PIMRL: Physics-Informed Multi-Scale Recurrent Learning for Burst-Sampled Spatiotemporal Dynamics

Han Wan*, Qi Wang*, Yuan Mi, Rui Zhang†, Hao Sun†

Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{wanhan2001, 2021000184, miyuan, rayzhang, haosun}@ruc.edu.cn

Abstract

Deep learning has shown strong potential in modeling complex spatiotemporal dynamics. However, most existing methods depend on densely and uniformly sampled data, which is often unavailable in practice due to sensor and cost limitations. In many real-world settings, such as mobile sensing and physical experiments, data are burst-sampled with short high-frequency segments followed by long gaps, making it difficult to learn accurate dynamics from sparse observations. To address this issue, we propose Physics-Informed Multi-Scale Recurrent Learning (PIMRL), a novel framework specifically designed for burst-sampled spatiotemporal data. PIMRL combines macro-scale latent dynamics inference with micro-scale adaptive refinement guided by incomplete prior information from partial differential equations (PDEs). It further introduces a temporal message-passing mechanism to effectively propagate information across burst intervals. This multi-scale architecture enables PIMRL to model complex systems accurately even under severe data scarcity. We evaluate our approach on five benchmark datasets involving 1D to 3D multi-scale PDEs. The results show that PIMRL consistently outperforms state-of-the-art baselines, achieving substantial improvements and reducing errors by up to 80% in the most challenging settings, which demonstrates the clear advantage of our model. Our work demonstrates the effectiveness of physics-informed recurrent learning for accurate and efficient modeling of sparse spatiotemporal systems.

Introduction

Learning dynamical systems from data is a central challenge in scientific machine learning. PDE-governed physical systems are ubiquitous in disciplines such as biology, chemistry, and meteorology (Anderson and Wendt 1995; Moukalled, Mangani, and Darwish 2016). While traditional numerical solvers are reliable, direct numerical simulation (DNS) is often computationally prohibitive. It typically requires high spatial and temporal resolutions, long runtimes, and complete knowledge of the governing PDEs, model parameters, and initial/boundary conditions (Ferziger, Perić, and Street 2019; Goc et al. 2021).

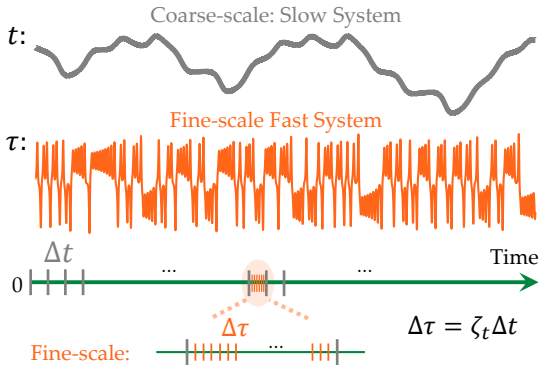

Figure 1: Multi-scale sampling, where $\Delta\tau$ denotes the micro-scale time interval for fast dynamics, Δt the macro-scale time interval for slow dynamics, and ζ_t the scale separation variable (typically $\zeta_t < 1$ or $\zeta_t \ll 1$).

To alleviate the high computational cost of DNS, data-driven deep learning methods such as DeepONet (Lu et al. 2021), FNO (Li et al. 2021), and DPOT (Hao et al. 2024; Azizzadenesheli et al. 2024) aim to learn solution operators directly from data. These approaches bypass the need for explicit PDEs and enable fast inference by learning mappings between function spaces. During rollout, they operate with relatively large time steps, thereby reducing simulation time significantly compared to classical solvers. However, their success critically depends on the availability of large, high-quality datasets, which are often expensive to obtain via numerical simulations or physical experiments (Kanov et al. 2015; Li et al. 2024a).

To reduce data dependency, physics-informed learning has emerged as an effective paradigm. These methods embed physical knowledge into the training process either softly, by augmenting the loss function with PDE residuals, e.g., PINNs (Raissi, Perdikaris, and Karniadakis 2019), PINO (Li et al. 2024b), MCNP (Zhang et al. 2025), or more strictly, by designing model architectures that obey physical rules through discrete approximations, e.g., PeRCNN (Rao et al. 2023, 2022), P^{2}C^{2}Net (Wang et al. 2024), LDSolver (Yan et al. 2025), and TSM (Sun, Yang, and Yoo 2023). While these approaches improve physical fidelity and reduce labeled data requirements, they still face challenges

*These authors contributed equally.
†Corresponding authors: Rui Zhang and Hao Sun.

in long-term forecasting. Due to their reliance on local update schemes, they typically require small time steps for numerical stability, which can lead to cumulative errors and increased computational cost over long horizons.

Since the performance of neural solvers is closely tied to the quality and quantity of training data, the strategy employed during data acquisition plays a critical role. However, most existing works assume that data are sampled uniformly or continuously over time. This assumption does not hold in many real-world scenarios, where data collection is often constrained by factors such as high experimental costs, limited temporal resolution, and hardware limitations. In practical applications including mobile sensing, in situ physical experiments, and computational fluid dynamics simulations, data are frequently gathered in a sparse and non-uniform manner, often across multiple temporal scales. Such irregular sampling patterns introduce information gaps and temporal inconsistencies, which can significantly degrade the performance of learning-based models that rely on temporally regular inputs (Champion, Brunton, and Kutz 2019).

Among alternative sampling strategies, burst sampling has recently emerged as a promising approach. It captures multiple high-frequency samples within short time intervals to record fast-changing or transient dynamics, while alternating with low-frequency or missing data periods to conserve resources. This scheme has demonstrated the ability to capture both rapid and slow dynamics efficiently (see Figure 1). However, despite its growing practical relevance, burst-sampled, multi-scale data remains largely unexplored in the context of deep learning for spatiotemporal systems.

