How knowledge discovery and embedded paradigm transform industrial process management: exploring pipeline hydraulic dynamic identification

Jian Du[1,2*], **Haochong Li**[2], **Jianqin Zheng**[3], **Qi Liao**[2†], **Jun Shen**[4], **Pengtao Niu**[2],

Shiyuan Pan[2], **Yongtu Liang**[2]

[1]Dipartimento di Energia, Politecnico di Milano, Milano, Italy
[2]China University of Petroleum-Beijing, Changing District, Beijing, PR China
[3]PetroChina Planning & Engineering Institute, Haidian District, Beijing, P.R. China
[4]School of Computing and Information Technology, University of Wollongong, NSW, Australia

Abstract

An effective hydraulic parameter identification underpins process simulation for pipeline optimization. However, current studies often overlook the hydraulic spatiotemporal dynamics and multi-frequency variations of simulation parameters, limiting accuracy and interpretability. Here, by exploring the opportunity of bridging industrial process simulation relying on theoretical research paradigms in scientific discovery, we propose a knowledge discovery and embedded framework to identify optimal friction coefficient and capture multi-frequency online variations of friction. The proposed framework identifies the optimal friction coefficient by discovering hydraulic spatiotemporal dynamics based on partial derivative differences within pipeline hydraulic state matrices. By embedding explicit hydraulic physical theory into forward propagation, a physics-constrained autoregressive neural network is developed as an efficient, interpretable surrogate model. Then, a self-coordination framework is designed for synchronous friction updating. The proposed framework can achieve precise online hydraulic simulation by performing knowledge-discovery identification and knowledge-embedded modeling. Results confirm accuracy and robustness of the proposed framework across varying pipeline and fluid properties. By integrating bottom-up knowledge discovery with top-down embedding, this approach forms a self-improving loop, offering strong potential for industrial pipeline digital twins and efficient decision-making.

1 Introduction

Pipeline transportation systems, fundamental to various industrial sectors, have emerged as the most economical and energy-efficient solution for liquid media distribution, including urban water supply [1, 2] and petroleum product delivery [3, 4]. However, with infrastructure aging and replacement often delayed, frequent switching of operation conditions [5] can induce extreme pressure surges. These transient processes heighten the risk of structural failure and potential explosions [6], particularly in hydrocarbon pipelines due to their flammable and explosive nature [7]. Therefore, effective monitoring of pressure and flowrate is critical to maintaining system safety and reliability.

*jiandu1997@163.com
†qliao@cup.edu.cn

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Science workshop (NeurIPS 2025).

High-precision sensors are commonly employed to monitor hydraulic states such as pressure and flowrate within pipelines. However, their widespread deployment is limited by substantial installation and maintenance costs, restricting sensor placement to locations such as pipeline inlets and outlets [8]. This sparse arrangement creates extensive non-detection zones within the pipeline system [9]. As a result, the development of efficient simulation tools capable of accurately estimating transient hydraulic states (pressure and flowrate) becomes essential for pipeline risk assessment [10, 11], system planning [12], and operational optimization [13, 14].

Hydraulic transients are governed by one-dimensional water hammer PDEs [15], which lack analytical solutions [16], prompting decades of research into numerical methods such as the Method of Characteristics (MOC) [17] and Finite Difference Method (FDM) [18]. In practice, wave propagation characteristic is influenced by uncertainties in PDE coefficients, including friction, wave speed, and pipe-wall viscoelasticity, which significantly impact simulation accuracy [19]. While most coefficients remain relatively constant and can be estimated empirically, the friction coefficient exhibits strong time dependence and is tightly coupled with transient dynamics. Accurate identification of friction coefficient play a pivotal role in reliable hydraulic simulation.

The optimization theory-based method, a prevalent framework for pipeline parameter identification, which identifies unknown parameters by aligning observed signals with associated numerical model outputs, was first introduced by Liggett and Chen in 1994 [20]. Optimization theory-based method primarily fall into two categories: mathematical statistical-based [21, 22] and evolutionary optimization-based [23, 24, 25], with the latter increasingly recognized as a leading technique due to its adaptability and robustness. However, conventional evolutionary strategies typically identify simulation parameters based solely on squared errors (SE) between measured and simulated responses. The essence that variations in parameters are often manifestations of deeper spatiotemporal hydraulic dynamics (***nonlinear time-delay characteristics and wave-propagation characteristics***) are neglected by conventional evolutionary strategies. This disconnect limits both the physical interpretability and the precision of hydraulic simulation under transient conditions. Additionally, the inherently iterative nature of evolutionary algorithms requires repeated PDE evaluation and population regeneration. This results in significant computational overhead, resulting in the incapability of parameter dynamics synchronous extraction [26].

Recently, time-series data-driven methods [27], such as the Nonlinear AutoRegressive neural network with eXogenous inputs (NARX) [25] and long short-term memory (LSTM) [28], have attracted significant interest in pipeline hydraulic simulation. These models can predict hydraulic parameters at pipeline inlets and outlets efficiently but struggle to infer states in non-detection zones. Moreover, the absence of hydraulic principles in their training leads to reduced accuracy and limited physical interpretability [29]. Physics-informed neural networks (PINNs) offer a promising solution by incorporating physical laws for more accurate and interpretable simulations [4]. However, their training is computationally intensive, which often requires several hours, thus limiting their practicality for real-time applications [30]. Meanwhile, although efficient hydraulic simulation can be achieved by data-driven methods, the high time cost of evolutionary iterations in parameter identification have not been fundamentally overcome.

The essence of identifying parameter is discovering the spatiotemporal hydraulic dynamics from pipeline system states. Motivated by the intrinsic spatiotemporal dynamics of hydraulic transient, this study explores the opportunity of bridging industrial process simulation relying on theoretical research paradigms in scientific discovery. We propose a novel knowledge discovery and embedded framework for interpretable parameter identification and precise hydraulic simulation. The key contributions of this work are as follows:

- To the best of our knowledge, we present the first spatiotemporal dynamic discovery-based parameter identification (STDD) algorithm (Sec 2.2.1) by designing a spatiotemporal partial derivatives to represent pipeline hydraulic dynamics. The proposed algorithm can identify friction coefficient effectively and overcome key limitations of existing techniques, including limited interpretability and degraded fidelity under transient conditions.
- We propose a physics-guided autoregressive neural network (PG-ARNN, Sec 2.2.2) that incorporates hydraulic transient theory to function as an efficient surrogate for parameter identification. Surrogate models can avoid the high iterative search costs of traditional parameter identification methods. This hybrid approach addresses the generalization limitations of purely data-driven models when exposed to previously unseen operating conditions.

- We develop a multi-frequency self-coordination simulation framework (Sec 2.2.3) by organically integrating knowledge discovery-driven identification with knowledge-informed modeling. This approach can capture multi-frequency synchronous variations in the friction coefficient. Then, the intrinsic asynchrony arising from fixed-interval parameter identification under both pseudo-steady and transient conditions can be better addressed.

2 Methodology

2.1 Problem description

As shown in Eqs. (1) and (2), the one-dimensional governing equations describe the transient hydraulic behavior in liquid pipelines as functions of both time (*t*) and space (*x*):

$$\frac{1}{gA}\left(\frac{\partial Q}{\partial t}+\frac{Q}{A}\frac{\partial Q}{\partial x}\right)+\frac{\partial H}{\partial x}+fQ|Q|^{1-m}=0 \tag{1}$$

$$\frac{\partial H}{\partial t}+\frac{Q}{A}\frac{\partial H}{\partial x}+\frac{a^2}{gA}\frac{\partial Q}{\partial x}=0 \tag{2}$$

where *H* is the head (pressure is the product of density, gravitational acceleration, and head), *Q* is the flowrate. *A* is the cross-sectional area of the pipeline, *g* is gravitational acceleration, *f* is the Darcy–Weisbach friction factor. The coefficient *m* is assigned as 0.25 for the hydraulically smooth zone and 0.125 for the mixed friction zone. $a=\sqrt{\frac{K/\rho}{1+\frac{K}{E}\frac{D}{\delta}C_1}}$ is the wave speed. For the other letters in the formula, see Reference [4].

Given the superior stability of pressure transmitters over ultrasonic flow meters, measured pressures are employed as inputs and boundary control conditions to drive the hydraulic simulation through MOC (Appendix A.1). The hydraulic simulation can be mathematically represented as $X_{t+1}=F\left(X_t,U_{t+1}\right)$. Where $X_t=[H_{0,t},H_{1,t},...,H_{M,t},Q_{0,t},Q_{1,t},...,Q_{M,t}]$ denotes the system state matrix at *t*, $U_{t+1}=[H_{0,t+1},H_{M,t+1}]$ is the boundary control conditions at *t*+1, $F\left(x\right)$ is the state update function. In practical applications, Δt is typically set to 1 second for real-time simulation, while Δx is derived from the wave speed and Δt.

As discussed in *Discussion of conventional parameter identification methods* in Appendix A.1, to tackle the identification asynchronism in real-time simulation and inherent overlook of spatiotemporal hydraulic dynamics of conventional methods, this study proposes an interpretable objective function and a multi-frequency self-coordination simulation framework.

2.2 Data-driven knowledge discovery and embedded framework

2.2.1 Spatiotemporal Dynamic Discovery-Based Parameter Identification Algorithm

The friction coefficient is closely linked to the Reynolds number, which depends on fluid properties and flowrate [18]. Under transient conditions, rapid fluctuations in pressure and flowrate cause corresponding changes in friction. Accurately identifying the optimal friction coefficient in real time is therefore crucial for precise hydraulic simulation.

