.297	2.489	3.560	2.261	3.268	**2.180**	3.154	2.477	3.390	2.513	3.384	2.221	3.275	2.607	3.401	2.277	3.201	2.181	3.130	2.222	3.142
	Cor	**2.373**	**3.286**	—	—	2.398	3.298	2.388	3.309	2.602	3.420	2.421	3.354	2.485	3.310	2.492	3.321	2.380	3.312	2.381	3.291	2.420	3.299	2.472	3.301	2.403	3.305
	Avg	**2.282**	**3.205**	—	—	2.417	3.298	2.439	3.435	2.432	3.344	2.301	3.254	2.481	3.350	2.503	3.352	2.301	3.294	2.494	3.346	2.349	3.250	2.327	3.215	2.312	3.223
Vis.o	MM	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4	125.7	176.4
	DD	**28.27**	**46.84**	40.80	61.89	29.56	47.67	28.75	47.29	29.43	48.21	28.48	47.25	29.79	47.38	30.33	47.88	28.67	46.87	30.04	48.48	28.63	46.87	29.49	47.27	29.29	47.32
	Cor	36.98	**53.48**	—	—	43.87	61.75	40.12	58.61	38.20	59.56	40.60	57.83	40.04	55.85	46.77	65.80	44.72	64.73	**36.36**	56.70	38.98	53.95	42.42	65.33	38.90	57.49
	Avg	**32.62**	**50.16**	—	—	36.72	54.71	34.44	52.95	33.82	53.88	34.54	52.54	34.92	51.61	38.55	56.84	36.69	55.80	33.20	52.59	33.81	50.41	35.96	56.30	34.09	52.40
Vis.l	MM	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4	445.6	477.4
	DD	**31.82**	**46.65**	76.17	96.65	32.91	49.42	34.46	50.31	35.66	50.63	33.83	49.85	34.89	50.48	34.23	50.38	36.68	52.83	37.40	53.44	35.06	50.92	33.13	48.37	37.59	53.15
	Cor	44.00	60.20	—	—	43.81	60.89	43.89	60.38	43.92	60.20	44.77	61.88	44.28	60.27	43.76	60.35	43.72	60.29	43.67	60.47	43.76	60.25	43.73	**60.15**	**43.64**	60.36
	Avg	**37.91**	**53.43**	—	—	38.36	55.15	39.17	55.34	39.79	55.42	39.30	55.86	39.58	55.38	38.99	55.36	40.20	56.56	40.53	56.95	39.41	55.59	38.43	54.26	40.62	56.75
AF	MM	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044	0.951	2.044
	DD	**0.087**	**0.125**	0.552	0.869	0.099	0.171	0.129	0.182	0.091	0.152	0.102	0.164	0.110	0.170	0.188	0.231	0.093	0.180	0.120	0.184	0.145	0.209	0.091	0.155	0.098	0.153
	Cor	0.279	0.377	—	—	0.347	0.422	**0.274**	**0.364**	0.288	0.382	0.318	0.408	0.298	0.382	0.312	0.394	0.328	0.421	0.291	0.377	0.306	0.387	0.313	0.410	0.292	0.375
	Avg	**0.183**	**0.251**	—	—	0.223	0.296	0.201	0.273	0.189	0.267	0.210	0.286	0.204	0.276	0.250	0.312	0.211	0.301	0.205	0.280	0.226	0.298	0.202	0.283	0.195	0.264
Stage II HL	MM	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199	5.417	7.199
	DD	**1.556**	**2.677**	1.602	2.679	1.758	2.905	1.728	2.789	1.651	2.757	1.574	2.698	1.852	2.871	1.644	2.805	1.670	2.813	1.622	2.734	1.752	2.818	1.894	2.969	1.647	2.765
P	MM	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2	77.55	131.2
	DD	**3.017**	**5.424**	14.60	29.03	9.067	13.73	7.471	10.12	10.41	22.39	3.527	6.234	4.488	7.173	4.695	7.881	9.281	17.39	4.426	7.390	9.885	14.69	33.51	69.99	20.51	41.01

4.3 Overall Performance

The overall performance is summarized in Tab.1. In Stage I, under the DNA architecture, each model is required not only to predict the target variable but also to produce a correction function to further refine the mechanistic model output. Therefore, we evaluate the predicted target values (DD), the corrected mechanistic model outputs (Cor), and their average (Avg). In Stage II, since models are only required to predict the target variable, we evaluate predictive performance only (DD). Furthermore, we include the mechanistic model (MM) results to highlight the improvements achieved by different models under the DNA architecture. **Stage I.** First, in the vertical comparison in the table, we observe that for almost all baselines, both the DD and Cor values show significant improvement compared to MM: the improvement in DD indicates that these baseline models possess strong predictive capabilities, validating the rationality of our baseline selection; the substantial gains in Cor can be attributed to the DNA architecture, under which data-driven models provide effective corrections to the mechanistic model. Regardless of which data-driven model is employed within the DNA architecture, the architecture consistently brings the mechanistic outputs to a substantially higher level of fidelity. Specifically, on approximately 44% of the datasets (Cp.g, Cp.o, Rho.o, Vis.l), Cor achieves an order-of-magnitude improvement over MM in terms of MAE. On the remaining 56% (Cp.w, Rho.l, Vis.w, Vis.o, AF), the MAE is reduced by a factor of 2 to 4. Second, in the horizontal comparison in the table, we find that our proposed data-driven framework consistently achieves the best Avg performance on at least one metric for every dataset. On 88.9% of the datasets (8 out of 9), it outperforms all baselines across all metrics, demonstrating the model's stability and strong generalization ability. In particular, our model achieves the best DD performance on at least one metric in 88.9% (8 out of 9) of the datasets, further demonstrating the superior predictive capability of the proposed data-driven framework in Stage I. **Stage II.** In this phase, we evaluate only the predicted target values of each model. DNA achieves the best results across all metrics on both datasets. Specifically, compared to the strongest baseline MultiPatchFormer Naghashi et al. (2025), DNA reduces MAE and RMSE on these two datasets by 7.80% and 6.89% on average, respectively.

4.4 Ablation Study

Stage I. The ablation study results for Stage I are summarized in Tab.2. To verify the superiority of the proposed MASA over the vanilla multi-head self-attention (MHSA), we first replace MASA with MHSA, and the experimental results (MASA $\rightarrow$ MHSA) show that replacing MASA with

Table 2: Ablation study results on Stage I.

Stage I		Cp.g		Cp.o		Cp.w		Rho.o		Rho.l		Vis.w		Vis.o		Vis.l	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Ours	DD	**0.763**	**1.073**	1.704	**2.335**	**0.724**	**1.017**	**0.793**	**1.367**	2.434	3.437	2.191	**3.125**	**28.27**	46.84	31.82	**46.65**
	Cor	0.831	**1.067**	1.895	**2.474**	1.219	1.611	**1.504**	2.397	**6.583**	**7.946**	**2.373**	**3.286**	**36.98**	53.48	44.00	**60.20**
	Avg	**0.797**	**1.070**	**1.799**	**2.404**	**0.972**	**1.314**	**1.148**	**1.882**	4.509	**5.692**	**2.282**	**3.205**	**32.62**	**50.16**	**37.91**	**53.43**
MASA → MHSA	DD	0.963	1.231	**1.702**	2.372	1.333	1.532	0.802	1.394	**2.144**	3.451	2.252	3.179	28.31	46.82	32.10	47.19
	Cor	**0.787**	1.091	1.962	2.543	1.167	1.542	1.575	**2.382**	6.726	8.151	2.520	3.311	59.74	85.06	**43.72**	60.46
	Avg	0.875	1.161	1.832	2.457	1.250	1.537	1.189	1.888	**4.435**	5.801	2.386	3.245	44.03	65.94	**37.91**	53.82
w/o Tri-E	DD	1.579	1.848	2.045	2.720	2.971	3.566	1.753	2.202	2.302	**3.359**	2.221	3.173	28.47	46.81	**31.43**	47.43
	Cor	0.883	1.130	1.910	2.491	**1.113**	**1.457**	1.724	2.424	6.741	8.115	2.770	3.678	39.61	58.03	45.73	62.89
	Avg	1.231	1.489	1.977	2.605	2.042	2.512	1.738	2.313	4.522	5.737	2.495	3.426	34.04	52.42	38.58	55.16
w/o MASA	DD	0.989	1.297	1.777	2.483	1.567	1.847	0.928	1.420	2.171	3.368	2.219	3.212	28.49	**46.80**	32.11	47.70
	Cor	0.846	1.226	**1.876**	2.475	1.615	2.010	1.559	**2.382**	6.794	8.243	2.513	3.315	38.58	55.10	44.97	61.88
	Avg	0.917	1.262	1.827	2.479	1.591	1.929	1.244	1.901	4.483	5.806	2.366	3.263	33.53	50.95	38.54	54.79
w/o CA&RB	DD	1.314	1.641	1.771	2.425	3.139	3.566	1.088	1.665	2.481	3.750	**2.177**	3.138	29.00	47.41	37.00	51.66
	Cor	0.886	1.133	2.029	2.555	1.487	1.985	1.623	2.386	6.889	8.391	2.511	3.330	37.74	**53.32**	43.79	60.48
	Avg	1.100	1.387	1.900	2.490	2.313	2.775	1.356	2.026	4.685	6.070	2.344	3.234	33.37	50.36	40.40	56.07

Table 3: Comparison of Stage II Outputs of the Original and Corrected Mechanistic Models.