To address this underexplored yet practically important challenge, we propose a framework named Physics-Informed Multi-Scale Recurrent Learning (PIMRL). PIMRL is designed to model sparse, non-uniform, and burst-sampled spatiotemporal data by coupling latent-space macro-scale inference with micro-scale physical correction. The macro module operates in a learned latent space to perform efficient temporal reasoning and accelerate simulation, while the micro module enforces fine-grained physical consistency using pretrained PDE priors. To integrate information across time and resolution scales, we introduce a dedicated cross-scale message passing mechanism that fuses information between the two modules, ensuring stability and accuracy during long-term rollouts. Our contributions are summarized as follows:

- We propose a Physics-Informed Multi-Scale Recurrent Learning (PIMRL) framework that effectively leverages information from multi-scale, burst-sampled data to improve long-term spatiotemporal dynamics prediction.
- We develop a cross-scale message passing mechanism that fuses physical information between macro and micro modules, ensuring stable latent state evolution and mitigating error accumulation during long rollouts.
- Our framework achieves optimal performance in effectively predicting the dynamics of diverse systems ranging from fluid flows to general physical phenomena, demonstrating scalability and generalizability under data-scarce, irregular sampling regimes.

Related Work

Accurate and efficient simulation of partial differential equations (PDEs) is essential for many scientific and engineering tasks, but traditional numerical methods are computationally costly. Although deep learning can accelerate simulations, these methods often require extensive physical knowledge or large datasets, resulting in poor performance when such resources are limited. In particular, current approaches struggle to model physical systems from sparse measurements without detailed physical priors.

Direct Numerical Simulation. Direct Numerical Simulation (DNS) is a computational method that solves the Navier–Stokes equations directly, capturing all spatial and temporal scales of turbulent flows without turbulence modeling (Anderson and Wendt 1995; Moukalled, Mangani, and Darwish 2016). Although DNS yields highly accurate results, it is extremely computationally expensive. This immense computational cost and memory demand often limit DNS to relatively simple or small-scale problems, restricting its practical applicability. Furthermore, when the governing equations contain unknown parameters or missing physics, DNS may still struggle to provide reliable predictions. To overcome these challenges, PIMRL is specifically designed to efficiently predict outcomes by leveraging limited physical prior knowledge and data.

Deep Learning Methods. With the advancement of AI, its application to physical system simulation has become increasingly diverse and profound. For example, classical convolutional neural networks (CNNs) (Bar-Sinai et al. 2019), ResNet (Lu et al. 2018), graph neural networks (GNNs) (Sanchez-Gonzalez et al. 2020; Pfaff et al. 2021), and transformer-based models (Wu et al. 2024; Hang et al. 2024; Janny et al. 2023; Li, Shu, and Barati Farimani 2024) have been widely adopted. Neural operators for learning mappings between function spaces have also seen rapid development, such as DeepONet (Lu et al. 2021), FNO (Li et al. 2021; Rahman, Ross, and Azizzadenesheli 2023; Wen et al. 2022), and PIANO (Zhang, Meng, and Ma 2024).

Physics-Informed Deep Learning Methods. To incorporate physical knowledge, researchers have developed various physics-informed approaches, such as PhyCRNet (Ren et al. 2022), PINN (Raissi, Perdikaris, and Karniadakis 2019), PeSANet (Wan et al. 2025), and PDE-Net (Long et al. 2018). These methods embed physical laws into deep models to improve prediction accuracy. PeRCNN (Rao et al. 2023, 2022) employs a hard encoding of prior knowledge, enabling strong predictive and generalization capabilities even with limited data. However, most existing methods still suffer from error accumulation, which limits their long-term stability and accuracy. Notably, PeRCNN's effective physical encoding strategy can be further leveraged in our proposed PIMRL framework.

Methodology

Partial Differential Equations Simulation Task

Simulation tasks are closely related to PDEs, which fundamentally describe and model various physical phenomena,

