REAL: REFINE NEURAL OPERATORS VIA ALIGNING DYNAMICAL FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural partial differential equation (PDE) surrogates trained with mean squared error (MSE) can capture dominant dynamics but often lose high-frequency information and fine-scale structures. Generative models (e.g., Flow Matching) can capture the distribution of fine-scale details, yet stochastic errors may accumulate during autoregressive rollouts and compromise long-term stability. To address these limitations, we introduce **Re**fine and **Al**ign (**ReAl**), a model-agnostic plug-in that augments pretrained PDE surrogates to improve the reconstruction of high-frequency structures associated with complex physical phenomena. In the refinement stage, a conditional flow-matching model learns the residual distribution around the deterministic base prediction, recovering missing fine-scale structures while retaining the dominant dynamics. In the alignment stage, to mitigate the generative discrepancies amplified in the autoregressive rollout, particularly in chaotic systems, we generate multiple trajectories from the same input and reinforce the refiner with trajectory-level rewards, thereby achieving long-horizon dynamical alignment. Experiments across diverse surrogate architectures and PDE tasks demonstrate that **ReAl** consistently improves high-frequency fidelity, forecasting accuracy, and long-horizon stability.

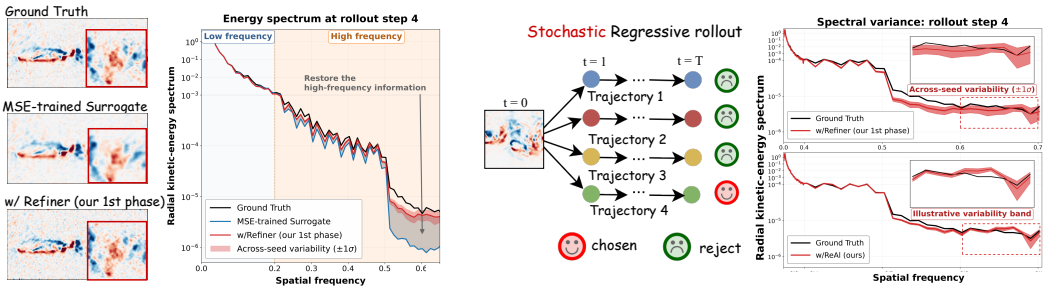
1 INTRODUCTION

Partial differential equations (PDEs) are fundamental to modeling physical systems across fluid mechanics (Davidson, 2004), heat and mass transfer (Bergman, 2011), electromagnetics (Schwinger et al., 1998), and chemical diffusion (Crank, 1979). Classical finite-difference (Morton & Mayers, 1994) and finite-element methods (Hughes, 2003) provide reliable solutions, but their computational costs grow rapidly with spatial resolution, dimensionality, and parameter-space size. This limitation has driven the development of deep-learning-based PDE surrogates and neural operators, including DeepONet (Lu et al., 2021), the Fourier Neural Operator (Li et al., 2020), and the Physics-Informed Neural Operator (Goswami et al., 2023). By approximating PDE solution operators, these methods enable substantially faster inference for large-scale simulation and real-time scientific prediction.

Most deep-learning-based PDE surrogates are trained with mean squared error (MSE) objective, which provides stable optimization and effectively captures large-scale, low-frequency dynamics. However, as a mean-regression objective, MSE tends to suppress high-frequency information, local vortices, and sharp interfaces (Lippe et al., 2023; Fotiadis et al., 2025), particularly on real-world datasets such as RealPDEBench (Hu et al., 2026), where measurement noise and unresolved physical processes further complicate the underlying dynamics. Generative models (Ho et al., 2020; Rombach et al., 2022; Lipman et al., 2023) can alleviate this limitation by modeling the conditional distribution of PDE solutions instead of producing only one deterministic prediction, enabling richer high-frequency details (Yao et al., 2026; Utkarsh et al., 2026; Hu et al., 2025). However, stochastic sampling may introduce predictive variance and compromise the dynamical consistency and long-term stability required by deterministic PDE systems.