Model		**DNA**		Transformer		FlowformerMH		iTransformer		MPFormer		GATv2		DGDL		TGCNS		RaDA		TimesNet		TimeXer		Umixer	
Metric		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HL	MM	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91	8.513	10.91
	Cor	5.417	7.199	5.445	7.175	5.449	7.180	5.424	7.129	5.418	7.129	5.419	7.133	5.414	7.131	5.416	7.136	**5.413**	7.126	5.426	7.142	5.419	**7.117**	5.440	7.148
P	MM	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5	124.2	214.5
	Cor	**77.55**	**131.2**	92.05	180.1	90.22	171.4	93.49	184.8	90.47	177.1	92.03	179.4	91.91	179.3	91.83	179.3	91.69	178.9	90.90	177.6	94.94	187.0	90.62	176.7

MHSA leads to a clear performance degradation. Specifically, the average MAE and RMSE across the eight datasets increase by 10.21% and 7.92%, respectively. In addition, we further evaluate the effectiveness of each component by removing them individually, including the Tri-Embedding (Tri-E), MASA, and the channel attention with ResBlocks (CA&RB). The results indicate that the full combination of Tri-E, MASA, and CA&RB achieves the most accurate predictions on nearly all datasets, while removing any single component consistently leads to degraded performance.

Stage II. The ablation study results for Stage II are summarized in Tab.4. These experimental results show that the model incorporating both the GCN-Embedding (GCN-E) and the gating mechanism (Gating) achieves the best performance. In contrast, removing either component results in varying degrees of performance degradation, demonstrating the effectiveness and necessity of each module.

Table 4: Ablation studys on Stage II.

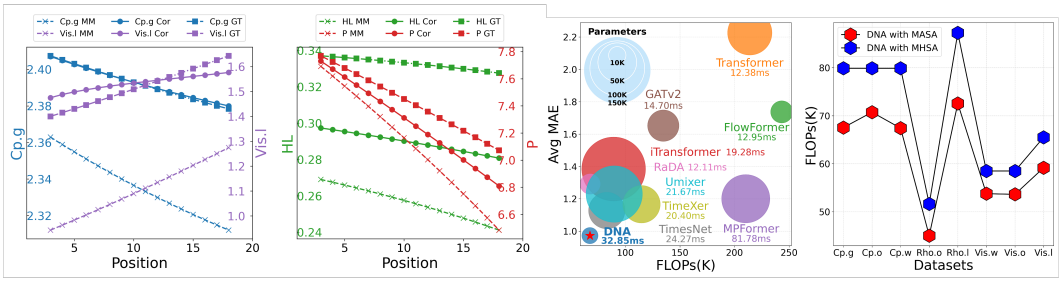
Stage II	HL		P	
	MAE	RMSE	MAE	RMSE
w/o GCN-E	1.612	2.726	4.509	7.641
w/o Gating	1.757	2.891	8.858	12.48
Ours	**1.556**	**2.677**	**3.017**	**5.424**

4.5 Analysis of Correction Function

The introduction of a unique interaction strategy, in which the correction function $\mathcal{F}$ is generated and fed back to the mechanistic model to improve its performance is one of the key contributions of this work. The correction functions considered in this work consist of three components: (1) an n order polynomial term with $n \in \{0, 1, 2, 3\}$; (2) a logarithmic term (with or without ln); and (3) weighting coefficients (with or without $gate$). We conducted extensive experiments to select the most suitable combination of the three components for each Stage I dataset, thereby determining a predefined function form. Under the DNA architecture, all baseline models were trained using the same predefined function forms to ensure a fair comparison. The predefined function form for each dataset and the correction functions obtained by our model are summarized in Tab.7.

4.6 Analysis of Gain Propagation Along the DNA Chains

We deploy the corrected mechanistic model $\mathcal{M}^I_{cor}$ obtained by applying DNA with each baseline in Stage I (green pipeline in Fig.1), keeping everything else unchanged, and repeat the experiment under identical operating conditions X, with one run for each baseline. The resulting Stage II outputs are reported in Tab.3, where MM denotes the outputs of the original mechanistic model and Cor those of the corrected model. As shown in Tab.1, under the DNA architecture, every baseline could substantially improve the predictive performance of $\mathcal{M}^I_{me}$, yielding $\mathcal{M}^I_{cor}$. Combined with Tab.3, when these $\mathcal{M}^I_{cor}$ models are deployed in Stage I, the final outputs are markedly improved even with the downstream model ($\mathcal{M}^{II}_{me}$) unchanged. This shows that, under DNA architecture, the



(a) (b) (c) (d)

Figure 3: (a,b) Visualization and efficiency comparison results. (c) Model complexity and efficiency comparison on the Cp.w dataset. (d) Computational cost comparison between MASA and MHSA across all Stage I datasets of the three-phase pipeline system

gains achieved upstream can be effectively propagated to downstream tasks. Specifically, our model achieves average MAE reductions of 36.37% and 37.56% for HL and P, respectively, and achieves the best performance on P, with a further 14.04% improvement in MAE over the second-best model. To intuitively demonstrate the improvement brought by this correction, we also visualize the results on 16 positions from a randomly selected case, as shown in Fig.3a-b.

4.7 Analysis of Model Complexity and Efficiency

Complexity. We compare the complexity of our model and other baselines in terms of total number of parameters and FLOPs required for a single forward inference. As shown in Fig.3c, under identical configurations, our model achieves the best predictive performance while also having the lowest total number of parameters and FLOPs among all models. This demonstrates that our model is highly lightweight. We further demonstrate the effectiveness of the proposed MASA in reducing computational cost, as shown in Fig.3d. Under strictly identical configurations, replacing MHSA with MASA leads to a clear reduction in the overall FLOPs across all Stage I datasets of the three-phase pipeline system, with an average reduction of 12.28%. **Efficiency.** We also evaluate efficiency by comparing the inference time required by different models to process a batch of samples. As shown in Fig.3c, although the experimental results indicate that DNA does not exhibit superior inference speed among these baselines, its favorable FLOPs performance suggests that DNA has the potential for efficient inference in theory. After careful investigation and analysis, we attribute this to the lack of an effective strategy for parallelizing MASA, which currently results in suboptimal inference efficiency. More experimental results and analyses are provided in the Appendix E.5.

5 Conclusion, Future Work and Limitations

In this work, we introduced a novel Double-Helix Hierarchical Deep Interaction Architecture (DNA) for industrial digital twin. The main idea is to enable effective interaction between mechanistic and data-driven models, ultimately building accurate models of complex industrial systems. A key objective of this work is to leverage data-driven models not only for accurate predictions, but also to assist in improving the performance of mechanistic models. This enhancement is of great practical and industrial value. We validate DNA through extensive experiments on the three-phase pipeline and acentric factor systems, comparing it with many competitive models. Results show DNA consistently outperforms baselines, and we believe DNA has the potential to contribute meaningfully to the advancement of industrial digital twins.

Other directions for future work are, in part, motivated by the limitations of this work: (1) DNA has the potential for efficient inference, but this potential remains unrealized. With the underlying cause identified, exploring how to parallelize MASA is a natural next step. (2) Currently, the forms of the correction functions are carefully designed in advance, and their end-to-end generation remains to be investigated. (3) DNA is limited to correcting mechanistic models formulated as explicit functions. We plan to extend it to mechanistic models governed by systems of differential equations. (4) Finally, we leave the exploration of applications in broader industrial scenarios to future work.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for [assist with translation]. We have not used generative AI tools for [generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results.] Additionally, we used generative AI tools for [edit a research paper to improve readability]. We have reviewed all AI-assisted work. [We used generative AI tools to polish and refine the language of the manuscript to improve its clarity and readability.]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

ETHICS STATEMENT

N/A

REPRODUCIBILITY STATEMENT

We provide all relevant code and datasets in the supplementary materials and detail the training configurations, hyperparameters, training procedures, and computational resources in the Appendix. These resources and details enable the reproduction of the experimental results presented in this work. More details are provided in the Appendix E.

REFERENCES

Obiechina I Abakporo. *General Thermodynamic Framework for Organic Rankine Cycles*. PhD thesis, Florida Agricultural and Mechanical University, 2021.

Luai M Al-Hadhrami, SM Shaahid, Lukman O Tunde, and A Al-Sarkhi. Experimental study on the flow regimes and pressure gradients of air-oil-water three-phase flow in horizontal pipes. *The Scientific World Journal*, 2014(1):810527, 2014.

Kamal IM Al-Malah. *Aspen plus: chemical engineering applications*. John Wiley & Sons, 2022.

Abolhasan Ameri. Predictive neural network model and empirical equations for the physico-chemical properties and solvent characteristics of potassium carbonate solutions in carbon capture processes. *Frontiers of Chemical Science and Engineering*, 19(4):1–26, 2025.

Dale H Beggs and James P Brill. A study of two-phase flow in inclined pipes. *Journal of Petroleum technology*, 25(05):607–617, 1973.