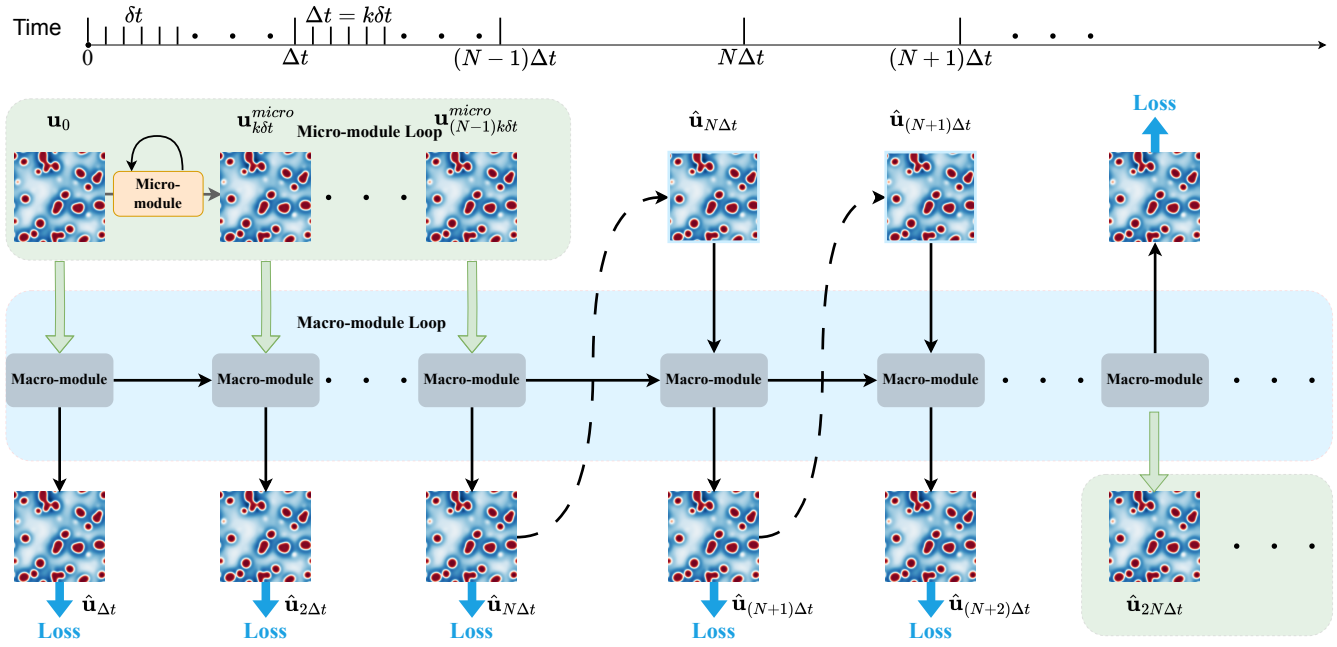

Figure 2: The overall framework architecture, which integrates physics-informed constraints with deep learning. The initial state of the system is denoted by $\mathbf{u}_0$. The state predicted by the micro-module after k iterations, where each iteration occurs at intervals of δt, is represented by $\mathbf{u}^{micro}_{k\delta t}$. The predicted value of the physical state from PIMRL is denoted by $\hat{\mathbf{u}}$.

with the general form of a time-dependent PDE:

$$\mathbf{u}_t = \mathcal{F}(t, x, \mathbf{u}, \nabla\mathbf{u}, \mathbf{u}\cdot\nabla\mathbf{u}, \nabla^2\mathbf{u}, \ldots; \mu), \qquad (1)$$

where $\mathbf{u}(x,t)$ denotes the spatiotemporal solution field, $\mathbf{u}_t$ is the first-order time derivative, $\mathcal{F}(\cdot)$ is a (possibly nonlinear) operator, ∇ represents the gradient (nabla) operator, ∇^2 is the Laplacian, and μ denotes the set of PDE parameters.

In addition, the initial and boundary conditions (ICs and BCs) are specified as, for example, $I[\mathbf{u}](x, t=0) = 0$ and $B[\mathbf{u}](x,t) = 0$, where $I[\cdot]$ and $B[\cdot]$ denote the operators for the initial and boundary conditions, respectively.

Overview

We aim to develop a model capable of predicting the evolution of spatiotemporal dynamics for nonlinear PDE systems over a long-term horizon, based on limited training data (e.g., a few trajectories generated from different initial conditions). As outlined in the introduction, we have designed the PIMRL to mitigate error accumulation and ensure that it effectively captures the intrinsic changes in the physical system, rather than overlooking significant physical information over extended time intervals. Furthermore, PIMRL is capable of efficiently utilizing multi-scale data.

As shown in Figure 2, PIMRL consists of two main components: a macro-scale module and a micro-scale module. The macro-module performs fast, coarse-grained inference in the latent space with large time steps, accelerating prediction and reducing error accumulation. The micro-module conducts fine-grained reasoning and adjusts the latent historical states. The message-passing mechanism enables interaction between these modules: the micro-module provides physical knowledge and correction signals to the macro-module, while the macro-module passes updated information to guide future micro-module iterations.

PIMRL operates in a recursive manner, involving rollouts at the micro-module level, the macro-module level, and the overall framework level. Only the output from the macro-module contributes to the final output of the PIMRL framework and is used to compute the loss for training the entire framework, whereas the output from the micro-module refines and updates the latent historical states. The message-passing mechanism we proposed is described as follows:

- Firstly, the micro-module rollout is a simple autoregressive process with time step δt, where the output at the previous time step serves as the input for the next step.
- When the micro-module is involved in the prediction, for every k steps of micro-module with δt like Equation 2, the final output of micro-module is passed to the macro-module, and at this point, the output from the macro-module serves as the output of the entire PIMRL model shown as Equation 3. When the micro-module is not involved in the prediction, the macro-module rollout is a simple autoregressive process with time step Δt.
- Finally, there is a PIMRL loop that operates in conjunction with the macro-module rollout. After every $N-1$ rollout of the macro-module, the micro-module stops participating in the prediction, and the macro-module performs N steps of autoregressive prediction on its own. This completes a total of $2N$ rollouts. Each output from the macro-module during these $2N$ rollouts serves as the output of PIMRL as depicted in Equation 4.

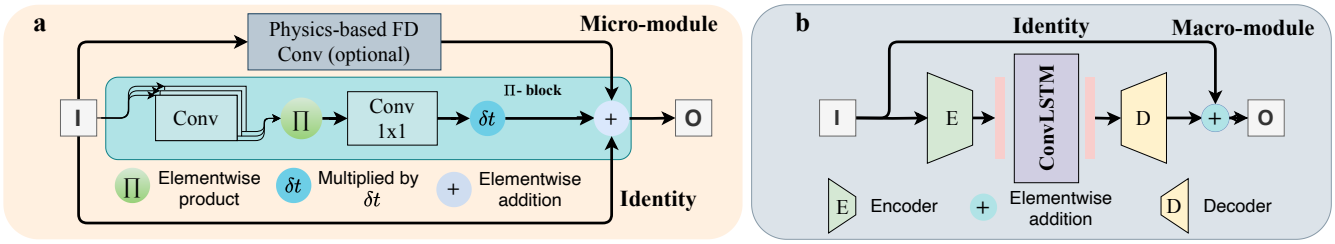

Figure 3: PIMRL includes two main modules: (a) the micro-module, designed to capture local features and small-scale dynamics; and (b) the macro-module, which captures long-term dependencies and global patterns using residual connections.