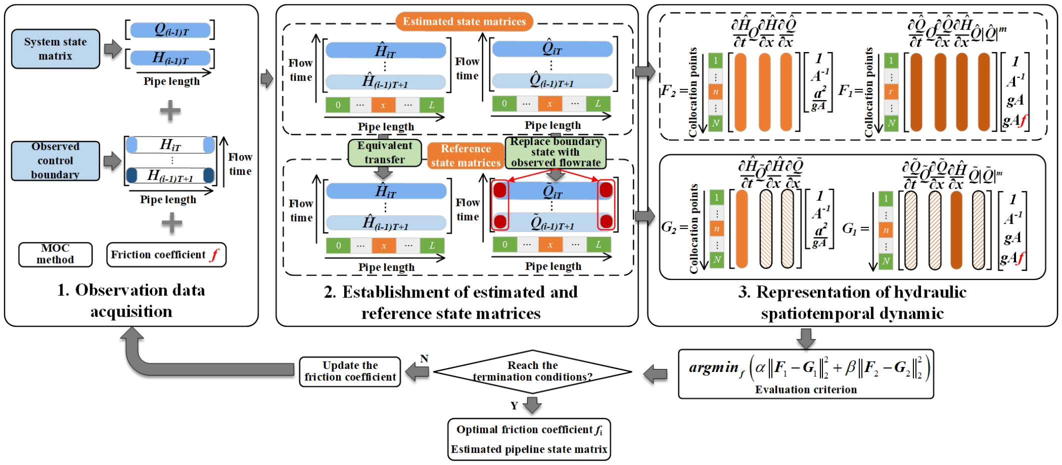


Figure 1: Spatiotemporal Dynamic Discovery-Based Parameter Identification Framework

As depicted in Figure 1, to identify the friction coefficient, the initial condition is treated as pseudo-steady, with hydraulic states $H(x,t)$ and $Q(x,t)$, $x \in [0, L]$, $t \in [-m, 0]$, where L is the pipeline length and m denotes the duration of pseudo-steady flow along the pipeline derived using Darcy's law. Let T represent the parameter identification interval. Given boundary conditions $H(x,t)$ in the i^{th} interval, with $x \in \{0, L\}$ and $t \in [(i-1)\,T+1, iT]$ as well as initial states $H(x,t)$ and $Q(x,t)$ with $x \in [0, L]$ and $t = (i-1)\,T$, the state matrices can be estimated via the hydraulic simulation described in Sec 2.1, as shown in Eqs. (3) and (4).

$$\hat{H} = \begin{bmatrix} H\left(0, iT\right) & H\left(1, T\right) & \cdots & H\left(L, T\right) \\ H\left(0, iT-1\right) & H\left(1, iT-1\right) & \cdots & H\left(L, iT-1\right) \\ \vdots & \vdots & \ddots & \vdots \\ H\left(0, (i-1)T+1\right) & H\left(1, (i-1)T+1\right) & \cdots & H\left(L, (i-1)T+1\right) \end{bmatrix} \tag{3}$$

$$\hat{Q} = \begin{bmatrix} Q(0, iT) & Q(1, T) & \cdots & Q(L, T) \\ Q(0, iT-1) & Q(1, iT-1) & \cdots & Q(L, iT-1) \\ \vdots & \vdots & \ddots & \vdots \\ Q\left(0, (i-1)T+1\right) & Q\left(1, (i-1)T+1\right) & \cdots & Q\left(L, (i-1)T+1\right) \end{bmatrix} \tag{4}$$

Observed flowrates are acquired from calibrated ultrasonic flow meters. Substituting the boundary flowrates in the estimated flowrate matrices (Eq. (4)) with observed values yields the reference flowrate matrix, as shown in Eq. (5).

$$\tilde{Q} = \begin{bmatrix} \tilde{Q}\left(0, iT\right) & Q\left(1, T\right) & \cdots & \tilde{Q}\left(L, T\right) \\ \tilde{Q}\left(0, iT-1\right) & Q\left(1, iT-1\right) & \cdots & \tilde{Q}\left(L, iT-1\right) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Q}\left(0, (i-1)\,T+1\right) & Q\left(1, (i-1)\,T+1\right) & \cdots & \tilde{Q}\left(L, (i-1)\,T+1\right) \end{bmatrix} \tag{5}$$

Notably, the reference head matrix is the same as the estimated head matrix. When the friction coefficient used in hydraulic simulation aligns well with actual hydraulic dynamics, the estimated and reference flowrate matrices should closely coincide. To evaluate hydraulic dynamic differences, the partial derivatives of elements in both estimated ($\hat{H}$ and $\hat{Q}$) and reference ($\hat{H}$ and $\tilde{Q}$) matrices over time and space are discretized using finite difference schemes, yielding the residuals of the water hammer PDEs, as shown in Eqs. (6)–(9).

$$F_1 = \Phi_1^{est} \bullet \xi_1^{est} = \left[\nabla_t \hat{Q}, \hat{Q} \bullet \nabla_x \hat{Q}, \nabla_x \hat{H}, \hat{Q}\left|\hat{Q}\right|^{0.75}\right] \left[1, \frac{1}{A}, gA, gAf\right]^T \tag{6}$$

$$F_2 = \Phi_2^{est} \bullet \xi_2^{est} = \left[\nabla_t \hat{H}, \hat{Q}\nabla_x \hat{H}, \nabla_x \hat{Q}\right] \left[1, \frac{1}{A}, \frac{a^2}{gA}\right]^T \tag{7}$$

$$G_1 = \Phi_1^{ref} \bullet \xi_1^{ref} = \left[\nabla_t \tilde{Q}, \tilde{Q} \bullet \nabla_x \tilde{Q}, \nabla_x \hat{H}, \tilde{Q}\left|\tilde{Q}\right|^{0.75}\right] \bullet \left[1, \frac{1}{A}, gA, gAf\right]^T \tag{8}$$

$$G_2 = \Phi_2^{ref} \bullet \xi_2^{ref} = \left[\nabla_t \hat{H}, \tilde{Q}\nabla_x \hat{H}, \nabla_x \tilde{Q}\right] \left[1, \frac{1}{A}, \frac{a^2}{gA}\right]^T \tag{9}$$

Here, ∇ denotes the gradient operator for head and flowrate, while Φ^{est} and Φ^{ref} form libraries of partial derivative and constant terms for the estimated and reference matrices, respectively. $F_1 \in \mathbb{R}^{N\times 4}$ and $F_2 \in \mathbb{R}^{N\times 3}$, with $N = L \times T$ being the number of collocation points, represent the momentum and continuity residuals for the estimated matrices; likewise, $G_1 \in \mathbb{R}^{N\times 4}$ and $G_2 \in \mathbb{R}^{N\times 3}$ are the corresponding residuals for the reference matrices. When the friction coefficient reflects the true hydraulic behavior, the spatiotemporal derivatives of both matrices coincide, resulting in equal residuals. Accordingly, an objective function (Eq. (10)) is proposed to quantify the residual differences.

$$\{f^*, H^*(x,t), Q^*(x,t)\} = argmin_f\left(\alpha \left\|F_1 - G_1\right\|_2^2 + \beta \left\|F_2 - G_2\right\|_2^2\right) \tag{10}$$

Here, a and β are tunable hyper-parameters used to enhance convergence effect. Solving the optimization yields the optimal friction coefficient f^*, along with the corresponding hydraulic states $H^*(x,t)$ and $Q^*(x,t)$ in non-detectable pipeline zones. Finally, the STDD algorithm is executed periodically to update the friction coefficient in different time intervals, ensuring the hydraulic simulation remains aligned with pipeline dynamics. In essence, STDD advances parameter identification method by quantifying hydraulic spatiotemporal dynamics (***time-delay and wave-propagation characteristics***), thereby enhancing physical interpretability. By replacing the objective function (Eqs. (3)-(10)) in Figure 1 as squared error (SE, Eq. (17)), conventional parameter identification methods can be acquired.

2.2.2 Physics-Guided Autoregressive Neural Network

While Sec 2.2.1 introduces a interpretable parameter identification algorithm, the real-time applicability of STDD is hindered by its high computational cost (on the order of minutes). To address this, we propose a computationally efficient surrogate model that replicates STDD's functionality. Based on the Reynolds Transport Theorem, liquid properties and present flowrate are primary determinants of the friction coefficient. However, as present flowrates are not boundary control conditions, they must be inferred from previous hydraulic states and present pressure data. To this end, a dual-layer neural network framework is constructed (Figure A.1), employing autoregressive neural networks in each layer to capture time-delay characteristics in hydraulic parameters. The first-level network takes the estimated boundary flowrate from (i–1)T to iT and the observed boundary pressure from (i–1)T to (i+1)T as inputs to predict flowrate in the interval iT to (i+1)T.

Then, the estimated present flowrate of the first-level network is concatenated with previous flowrate as input features. This composite input, along with the friction coefficient at the i^{th} interval, are fed into the second-level network. Additionally, fluid density and viscosity are processed via a fully connected (FC) layer to infer the friction coefficient at the (i+1)th interval. The forward propagation of the dual-lay neural network is detailed in Appendix A.2.

2.2.3 Multi-Frequency Self-Coordination Simulation Framework

The temporal variability of the friction coefficient, a key determinant of hydraulic simulation accuracy, differs markedly between pseudo-steady and transient conditions. In pseudo-steady states, friction changes minimally over minutes to hours, whereas transient conditions induce second-scale fluctuations. Optimization-based parameter identification approaches, including STDD, which require minutes for parameter identification, suffer from significant phase lag during transients. To address this, we propose an multi-frequency self-coordination simulation framework, detailed in Framework 1.

Initially, optimal friction coefficients and estimated states obtained via STDD at two different time intervals are used to train separate PG-ARNNs. Among these, STDD with time interval being 10 seconds for transient and 5 minutes for pseudo-steady conditions. Operation conditions are identified by comparing pressure fluctuation amplitudes against predefined thresholds, triggering the corresponding PG-ARNN. When the cumulative sampling time surpasses the designated interval, observed pressure and estimated states are used to identify the friction coefficient. Otherwise, the previous friction is retained. This updated coefficient is then applied for real-time hydraulic simulation, enabling an online rolling simulation.

Framework 1 Online hydraulic simulation by using a multi-frequency self-coordination framework

Offline Training:
1. **For time intervals of 10-second or 5-minutes:**
   1.1 Extract the observed boundary pressure and previous hydraulic states along pipeline
   1.2 Identify the optimal friction and simulate the hydraulic states
   1.3 Form friction databases **(D_1 for 10-second interval, D_2 for 5-minute interval)**
2. Train different PG-ARNNs using databases **($D_1 \rightarrow$PG-ARNN$_1$, $D_2 \rightarrow$PG-ARNN$_2$)**

Online Simulation:
1. Gather real-time pressure and previous estimated hydraulic states
2. If the pressure fluctuation exceeds predefined threshold
   2.1 Recognized as transient condition and a **10-second interval** is applied
   2.2 If cumulative sampling time surpasses the designated interval
      2.2.1 Identify friction coefficient using **PG-ARNN$_1$**
   2.3 Else, **using the historical friction coefficient**
   2.4 Hydraulic simulation
3. Else
   3.1 Recognized as pseudo-steady condition and a **5-minute interval** is applied
   3.2 If cumulative sampling time surpasses the designated interval
      3.2.1 Identify friction coefficient using **PG-ARNN$_2$**
   3.3 Else, **using the historical friction coefficient**
   3.4 Hydraulic simulation
4. Begin hydraulic simulation in next time step

3 Case studies with experiments

3.1 Experiment setting

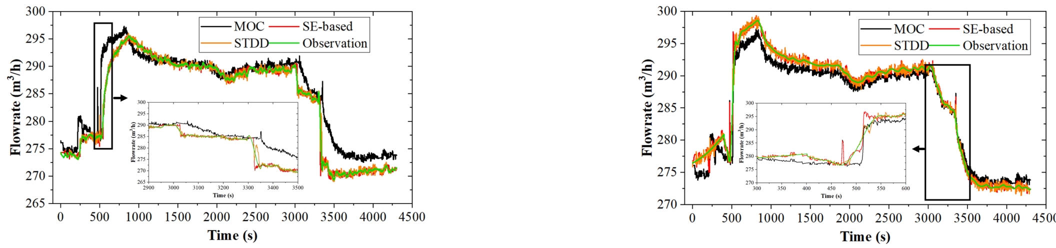
To evaluate the effectiveness and generalizability of the proposed framework, four real-world liquid pipelines with varying characteristics were selected, as detailed in Table B.1. All pipelines share a common elasticity modulus of 2.07×10^{11} Pa. As illustrated in Figure B.1, pressure signals recorded at the pipeline inlet and outlet by high-precision sensors were used as input to the hydraulic simulation model, enabling estimation of pressure and flowrate at Δx km intervals at 1-second resolution. Flowrate data from calibrated ultrasonic flowmeters at both ends were employed for STDD execution and validation of flowrate simulations. Pressure measurements from three intermediate valve chambers between pipeline inlet and outlet supported pressure simulation verification.