Sumit K Bishnu, Sabla Y Alnouri, and Dhabia M Al-Mohannadi. Computational applications using data driven modeling in process systems: A review. *Digital Chemical Engineering*, 8:100111, 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.

William Bradley, Jinhyeun Kim, Zachary Kilwein, Logan Blakely, Michael Eydenberg, Jordan Jalvin, Carl Laird, and Fani Boukouvala. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 166:107898, 2022.

Gabriela Souza Chaves, Hamidreza Karami, Virgilio Jose Martins Ferreira Filho, and Bruno Ferreira Vieira. A comparative study on the performance of multiphase flow models against offshore field production data. *Journal of Petroleum Science and Engineering*, 208:109762, 2022.

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

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.

Montserrat Dalmau, Nataša Atanasova, Sara Gabarrón, Ignasi Rodriguez-Roda, and Joaquim Comas. Comparison of a deterministic and a data driven model to describe mbr fouling. *Chemical Engineering Journal*, 260:300–308, 2015.

Wayne C Edmister and BK Lee. Applied hydrocarbon thermodynamics. vol. 1. 1984.

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.

Oliver J Fisher, Nicholas J Watson, Josep E Escrig, Rob Witt, Laura Porcu, Darren Bacon, Martin Rigley, and Rachel L Gomes. Considerations, challenges and opportunities when developing data-driven models for process manufacturing systems. *Computers & Chemical Engineering*, 140:106881, 2020.

Bin Gao and Yuekuan Zhou. A hybrid physical-data driven model on proton exchange membrane fuel cells for cooling, heating and power supply with gas heat recovery. *Applied Energy*, 411: 127588, 2026.

Shan Gao, Jicheng Xu, Feng Fu, Zhenhua Huang, Jean Francois Demonceau, and Jie Yang. Mechanistic-data-driven modeling of multi-material composite columns: Toward intelligent lightweight design. *Engineering Structures*, 352:122134, 2026.

Debanjan Ghosh, Jesus Moreira, and Prashant Mhaskar. Model predictive control embedding a parallel hybrid modeling strategy. *Industrial & Engineering Chemistry Research*, 60(6):2547–2562, 2021.

Jarka Glassey and Moritz Von Stosch. *Hybrid modeling in process industries*. CRC Press, 2018.

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*, pp. 770–778, 2016.

David Mautner Himmelblau and James B Riggs. *Basic principles and calculations in chemical engineering*. FT press, 2012.

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.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Aaron Klug. The discovery of the dna double helix. *Journal of molecular biology*, 335(1):3–26, 2004.

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.

Milo D Koretsky. *Engineering and chemical thermodynamics*. John Wiley & Sons, 2012.

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

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.

Yuancheng Lin, Junlong Tang, Jing Guo, Shidong Wu, and Zheng Li. Advancing ai-enabled techniques in energy system modeling: a review of data-driven, mechanism-driven, and hybrid modeling approaches. *Energies*, 18(4):845, 2025.

Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of manufacturing systems*, 58:346–361, 2021.

Yanbing Liu, Xinglin Yang, Jiaqi Zhang, and Zongyuan Zhu. Process simulation of preparing biochar by biomass pyrolysis via aspen plus and its economic evaluation. *Waste and Biomass Valorization*, pp. 1–14, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

Xiang Ma, Xuemei Li, Lexin Fang, Tianlong Zhao, and Caiming Zhang. U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting, 2024. URL https://arxiv.org/abs/2401.02236.

Samruddhi Mane, Ritesh R Dhote, Adityakumar Sinha, and Raja Thirumalaiswamy. Digital twin in the chemical industry: A review. *Digital Twins and Applications*, 1(2):118–130, 2024.

Seyed Soheil Mansouri, Abhishek Sivaram, Christopher J Savoie, and Rafiqul Gani. Models, modeling and model-based systems in the era of computers, machine learning and ai. *Computers & Chemical Engineering*, 194:108957, 2025.

Vahid Naghashi, Mounir Boukadoum, and Abdoulaye Banire Diallo. A multiscale model for multivariate time series forecasting. *Scientific Reports*, 15(1):1565, 2025.

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)*, pp. 807–814, 2010.

Jagan Nath, SS Das, and ML Yadava. On the choice of acentric factor. *Industrial & Engineering Chemistry Fundamentals*, 15(3):223–225, 1976.

Chibuzo Cosmas Nwanwe and Ugochukwu Ilozurike Duru. Comparison and performance analysis of models for predicting multiphase flow behaviours in wellbores. *International Journal of Petroleum and Geoscience Engineering*, 2022:1–20, 2022.

Pratyush Kumar Pal, Abhiram Hens, Narottam Behera, and Sandip Kumar Lahiri. Digital twins: Transforming the chemical process industry—a review. *The Canadian Journal of Chemical Engineering*, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Luisa Peterson, Ion Victor Gosea, Peter Benner, and Kai Sundmacher. Digital twins in process engineering: An overview on computational and numerical methods. *Computers & Chemical Engineering*, 193:108917, 2025.

Ward Quaghebeur, Elena Torfs, Bernard De Baets, and Ingmar Nopens. Hybrid differential equations: Integrating mechanistic and data-driven techniques for modelling of water systems. *Water Research*, 213:118166, 2022.

MR Riazi. *Characterization and properties of petroleum fractions*, volume 50. ASTM international, 2005.

Artur M Schweidtmann, Dongda Zhang, and Moritz Von Stosch. A review and perspective on hybrid modeling methodologies. *Digital Chemical Engineering*, 10:100136, 2024.

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.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

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

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

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

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)*, pp. 3–19, 2018.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022a.

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows, 2022b. URL https://arxiv.org/abs/2202.06258.

Haoran Yang, Yinan Zhang, Qunshan He, Yuqi Ye, Jing Zhao, and Wenhai Wang. Ultramodel: a modeling paradigm for industrial objects. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 7876–7884, 2025.

Muhammad Waqas Yaqub, Ramasamy Marappagounder, Risza Rusli, Reddy Prasad DM, and Rajashekhar Pendyala. Review on gas–liquid–liquid three–phase flow patterns, pressure drop, and liquid holdup in pipelines. *Chemical Engineering Research and Design*, 159:505–528, 2020.

Hong-Quan Zhang and Cem Sarica. Unified modeling of gas/oil/water-pipe flow-basic approaches and preliminary validation. *SPE Projects, Facilities & Construction*, 1(02):1–7, 2006.

Jun Zhang, Zhi-zhong Mao, Run-da Jia, and Da-kuo He. Real time optimization based on a serial hybrid model for gold cyanidation leaching process. *Minerals Engineering*, 70:250–263, 2015.

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.

Denghong Zhou and Yilin Fan. Modeling pressure gradient of gas–oil–water three-phase flow in horizontal pipes downstream of restrictions. *Energies*, 17(12):2849, 2024.

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.

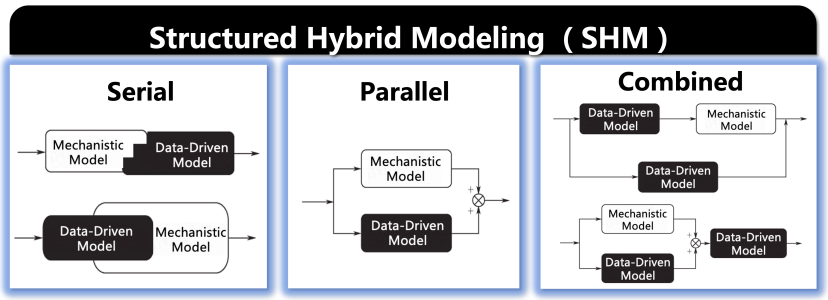


Figure 4: Three common paradigms of structured hybrid modeling.

SUPPLEMENTAL MATERIAL

This supplemental material is divided into the following five appendices:

- **Appendix A**: Related works on existing structured hybrid modeling.
- **Appendix B**: Additional details of the proposed data-driven modeling frameworks in Stages I and II in the Methodology section.
- **Appendix C**: Details of the three-phase pipeline system, including an introduction to three-phase pipeline, dataset acquisition, mechanistic models, and the data flow within the system.
- **Appendix D**: Details of the acentric factor system, including an introduction to acentric factor, dataset acquisition and the mechanistic models.
- **Appendix E**: Additional details on the Experiment section.

A RELATED WORK

Industrial modeling can be broadly categorized into three paradigms: mechanistic modeling, data-driven modeling, and hybrid modeling Glassey & Von Stosch (2018); Fisher et al. (2020); Lin et al. (2025). Here, we focus on hybrid modeling methods.

Serial architecture. This architecture is particularly useful when a certain dynamics is not accurately described by the mechanistic model. For example, Zhang et al. Zhang et al. (2015) developed a BP ANN to estimate unknown kinetic reaction rates in a mechanistic model, and then fed the predicted results into a subsequent mechanistic component to construct a hybrid model for the gold cyanidation leaching process. Similarly, Dalmau et al. Dalmau et al. (2015) modeled the fouling process in a membrane bioreactor using a tree-based model, which was subsequently integrated into a mechanistic framework. Gao et al. Gao & Zhou (2026) first used an MLP to estimate a subset of process parameters in the mechanistic model, and then fed these estimates into a downstream mechanistic component for computation.