Cross-scale Message Passing

We conceptualize the interaction between the micro-scale module (F_{micro}) and the macro-scale module (F_{macro}) as a **cross-scale message passing** framework, illustrated in Figure 2. The process is governed by the following operations:

First, during a data-rich burst, the micro-module generates a physically-informed "message" by performing a high-fidelity rollout over k fine steps. This message generation process is defined by the **Micro-module Loop**:

$$\text{Micro-module Loop: } \mathbf{u}^{\text{micro}}_{t+k\delta t} = \underbrace{F_{\text{micro}}(...F_{\text{micro}}}_{\times k}(\mathbf{u}_t)) \quad (2)$$

Next, this message, $\mathbf{u}^{\text{micro}}_{t+k\delta t}$, is passed to the macro-module, which consumes it to correct its own state. This state update is described by the **Macro-module Loop**:

$$\text{Macro-module Loop: } \hat{\mathbf{u}}_{t+2k\delta t} = F_{\text{macro}}(\mathbf{u}^{\text{micro}}_{t+k\delta t}) \quad (3)$$

Finally, after being corrected by the message, the macro-module evolves autoregressively for efficient long-term forecasting. This complete, multi-step prediction cycle is summarized by the overall **PIMRL** process:

$$\text{PIMRL: } \hat{\mathbf{u}}_{t+2N\Delta t} = \underbrace{F_{\text{macro}}(...F_{\text{macro}}}_{\times N}(\hat{\mathbf{u}}_{t+Nk\delta t})) \quad (4)$$

where the $\mathbf{u}$ denotes the physical state. The relationship between the micro-time step δt and the macro-time step Δt is given by $\Delta t = k\delta t$, where k is an adjustable parameter.

Micro-scale Module

The micro-scale module is designed to learn underlying physical laws that govern the spatiotemporal dynamics from micro-scale data with small time stepping, where we adopt the PeRCNN model (Rao et al. 2023) with the architecture of Π-block shown in Figure 3(a). In a forward Euler scheme: $\mathbf{u}_{(k+1)\delta t} = \hat{\mathcal{F}}(\mathbf{u}_{k\delta t}) \cdot \delta t + \mathbf{u}_{k\delta t}$, where δt denotes that the module predicting in micro-scale time stepping. We can then approximate the $\mathcal{F}$ by $\hat{\mathcal{F}}$ described as follows:

$$\hat{\mathcal{F}}(\mathbf{u}_{k\delta t}) = \sum_{c=1}^{N_c} W_c \cdot \left[\prod_{l=1}^{N_l}(K_{c,l} \star \hat{\mathbf{u}}_{k\delta t} + b_l)\right], \quad (5)$$

where N_c denotes the channel count, and N_l the total number of parallel convolutional layers. The symbol $\star$ denotes the convolutional operation. For each layer l and channel c, $K_{c,l}$ designates the specific filter weight, while b_l stands for the bias term of that layer l. In the context of a 1×1 convolutional layer, W_c denotes the weight assigned to the c^{th} channel, with the bias term being omitted for the sake of simplicity and brevity. When a certain term in the governing PDE remains known (e.g., the Laplace operator $\nabla^2\mathbf{u}$), its discretization can be directly embedded in PeRCNN (called the physics-based Conv layer as shown in Figure 3(a)). The convolutional kernel in such a layer can be set according to the corresponding finite difference (FD) stencil. In essence, the physics-based Conv connection is constructed to incorporate known physical principles, whereas the Π-block is aimed at capturing the complementary unknown dynamics. The details of the physics-based FD Conv are provided in the Physical Filter part of Appendix A.

Macro-scale Module

The macro-scale module serves as the long-term predictive module of PIMRL, specifically engineered to bridge the large temporal gaps inherent in burst-sampled data. Instead of operating in the computationally expensive physical space, it performs inference within a compact latent space. This design choice is motivated by the need for both computational efficiency and the expressive power to capture complex, long-range dependencies that are intractable for traditional, fine-grained solvers.

The module is implemented as a residual ConvLSTM autoencoder (Figure 3; Appendix B), with components operating together within a message-passing framework. An encoder maps the input to a latent space, where the ConvLSTM cell propagates the system's state forward in time with large, efficient steps (Δt). Crucially, this latent-space evolution is not a pure black-box rollout. It is periodically corrected by physics-informed messages from the micro-module. This mechanism allows the macro-module to anchor its latent trajectory to physical reality, preventing the error accumulation and drift that plague purely data-driven recurrent models during long-term forecasting. By integrating the efficiency of latent-space dynamics with the physical fidelity of micro-scale corrections, our macro-scale module achieves a robust and accurate forecasting capability.

Experiments

To validate the effectiveness of our proposed PIMRL framework, we conducted extensive experiments on a diverse set of fluid dynamics and reaction-diffusion systems.

Case	Numerical Methods	Spatial Grid	Macro Step	Micro Step	Training Trajectories	Test Trajectories
KdV	FVM	256	0.01s	0.01s	5	2
Burgers	FDM	128^2	0.001s	0.01s	13	3
FN	FDM	128^2	0.5s	0.01s	5	3
2D GS	FDM	128^2	0.002s	0.01s	2	3
3D GS	FDM	96^3	0.25s	0.01s	3	2

Table 1: Summary of experimental settings for different cases. The 3D GS case is downsampled from 96^3 to 48^3 during training.