The optimal network parameters of PG-ARNN were determined via trial and error, as shown in Table B.2. The model was implemented using the PyTorch framework. To optimize the balance between performance and computational efficiency of STDD, 50 iterations were performed with 50 candidate coefficients evaluated per iteration. PG-ARNN was trained over 2000 epochs with an initial learning rate of 0.0001. The experiments were conducted on a workstation equipped with a single NVIDIA GeForce RTX 3090 GPU.

3.2 Evaluation and analysis of the proposed STDD algorithm

Optimization-based parameter identification methods primarily differ in their iterative strategies and evolutionary mechanisms, as their underlying objective functions are essentially the same. In this study, we mainly focus on proposing a novel objective function for evolution iteration process. Consequently, this subsection only analyzes identification results derived from different objective functions. The SE-based method, which replaces the objective function in STDD as SE are selected as benchmark. To demonstrate the significance of parameter identification, the hydraulic simulation model based on MOC without parameter identification is used as comparative model.

The hydraulic simulations on Case 1 are illustrated in Figure 2. and 3. STDD delivers the most accurate hydraulic simulation, closely aligning with observed profiles and outperforming all comparative methods. In contrast, the MOC approach exhibits the largest discrepancies. While SE-based and STDD methods show similar accuracy during pseudo-steady states, STDD demonstrates markedly superior performance under transient conditions. Flowrate and pressure simultion on other three cases are shown in Figure B.2 and B.4. Figure B.3, B.5, Table 1, and 2 further support this, with STDD achieving the lowest pressure residuals, with average errors being 0.0013 MPa, 0.0011 MPa, 0.0034 MPa, and 0.0042 MPa across four cases. By exploiting hydraulic dynamics characteristics of spatiotemporal derivatives, STDD reduces residuals by 81.9%, 60.7%, 87.5%, and 40.0% compared to SE-based method.



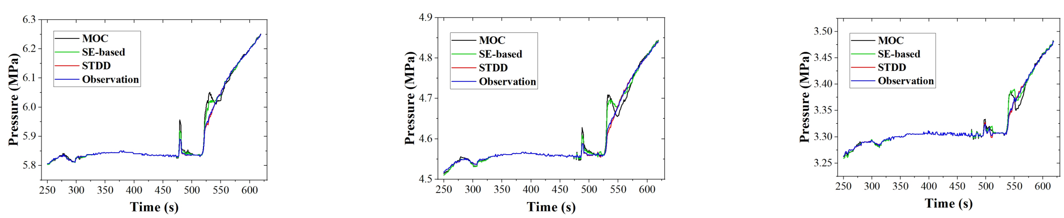
(a): Outlet flowrate estimation

(b): Inlet flowrate estimation

Figure 2: Comparison of flowrate estimation between SE-based method and STDD on Case 1

Table 1: MAPE comparisons of flowrate estimation between SE-based method and STDD

Methods	Inlet flowrate estimation (%)			Outlet flowrate estimation (%)		
	Whole flow process	Transient condition	Pseudo-steady condition	Whole flow process	Transient condition	Pseudo-steady condition
STDD	**0.085**	**0.299**	**0.082**	**0.079**	**0.193**	**0.070**
MOC	0.415	1.084	0.344	0.700	2.417	0.603
SE-based	0.108	0.795	0.098	0.121	0.463	0.109



(a): 9.6 km

(b): 19.6 km

(c): 29.6 km

Figure 3: Comparison of flowrate estimation between SE-based method and STDD on Case 1

Table 2: Average absolute residuals of pressure simulation between SE-methods and STDD

Cases	Average absolute residuals (MPa)		
	MOC	SE-based	STDD
Case 1	0.0135	0.0072	**0.0013**
Case 2	0.0043	0.0028	**0.0011**
Case 3	0.0338	0.0272	**0.0034**
Case 4	0.0090	0.0070	**0.0042**

Table 3: Flowrate estimation errors of different frameworks

Flowrate estimation	Frameworks	Inlet flowrate estimation (%)			Outlet flowrate estimation (%)		
		Whole flow process	Transient condition	Pseudo-steady condition	Whole flow process	Transient condition	Pseudo-steady condition
Outlet flowrate	Proposed Framework (PG-ARNN)	**0.283**	**1.204**	**0.296**	**0.097**	**0.291**	**0.169**
	Proposed Framework (NARX)	0.402	1.683	0.530	0.391	0.875	0.624
	Conventional Framework	0.609	3.803	0.708	0.639	1.017	0.727
Inlet flowrate	Proposed Framework (PG-ARNN)	**0.395**	**0.693**	**0.422**	**0.112**	**0.351**	**0.151**
	Proposed Framework (NARX)	0.443	0.837	0.533	0.176	0.465	0.195
	Conventional Framework	0.565	1.447	0.589	0.455	1.956	0.620

Table 4: Average absolute residuals of pressure simulation between various frameworks

Cases	Average absolute residuals (MPa)		
	Conventional Framework	Proposed Framework (NARX)	Proposed Framework (PG-ARNN)
Case 3	0.0277	0.0114	**0.0025**
Case 4	0.0194	0.0124	**0.0024**

To provide a comprehensive interpretability analysis of the spatiotemporal dynamics identification algorithm, the identification results and the trend of hydraulic parameters over time are discussed in the appendix C.1.

3.3 Evaluation of PG-ARNN and multi-frequency simulation framework

To demonstrate the importance of multi-frequency parameter identification for online simulation, we compare the proposed frameworks (with either PG-ARNN or the benchmark NARX model) to a conventional fixed-interval (5 min) parameter identification approach. Neural networks are trained on Cases 1–2 and tested on Cases 3–4, ensuring a balanced dataset split.

As shown in Figure B.6 and B.8, the simulation results of conventional framework on Case 3 exhibits the largest deviations from observed values. These errors stem from phase delays in fixed-interval parameter identification, which fail to capture rapid friction coefficient fluctuations during transients. In contrast, the proposed multi-frequency approach enables high-frequency updates during transients, enhancing estimation accuracy. Residual error comparisons in Figure B.7 and B.9 further support this. Quantitative results in Table 3 and 4 confirm that PG-ARNN achieves the lowest errors, reducing inlet flowrate MAPE by 68.4% and 71.4% compared to the conventional framework. For the pressure simulation, PG-ARNN-based framework suggests a residual reduction of 91.0% and 87.7% compared to the conventional framework, and 78.0% and 80.7% compared to NARX-based framework.

4 Discussion and Conclusion

In this study, we propose a innovative data-driven knowledge discovery and embedded framework for interpretable parameter identification and accurate pipeline hydraulic simulation. The primary advantages and innovations are summarized as follows:

- Our approach constructs partial derivative residuals across spatial and temporal domains of system state matrices for discovering hydraulic spatiotemporal dynamics. These form the basis of the parameter identification algorithm with an interpretable objective function. The proposed algorithm tackles the limitations that estimate hydraulic states inaccurately, especially under transient conditions. Real-world cases show that pipeline transients exhibit substantial hydraulic spatiotemporal dynamic variability. Across four benchmark cases, our algorithm reduces MAPE in transient inlet and outlet flowrate by 53.1% and 63.5%, respectively, and improves pressure prediction by 81.9%, 60.7%, 87.5%, and 40.0%.

- We develop a physics-constrained neural network by embedding discovered hydraulic laws into forward propagation, serving as an efficient surrogate for parameter identification. Building on this, a multi-frequency online simulation framework is introduced to enable synchronous parameter updates. Dynamic adjustment of identification intervals effectively eliminates the phase delay issues inherent in fixed-interval-based methods. Compared to conventional frameworks, the proposed method reduces MAPE by 68.35% (inlet) and 52.15% (outlet) in Case 3, further improving to 71.37% and 82.08% in Case 4. For pressure prediction, residuals drop by 91.0% and 87.7%, respectively.

This approach offers a novel pathway for digital twin development in process simulation of pipeline operation by uncovering hidden physics and reintegrating it to enhance model fidelity. Future efforts will aim to develop efficient multi-dimensional hydrodynamic interaction tensors and advance high-accuracy, low-cost simulation algorithms.

5 Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (52202405), the ARC Linkage Project LP230100083, and the China Scholarship Council program (Project ID: 202406440090). The authors are grateful to all study participants.

References

[1] Huan-Feng Duan, Bin Pan, Manli Wang, Lu Chen, Feifei Zheng, and Ying Zhang. State-of-the-art review on the transient flow modeling and utilization for urban water supply system (uwss) management. *Journal of Water Supply: Research and Technology—AQUA*, 69(8):858–893, 2020.

[2] Ying Zhang, Huan-Feng Duan, Alireza Keramat, and Tong-Chuan Che. On the leak-induced transient wave reflection and dominance analysis in water pipelines. *Mechanical Systems and Signal Processing*, 167:108512, 2022.

[3] Qi Liao, Yongtu Liang, Renfu Tu, Liqiao Huang, Jianqin Zheng, Guotao Wang, and Haoran Zhang. Innovations of carbon-neutral petroleum pipeline: A review. *Energy Reports*, 8:13114–13128, 2022.

[4] Jian Du, Haochong Li, Kaikai Lu, Jun Shen, Qi Liao, Jianqin Zheng, Rui Qiu, and Yongtu Liang. Deeppipe: A multi-stage knowledge-enhanced physics-informed neural network for hydraulic transient simulation of multi-product pipeline. *Journal of Industrial Information Integration*, 42:100726, 2024.