Parallel architecture. This architecture is typically used to model the residual error of mechanistic models in a post-hoc correction manner. For instance, Gao et al. Gao et al. (2026) developed an accurate model for multi-material composite columns by first using XGBoost to predict the residual between mechanistic predictions and ground truth, and then adding the estimated residual to the theoretical output of the mechanistic model. Similarly, Ghosh et al. Ghosh et al. (2021) designed a data-driven residual model to estimate the error of a mechanistic model, which was then added to its output to enable model predictive control in a seeded batch crystallization process.

Combined architecture. Combined architecture integrates both serial and parallel coupling strategies, thus combining their respective advantages. For example, in the Neural ODE model proposed by Quaghebeur Quaghebeur et al. (2022) for a water resource recovery facility, the data-driven component is first coupled with the mechanistic component in a parallel manner, forming an intermediate hybrid module, which is then integrated into a subsequent serial computational structure. Chen et

al. Chen & Ierapetritou (2020) systematically explored serial, parallel, and hybrid coupling strategies for modeling a continuous stirred-tank reactor (CSTR) and two pharmaceutical case studies. By comparing the predictive performance of these three hybrid paradigms, they demonstrate that the hybrid structure achieves superior robustness against disturbances as well as better extrapolation capability.

Although these hybrid modeling approaches can effectively integrate mechanistic models with data-driven counterparts and leverage the strengths of both components, they largely rely on a simple stacking of the mechanistic and data-driven models where information flows in a unidirectional manner without enabling bidirectional interaction. Moreover, their one-shot modeling strategies often become inadequate when dealing with highly complex industrial systems.

B METHODOLOGY

B.1 STAGE I

B.1.1 TRI-EMBEDDING

In this paper, we propose a novel embedding method, tripartite-embedding (Tri-Embedding), which generates high-dimensional representations for low-dimensional variables. This method is particularly suitable for scenarios where the mechanistic model underlying the target system exhibits a complex topological structure, as its embedding generation involves only a few nonlinear projections, resulting in minimal computational overhead. The Tri-Embedding process consists of three components, each detailed as follows:

(i) **Variable Type Encoding ($\mathcal{A}_1$).** We first assign a real-valued type token v to each variable according to its physical meaning. Variables sharing the same physical meaning receive the same token. For example, in the $Cp.o$ dataset, x_2 denotes the boiling point (Tb) of the substance and x_3 represents the actual temperature (T); both belong to the temperature category and are therefore assigned the same value. In contrast, variables of different types, such as x_2 for the boiling point (Tb) and x_1 for the specific gravity (SG), are assigned distinct values. Subsequently, these type tokens are passed through an embedding layer (EL) followed by layer normalization (LN) to produce the first representation, $e_i^{(1)} = LN(EL(v))$.

(ii) **Nonlinear Projection ($\mathcal{A}_2$).** To enrich the representational capacity of each variable, we apply a nonlinear projection that maps the original one-dimensional real value x_i or z_i into a higher-dimensional latent space. This high-dimensional embedding is then processed with layer normalization to yield the second representation, $e_i^{(2)} = LN(\sigma(Linear(x_i)))$.

(iii) **Variable Context Encoding ($\mathcal{A}_3$).** We abstract the contextual information of each variable within the topological graph as a 4-element array, $context_info : [level, IO, in_degree, out_degree]$, where $level$ denotes the depth level of the variable in the graph, IO indicates the variable type {input(1), output(2), intermediate(0), or both input and output(3)}, in_degree and out_degree represent the in-degree and out-degree of the node, respectively. Each of the four elements is independently embedded using separate embedding layers followed by layer normalization. The resulting vectors are concatenated to form the third part of the representation, $e_i^{(3)} = CAT(LN(EL(context_info)))$.

After obtaining the output of each component, we concatenate all three parts to generate the full high-dimensional representation of the variable i, $e_i = CAT(e_i^{(1)}, e_i^{(2)}, e_i^{(3)})$.

B.1.2 RESBLOCKS

The design of the ResBlocks is illustrated in Fig.5. We first apply a 1×1 convolution to the input $E_{exp}^{I'} \in \mathbb{R}^{n \times d_m \times 1}$ to expand the channel dimension, yielding an intermediate representation $\mathcal{Z} \in \mathbb{R}^{c \times d_m \times 1}$. The resulting features are then processed by three consecutive ResBlocks ($\mathcal{P}(\cdot)$), producing the output $\mathcal{Z}'$. Finally, another 1×1 convolution is applied to project the channel dimension back to that of the input, resulting in the final output H.

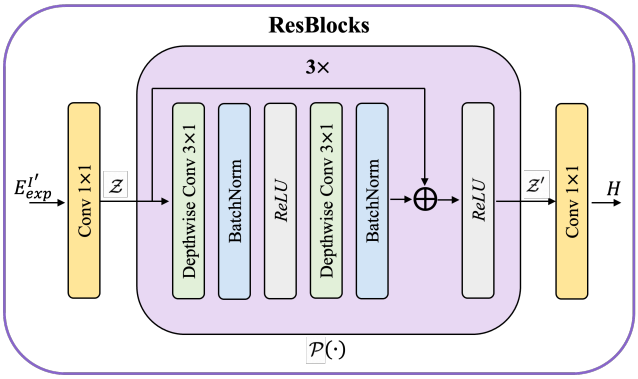


Figure 5: Diagram of ResBlocks.

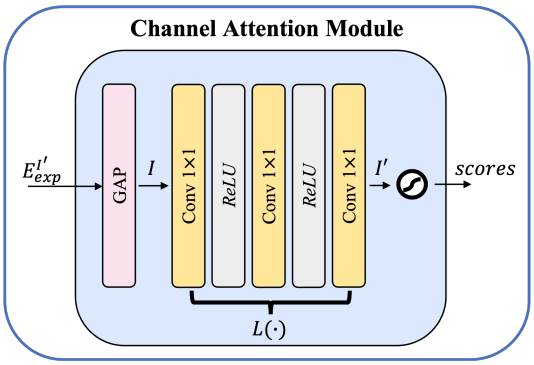


Figure 6: Diagram of Channel Attention Module.

Notably, we employ depthwise convolutions instead of standard convolutions within each ResBlock to reduce the number of model parameters and the computational cost. Experimental results indicate that this replacement leads to no noticeable degradation in model performance. The overall process can be formulated as follows:

$$\mathcal{Z} = Conv_{1\times1}(E^{I'}_{exp}) \in \mathbb{R}^{c\times d_m\times 1}, n \to c, \tag{8}$$

$$\mathcal{Z}' = \mathcal{P}(\mathcal{Z}) \in \mathbb{R}^{c\times d_m\times 1}, c \to c, \tag{9}$$

$$H = Conv_{1\times1}(\mathcal{Z}') \in \mathbb{R}^{n\times d_m\times 1}, c \to n, \tag{10}$$

where c denotes the number of channels after expansion, and we set it to 32 in this work. If we denote the entire process by $\mathcal{R}(\cdot)$, then Eqs. 8-10 correspond exactly to Eq. 2 in the main text.

B.1.3 Channel Attention Module

The channel attention module is illustrated in Fig.6, and its calculation process is described as follows:

$$I = GAP(E^{I'}_{exp}, dim = (-2, -1)) \in \mathbb{R}^{n\times1\times1}, \tag{11}$$

$$I' = L(I) \in \mathbb{R}^{n\times1\times1}, n \to d_m \to \frac{d_m}{2} \to n, \tag{12}$$

$$scores = Sigmoid(I') \in \mathbb{R}^{n\times1\times1}, \tag{13}$$

where n is the total number of variables in the input, and d_m is the embedding dimension. The steps described in Eqs. 11-13 correspond exactly to those in Eq. 3 of the main paper.

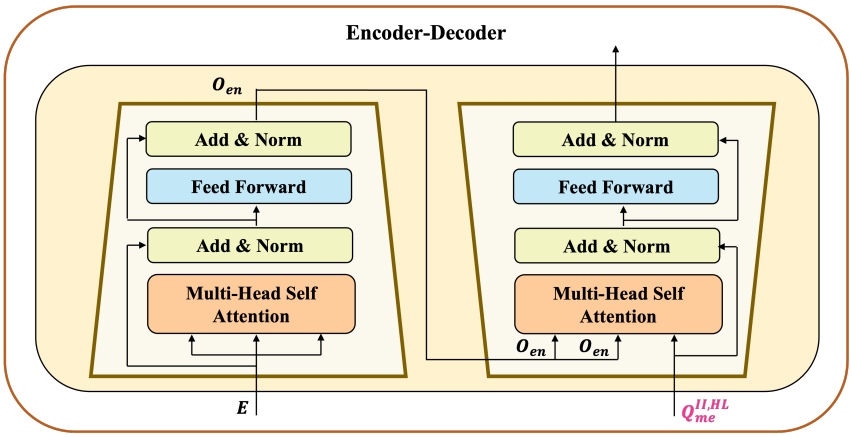


Figure 7: Diagram of Encoder-Decoder Structure in Stage II.

B.2 Stage II

B.2.1 Encoder-Decoder Structure

The encoder–decoder structure in Stage II is illustrated in Fig.7. We largely follow the Transformer backbone, except that positional encoding is removed, and only the components after the cross-attention module in the decoder are retained.