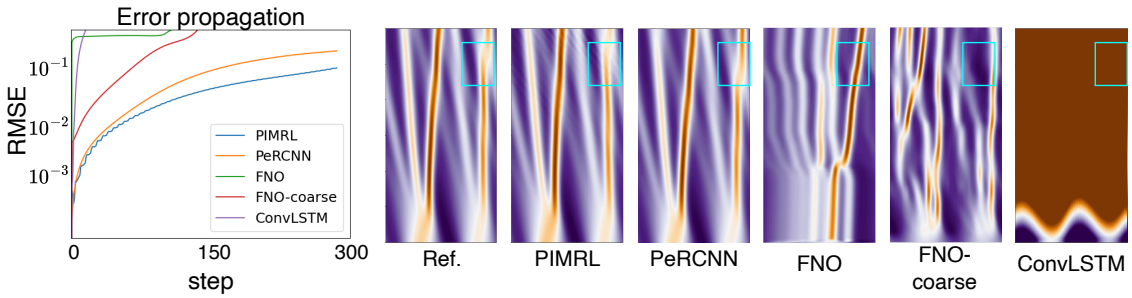

Figure 4: Error propagation curves and final prediction plots for the KdV case, comparing the PIMRL and baseline models.

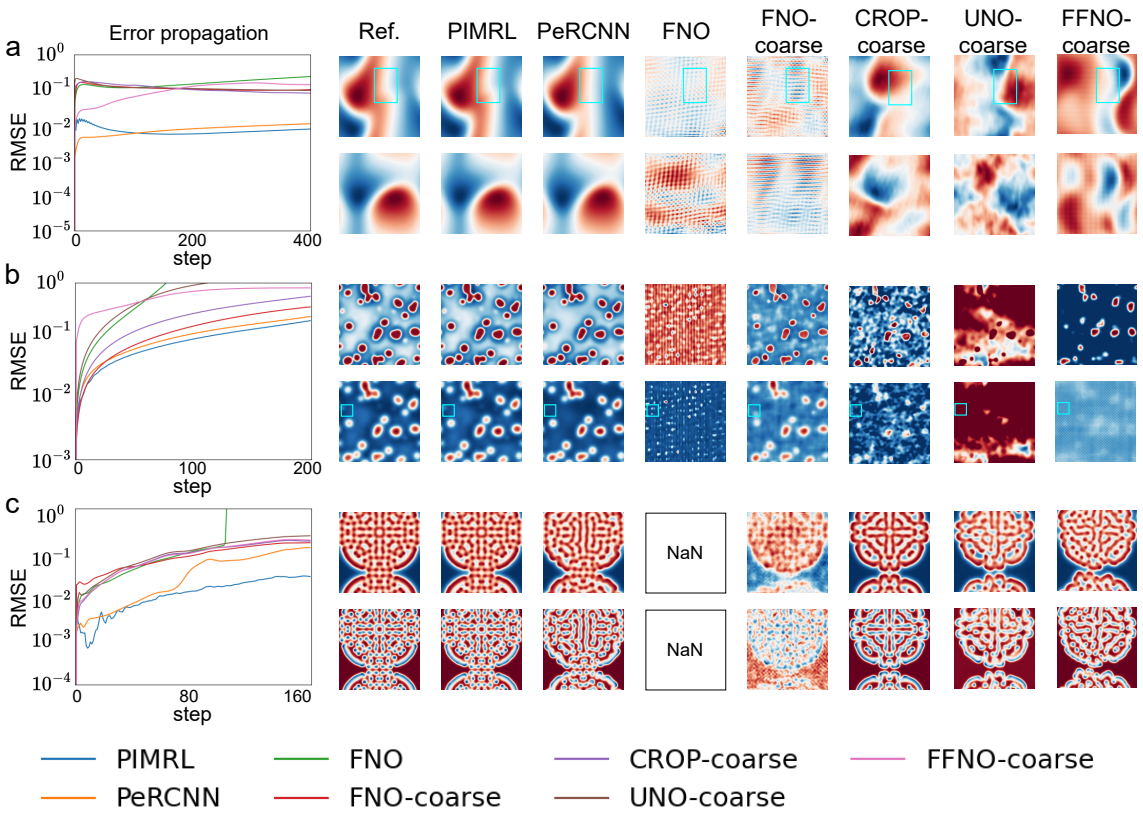

Figure 5: Figures (a)–(c) summarize qualitative comparisons with baselines on Burgers, FN, and 2D GS, showing error propagation and final predictions.

Datasets. We conducted experiments on five datasets, including Korteweg-de Vries (KdV), Burgers, FitzHugh-Nagumo (FN), 2D Gray-Scott (2D GS), and 3D Gray-Scott (3D GS). The IC for the KdV equation is created by summing multiple sine waves with random amplitudes, phases, and frequencies, resulting in a complex waveform. ICs for the Burgers' equation are generated randomly according to a Gaussian distribution. The FN equation is initialized with

random Gaussian noise for a warm-up period, after which time sequences are extracted to form the dataset. The GS equation starts the reaction from random initial positions and then diffuses. In these cases, except for the KdV case, which was solved using the Finite Volume Method (FVM), the rest were solved using the Finite Difference Method (FDM).

Additionally, we have two sets of data with different time scales originating from the same ICs. The micro-scale data $\mathbf{U}^{\text{micro}} = \{\mathbf{u}_0, \mathbf{u}_{\delta t}, \mathbf{u}_{2\delta t} \cdots\} \in \mathbf{R}^{\text{micro}}$ is characterized by short and scattered continuous time intervals, while the macro-scale data $\mathbf{U}^{\text{macro}} = \{\mathbf{u}_0, \mathbf{u}_{\Delta t}, \mathbf{u}_{2\Delta t} \cdots \mathbf{u}_{T_{\text{end}}}\} \in \mathbf{R}^{\text{macro}}$ exhibits persistent continuity until the end. The validation and test sets are established based on different ICs but with the same parameters, making it more challenging than extrapolation under the same ICs.