[5] Jianqin Zheng, Jian Du, Yongtu Liang, Qi Liao, Zhengbing Li, Haoran Zhang, and Yi Wu. Deeppipe: A semi-supervised learning for operating condition recognition of multi-product pipelines. *Process Safety and Environmental Protection*, 150:510–521, 2021.

[6] Jian Du, Jianqin Zheng, Yongtu Liang, Ning Xu, Qi Liao, Bohong Wang, and Haoran Zhang. Deeppipe: Theory-guided prediction method based automatic machine learning for maximum pitting corrosion depth of oil and gas pipeline. *Chemical Engineering Science*, 278:118927, 2023.

[7] Ying Zhang, Alireza Keramat, and Huan-Feng Duan. Formulation and analysis of transient flows in fluid pipelines with distributed leakage. *Mechanical Systems and Signal Processing*, 212:111294, 2024.

[8] G.H. Roshani, S.A.H. Feghhi, and S. Setayeshi. Dual-modality and dual-energy gamma ray densitometry of petroleum products using an artificial neural network. *Radiation Measurements*, 82:154–162, 2015.

[9] Jianqin Zheng, Jian Du, Yongtu Liang, Chang Wang, Qi Liao, and Haoran Zhang. Deeppipe: Theory-guided lstm method for monitoring pressure after multi-product pipeline shutdown. *Process Safety and Environmental Protection*, 155:518–531, 2021.

[10] Yunlu Ma, Jianqin Zheng, Yongtu Liang, Jiří Jaromír Klemeš, Jian Du, Qi Liao, Hongfang Lu, and Bohong Wang. Deeppipe: Theory-guided neural network method for predicting burst pressure of corroded pipelines. *Process Safety and Environmental Protection*, 162:595–609, 2022.

[11] Ming-ming Sun, Hong-yuan Fang, Nian-nian Wang, Xue-ming Du, Hai-sheng Zhao, and Ke-Jie Zhai. Limit state equation and failure pressure prediction model of pipeline with complex loading. *Nature Communications*, 15(1):4473, 2024.

[12] Mouad Sidki, Nikolay Tchernev, Pierre Féniès, Libo Ren, and Selwa Elfirdoussi. Heuristic based decision approach for an integrated slurry pipeline network scheduling in the phosphate industry. *Expert Systems with Applications*, 269:126495, 2025.

[13] Qi Liao, Haoran Zhang, Ning Xu, Yongtu Liang, and Junao Wang. A milp model based on flowrate database for detailed scheduling of a multi-product pipeline with multiple pump stations. *Computers & Chemical Engineering*, 117:63–81, 2018.

[14] Renfu Tu, Hao Zhang, Bin Xu, Xiaoyin Huang, Yiyuan Che, Jian Du, Chang Wang, Rui Qiu, and Yongtu Liang. Machine learning application in batch scheduling for multi-product pipelines: A review. *Journal of Pipeline Science and Engineering*, 4(3):100180, 2024.

[15] Muhammad Waqar, Azhar M. Memon, Moez Louati, Mohamed S. Ghidaoui, Luai M. Alhems, Silvia Meniconi, Bruno Brunone, and Caterina Capponi. Pipeline leak detection using hydraulic transients and domain-guided machine learning. *Mechanical Systems and Signal Processing*, 224:111967, 2025.

[16] J.A. Delgado-Aguiñaga, V. Puig, and F.I. Becerra-López. Leak diagnosis in pipelines based on a kalman filter for linear parameter varying systems. *Control Engineering Practice*, 115:104888, 2021.

[17] Mohamed S. Ghidaoui, Ming Zhao, Duncan A. McInnis, and David H. Axworthy. A review of water hammer theory and practice. *Applied Mechanics Reviews*, 58(1):49–76, 03 2005.

[18] M. Hanif Chaudhry. *Characteristics and Finite-Difference Methods*, pages 65–113. Springer New York, New York, NY, 2014.

[19] Jiawei Ye, Nhu Cuong Do, Wei Zeng, and Martin Lambert. Physics-informed neural networks for hydraulic transient analysis in pipeline systems. *Water Research*, 221:118828, 2022.

[20] James A. Liggett and Li-Chung Chen. Inverse transient analysis in pipe networks. *Journal of Hydraulic Engineering*, 120(8):934–955, 1994.

[21] Xun Wang and Mohamed S. Ghidaoui. Identification of multiple leaks in pipeline: Linearized model, maximum likelihood, and super-resolution localization. *Mechanical Systems and Signal Processing*, 107:529–548, 2018.

[22] Doosun Kang and Kevin Lansey. Demand and roughness estimation in water distribution systems. *Journal of Water Resources Planning and Management*, 137(1):20–30, 2011.

[23] Ahmad Malekpour and Yuntong She. Real-time leak detection in oil pipelines using an inverse transient analysis model. *Journal of Loss Prevention in the Process Industries*, 70:104411, 2021.

[24] BS Jung and BW Karney. Systematic exploration of pipeline network calibration using transients. *Journal of hydraulic research*, 46(sup1):129–137, 2008.

[25] Lei He, Kai Wen, Changchun Wu, Jing Gong, and Xie Ping. Hybrid method based on particle filter and narx for real-time flow rate estimation in multi-product pipelines. *Journal of Process Control*, 88:19–31, 2020.

[26] Chi Zhang and Abdollah Shafieezadeh. Nested physics-informed neural network for analysis of transient flows in natural gas pipelines. *Engineering Applications of Artificial Intelligence*, 122:106073, 2023.

[27] Jianqin Zheng, Jian Du, Bohong Wang, Jiří Jaromír Klemeš, Qi Liao, and Yongtu Liang. A hybrid framework for forecasting power generation of multiple renewable energy sources. *Renewable and Sustainable Energy Reviews*, 172:113046, 2023.

[28] Jian Du, Jianqin Zheng, Yongtu Liang, Xinyi Lu, Jiří Jaromír Klemeš, Petar Sabev Varbanov, Khurram Shahzad, Muhammad Imtiaz Rashid, Arshid Mahmood Ali, Qi Liao, and Bohong Wang. A hybrid deep learning framework for predicting daily natural gas consumption. *Energy*, 257:124689, 2022.

[29] Jian Du, Jianqin Zheng, Yongtu Liang, Ning Xu, Jiří Jaromír Klemeš, Bohong Wang, Qi Liao, Petar Sabev Varbanov, Khurram Shahzad, and Arshid Mahmood Ali. Deeppipe: A two-stage physics-informed neural network for predicting mixed oil concentration distribution. *Energy*, 276:127452, 2023.

[30] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

A Supplementary Material of Proposed Method

A.1 Hydraulic simulation of liquid pipeline based on MOC

Hydraulic simulation process Using the MOC method, the transient hydraulic characteristic equations are derived through simplification and linear combination, as presented in Eqs. (11) and (12).

$$C^{+}\begin{cases}\frac{dx}{dt}=a\\ \frac{a}{gA}\frac{dQ}{dt}+\frac{dH}{dt}+fQ\left|Q\right|^{1-m}a=0\end{cases}\tag{11}$$

$$C^{-}\begin{cases}\frac{dx}{dt}=-a\\ \frac{a}{gA}\frac{dQ}{dt}-\frac{dH}{dt}+fQ\left|Q\right|^{1-m}a=0\end{cases}\tag{12}$$

Due to the non-differentiability of the friction terms in the characteristic equations, finite difference schemes are employed to approximate the spatial derivatives in Eqs. (11) and (12), as shown in Eqs. (13) and (14). The pipeline is discretized into segments (Δx = L/M), and the flow is computed over time steps (Δt = $\Delta x/a$).

$$C^{+}: \frac{a}{gA}\left(Q_{i,t+1}-Q_{i-1,t}\right)+\left(H_{i,t+1}-H_{i-1,t}\right)+fQ_{i,t+1}|Q_{i-1,t}|^{1-m}a\Delta t=0 \tag{13}$$

$$C^{-}: \frac{a}{gA}\left(Q_{i,t+1}-Q_{i+1,t}\right)-\left(H_{i,t+1}-H_{i+1,t}\right)+fQ_{i,t+1}|Q_{i+1,t}|^{1-m}a\Delta t=0 \tag{14}$$

Simultaneously solving Eqs. 13 and 14 establishes the relationship between variables at time t+1 and t, as expressed in Eqs. 15 and 16.

$$Q_{i,t+1}=\frac{R^{+}-R^{-}}{S^{+}+S^{-}}\tag{15}$$

$$H_{i,t+1}=R^{+}-S^{+}Q_{i,t+1}\tag{16}$$

where $R^{+} = H_{i-1,t} + C_W Q_{i-1,t},\, R^{-} = H_{i+1,t} - C_W Q_{i+1,t},\, S^{+} = C_W + f\left|Q_{i-1,t}\right|^{1-m}a\Delta t,\, S^{-} = C_W + f\left|Q_{i+1,t}\right|^{1-m}a\Delta t, C_W = \frac{a}{\mathrm{g}A}$. Thus, by specifying any two of the flowrate and pressure values at the pipeline inlet and outlet boundaries at time t+1, the hydraulic state along the pipeline at that moment can be determined.

Discussion of conventional parameter identification methods To estimate the friction coefficient and optimal states, existing parameter identification methods address the inverse problem by minimizing the squared error (SE) between estimated and observed flowrates over a time interval T:

$$\hat{f}=\underset{f}{\operatorname{argmin}}\left(\sum_{t=1}^{T}\left(Q_{0,t}^{est}-Q_{0,t}^{obs}\right)^{2}+\sum_{t=1}^{T}\left(Q_{M,t}^{est}-Q_{M,t}^{obs}\right)^{2}\right)\tag{17}$$

Conventional objective function (Eq. (17)) overlook the central role of the friction factor in governing spatiotemporal hydraulic dynamics. Existing parameter identification methods typically identify parameters using data from a fixed interval (e.g., 20 minutes) and apply them to the subsequent interval, leading to estimation deviations, especially in real-time transients where hydraulic conditions evolve within seconds. This study thus proposes an interpretable evaluation criterion (Eq. (17)) to enable synchronous parameter identification for real-time simulation.