B.3 Model Optimization

Stage I of DNA is actually a multi-task learning problem. The network simultaneously outputs the predicted target Y_{dd}^I and a set of coefficients $coefs$, which are used to construct a correction function $\mathcal{F}(\cdot)$. This function refines the original output Y_{me}^I of the mechanistic model and produces a corrected result, $Y_{cor}^I = \mathcal{F}(Y_{me}^I; coefs)$. Therefore, to train this model well, it is necessary to minimize the following two loss functions at the same time:

$$\mathcal{L}_1 = \frac{1}{N}\sum \left\| Y_{dd}^I - Y_{true}^I \right\|^2, \tag{14}$$

$$\mathcal{L}_2 = \frac{1}{N}\sum \left\| Y_{cor}^I - Y_{true}^I \right\|^2, \tag{15}$$

where, N represents the total number of samples. Since different tasks vary in learning difficulty, simply summing the two loss terms may lead to training imbalance, where one task dominates the optimization process and achieves significantly better performance at the expense of the other Vandenhende et al. (2021). To address this issue, we follow the work of Kendall et al. (2018) and introduce task-dependent uncertainty to adaptively weight the losses. The resulting optimization objective is formulated as follows:

$$w^* = \arg\min_w [\frac{1}{2\sigma_1^2}\mathcal{L}_1(w) + \frac{1}{2\sigma_2^2}\mathcal{L}_2(w) + ln(\sigma_1\sigma_2)], \tag{16}$$

here, w represents the set of all parameters in the data-driven model; σ_1 and σ_2 represent the task-dependent uncertainties of the two tasks. They are learnable parameters that are updated dynamically during training. However, when σ_1 or σ_2 becomes too small, the loss term can become negative, potentially causing training instability or failure. In this case, we adopt a more pragmatic alternative by assigning a weight ($\alpha = 0.5$) to one loss term. The revised optimization objective is defined as follows:

$$w^* = \arg\min_w [\mathcal{L}_1(w) + \alpha\mathcal{L}_2(w)]. \tag{17}$$

Stage II of DNA does not involve correction function generation, model in this phase is trained using only Eq. 14.

Table 5: Description of all the datasets. For certain datasets, Z denotes intermediate variables specific to each case; although some symbols may appear identical across datasets, their meanings and computational definitions are entirely distinct.

	Modeled object $\mathcal{O}$	X	Z	y_{me}, y_{true}	Unit
Stage I	Cp.g	T, Mw_g	z_1, z_2, z_3, z_4, z_5	$Cp.g_{me}, Cp.g_{true}$	$J/(kg \cdot K)$
	Cp.o	T, SG_o, Tb_o	K, A_1, A_2, A_3, CpR	$Cp.o_{me}, Cp.o_{true}$	$J/(kg \cdot K)$
	Cp.w	T, Mw_w	z_1, z_2, z_3, z_4, z_5	$Cp.w_{me}, Cp.w_{true}$	$J/(kg \cdot K)$
	Rho.o	T, SG_o	$alpha$	$Rho.o_{me}, Rho.o_{true}$	kg/m^3
	Rho.l	$T, P, Mw_{og}, Mw_l, x_{mol}[2], Rho.w$	$Rho.og, z_1$	$Rho.l_{me}, Rho.l_{true}$	kg/m^3
	Vis.w	T	z_1, z_2, z_3	$Vis.w_{me}, Vis.w_{true}$	$N \cdot s/mm^2$
	Vis.o	T, SG_o	API, n	$Vis.o_{me}, Vis.o_{true}$	$N \cdot s/mm^2$
	Vis.l	$T, x_{mol}[2]$	Vis_g, Vis_w, Vis_o	$Vis.l_{me}, Vis.l_{true}$	$N \cdot s/mm^2$
II	HL	——	$Q_g, Q_l, R_l, v, N_{Fr}, HL_{me}$	HL_{me}, HL_{true}	%
	P	——	$Rho_g, Rho_l, Mw_l, phi, v_{sg}, z_1, z_2, v, Rho, z_3, M_{flow}, P_{me}$	P_{me}, P_{true}	kPa

C Industrial System: Three-Phase Pipeline

C.1 Three-Phase Pipeline

In the petrochemical industries, the three-phase pipeline typically refers to a multiphase flow system capable of simultaneously transporting crude oil, natural gas, and water Yaqub et al. (2020). Such systems serve as critical links between oil production sites, processing facilities, and transportation infrastructure, and are widely applied in scenarios such as field development, gathering and distribution, and long-distance pipeline transportation. By enabling the co-transport of oil, gas, and water without prior separation, these systems can significantly reduce the infrastructure and energy costs associated with phase separation and transportation, while enhancing production efficiency and flexibility in resource allocation Zhang & Sarica (2006).

The mechanistic modeling of three-phase pipelines typically relies on fundamental physical conservation laws, such as the conservation of mass, momentum, and energy. These conservation equations are often coupled with empirical formulas to describe interphase interactions as well as flow-related factors such as frictional losses. During the computation process, it is essential to rigorously define the properties of each phase, such as density, viscosity, and specific heat capacity, and their dependencies on temperature and pressure. However, the composition of three-phase fluids is often complex, and the formation of emulsions (e.g., oil-in-water or water-in-oil) further complicates the system. These phenomena make it difficult to accurately characterize the relevant physical properties Zhou & Fan (2024). Moreover, variations in pipeline geometry and operating conditions can induce changes in flow patterns, resulting in strongly nonlinear and coupled behaviors, such as interphase interactions, energy exchange, and phase transitions. These complexities hinder the formulation and solution of core conservation equations for momentum, mass, and energy. To ensure tractability, many empirical formulas, based on specific experiments or simplified assumptions, are introduced, but their limited generalizability makes it difficult to adapt to the diverse conditions of real-world operations Yaqub et al. (2020).

In summary, the difficulty in accurately characterizing physical properties, coupled with the structural complexity of the systems, often leads to prohibitively high modeling costs for constructing mechanistic models of three-phase pipelines, yet satisfactory performance remains elusive Al-Hadhrami et al. (2014).

C.2 Datasets

As described in the main text (Section 2.1), we constructed a mechanistic model of the three-phase pipeline based on the Beggs–Brill theory Beggs & Brill (1973); Chaves et al. (2022) and ran it under the same operating conditions X as those of the real-world three-phase pipeline from which our data were collected, in order to obtain the mechanistic predictions (Y^I_{me}, Y^{II}_{me}) together with a set of key intermediate variables (Z^I). Using these data, we formed eight datasets for Stage I and two datasets for Stage II.

In the three-phase pipeline system, we select eight physical properties as Stage I modeled objects: gas-phase specific heat capacity **Cp.g**, oil-phase specific heat capacity **Cp.o**, water-phase specific

heat capacity **Cp.w**, oil-phase density **Rho.o**, liquid-phase density **Rho.l**, water-phase viscosity **Vis.w**, oil-phase viscosity **Vis.o**, and liquid-phase viscosity **Vis.l**. In Stage II, we further model two downstream objects—liquid holdup **HL** and pressure **P** at the pipeline outlet—**which are derived from the aforementioned physical properties through additional calculations**. As a result, this study involves a total of ten modeled objects, each associated with a distinct dataset. **It is important to note that although these datasets originate from the same industrial system, they differ significantly in terms of input-output variables and data distributions.** Detailed information about these datasets is provided in Tab.5.

C.3 Mechanistic Model

C.3.1 Introduction

In three-phase pipeline, the dominant flowing media consist of **gas**, **water**, and **oil**. The oil and water phases are immiscible, and no mass transfer occurs between the gas and water phases. The dissolution and separation processes between the oil and gas phases are assumed to be instantaneous. The definitions of gas, water, and oil used in this work are given below:

- The gas phase is composed of methane CH_4.
- The water phase consists of water H_2O.
- The oil phase is represented by a pseudo-component of representative oil (with a specific gravity $SG_o = 0.84$ and a boiling point $Tb_o = 569.65K$) mixed with a small amount of dissolved methane.
- The liquid phase comprises both the oil and water phases.

Consequently, during the Stage I computation, the gas-phase physical property ($Cp.g$) is taken as that of methane; the water-phase physical properties ($Cp.w, Vis.w$) correspond to those of water; the oil-phase physical properties ($Cp.o, Rho.o, Vis.o$) are calculated as the mixture properties of the representative oil pseudo-component and dissolved methane; and the liquid-phase physical properties ($Rho.l, Vis.l$) are defined as the properties of the liquid-phase mixture.