Motivated by these observations, we adopt a complementary modeling strategy: retain the MSE-trained surrogate as a stable predictor of dominant dynamics and apply a conditional flow-based refiner to its output. Rather than regenerating the entire PDE state, the refiner is conditioned on the base prediction and learns to generate its fine-scale residual correction (Fotiadis et al., 2025). As shown in Fig.1 (a), this strategy recovers high-frequency information lost by MSE training

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



(a) Necessity of refinement

(b) Necessity of alignment

Figure 1: (a) MSE training leads to high-frequency information loss, while stochastic refiner recovers the structures. (b) Sampling variability accumulates during stochastic autoregressive rollout, especially in the high-frequency regime. Our **ReAl** mitigates this accumulation of sampling variability.

while preserving low-frequency dynamics consistent in PDE predictions. Despite these benefits, a generative refiner can only partially mitigate the rollout shift rather than completely eliminate it. Stochastic perturbations can accumulate during autoregressive rollout, creating a gap between recovering fine-scale structures at individual steps and maintaining forecasting accuracy over the resulting trajectory (Luo, 2025; Yoo et al., 2026). As shown in Fig.1 (b), the base model preserves low-frequency dynamics over longer horizons, while high-frequency errors progressively accumulate and amplify, degrading long-term stability and physical consistency.

These observations motivate optimizing the refiner beyond individual prediction quality by accounting for its cumulative effect on the entire autoregressive trajectory. To this end, inspired by reinforcement learning for diffusion models (Liu et al., 2025; Zheng et al., 2026), we introduce trajectory-alignment post-training for the flow-based refiner. Starting from the same input state, we sample multiple autoregressive trajectories using independent random seeds and evaluate their long-term quality with trajectory-level rewards. These rewards guide flow-matching updates to the same refiner, favoring refinements that yield accurate long-horizon forecasts. This extends the optimization target from one-step accuracy to long-term trajectory quality without sacrificing the refiner's ability to recover fine-scale structures. Our main contributions are summarized as follows:

- We propose **Re**fine and **Al**ign **(ReAl)**, a lightweight, model-agnostic PDE forecasting plug-in based on Flow Matching. **ReAl** addresses the loss of high-frequency information and fine-scale structures commonly observed in MSE-pretrained surrogates on real-world data, refining their predictions without modifying the underlying architectures.
- To the best of our knowledge, we are the first to post-train long-term autoregressive PDE trajectories from a reinforcement learning perspective. **ReAl** samples multiple stochastic rollouts from the same initial state and optimizes the Flow refiner with trajectory-level rewards, aligning local residual corrections with long-term dynamical evolution.
- Extensive experiments demonstrate that **ReAl** improves both prediction accuracy and long-term stability. Compared with the base models, it reduces the average RMSE by **34.3**% at the 1-st rollout step and by **45.4**% at the 20-th step, with larger gains over longer horizons on RealPDEBench. **ReAl** also recovers high-frequency physical structures, suppresses long-term error accumulation, and mitigates numerical dissipation.

2 Related Works

Neural PDE Surrogates. Neural PDE surrogates approximate expensive numerical solvers by learning mappings between physical fields. Representative approaches include DeepONet (Lu et al., 2021), the Fourier Neural Operator (FNO) (Li et al., 2021), and physics-informed neural operators (Goswami et al., 2023). Subsequent architectures have explored latent spectral representations, geometry-aware attention, hierarchical modeling, and large-scale pretraining, including LSM (Wu et al., 2023), DPOT (Hao et al., 2024), Transolver (Wu et al., 2024), Poseidon (Herde et al., 2024), and HINE (Jiang et al., 2026). Although these models are effective for autoregressive forecasting,

prediction errors and insufficiently modeled frequency components can accumulate over long rollouts (Lippe et al., 2023; Yoo et al., 2026). We retain such models as deterministic backbones and improve their predictions through a model-agnostic refinement plug-in.