A.2 Forward and backward propagation in PG-ARNN

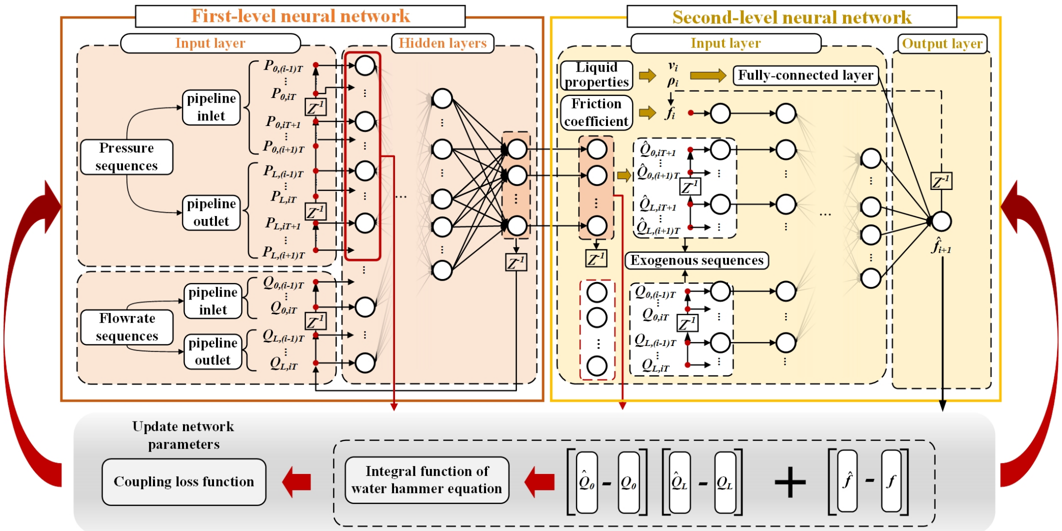


Figure A.1: Schematic diagram of PG-ARNN

As depicted in Figure A.1, the first-level network takes the estimated boundary flowrate from $(i–1)T$ to iT and the observed boundary pressure from $(i–1)T$ to $(i+1)T$ as inputs to predict flowrate in the interval iT to $(i+1)T$, as defined in Eq. (18).

$$\begin{aligned}\left(\boldsymbol{Q}_{0,\boldsymbol{iT}+1},\cdots,\boldsymbol{Q}_{0,(\boldsymbol{i}+1)\boldsymbol{T}},\boldsymbol{Q}_{\boldsymbol{L},\boldsymbol{iT}+1},\cdots,\boldsymbol{Q}_{\boldsymbol{L},(\boldsymbol{i}+1)\boldsymbol{T}}\right) &= \\ MAN_1\left(\begin{matrix}\boldsymbol{Q}_{0,(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{Q}_{0,\boldsymbol{iT}},\boldsymbol{Q}_{\boldsymbol{L},(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{Q}_{\boldsymbol{L},\boldsymbol{iT}}, \\ \boldsymbol{P}_{0,(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{P}_{0,(\boldsymbol{i}+1)\boldsymbol{T}},\boldsymbol{P}_{\boldsymbol{L},(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{P}_{\boldsymbol{L},(\boldsymbol{i}+1)\boldsymbol{T}};\theta_1\end{matrix}\right)\end{aligned} \tag{18}$$

where MAN_1 is the multilayer autoregressive neural network in the first layer, and θ_1 is the trainable parameters in the neural network of the first layer. $Q_{0,t}$ and $Q_{L,t}$ represent the flowrate in the pipeline inlet and outlet. $P_{0,t}$ and $P_{l,t}$ represent the pressure in the pipeline inlet and outlet.

The estimated present flowrate of the first-level network is concatenated with previous flowrate as input features. This composite input, along with the friction coefficient at the i^{th} interval, is fed into the second-level network. Additionally, fluid density and viscosity are processed via a fully connected (FC) layer to infer the friction coefficient at the $(i+1)^{th}$ interval. The forward propagation of this second-level network is detailed from Eqs. (19)–(21).

$$\boldsymbol{Z}_1 = MAN_2\left(\boldsymbol{Q}_{0,(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{Q}_{0,(\boldsymbol{i}+1)\boldsymbol{T}},\boldsymbol{Q}_{\boldsymbol{L},(\boldsymbol{i}-1)\boldsymbol{T}},\cdots,\boldsymbol{Q}_{\boldsymbol{L},(\boldsymbol{i}+1)\boldsymbol{T}},\boldsymbol{f}_{\boldsymbol{i}};\theta_2\right) \tag{19}$$

$$Z_2 = W_{fc}X + b_{fc} \tag{20}$$

$$\hat{f}_{i+1} = W_o(Z_1 \oplus Z_2) + b_o \tag{21}$$

where X represents the input matrix consisting of liquid properties elements. θ_2 is the trainable parameters in the neural network of the second layer. (W_{fc}, b_{fc}) and (W_0, b_0) are the weights and biases in the FC layer and output layer. $\oplus$ represents the feature-wise concatenation. where $\hat{f}_{i+1}$ is the observed friction coefficient.

The mean squared errors (MSE) between predicted and observed results of first and second-level networks can be represented as:

$$L_{first} = \frac{1}{N} \left(\|FLN_{Q_L} - Q_L\|_2^2 + \|FLN_{Q_0} - Q_0\|_2^2 \right) \tag{22}$$

$$L_{seond} = \frac{1}{N} \|SLN_f - f\|_2^2 \tag{23}$$

To end with, the coupling loss function $L = L_{seond} + L_{first}$ can be used to train the dual-layer neural network.

B Supplementary Figures and Tables

Table B.1: The properties of example pipelines and transported liquids

Cases	Pipeline properties			Liquid properties		
	Outer diameter (mm)	Length (km)	Wall thickness (mm)	Density ($kg \cdot m^{-3}$)	Viscosity (mmPa·s)	Volume elasticity modulus(Pa)
Case 1	406.4	39.9	7.1	742	0.72	9.2×10^8
Case 2	323.9	55.1	6.4	753	1.12	4.2×10^8
Case 3	219.1	32.1	5.6	825	5.33	1.5×10^9
Case 4	219.1	45.7	5.6	821	5.27	1.3×10^9

Table B.2: The hyper-parameter setting of PG-ARNN

Network section	Hyper-parameters	Range	Time interval	
			10 seconds	5 minutes
First-level network	Number of layers	1-10	3	5
	Neural units	10-3000	[100, 50]	[2000, 500, 10]
	Batch size	16-512	256	512
	Activation function	[Relu, Tanh, Sigmoid]	Relu	Relu
	Dropout	0-0.5	0.1	0.1
Second-level network	Number of layers	1-10	2	2
	Neural units	8-128	[60, 20]	[700, 100]
	Batch size	10-1000	256	512
	Activation function	[Relu, Tanh, Sigmoid]	Relu	Relu
	Dropout	0-0.5	0.1	0.1

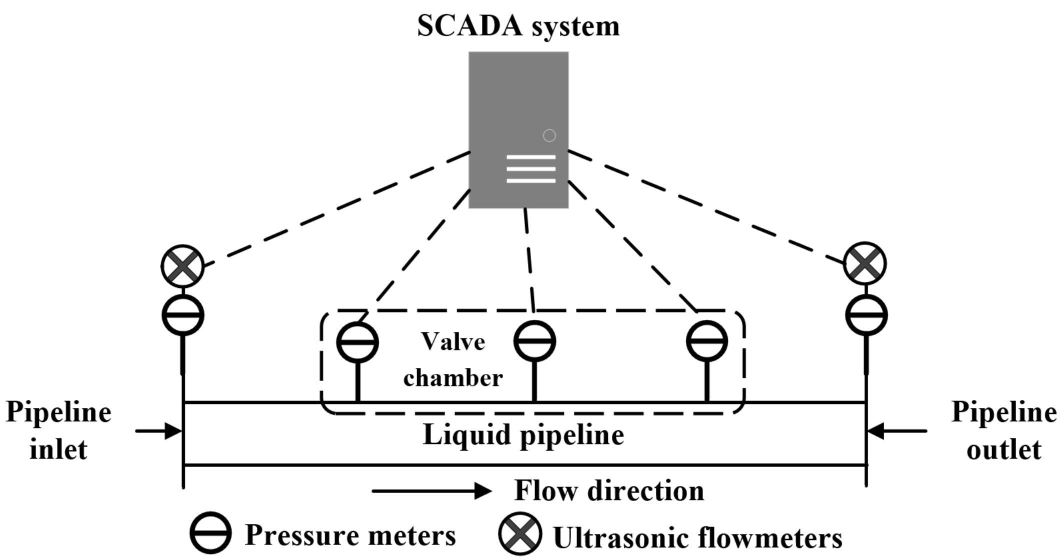
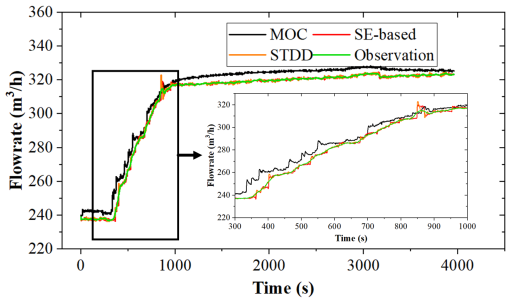
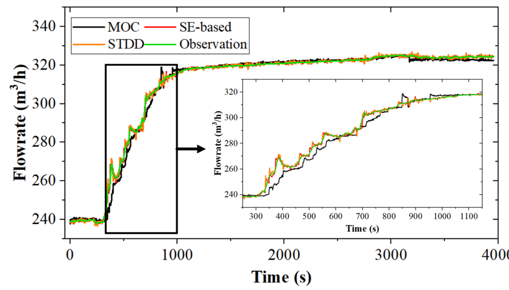


Figure B.1: Schematic diagram of liquid pipeline (SCADA: Supervisory Control and Data Acquisition system)