C.3.2 Mechanistic Model of **Cp.g**

The mechanistic model of Cp.g can be expressed as follows:

$$Cp.g = [VapCp[1] + VapCp[2] \times (\frac{VapCp[3]}{T \times \sin \frac{VapCp[3]}{T}})^2 + VapCp[4] \times (\frac{VapCp[5]}{T \times \cos \frac{VapCp[5]}{T}})^2]/Mw_g \tag{18}$$

where T denotes the actual temperature, $\{VapCp[i]\}_{i=1}^{5}$ represents the specific heat parameter of the gas phase, and Mw_g is the relative molecular weight of the gas phase (methane). The intermediate variables in this dataset are defined as follows:

$$z_1 = T \times \sin \frac{VapCp[3]}{T}, \tag{19}$$

$$z_2 = T \times \cos \frac{VapCp[5]}{T}, \tag{20}$$

$$z_3 = VapCp[2] \times (\frac{VapCp[3]}{z_1})^2, \tag{21}$$

$$z_4 = VapCp[4] \times (\frac{VapCp[5]}{z_2})^2, \tag{22}$$

$$z_5 = VapCp[1] + z_3 + z_4. \tag{23}$$

C.3.3 MECHANISTIC MODEL OF **CP.O**

The mechanistic model of Cp.o can be expressed as follows:

$$Cp.o = 4.1868 \times (A_1 + A_2 \times TR + A_3 \times TR^2), \tag{24}$$

where A_1, A_2, A_3, TR are fitting parameters used in engineering practice with no explicit physical meaning. The intermediate variables in this dataset are defined as follows:

$$A_1 = -1.17126 + (0.023722 + 0.024907 \times SG_o) \times K + \frac{1.14982 - 0.046535 \times K}{SG_o}, \tag{25}$$

$$A_2 = 10^{-4} \times (1 + 0.82463 \times K) \times (1.12172 - \frac{0.27634}{SG_o}), \tag{26}$$

$$A_3 = -10^{-8} \times (1 + 0.82463 \times K) \times (2.9027 - \frac{0.70958}{SG_o}), \tag{27}$$

$$K = \frac{(1.8 \times Tb_o)^{1/3}}{SG_o}, \tag{28}$$

$$TR = 1.8 \times T, \tag{29}$$

$$CpR = A_1 + A_2 \times TR + A_3 \times TR^2, \tag{30}$$

where K, SG_o, Tb_o denote the characterization factor, specific gravity, and boiling point of the pseudo-component of representative oil, respectively. The correspondence between these variables and those in Fig.1 of the main paper is as follows: $\{x_1, x_2, x_3, z_1, z_2, z_3, z_4, z_5\}$ in Fig. 1 correspond to $\{SG, Tb_o, T, K, A_1, A_2, A_3\}$, respectively.

C.3.4 MECHANISTIC MODEL OF **CP.W**

The mechanistic model of Cp.w can be expressed as follows:

$$Cp.w = [LiqCp[1] + LiqCp[2] \times T + LiqCp[3] \times T^2 + LiqCp[4] \times T^3 + LiqCp[5] \times T^4]/Mw_w, \tag{31}$$

where $\{LiqCp[i]\}_{i=1}^{5}$ represents the specific heat parameter of the water phase. The intermediate variables in this dataset are defined as follows:

$$z_1 = LiqCp[2] \times T, \tag{32}$$

$$z_2 = LiqCp[3] \times T^2, \tag{33}$$

$$z_3 = LiqCp[4] \times T^3, \tag{34}$$

$$z_4 = LiqCp[5] \times T^4, \tag{35}$$

$$z_5 = LiqCp[1] + z_1 + z_2 + z_3 + z_4. \tag{36}$$

C.3.5 MECHANISTIC MODEL OF **RHO.O**

The mechanistic model of Rho.o can be expressed as follows:

$$Rho.o = \frac{SG_o \times 1000}{1 + alpha \times (T - 20)}, \tag{37}$$

$$alpha = (3.083 - 2.638 \times SG_0) \times 10^{-3} \tag{38}$$

C.3.6 MECHANISTIC MODEL OF **RHO.L**

The mechanistic model of Rho.l can be expressed as follows:

$$Rho.l = \frac{1}{\frac{x_{mol}[2]}{Rho.w} + \frac{1-x_{mol}[2]}{Mw_{og} \times Rho.og}} \times Mw_l, \tag{39}$$

where $Mw_{og}, Mw_l, x_{mol}[2], Rho.w, Rho.og$ denote the relative molecular weight of the oil–gas mixture, the relative molecular weight of the liquid phase, the molar fraction of the water phase in the liquid phase, the density of the water phase, and the density of the oil-gas mixture respectively. The computation of $Rho.og$ and another intermediate variable z_1 is given as follows:

$$Rho.og = \frac{Rho.o + Rs \times (Rs \times (0.00379 \times do - 0.00393) - 4.08779 \times do + 4.43818)1.29}{0.972 + 0.000147 \times (5.62 \times Rs \times (\frac{dg}{do})^{0.5} + 2.25 \times T + 40)^{1.175}}, \tag{40}$$

$$Rs = 2.2 \times dg \times (\frac{P}{10} \times 10^{(\frac{1.77}{do} - 0.001638 \times T - 1.67)})^{1.205}, \tag{41}$$

$$do = \frac{Rho.o}{Rho.w}(T = 288.15K), \tag{42}$$

$$dg = 0.86, \tag{43}$$

$$z_1 = \frac{1}{\frac{x_{mol}[2]}{Rho.w} + \frac{1-x_{mol}[2]}{Mw_{og} \times Rho.og}}, \tag{44}$$

where T, P, R_s, d_g, d_o denote the actual temperature, actual pressure, the gas–oil ratio of oil (unit: $\frac{m^3}{m^3}$), the relative density of methane with respect to air, and the oil–water density ratio at a temperature of 288.15 K, respectively.

C.3.7 MECHANISTIC MODEL OF **VIS.W**

The mechanistic model of Vis.w can be expressed as follows:

$$Vis.w = \exp\left(LiqVis[1] + \frac{LiqVis[2]}{T} + LiqVis[3] \times ln(T) + LiqVis[4] \times T^{LiqVis[5]}\right), \tag{45}$$

where $\{LiqVis[i]\}_{i=1}^{5}$ represents liquid-phase viscosity parameter. The intermediate variables in this dataset are defined as follows:

$$z_1 = \frac{LiqVis[2]}{T}, \tag{46}$$

$$z_2 = LiqVis[3] \times ln(T), \tag{47}$$

$$z_3 = LiqVis[4] \times T^{LiqVis[5]}. \tag{48}$$

C.3.8 MECHANISTIC MODEL OF **VIS.O**

The mechanistic model of Vis.o can be expressed as follows:

$$Vis.o = 3.141 \times 10^7 \times |1.8 \times T - 460|^{-3.444} \times (log_{10}(API))^n, \tag{49}$$

API is another representation of the specific gravity of the pseudo-component of representative oil, and n is a fitting parameter

$$n = 10.313 \times log_{10}(|1.8 \times T - 460|) - 36.447, \tag{50}$$

$$API = \frac{141.5}{SG_o} - 131.5, \tag{51}$$

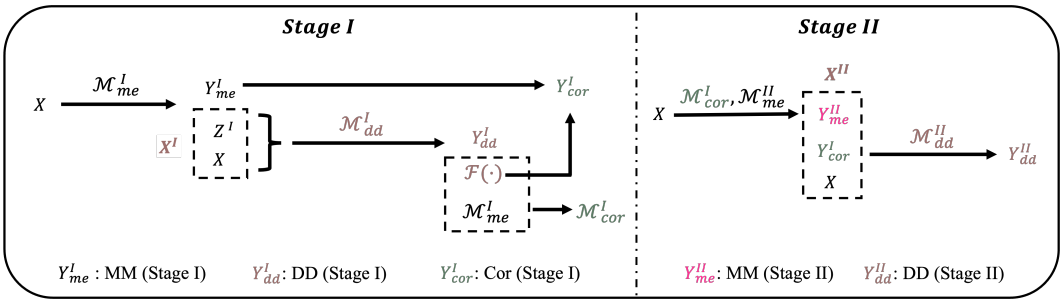


Figure 8: Data flow in this work.

C.3.9 MECHANISTIC MODEL OF VIS.L

The mechanistic model of Vis.l can be expressed as follows:

$$Vis.l = x_{mol}[1] \times Vis.g + x_{mol}[2] \times Vis.w + x_{mol}[3] \times Vis.o \tag{52}$$

where $x_{mol}[1], x_{mol}[2], x_{mol}[3]$ denote the molar fractions of the dissolved gas, water phase, and oil phase in the liquid phase, respectively.

C.4 THE DATA FLOW IN THREE-PHASE PIPELINE

In this work, the modeling targets in Stage II include the liquid holdup (HL) in the three-phase pipeline and the pressure (P) at the pipeline outlet. Mechanistically, these variables are computed based on the physical properties obtained in Stage I through further calculations. To present the logic of this work more clearly and intuitively, the complete data flow is illustrated in Fig.8.

In Stage I, the operational conditions of the three-phase pipeline collected from the real world (e.g., T_{inlet}, P_{inlet}, HL_{inlet}, etc.) serve as the raw inputs X to our model. We then run the mechanistic model $\mathcal{M}^I_{me}$ under exactly the same operating conditions to obtain the mechanistic outputs Y^I_{me} and Z^I. The set $X^I : \{X, Z^I\}$ is subsequently used as input to the Stage I data-driven model $\mathcal{M}^I_{dd}$, producing the data-driven outputs Y^I_{dd} and a set of correction formulas $\mathcal{F}(\cdot)$. Applying $\mathcal{F}(\cdot)$ to the original mechanistic model $\mathcal{M}^I_{me}$ yields the Stage I corrected mechanistic model $\mathcal{M}^I_{cor}$. Y^I_{me} is corrected by the correction functions to obtain Y^I_{cor}, which is more accurate than Y^I_{me}.