Baseline Models. To validate the effectiveness of the proposed PIMRL framework, we introduced several baseline models. Firstly, we considered the widely recognized high-performing data-driven model FNO (Li et al. 2021), which has been trained on datasets with two different time intervals, denoted as FNO (trained on fine-scale data with small time steps) and FNO-coarse (trained on coarse-scale data with large time steps). Secondly, we included the PeRCNN model (Rao et al. 2023), which embeds physical knowledge in a hard way. Due to constraints on the time stepping of the model, PeRCNN is trained on datasets with small time steps. Lastly, we incorporated CROP (Gao et al. 2025)(CROP-coarse trained on coarse-scale data with large time steps), a method proposed to address problems with generalization and discretization mismatch errors in existing neural operators across different data resolutions. Since the original CROP paper did not include 1D and 3D cases, we employ AE-ConvLSTM (Vlachas et al. 2022) as a substitute. The details are shown in Appendix C.

Evaluation Metrics. To comprehensively evaluate the performance of our model, we adopted metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and High Correction Time (HCT). RMSE is calculated on the macro-scale data to facilitate comparisons between models operating at different granularities. MAE provides a measure of the average absolute difference between the predicted and actual values. HCT evaluates the time it takes for the model to correct its predictions to a high level of accuracy. Detailed formulas for these metrics are provided in the Evaluation Metrics part of Appendix E.

Main Results

Figure 5 compares our framework with baseline models across cases using box plots of predicted quantities. Table 2 reports the corresponding quantitative metrics.

1D KdV Equation. The KdV equation governs nonlinear wave evolution. Notably, significant discrepancies between PeRCNN predictions and ground truth emerge in the blue box of Figure 4, whereas other baselines exhibit poor initial performance attributable to KdV complexity. Conversely, PIMRL demonstrates substantial advantages in predictive accuracy over baseline models, achieving 30%–50% improvements in evaluation metrics (Table 2) and thereby highlighting significant field advancement.

Case	Model	RMSE ↓	MAE ↓	HCT (s) ↑
KdV	ConvLSTM	5.8507	7.6036	9.6
	FNO	0.4891	0.3300	0.45
	FNO-coarse	0.5461	0.4167	7.8
	PeRCNN	0.0942	0.0941	30
	PIMRL(Ours)	**0.0457**	**0.0607**	**46.2**
	Promotion	51.5%	35.5%	54.0%
Burgers	CROP-coarse	0.1103	0.0887	0.064
	FNO	0.1561	0.1301	0.104
	FNO-coarse	0.1094	0.0879	0.064
	PeRCNN	0.0075	0.0058	3.216
	PIMRL(Ours)	**0.0068**	**0.0049**	**3.216**
	Promotion	9.3%	15.6%	0%*
FN	CROP-coarse	0.3950	0.2734	3.03
	FNO	937	2393980	1.65
	FNO-coarse	0.1878	0.1643	4.98
	PeRCNN	0.1591	0.1139	6.99
	PIMRL(Ours)	**0.1349**	**0.0990**	**7.74**
	Promotion	15.2%	13.1%	10.7%
2D GS	CROP-coarse	0.1027	0.0579	1260
	FNO	NaN	NaN	810
	FNO-coarse	0.0884	0.0629	1335
	PeRCNN	0.0455	0.0268	1379.5
	PIMRL(Ours)	**0.0133**	**0.0072**	**1965***
	Promotion	70.8%	73.1%	42.5%
3D GS	ConvLSTM	0.2081	0.2009	56.5
	FNO	0.2798	0.1950	112.5
	FNO-coarse	0.1042	0.0611	360
	PeRCNN	0.0532	0.0977	510
	PIMRL(Ours)	**0.0381**	**0.0190**	**731.25**
	Promotion	28.4%	80.6%	43.4%

Table 2: Quantitative results of our model and baselines, where $*$ denotes that the prediction lasts to the end.

2D Burgers Equation. Figure 5(a) demonstrates that only PeRCNN and our framework attain satisfactory post-training results on micro-scale data, while other baselines fail to capture physical changes. In the blue-boxed regions, PeRCNN shows significant long-term forecast errors compared to PIMRL's sustained accuracy. Corresponding quantitative improvements in RMSE and MAE are documented in Table 2, with PeRCNN's divergence further confirmed at the final time step.

2D FitzHugh-Nagumo Equation. In Figure 5(b), the purely data-driven method fails to achieve its original performance on this relatively small dataset. Quantitatively, our model outperforms the best existing model by at least 10%. Additionally, Table 2 shows that while the RMSE and other metric curves of our model and PeRCNN are similar, our model consistently achieves better results.

2D Gray-Scott Equation. As shown in the left part of Figure 5(c), only PIMRL effectively carried out long-term predictions, showcasing a clear demonstration of the cumulative errors of PeRCNN in this case. In this case, Table 2 further validates the superior performance of PIMRL with

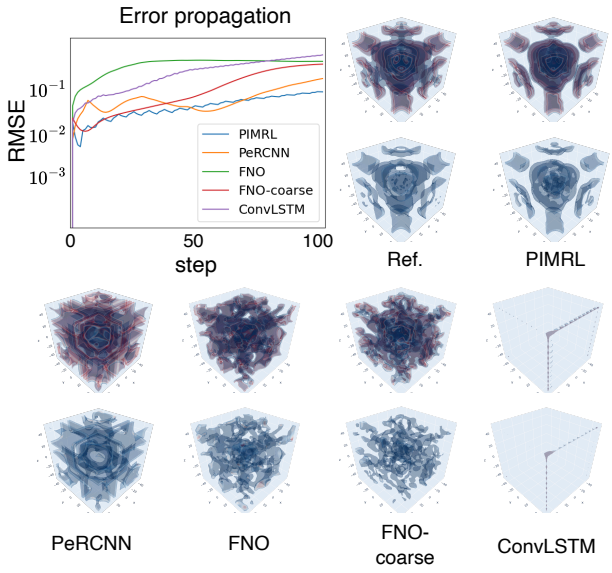

Figure 6: Error propagation curves and final prediction plots for the 3D GS, comparing the PIMRL and baseline models.