Generative Models and Refinements for PDEs. Diffusion and Flow Matching models (Ho et al., 2020; Rombach et al., 2022; Lipman et al., 2023) have been widely applied to PDE simulation, data assimilation, and physics-constrained generation (Shysheya et al., 2024; Yao et al., 2026; Utkarsh et al., 2026). DiffusionPDE addresses PDE solving under partial observations (Huang et al., 2024), DYffusion targets probabilistic multi-step forecasting (Rühling Cachay et al., 2023), and autoregressive conditional diffusion models have been studied for turbulent-flow prediction (Kohl et al., 2026). Other methods improve multiscale reconstruction through wavelet or latent representations (Hu et al., 2025; Li et al., 2026). PDE-Refiner (Lippe et al., 2023), Adaptive Flow Matching (Fotiadis et al., 2025), PDE-SpectralRefiner (Luo, 2025), and FlowRefiner (Dai et al., 2026) further recover non-dominant frequency components through generative refinement. Unlike those methods based primarily on short-horizon objectives, ReAl directly optimizes long-term autoregressive rollouts.

Post-Training of Diffusion Models. Reward-based post-training aligns diffusion and Flow Matching models with task-specific objectives. DDPO (Black et al., 2024) and DPOK (Fan et al., 2023) optimize diffusion sampling through reinforcement learning, Flow-GRPO (Liu et al., 2025) compares groups of samples, and DiffusionNFT (Zheng et al., 2026) employs negative-aware online reinforcement learning. We extend these ideas to PDE forecasting by post-training the refiner with trajectory-level rewards for long-term accuracy, stability, and physical consistency.

3 Method

In this section, we introduce **Re**fine and **Al**ign (**ReAl**), a model-agnostic plug-in for stable and accurate long-term partial differential equation forecasting, as illustrated in Fig. 2. During refinement, we pretrain a flow matching refiner to correct the residual distribution of the base predictions. During alignment, we employ trajectory-level post-training to improve long-term forecasting stability.

3.1 Preliminaries

PDE surrogates for forecasting. Consider a nonlinear dynamical system governed by a partial differential equation (PDE):

$$\frac{\partial \boldsymbol{u}}{\partial t} = \mathcal{G}(\boldsymbol{u}; \phi), \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ denotes a continuous physical field depending on spatial coordinate $\boldsymbol{x}$ and time t, ϕ represents the physical parameters governing the dynamics of the system, and $\mathcal{G} : \mathcal{U} \times \Phi \to \mathcal{U}$ is a differential operator. By applying numerical discretization techniques, such as finite-difference or spectral methods, we obtain a discrete trajectory of consecutive observed states:$\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$. Our goal is to learn a neural surrogate model f_θ, parameterized by θ, that approximates the discrete evolution operator and predicts the next state $\boldsymbol{u}_{n+1}$ from the current state $\boldsymbol{u}_n$[1], $\hat{\boldsymbol{u}}_{n+1} = f_\theta(\boldsymbol{u}_n)$. For multi-step forecasting, the surrogate model is applied recursively in an autoregressive manner. The prediction at the k-th future step is given by $\hat{\boldsymbol{u}}_{n+k} = f_\theta^{\circ k}(\boldsymbol{u}_n)$, where $f_\theta^{\circ k}(\cdot)$ denotes the k-fold composition of $f_\theta(\cdot)$.

Conditional Flow Matching. Conditional flow matching (CFM) (Lipman et al., 2023) learns a conditional vector field that transports samples from a Gaussian noise distribution to the target data distribution conditioned on $\boldsymbol{c}$. During training, an intermediate state is sampled along a linear interpolation between a noise sample and a target sample, and the vector field $\boldsymbol{v}_\psi$ is optimized to predict the corresponding transport velocity. At inference, the learned vector field is integrated from noise to data to generate a sample under the given condition. Here, the flow time $\tau \in [0, 1]$ parameterizes the generative transport process and is distinct from the physical time of PDE evolution.

[1] To simplify the notation, we consider one-step forecasting as a representative setting. Our method is agnostic to the temporal input–output configuration and can be applied to surrogates that use multiple historical states or predict multiple future states.