In Stage II, we run $\mathcal{M}^I_{cor}, \mathcal{M}^{II}_{me}$ under the same real-world operating conditions X, obtaining the outputs Y^{II}_{me}. Then, the set $X^{II} : \{Y^{II}_{me}, Y^I_{cor}, X\}$ are used as inputs to the Stage II data-driven model $\mathcal{M}^{II}_{dd}$ to obtain the predictions Y^{II}_{dd}.

This completes the full data flow in this work. In this process, Y^I_{me}, Y^I_{dd}, and Y^I_{cor} correspond to the Stage I MM, DD, and Cor in Tab. 1 of the main paper; while Y^{II}_{me} and Y^{II}_{dd} correspond to the Stage II MM and DD in the same table.

D INDUSTRIAL SYSTEM: ACENTRIC FACTOR

D.1 ACENTRIC FACTOR AND THE MECHANISTIC MODEL

The acentric factor (ω) is an unmeasurable dimensionless parameter used to describe the extent to which real gas deviate from ideal gas behavior. It plays a significant role in thermodynamics and fluid mechanics Biswas et al. (2023). Although there are numerous methods for calculating the acentric factor, such as the Edmister method Edmister & Lee (1984), Nath method Nath et al. (1976), and Lee-Kesler method Lee & Kesler (1975), these mechanistic formulas are derived from empirical data and thus have limitations in their applicability. When the values fall within the range for which these formulas are valid, the deviations are minimal; however, outside this range, the deviations can be significant Abakporo (2021). Among all these methods, the Lee-Kesler method is the most commonly used due to its simple structure. Therefore, we adopt the Lee-Kesler method as

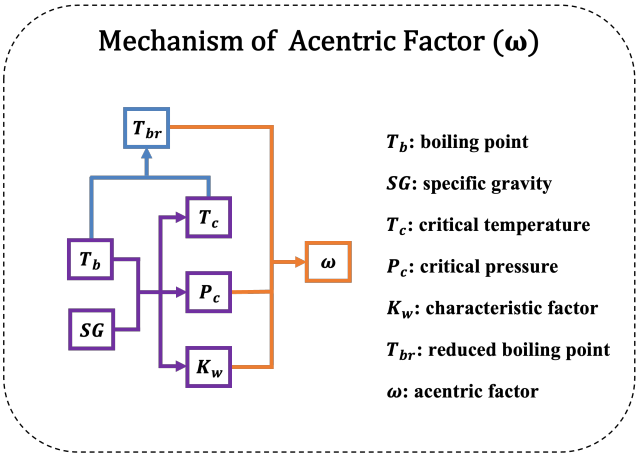


Figure 9: The mechanism of acentric factor.

the mechanistic model for the acentric factor. The Lee-Kesler method is presented as follows:

$$\psi = -\ln\left(\frac{P_c}{1.01325} - 5.92714 + \frac{6.09648}{T_{br}} + 1.28862 \times \ln T_{br} - 0.169347 \times T_{br}^6\right), \tag{53}$$

$$\phi = 15.2518 - \frac{15.6875}{T_{br}} - 13.4721 \times \ln T_{br} + 0.43577 \times T_{br}^6, \tag{54}$$

$$\rho = -7.904 + 0.1352 \times K_w - 0.007465 \times K_w^2 + 8.359 \times T_{br} + \frac{1.408 - 0.01063 \times K_w}{T_{br}}, \tag{55}$$

$$\omega = \begin{cases} \frac{\psi}{\phi}, & T_{br} \leq 0.8, \\ \rho, & T_{br} > 0.8, \end{cases} \tag{56}$$

where P_c, T_{br}, and K_w represent the critical pressure, reduced boiling temperature, and characteristic factor, respectively. All of them can be calculated from the boiling point (T_b) and specific gravity (SG), with the specific calculation formulas to be introduced later.

D.2 Dataset

During dataset acquisition, a pseudo-component approach Riazi (2005) is applied to petroleum with boiling points ranging from 300K to 1100K, resulting in pseudo-components characterized by their boiling point (T_b) and specific gravity (SG). Each pseudo-component represents a sample in this dataset, with the ground truth ω_{gt} obtained using Aspen Plus, a famous and efficient simulation software Liu et al. (2022); Al-Malah (2022). The input for each sample consists of six variables $\{T_b, SG, K_w, T_{br}, T_c, P_c\}$, where the characteristic factors (K_w), reduced boiling point (T_{br}), critical temperature (T_c), and critical pressure (P_c) for each pseudo-component can be calculated from T_b and SG using the equations provided below Riazi (2005):

$$T_{br} = \frac{T_b}{T_c}, \tag{57}$$

$$K_w = \frac{(1.8T_b)^{1/3}}{SG}, \tag{58}$$

$$T_c = 19.06232 \times T_b^{0.58848} \times SG^{0.3596}, \tag{59}$$

$$P_c = 5.53027 * 10^7 * {T_b}^{-2.3125} * SG^{2.3201}. \tag{60}$$

The underlying mechanism of the acentric factor is illustrated in Fig.9. The mechanistic model of the acentric factor is composed of explicit functions and exhibits limited accuracy; therefore, in this experiment, it is modeled using the Stage I approach of DNA, which also improves the performance of the mechanistic model itself.

E EXPERIMENTS

E.1 BASELINES

As described in Section2.1, the industrial digital twin task considered in this work can be formulated as follows:

$$\min_{M} \ \mathbb{E}_{x\sim\mathcal{X}} \left[|\mathcal{M}(x) - \mathcal{O}(x)|\right], \tag{61}$$

where $\mathbb{E}[\cdot]$ denotes the mathematical expectation and $\mathcal{X}$ denotes the distribution of operating conditions. Therefore, the task considered in this work is essentially a regression problem. To thoroughly evaluate DNA, we select four categories of models with strong regression capabilities: tree-based, transformer-based, graph-based, and other representative models. Below, we introduce each model individually:

- LightGBM Ke et al. (2017) is a boosting decision tree, with gradient-based one-side sampling and exclusive feature bundling.
- Transformer Vaswani (2017) is a sequence-based model, and here, we utilize only its encoder component for industrial modeling.
- FlowformerMH Wu et al. (2022b) is a linear transformer that models global interactions through flow conservation and efficient attention mechanisms, reducing the computational complexity of conventional self-attention.
- iTransformer Liu et al. (2024) inverts the conventional Transformer architecture by treating individual variables as tokens, enabling the model to capture multivariate correlations more effectively.
- MultiPatchFormer Naghashi et al. (2025) is a multi-scale patch-based Transformer that captures underlying patterns at different resolutions by modeling patches with multiple scales.
- GATv2 Veličković et al. (2017) leverages attention mechanisms to dynamically assign different importance weights to a node's neighbors ,enabling more effective learning on graph-structured data.
- DGDL Zhu & Zhao (2022) uses a dynamic graph to realize adaptive learning and automatic inference and aggregate node feature representations by a multihop attention graph convolution network.
- TGCNS Kong et al. (2023) emphasis lies in using a graph structure learning module to learn potential intervariable relationships from data.
- RaDA Chen et al. (2024) is a residual-aware deep attention graph convolutional network which uses a residual-aware connection module to reduce data uncertainty and alleviate over-smoothing.
- TimesNet Wu et al. (2022a) transforms one-dimensional input variations into two-dimensional tensors and employs convolutional operations to capture complex patterns and interactions, providing strong representation and regression capabilities.
- TimeXer Wang et al. (2024) is a Transformer-based architecture that employs patch-wise modeling and cross-attention to capture complex dependencies and interactions among variables, enabling effective representation learning for regression tasks.
- Umixer Ma et al. (2024) is a U-Net style MLP-based architecture that employs hierarchical mixing operations to capture multiscale patterns and learn effective representations for regression tasks.

E.2 IMPLEMENTATION DETAILS

We split all datasets into training, validation, and testing sets with a ratio of 7:2:1. To prevent overfitting, all models are trained using an early stopping strategy. The maximum number of training epochs is uniformly set to 2000, while the actual stopping epoch varies across different datasets. All

Table 6: General hyperparameters setting.