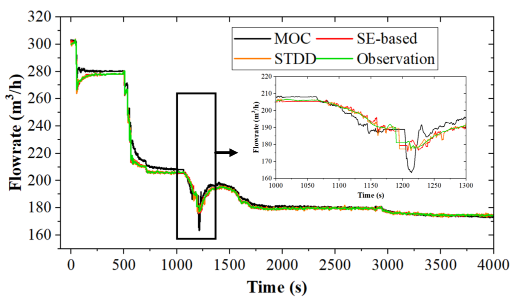


(i) Inlet flowrate estimation

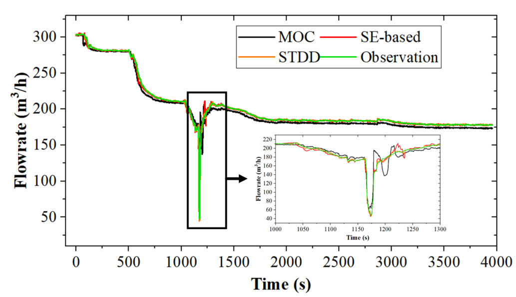


(ii) Outlet flowrate estimation

(a) Case2

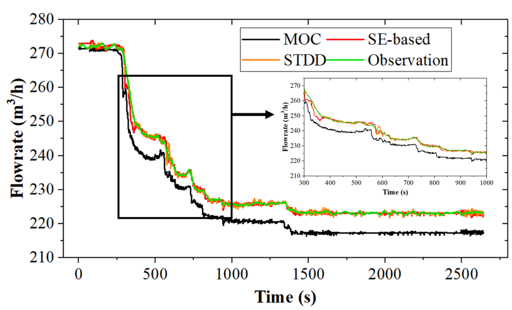


(i) Inlet flowrate estimation

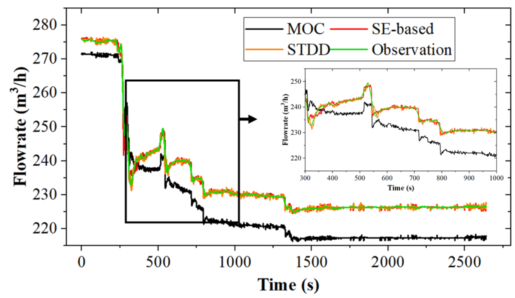


(ii) Outlet flowrate estimation

(b) Case 3



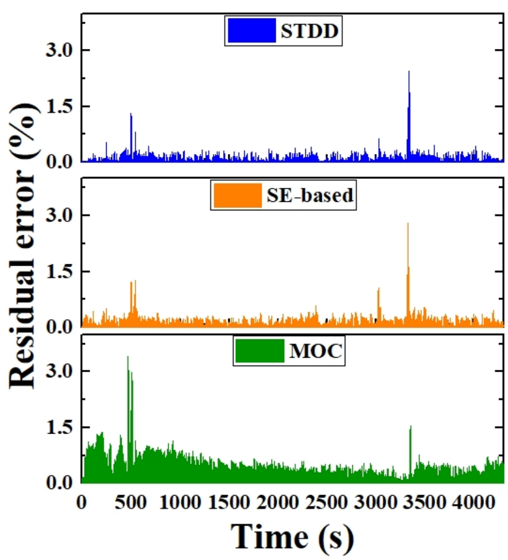
(i) Inlet flowrate estimation



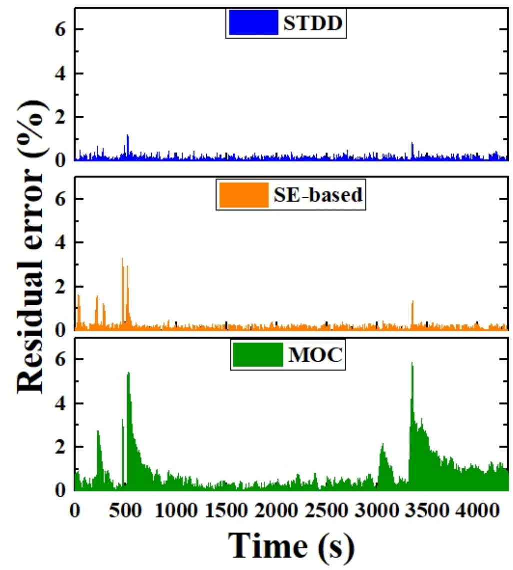
(ii) Outlet flowrate estimation

(c) Case4

Figure B.2: Visualization comparison of flowrate estimation between SE-based method and STDD

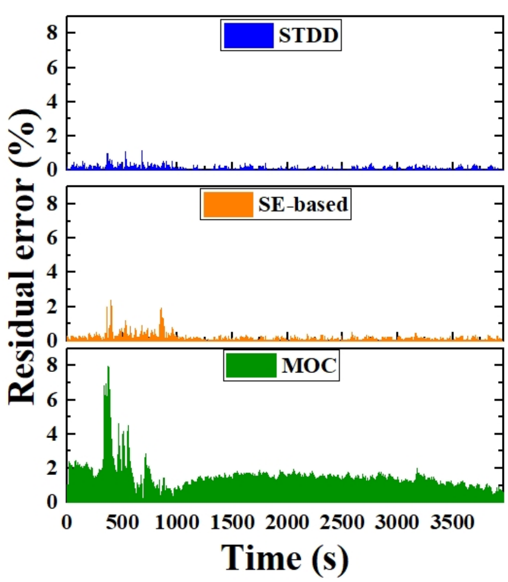


(i) Inlet flowrate estimation

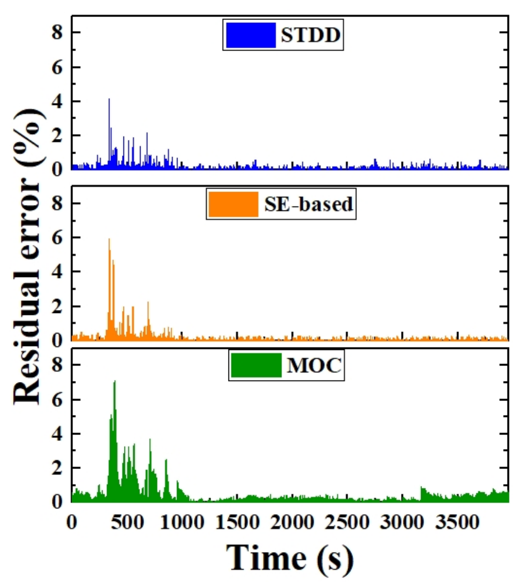


(ii) Outlet flowrate estimation

(a) Case1

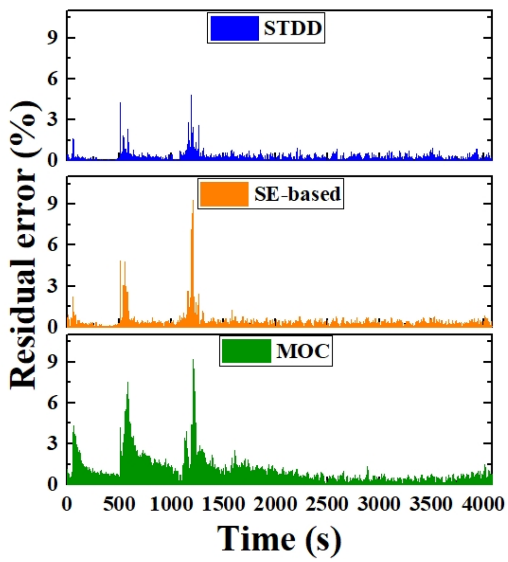


(i) Inlet flowrate estimation

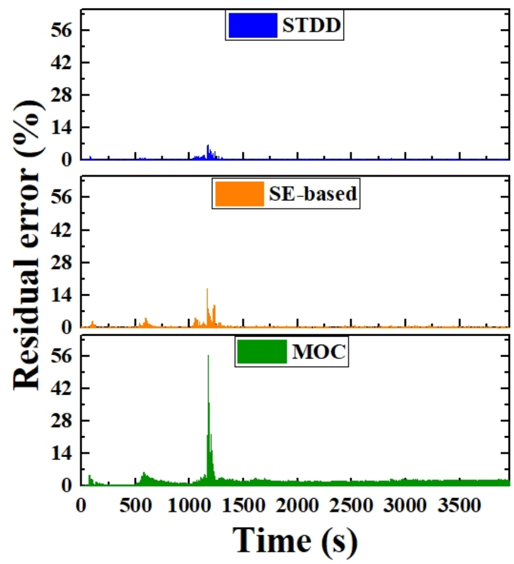


(ii) Outlet flowrate estimation

(b) Case2

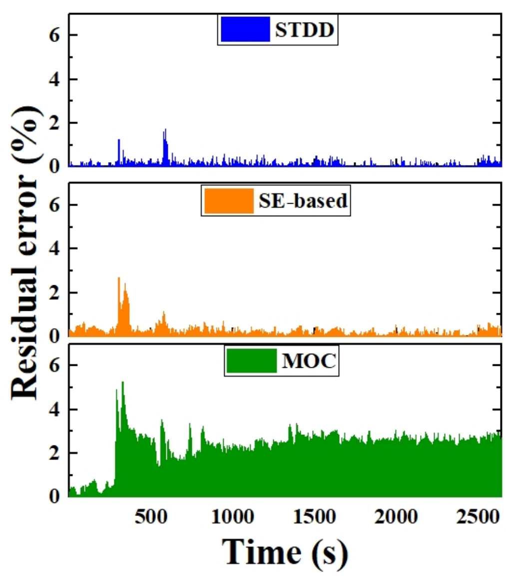


(i) Inlet flowrate estimation

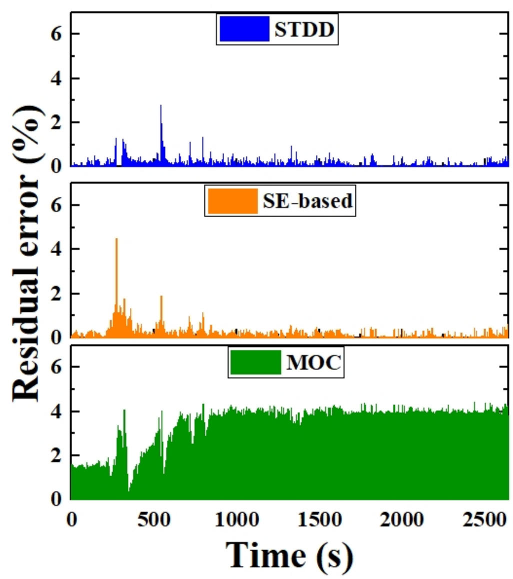


(ii) Outlet flowrate estimation

(c) Case 3



(i) Inlet flowrate estimation



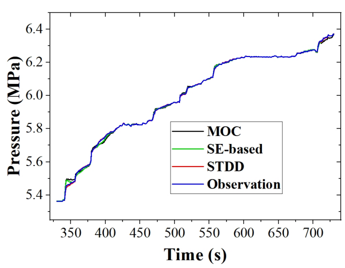
(ii) Outlet flowrate estimation

(d) Case4

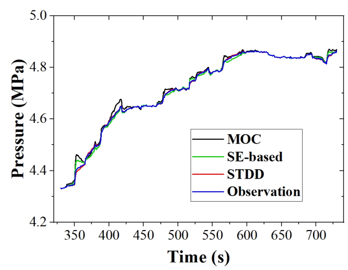
Figure B.3: Visualization comparison of absolute residual errors between SE-based method and STDD

Table B.3: MAPE comparisons of flowrate estimation between SE-based method and STDD