Model	RMSE
PIMRL w/o Connect	0.1975
FNO-MRL	0.7854
PIMRL w/o Pretraining	0.2599
PIMRL w/o Physics-based FD Conv	0.1738
PIMRL	0.1349

Table 3: Results for ablation study in FN-case.

both RMSE and MAE showing an improvement of over 70%, and the HCT similarity of PIMRL at 0.99 at the final prediction step. This highlights the effectiveness of PIMRL in improving predictive accuracy and consistency throughout forecasting, demonstrating its potential for tackling complex challenges.

3D Gray-Scott Equation. As shown on the right side of Figure 6, the predictions under our framework closely align with the ground truth. Through box plots and error evolution graphs, it is evident that PeRCNN, as the best-performing model among the baselines in our study, outperformed the other models lacking physical knowledge embeddings, especially when compared to FNO trained on the same dataset. In quantitative analysis of Table 2, the improvements are also significant, not only in the overall evaluation metrics of RMSE and MAE in 28.4% and 80.6% but also in the substantial growth of the HCT in 43.4%.

Ablation Study

We designed five novel models and report them in Table 3.

(1) The ablation study with the "PIMRL w/o Connect", where the connections between the micro-module and the macro-module are removed, is designed to demonstrate the effectiveness of the PIMRL framework's structural design. This experiment, which leaves only the serial structure of the two modules, shows that the connections within the PIMRL framework are essential. (2) The "FNO-MRL" replaces the micro-scale module containing physical information, PeRCNN, with the data-driven model FNO, aiming to validate the efficacy of the physical embedding in the micro-scale module within our framework. (3) "PIMRL w/o Pretraining" provides a perspective on the training method by eliminating the pretraining step for the micro module. In this ablation study, the absence of pretraining leads to inferior performance compared to the PIMRL. This demonstrates that directly introducing large time intervals for training can deprive the micro module of the opportunity to learn fine-grained changes, similar to why PeRCNN cannot be directly used with large time intervals. (4) "PIMRL w/o Physics-based FD Conv" indicates the removal of the Physics-based FD Convolution. This ablation study emphasizes the effectiveness of the Physics-based FD Conv by showing the performance degradation when it is omitted. (5) "PeRCNN w/o Physics-based FD Conv" is a version of PeRCNN without the Physics-based Finite Difference (FD) Convolution.

Addition Results

To comprehensively evaluate PIMRL, we conducted a series of supplementary experiments, the details of which are provided in Appendix G. In particular, we assessed computational efficiency by comparing runtime, parameter count, and memory usage with other physics-embedded methods, and examined the model's scalability and practicality in various scenarios. Furthermore, we investigated the robustness of PIMRL under different training set sizes and analyzed the reliability of its predictions using error bar analyses. These additional experiments collectively demonstrate the promising efficiency, scalability, and robustness of PIMRL.

Conclusion

In this paper, we introduced the Physics-Informed Multi-Scale Recurrent Learning (PIMRL) framework, the first, to our knowledge, specifically designed to predict the evolution of spatiotemporal systems from multi-scale, burst-sampled data. Its novel architecture synergistically combines a macro-scale module for efficient long-term forecasting with a physics-informed micro-scale module for periodic correction, a design that effectively mitigates cumulative error while capturing underlying physical laws from extremely limited observations. Rigorous evaluations confirmed its substantial performance gains over state-of-the-art baselines, including a reduction in RMSE of up to 70% on the 2D Gray-Scott dataset, thereby validating the effectiveness of this multi-scale, physics-informed approach.

While we acknowledge the framework's current limitations, such as the yet unexplored sensitivity to the duration of data gaps and the presence of input noise, as detailed in Appendix H, we view them as clear and important avenues for future research. Going forward, we will improve the macro-module's efficiency and integrate advanced super-resolution to better exploit the model's potential, enhance generalization, and widen its impact in scientific computing.

Acknowledgments

The work is supported by the Beijing Natural Science Foundation (No. 1232009) and the National Natural Science Foundation of China (No. 62276269 and No. 62506367). R.Z. would like to acknowledge the supported by the China Postdoctoral Science Foundation under Grant Number 2025M771582 and the Postdoctoral Fellowship Program of CPSF under Grant Number GZB20250408. Additional experimental details and full appendices are available in the arXiv version of this paper at https://arxiv.org/abs/2503.10253.

References

Anderson, J. D.; and Wendt, J. 1995. *Computational Fluid Dynamics*, volume 206. Springer.

Azizzadenesheli, K.; Kovachki, N.; Li, Z.; Liu-Schiaffini, M.; Kossaifi, J.; and Anandkumar, A. 2024. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5): 320–328.

Bar-Sinai, Y.; Hoyer, S.; Hickey, J.; and Brenner, M. P. 2019. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31): 15344–15349.

Champion, K. P.; Brunton, S. L.; and Kutz, J. N. 2019. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1): 312–333.

Ferziger, J. H.; Perić, M.; and Street, R. L. 2019. *Computational methods for fluid dynamics*. springer.