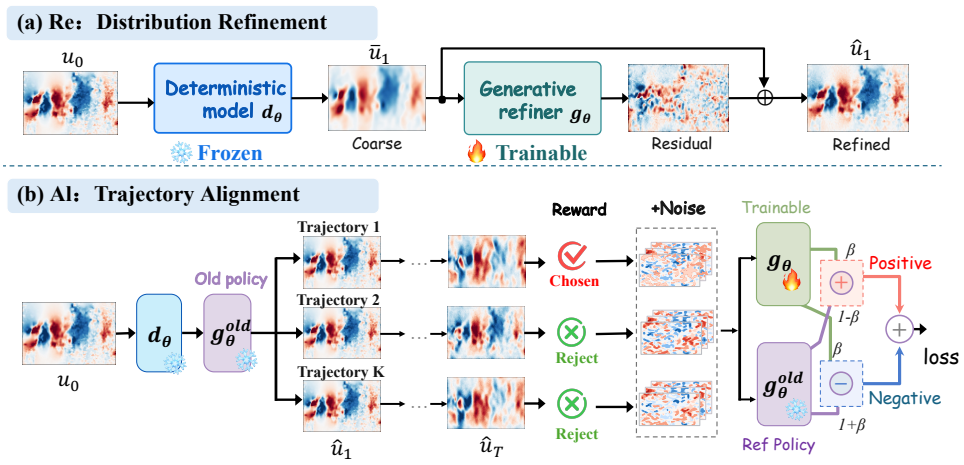


Figure 2: Overview of **Re**fine and **Al**ign (**ReAl**), a model-agnostic plug-in for pretrained PDE surrogates. (a) **Refine**: a conditional flow matching refiner generates fine-scale residuals for the frozen deterministic prediction. (b) **Align**: trajectory rewards from multiple stochastic rollouts post-train the refiner for stable long-term PDE forecasting. Both stages optimize the same conditional velocity field, with trajectory rewards weighting its flow-matching objectives during alignment.

3.2 Distribution Refinement with Conditional Flow Matching

We first introduce the distribution refinement phase of **ReAl**, as shown in Fig. 2(a). Most existing PDE surrogates are trained with MSE, enabling the deterministic predictor $\boldsymbol{d}_\theta$ to stably capture dominant dynamics but often suppressing high-frequency structures through mean regression. To address this limitation, we introduce a conditional flow matching refiner $\boldsymbol{g}_\psi$ that learns the residual distribution between the base prediction and ground truth, recovering fine-scale structures.

Deterministic Base Prediction. Given the current PDE state $\boldsymbol{u}_n$, the MSE-based deterministic predictor produces the next-state estimate, $\bar{\boldsymbol{u}}_n$, and is pretrained by minimizing its discrepancy from the ground truth, $\boldsymbol{u}_{n+1}$:

$$\bar{\boldsymbol{u}}_{n+1} = \boldsymbol{d}_\theta(\boldsymbol{u}_n), \quad \mathcal{L}_{\text{MSE}}(\theta) = \mathbb{E}\left[\|\bar{\boldsymbol{u}}_{n+1} - \boldsymbol{u}_{n+1}\|^2\right]. \tag{2}$$

The optimal MSE predictor approaches the conditional mean, $\mathbb{E}\left[\boldsymbol{u}_{n+1} \mid \boldsymbol{u}_n\right]$. Although this reliably captures the dominant dynamics, conditional averaging tends to suppress non-dominant fine-scale structures, resulting in overly smooth predictions.

Conditional Flow Refinement. To recover non-dominant structures suppressed by MSE training, we introduce a conditional flow matching refiner $\boldsymbol{g}_\psi$ that models the conditional distribution of fine-scale residuals. Specifically, we define the target residual, $\boldsymbol{r}_{n+1}$, and the condition, $\boldsymbol{c}_n$, as:

$$\boldsymbol{r}_{n+1} = \boldsymbol{u}_{n+1} - \bar{\boldsymbol{u}}_{n+1}, \quad \boldsymbol{c}_n = \text{Concat}\left[\boldsymbol{u}_n, \bar{\boldsymbol{u}}_{n+1}\right]. \tag{3}$$