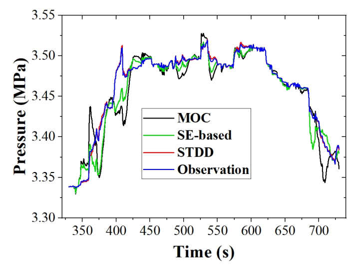
Cases	Methods	Inlet flowrate estimation (%)			Outlet flowrate estimation (%)		
		Whole flow process	Transient condition	Pseudo-steady condition	Whole flow process	Transient condition	Pseudo-steady condition
Case 2	STDD	**0.073**	**0.149**	**0.056**	**0.069**	**0.134**	**0.054**
	MOC	0.452	1.358	0.247	1.441	2.062	1.531
	SE-based	0.104	0.283	0.064	0.125	0.319	0.081
Case 3	STDD	**0.201**	**0.720**	**0.129**	**0.174**	**0.334**	**0.152**
	MOC	2.210	3.932	1.997	0.930	1.905	0.753
	SE-based	0.357	1.375	0.222	0.268	0.797	0.198
Case 4	STDD	**0.110**	**0.321**	**0.091**	**0.094**	**0.188**	**0.076**
	MOC	3.366	3.559	1.805	2.281	2.505	2.237
	SE-based	0.158	0.714	0.115	0.292	0.913	0.172



(i) 14.4 km

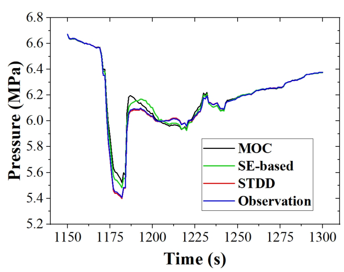


(ii) 28.7 km

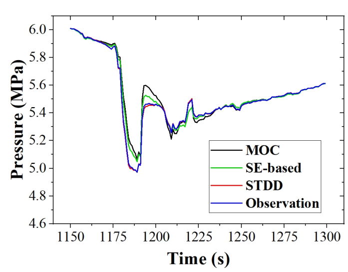


(iii) 42.6 km

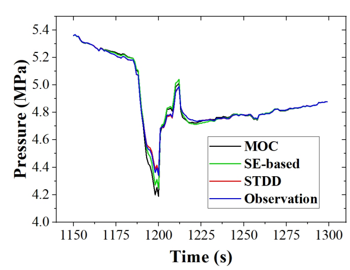
(b) Case2



(i) 5.5 km

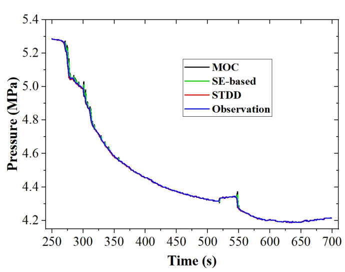


(ii) 15.6 km

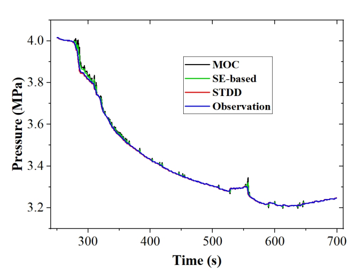


(iii) 25.6 km

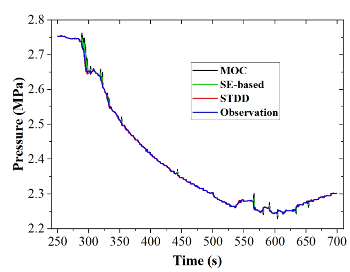
(c) Case 3



(i) 11.5 km



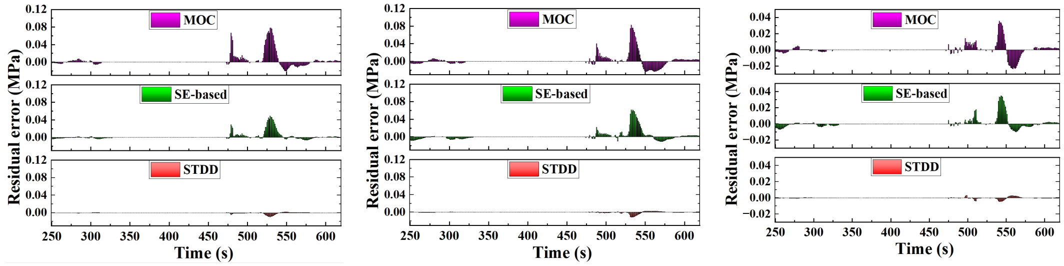
(ii) 23.6 km



(iii) 33.8 km

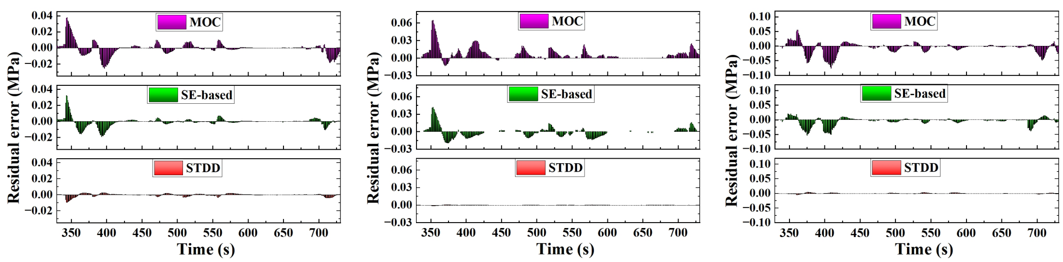
(d) Case4

Figure B.4: Results comparison of simulated pressure at different locations between SE-based methods and STDD



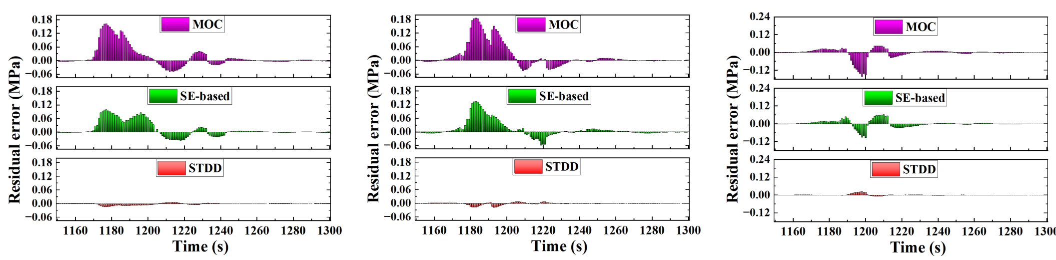
(i) 9.6 km (ii) 19.6 km (iii) 29.6 km

(a) Case1



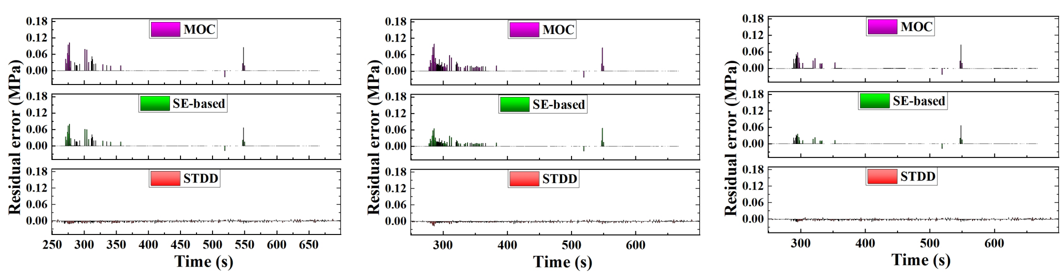
(i) 14.4 km (ii) 28.7 km (iii) 42.6 km

(b) Case2



(i) 5.5 km (ii) 15.6 km (iii) 25.6 km

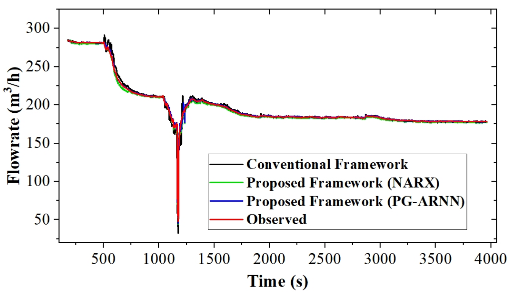
(c) Case 3



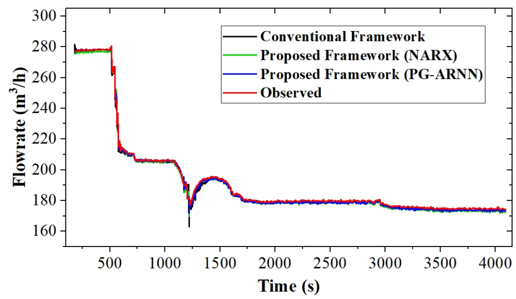
(i) 11.5 km (ii) 23.6 km (iii) 33.8 km

(d) Case4

Figure B.5: Residuals comparison of simulated pressure at different locations between SE-based method and STDD

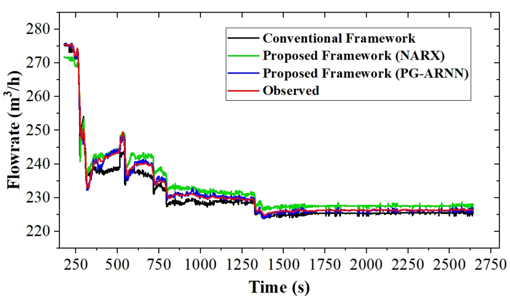


(i) Inlet flowrate estimation

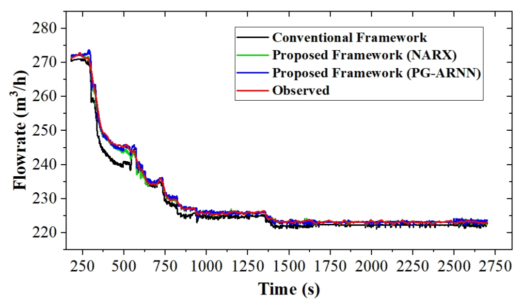


(ii) Outlet flowrate estimation

(c) Case 3



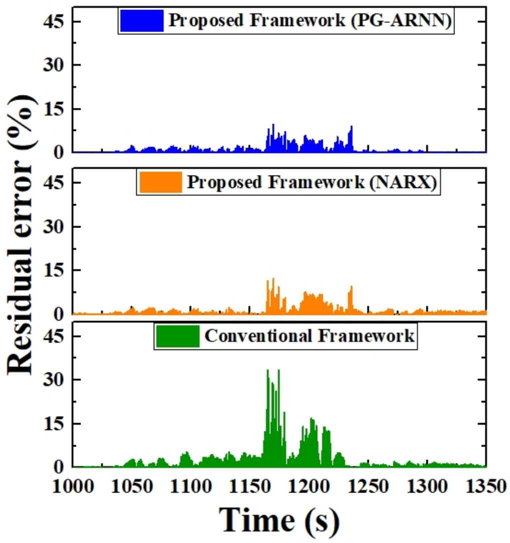
(i) Inlet flowrate estimation



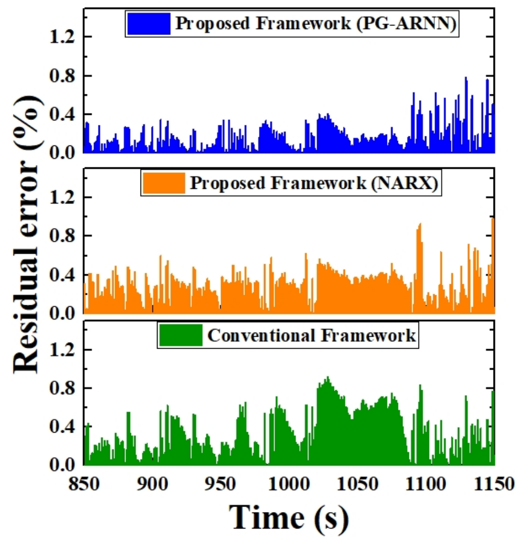
(ii) Outlet flowrate estimation

(d) Case4

Figure B.6: Results comparison of different online real-time simulation frameworks