Gao, W.; Xu, R.; Deng, Y.; and Liu, Y. 2025. Discretization-invariance? On the Discretization Mismatch Errors in Neural Operators. In *International Conference on Learning Representations*, 1–9.

Goc, K. A.; Lehmkuhl, O.; Park, G. I.; Bose, S. T.; and Moin, P. 2021. Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. *Flow*, 1: E14.

Hang, Z.; Ma, Y.; Wu, H.; Wang, H.; and Long, M. 2024. Unisolver: PDE-Conditional Transformers Are Universal PDE Solvers. *arXiv preprint arXiv:2405.17527*.

Hao, Z.; Su, C.; Liu, S.; Berner, J.; Ying, C.; Su, H.; Anandkumar, A.; Song, J.; and Zhu, J. 2024. DPOT: Auto-regressive denoising operator transformer for large-Scale PDE pre-training. In *International Conference on Machine Learning*, volume 703, 17616 – 17635. PMLR.

Janny, S.; Benetteau, A.; Thome, N.; Nadri, M.; Digne, J.; and Wolf, C. 2023. EAGLE: Large-scale learning of turbulent fluid dynamics with mesh transformers. In *International Conference on Learning Representations*, 1–9.

Kanov, K.; Burns, R.; Lalescu, C.; and Eyink, G. 2015. The Johns Hopkins turbulence databases: an open simulation laboratory for turbulence research. *Computing in Science & Engineering*, 17(5): 10–17.

Li, Z.; Han, W.; Zhang, Y.; Fu, Q.; Li, J.; Qin, L.; Dong, R.; Sun, H.; Deng, Y.; and Yang, L. 2024a. Learning spatiotemporal dynamics with a pretrained generative model. *Nature Machine Intelligence*, 6(12): 1566–1579.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2021. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 1–9.

Li, Z.; Shu, D.; and Barati Farimani, A. 2024. Scalable transformer for pde surrogate modeling. In *Advances in Neural Information Processing Systems*, volume 1216, 28010 – 28039.

Li, Z.; Zheng, H.; Kovachki, N.; Jin, D.; Chen, H.; Liu, B.; Azizzadenesheli, K.; and Anandkumar, A. 2024b. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3): 1–27.

Long, Z.; Lu, Y.; Ma, X.; and Dong, B. 2018. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, 3208–3216. PMLR.

Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; and Karniadakis, G. E. 2021. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3): 218–229.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, 3276–3285. PMLR.

Moukalled, F.; Mangani, L.; and Darwish, M. 2016. *The finite volume method in computational fluid dynamics*. Springer.

Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; and Battaglia, P. W. 2021. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*, 1–9.

Rahman, M. A.; Ross, Z. E.; and Azizzadenesheli, K. 2023. U-no: U-shaped neural operators. *Transactions on Machine Learning Research*.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. 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.

Rao, C.; Ren, P.; Liu, Y.; and Sun, H. 2022. Discovering nonlinear PDEs from scarce data with physics-encoded learning. In *International Conference on Learning Representations*, 1–9.

Rao, C.; Ren, P.; Wang, Q.; Buyukozturk, O.; Sun, H.; and Liu, Y. 2023. Encoding physics to learn reaction–diffusion processes. *Nature Machine Intelligence*, 5(7): 765–779.

Ren, P.; Rao, C.; Liu, Y.; Wang, J.-X.; and Sun, H. 2022. PhyCRNet: Physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. *Computer Methods in Applied Mechanics and Engineering*, 389: 114399.

Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; and Battaglia, P. 2020. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, 8459–8468. PMLR.

Sun, Z.; Yang, Y.; and Yoo, S. 2023. A neural pde solver with temporal stencil modeling. In *International Conference on Machine Learning*, 33135–33155. PMLR.

Vlachas, P. R.; Arampatzis, G.; Uhler, C.; and Koumoutsakos, P. 2022. Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence*, 4(4): 359–366.

Wan, H.; Zhang, R.; Wang, Q.; Liu, Y.; and Sun, H. 2025. PeSANet: Physics-encoded Spectral Attention Network for Simulating PDE-Governed Complex Systems. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 7751–7759.

Wang, Q.; Ren, P.; Zhou, H.; Liu, X.-Y.; Deng, Z.; Zhang, Y.; Chengze, R.; Liu, H.; Wang, Z.; Wang, J.-X.; Wen, J.-R.; Sun, H.; and Liu, Y. 2024. P^2C^2Net: PDE-Preserved Coarse Correction Network for efficient prediction of spatiotemporal dynamics. In *Advances in Neural Information Processing Systems*, volume 38.

Wen, G.; Li, Z.; Azizzadenesheli, K.; Anandkumar, A.; and Benson, S. M. 2022. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163: 104180.

Wu, H.; Luo, H.; Wang, H.; Wang, J.; and Long, M. 2024. Transolver: A Fast Transformer Solver for PDEs on General Geometries. In *International Conference on Machine Learning*, volume 2200, 53681 – 53705.

Yan, M.; Wang, Q.; Wang, H.; Chengze, R.; Zhang, Y.; Liu, H.; Wang, Z.; Yu, F.; Qi, Q.; and Sun, H. 2025. Learnable-Differentiable Finite Volume Solver for Accelerated Simulation of Flows. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhang, R.; Meng, Q.; and Ma, Z.-M. 2024. Deciphering and integrating invariants for neural operator learning with various physical mechanisms. *National Science Review*, 11(4): nwad336.

Zhang, R.; Meng, Q.; Zhu, R.; Wang, Y.; Shi, W.; Zhang, S.; Ma, Z.-M.; and Liu, T.-Y. 2025. Monte Carlo neural PDE solver for learning PDEs via probabilistic representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6): 5059–5075.