The refiner learns to generate $\boldsymbol{r}_{n+1}$ conditioned on the current state, $\boldsymbol{u}_n$, and deterministic base prediction, $\bar{\boldsymbol{u}}_{n+1}$. Given a Gaussian sample $\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, we construct a linear flow path toward the target residual:

$$\boldsymbol{z}_\tau = (1-\tau)\boldsymbol{z}_0 + \tau \boldsymbol{r}_{n+1}, \qquad \tau \sim \mathcal{U}(0,1). \tag{4}$$

The conditional velocity field $\boldsymbol{v}_\psi$ is trained to match the corresponding transport velocity:

$$\mathcal{L}_{\text{refine}}(\psi) = \mathbb{E}\left[\left\|\boldsymbol{v}_\psi\left(\boldsymbol{z}_\tau, \tau \mid \boldsymbol{c}_n\right) - \left(\boldsymbol{r}_{n+1} - \boldsymbol{z}_0\right)\right\|^2\right]. \tag{5}$$

At inference, the generative mapping $\boldsymbol{g}_\psi$ is obtained by integrating $\boldsymbol{v}_\psi$ from $\tau = 0$ to $\tau = 1$, starting from $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The generated residual and refined prediction are:

$$\hat{\boldsymbol{r}}_{n+1} = \boldsymbol{g}_\psi\left(\boldsymbol{\epsilon} \mid \boldsymbol{c}_n\right), \quad \hat{\boldsymbol{u}}_{n+1} = \bar{\boldsymbol{u}}_{n+1} + \hat{\boldsymbol{r}}_{n+1}. \tag{6}$$

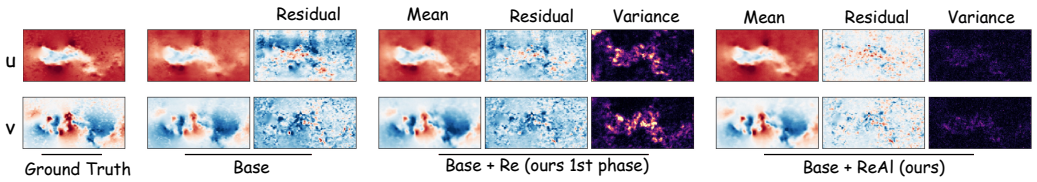


Figure 3: Qualitative results on Controlled Cylinder *Real2Real* for velocity components u and v. Residuals are computed against the ground truth, while variance is measured across random seeds. Base+ReAl produces more accurate and consistent predictions.

The deterministic predictor and stochastic refiner are complementary: the former captures stable dominant dynamics, while the latter restores residual fine-scale structures without imposing a strict low- and high-frequency decomposition, as illustrated in Fig. 2(a).

3.3 Trajectory Alignment for PDE Forecasting

Although combining the deterministic predictor $\boldsymbol{d}_\theta$ with a conditional flow matching refiner $\boldsymbol{g}_\psi$ recovers non-dominant fine-scale information missing from the base prediction, perturbations introduced by stochastic flow sampling may accumulate during autoregressive rollouts and degrade long-term stability. To address this issue, we introduce the trajectory-alignment post-training phase of **ReAl**, as shown in Fig. 2(b). Specifically, we keep $\boldsymbol{d}_\theta$ frozen and post-train the refiner $\boldsymbol{g}_\psi$ using trajectory-level rewards from multiple stochastic rollouts, thereby improving the quality of long-term PDE trajectories.

Trajectory rollout. Given an initial state $\boldsymbol{u}_n$, we generate K candidate trajectories using the frozen deterministic predictor $\boldsymbol{d}_\theta$ and a snapshot of the refiner $\boldsymbol{g}_\psi^{\text{old}}$. All candidates start from the same state but use independent noise sequences. For the i-th candidate, we initialize $\hat{\boldsymbol{u}}_n^{(i)} = \boldsymbol{u}_n$ and recursively apply the refined transition:

$$\begin{aligned}
\bar{\boldsymbol{u}}_{n+k}^{(i)} &= \boldsymbol{d}_\theta\left(\hat{\boldsymbol{u}}_{n+k-1}^{(i)}\right), \boldsymbol{c}_{n+k-1}^{(i)} = \text{Concat}\left[\hat{\boldsymbol{u}}_{n+k-1}^{(i)}, \bar{\boldsymbol{u}}_{n+k}^{(i)}\right], \\
\hat{\boldsymbol{r}}_{n+k}^{(i)} &= \boldsymbol{g}_{\psi^{\text{old}}}\left(\boldsymbol{\epsilon}_k^{(i)} \mid \boldsymbol{c}_{n+k-1}^{(i)}\right), \quad \boldsymbol{\epsilon}_k^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\
\hat{\boldsymbol{u}}_{n+k}^{(i)} &= \bar{\boldsymbol{u}}_{n+k}^{(i)} + \hat{\boldsymbol{r}}_{n+k}^{(i)}, \quad k = 1, \ldots, h.
\end{aligned} \tag{7}$$

After h autoregressive steps, the resulting trajectory is:

$$\widehat{\mathcal{T}}_n^{(i)} = \left[\hat{\boldsymbol{u}}_{n+1}^{(i)}, \hat{\boldsymbol{u}}_{n+2}^{(i)}, \ldots, \hat{\boldsymbol{u}}_{n+h}^{(i)}\right], \qquad i = 1, \ldots, K. \tag{8}$$

Despite sharing the same initial state and deterministic predictor, independent residual samples produce different long-term trajectories. As shown in Fig. 3, stochastic refinement (Base + Re) introduces high-frequency variance that can accumulate during autoregressive rollouts.

Trajectory-level reward. Each candidate trajectory $\widehat{\mathcal{T}}_n^{(i)}$ is evaluated against the corresponding ground-truth trajectory $\mathcal{T}_n$ to obtain its trajectory-level reward:

$$R^{(i)} = \mathcal{R}\left(\widehat{\mathcal{T}}_n^{(i)}, \mathcal{T}_n\right), \quad R_{\text{MSE}}^{(i)} = -\left\|\hat{\boldsymbol{u}}_{n+h}^{(i)} - \boldsymbol{u}_{n+h}\right\|^2, \tag{9}$$

where $R_{\text{MSE}}^{(i)}$ denotes the negative mean squared error at the terminal state, such that a larger reward indicates a more accurate terminal prediction. Although the reward is evaluated only at the endpoint, the terminal state reflects the accumulated effects of stochastic residual corrections propagated throughout the multi-step autoregressive rollout. It therefore serves as a measure of long-term trajectory quality. More generally, $\mathcal{R}$ may be defined using discrepancies in physically relevant quantities, such as vorticity, divergence, and kinetic energy, or by combining multiple criteria to evaluate complementary properties of the predicted dynamics.

Since reward magnitudes can vary across initial conditions, we standardize the rewards within each group of K trajectories. Let μ_R and σ_R denote the group mean and standard deviation, respectively. We then clip the standardized reward and linearly rescale it to $[0, 1]$:

$$w^{(i)} = \frac{1}{2}\left[1 + \frac{1}{a_{\max}} \operatorname{clip}\left(\frac{R^{(i)} - \mu_R}{\sigma_R + \epsilon_{\text{num}}}, -a_{\max}, a_{\max}\right)\right]. \tag{10}$$

Here, ϵ_{num} is a small constant that ensures numerical stability and $a_{\max} > 0$ controls the clipping range. Consequently, higher-quality trajectories receive weights closer to 1, whereas lower-quality receive weights closer to 0.

Trajectory reward-Guided Flow Alignment. Trajectory alignment post-trains the conditional velocity field learned during refinement (Sec. 3.2). We incorporate feedback from autoregressive rollouts through positive and negative flow-matching terms constructed from the current and old velocity fields, weighted by trajectory rewards. For the k-th autoregressive transition of the i-th candidate trajectory, $\boldsymbol{z}_\tau^{(i,k)}$ denotes the intermediate residual state at flow time τ, following Eq. equation 4. The velocity fields predicted by the current policy, old policy, and reference model are uniformly written as:

$$\boldsymbol{v}^{s,(i,k)} = \boldsymbol{v}_{\psi^s}\left(\boldsymbol{z}_\tau^{(i,k)}, \tau \mid \boldsymbol{c}_{n+k-1}^{(i)}\right), \qquad s \in \{\text{cur}, \text{old}, \text{ref}\}. \tag{11}$$