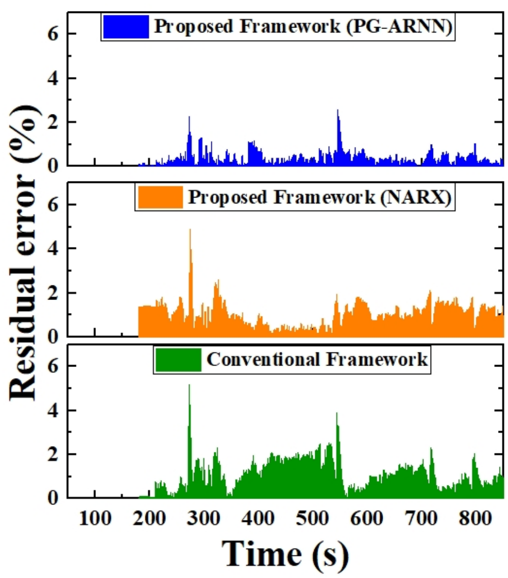


(i) Inlet flowrate estimation

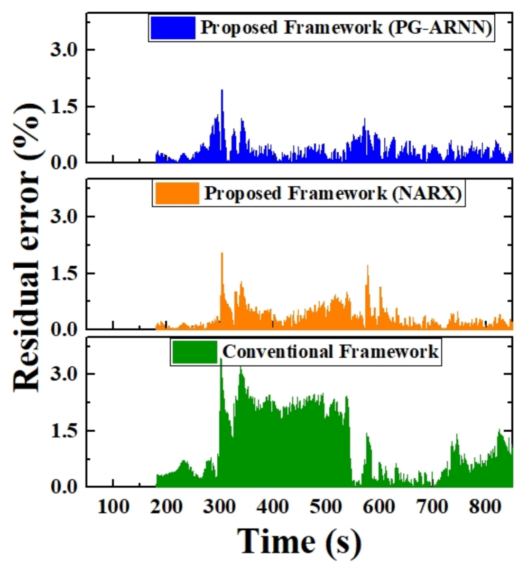


(ii) Outlet flowrate estimation

(c) Case 3



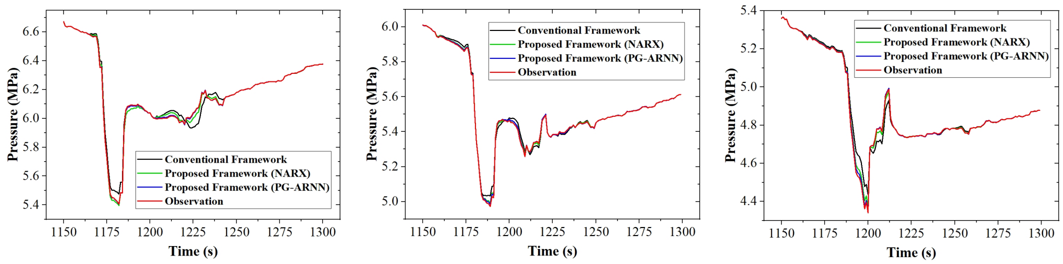
(i) Inlet flowrate estimation



(ii) Outlet flowrate estimation

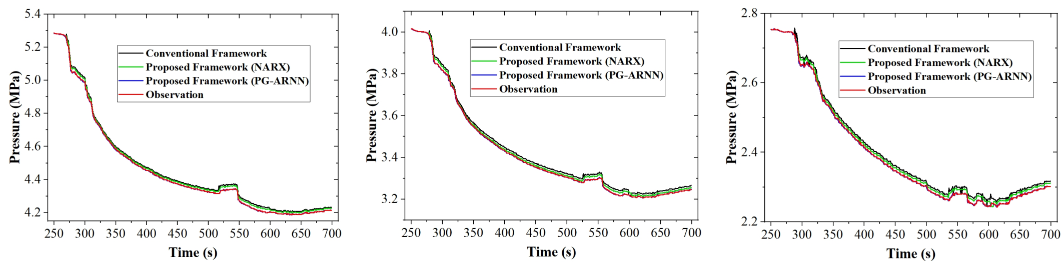
(d) Case4

Figure B.7: Absolute residual errors of different online real-time simulation frameworks



(i) 5.5 km (ii) 15.6 km (iii) 25.6 km

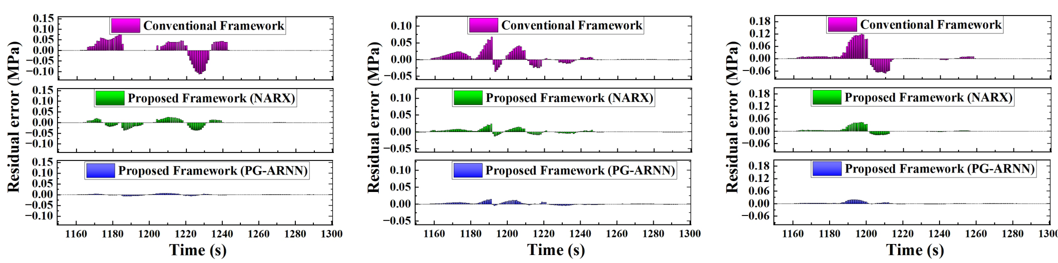
(a) Case3



(i) 11.5 km (ii) 23.6 km (iii) 33.8 km

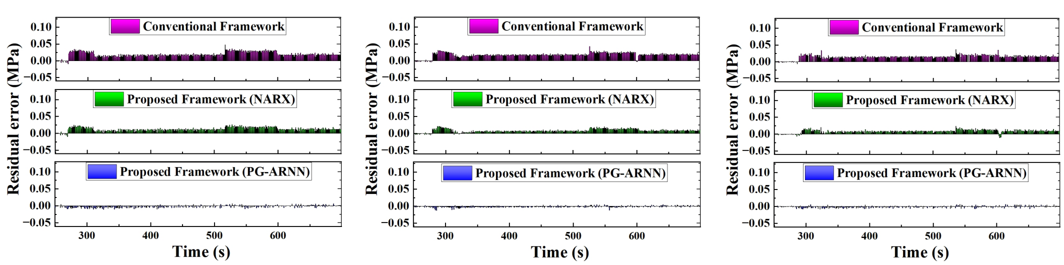
(b) Case4

Figure B.8: Results comparison of simulated pressure at different locations between various frameworks



(i) 5.5 km (ii) 15.6 km (iii) 25.6 km

(a) Case3



(i) 11.5 km (ii) 23.6 km (iii) 33.8 km

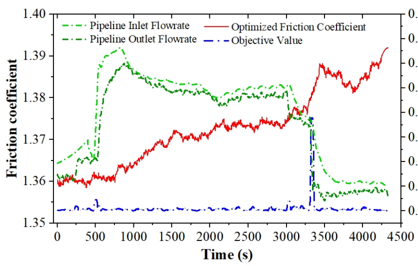
(b) Case4

Figure B.9: Residual errors of simulated pressure at different locations between various frameworks

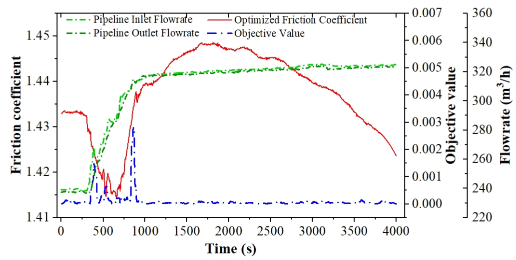
C Additional Experimental Results

C.1 Interpretability analysis of hydraulic spatiotemporal dynamics identification

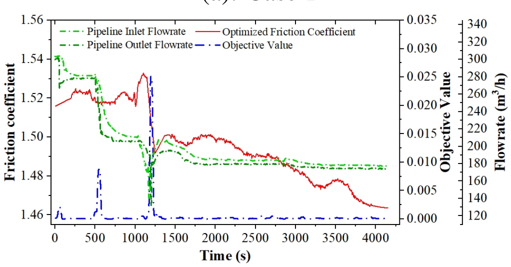
As illustrated in Figure C.1, STDD is conducted at 10-second interval to identify friction coefficients across various cases. The results reveal a strong correlation between the friction coefficient and hydraulic parameters, with clear phase synchronization to flowrate dynamics. Notably, under pronounced flowrate fluctuations, the friction coefficient exhibits rapid transient changes. This highlights the physical interpretability of the proposed STDD, which captures variations in transient hydraulic behavior induced by friction changes through spatiotemporal derivative residuals. Furthermore, the larger objective values can be found during fast-transient process, which demonstrates the significant challenge in precisely reconstructing abrupt hydraulic dynamic features.



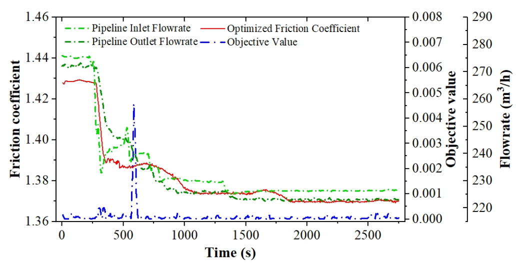
(a): Case 1



(b): Case 2



(c): Case 3



(d): Case 4

Figure C.1: Optimization results of friction coefficient with time interval being 10 seconds

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: Data and code will be available if request.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: This paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.