Stage	Dataset	Batch Size	Dim	lr	Max Training Epochs	Early Stopping at Epoch
Stage I	Cp.g	256	32	1e-4	2000	1733
	Cp.o	256	32	1e-4	2000	1140
	Cp.w	256	32	5e-4	2000	502
	Rho.o	256	32	1e-4	2000	1236
	Rho.l	256	32	5e-4	2000	2000
	Vis.w	256	32	1e-4	2000	1636
	Vis.o	256	32	5e-4	2000	1598
	Vis.l	256	32	1e-4	2000	1238
	ω	256	32	1e-3	2000	298
Stage II	HL	256	32	1e-3	2000	439
	P	256	64	1e-3	2000	940

Table 7: Correction functions of all the dataset in Stage I

Dataset	n	ln	$gate$	Correction Function
Cp.g	2	✗	✓	$y_{cor} = 0.8342991 \times 1.472976 + 0.52315223 \times 0.8895244 \times y_{me} + 0.30744648 \times 0.0457618 \times y_{me}^2$
Cp.o	2	✗	✓	$y_{cor} = 0.7068145 \times 1.2715807 + 0.5873085 \times 0.5926248 \times y_{me} + 0.59426093 \times 0.17929147 \times y_{me}^2$
Cp.w	0	✓	✗	$y_{cor} = 0.82825834 + 1.9789125 \times ln(y_{me} + 1.278508)$
Rho.o	2	✓	✓	$y_{cor} = 0.4679038 \times 0.42148522 + 0.6566006 \times 0.4409263 \times y_{me} + 0.6441115 \times 0.55001056 \times y_{me}^2 + 0.4212044 \times \ln(\lvert -0.82146084 \times y_{me} - 0.7055963 \rvert)$
Rho.l	2	✓	✗	$y_{cor} = 0.05121769 + 0.87679595 \times y_{me} - 0.5762174 \times y_{me}^2 + ln(\lvert 1.5252085 \times y_{me} + 0.34641176 \rvert)$
Vis.w	2	✓	✗	$y_{cor} = 0.4235715 + 0.5907204 \times y_{me} - 0.26162362 \times y_{me}^2 + ln(\lvert 0.3140283 \times y_{me} + 0.6718855 \rvert)$
Vis.o	3	✓	✓	$y_{cor} = 0.6880405 \times 2.018896 + 0.49424344 \times 1.8307866 \times y_{me} - 0.43409765 \times 1.4872801 \times y_{me}^2 + 0.19532314 \times 0.73345625 \times y_{me}^3 + 0.14005646 \times ln(\lvert -2.045505 \times y_{me} + 3.0183353 \rvert)$
Vis.l	3	✓	✗	$y_{cor} = 2.210485 - 3.1021433 \times y_{me} + 1.5562956 \times y_{me}^2 - 0.24822733 \times y_{me}^3 + ln(\lvert -4.001664 \times y_{me} + 0.9682727 \rvert)$
ω	2	✗	✗	$y_{cor} = 0.17715707 + 0.4112167 \times y_{me} + 0.43530726 \times y_{me}^2$

experiments are implemented in PyTorch Paszke et al. (2019) and conducted on a machine equipped with four NVIDIA GeForce RTX 4090 GPUs. We adopt the Adam optimizer Kingma & Ba (2014) for all experiments. Detailed hyperparameter settings for each dataset are reported in Tab. 6.

E.3 Choice of Correction Function Form

As described in the main paper, the predefined function form for each dataset and the correction functions ultimately obtained by our model are summarized in Tab.7.

E.4 More Discussion on Stage I Experimental Results of the Three-Phase Pipeline

An important point that warrants clarification is that, in Stage I, it is not necessary to directly compare the results of corrected mechanistic model (Cor) with those of the data-driven model (DD). In industrial practice, data-driven models are often difficult to deploy at scale due to their black-box nature, particularly in safety-critical applications, whereas mechanistic models remain widely adopted because they are grounded in physical principles and offer physical transparency and stability. However, mechanistic models often suffer from limited accuracy due to their complex formulation, reliance on expert knowledge, and idealizing assumptions. This perspective motivates us to develop the DNA architecture, where the enhanced mechanistic model is intended for industrial deployment, filling the gap left by data-driven models that are difficult to deploy at scale while partially overcoming the accuracy limitations of itself. Therefore, a substantial improvement of the enhanced mechanistic model over the original mechanistic model indicates that our objective has been achieved, regardless of whether it outperforms the data-driven model. In this context, compar-

Table 8: Comparison of model parameters, FLOPs, and batch inference time on the Cp.w dataset.

Models	Parameters	FLOPs	Inference Time (ms)
DNA	**8726**	**67540**	32.85
Transformer	65476	213984	12.38
FlowFormer	16253	243241	12.95
iTransformer	135460	89264	19.28
MPFormer	77444	210288	81.78
GATV2	32722	134752	14.70
RaDa	13652	67730	**12.11**
TimesNet	44264	83008	24.27
TimeXer	46628	114784	20.40
Umixer	105758	89768	21.67

ing $\mathcal{M}^I_{cor}$ with $\mathcal{M}^I_{dd}$ is not essential: $\mathcal{M}^I_{cor}$ does not need to outperform $\mathcal{M}^I_{dd}$; improvement over $\mathcal{M}^I_{me}$ is sufficient.

E.5 Further Analysis of Model Complexity and Efficiency

We report model complexity and efficiency metrics in Tab.8. In terms of total number of parameters and FLOPs required for a single forward inference, DNA shows a clear advantage over other baselines, indicating that our model is highly lightweight. Moreover, due to its reduced FLOPs, DNA theoretically possesses the potential for efficient inference.

However, as shown by the inference time in Tab.8, DNA does not yet achieve the ideal inference efficiency. This is because, in MASA, we assign different attention scopes to different variables based on the underlying mechanism of the modeled object. Each variable performs computations within its own attention scope to produce the corresponding updated representations. Since the attention scopes differ across variables, it is difficult to perform fully parallel computation as in vanilla self-attention, which can generate updated representations for all variables simultaneously. Instead, each variable's representation must be computed sequentially according to its assigned scope, which significantly increases the inference time. This explains the current suboptimal inference efficiency of DNA.

Nonetheless, precisely because different attention scopes are assigned to different variables, the data-driven model constructed by DNA both adheres to the intrinsic mechanism of the modeled object and avoids unnecessary attention computations. As a result, when replacing MHSA with MASA, the FLOPs are substantially reduced, as shown in Tab.9.

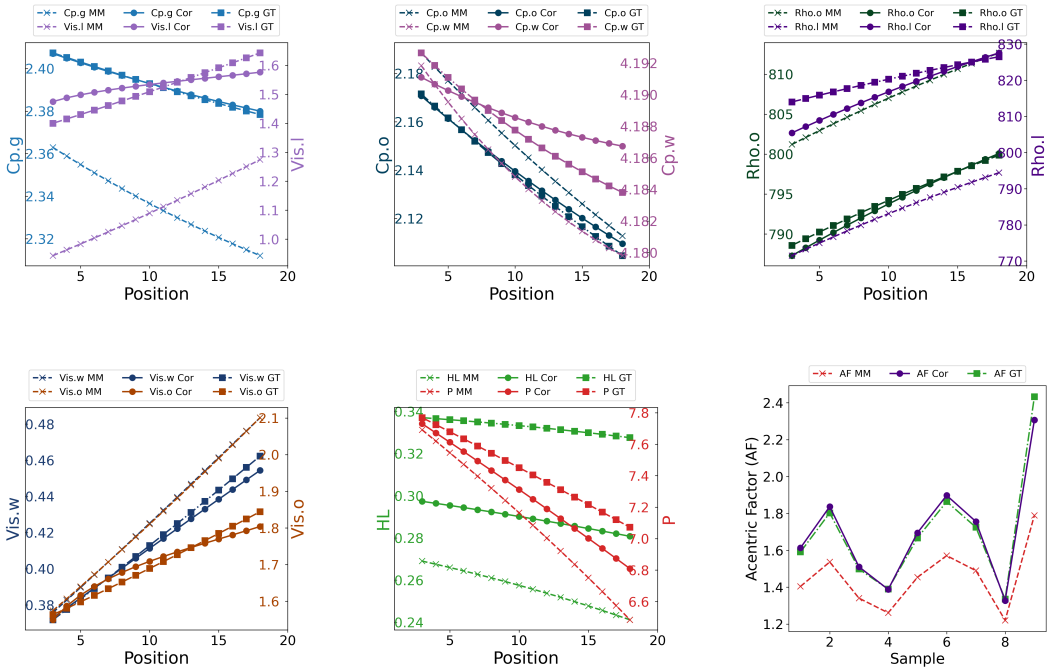
In summary, the current suboptimal inference efficiency of DNA is primarily due to the fact that we have not yet found an effective way to parallelize MASA. Addressing this implementation-related issue will be one of our key focuses in future work.

E.6 Visualization

To intuitively demonstrate the improvement brought by the Stage I correction, we visualize the results at 16 representative positions from a randomly selected case. Due to space constraints, the main paper only presents visualizations for four datasets (Cp.g, Vis.l, HL, P). Here, we supplement the remaining six visualization results from the three-phase pipeline experiments, as well as those from the acentric factor experiments, as shown in the figures below.

Table 9: FLOPs comparison between DNA (with MASA) and DNA (with MHSA) on Stage I datasets.

Dataset	FLOPs(DNA with MASA)	FLOPs(DNA with MHSA)	FLOPs Reduction (%)
Cp.g	67540	79892	15.46
Cp.o	70740	79892	11.46
Cp.w	67412	79892	15.62
Rho.o	45028	51620	12.77
Rho.l	72560	87280	16.87
Vis.w	53760	58496	8.10
Vis.o	53632	58496	8.32
Vis.l	59164	65500	9.67



In summary, this work lays the groundwork for future advances in developing high-fidelity models for industrial digital twins, particularly from a symbiotic modeling perspective in which mechanistic models and data-driven models interact effectively, mutually influence each other, and jointly improve their respective performance. While we acknowledge that there remains a long way to go, DNA represents a pioneering effort that will be continuously refined, aiming to offer an effective pathway toward a new paradigm of next-generation symbiotic modeling.