Here, $\psi^{\text{cur}} \equiv \psi$ denotes the parameters of the refiner being updated, ψ^{old} denotes the old-policy parameters used to generate the candidate trajectories, and ψ^{ref} denotes the parameters of the pre-trained reference model, which remain frozen throughout post-training. Following the negative-aware forward-process formulation of DiffusionNFT (Zheng et al., 2026), we construct positive and negative velocity fields from the current- and old-policy predictions:

$$\boldsymbol{v}^{+,(i,k)} = \beta \boldsymbol{v}^{\text{cur},(i,k)} + (1-\beta)\boldsymbol{v}^{\text{old},(i,k)}, \quad \boldsymbol{v}^{-,(i,k)} = (1+\beta)\boldsymbol{v}^{\text{old},(i,k)} - \beta \boldsymbol{v}^{\text{cur},(i,k)}, \tag{12}$$

where $\beta > 0$ controls the deviation of the positive and negative velocity fields from the old policy. The corresponding positive and negative flow matching losses are:

$$\ell_\pm^{(i,k)} = \left\| \boldsymbol{v}^{\pm,(i,k)} - \left(\boldsymbol{r}_{n+k}^{(i)} - \boldsymbol{z}_0^{(i,k)}\right)\right\|^2. \tag{13}$$

The final alignment objective averages over both candidate trajectories i and autoregressive steps k:

$$\mathcal{L}_{\text{align}} = \frac{1}{\beta}\mathbb{E}_{i,k}\left[w^{(i)}\ell_+^{(i,k)} + \left(1 - w^{(i)}\right)\ell_-^{(i,k)}\right]. \tag{14}$$

Same trajectory-level reward $w^{(i)}$ is applied to every autoregressive step of the i-th candidate. A high trajectory reward produces a weight close to 1, emphasizing the positive objective and encouraging the associated residual corrections. A low reward produces a weight closer to 0, emphasizing the negative objective. Because the same trajectory-level weight is shared at every rollout step, the terminal feedback supervises residual generation along the entire trajectory.

To stabilize post-training, we further regularize the refiner toward the frozen pretrained reference model while retaining the supervised residual-learning objective. The training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda_{\text{ref}}\mathbb{E}_{i,k}\left[\left\|\boldsymbol{v}^{\text{cur},(i,k)} - \boldsymbol{v}^{\text{ref},(i,k)}\right\|^2\right] + \lambda_{\text{sup}}\mathbb{E}_{i,k}\left[\left\|\hat{\boldsymbol{r}}_{n+k}^{(i)} - \boldsymbol{r}_{n+k}^{(i)}\right\|^2\right]. \tag{15}$$

The reference-model regularization prevents excessive deviation from the pretrained flow dynamics, while the supervised term preserves the refiner's residual-modeling capability. Throughout post-training, only the current refiner parameters ψ are updated; the deterministic predictor $\boldsymbol{d}_\theta$, old-policy parameters $\boldsymbol{g}_\psi^{\text{old}}$, and reference-model parameters $\boldsymbol{g}_\psi^{\text{ref}}$ remain frozen.

Overall, **ReAl** produces more accurate long-term predictions with lower across-seed variance. By favoring residual corrections that remain reliable over long horizons, it produces more accurate predictions with lower across-seed variance, as shown in Fig. 3 (Base+ReAl).

Table 1: Main experimental results on RealPDEBench (Hu et al., 2026). We report RMSE over the 10 predicted future frames on test set lower is better. The small percentage in the bottom-right corner indicates the relative RMSE reduction over the corresponding baseline.

Benchmark	Setting	Task	UNet		FNO		DPOT		Transolver	
			Baseline	+ Ours	Baseline	+ Ours	Baseline	+ Ours	Baseline	+ Ours
RealPDEBench	*Sim2Real*	Cylinder	0.0256	**0.0216** +16%	0.0425	**0.0261** +39%	0.0235	**0.0236** +1%	0.0315	**0.0244** +23%
		Controlled Cylinder	0.0219	**0.0155** +29%	0.0309	**0.0168** +46%	0.0183	**0.0144** +21%	0.0277	**0.0147** +47%
		Foil	0.0167	**0.0136** +19%	0.0197	**0.0122** +38%	0.0290	**0.0169** +42%	0.0345	**0.0276** +20%
		FSI	0.0247	**0.0161** +35%	0.0452	**0.0132** +71%	0.0201	**0.0116** +42%	0.0273	**0.0128** +53%
	Real2Real	Cylinder	0.0082	**0.0073** +11%	0.0135	**0.0089** +34%	0.0115	**0.0092** +20%	0.0154	**0.0092** +41%
		Controlled Cylinder	0.0089	**0.0073** +19%	0.0102	**0.0086** +18%	0.0114	**0.0081** +30%	0.0137	**0.0083** +39%
		Foil	0.0082	**0.0069** +14%	0.0134	**0.0070** +48%	0.0144	**0.0074** +49%	0.0241	**0.0101** +58%
		FSI	0.0058	**0.0053** +8%	0.0089	**0.0056** +37%	0.0106	**0.0056** +47%	0.0119	**0.0056** +53%

4 Experiment

Benchmarks. We evaluate on RealPDEBench (Hu et al., 2026), comprising four physical sub-datasets: Cylinder, Controlled Cylinder, Foil, and FSI, which provides paired numerical simulations and real-world measurements. Real observations contain noise, unresolved effects, and finer-scale dynamics absent from simulations. We consider two settings: *Sim2Real*, training on simulations and testing on real data; and *Real2Real*, training and testing on real data[2].

Evaluation metircs. To evaluate quality of prediction , we report the root mean squared error (RMSE) and relative (Rel. L_2) error. To further assess physical fidelity, we use the Fourier-space RMSE (fRMSE), frequency error (FE), and kinetic energy error (KE). Detailed definitions of these metrics are provided in the Appendix A.

Experiment setup. In our experiments, following the task setting of RealPDEBench, we use 10 consecutive input frames to directly predict the subsequent 10 frames. We adopt U-Net (Ronneberger et al., 2015), FNO (Li et al., 2021), DPOT-S (Hao et al., 2024), and Transolver (Wu et al., 2024) as the base architectures. We then freeze the base model and train a lightweight time-conditioned U-Net flow matching Refiner (approximately 5.04M parameters, typically under one-tenth of the base model size) for 50,000 steps with a batch size of 16 and a learning rate of 1×10^{-4}. Details are provided in Appendix B. Subsequently, we initialize ReAl from the best Refiner checkpoint while keeping the base model frozen, and optimize ReAl for 10,000 steps using the trajectory-alignment objective. We use AdamW with a batch size of 4 and a learning rate of 2×10^{-5}. For each condition, we sample $K = 4$ complete trajectories. Each trajectory consists of five autoregressive rollout blocks, with each block generated by integrating the flow velocity field using four Euler flow steps. We use the negative MSE of the final, fifth prediction block as the trajectory reward. We set $\beta = 0.1$ in Eq. 14, and set $\lambda_{\text{ref}} = 0.01$ and $\lambda_{\text{sup}} = 0.1$ in Eq. 15.

4.1 Main Results

Prediction Accuracy. We first evaluate **ReAl** after a single rollout, i.e., direct prediction of one 10-frame future block, under the *Sim2Real* and *Real2Real* settings of RealPDEBench. As shown in Table 1, Base+ReAl consistently improves diverse base architectures across physical tasks. Averaged over all models, tasks, and settings, Base+ReAl reduces RMSE by 33.9%, including reductions of 33.7% on *Sim2Real* and 34.1% on *Real2Real*. Notably, ReAl achieves a 71% relative improvement for FNO on the *Sim2Real* FSI task. These results demonstrate that Base+ReAl provides effective post-training refinement even after a single rollout.

Long-Term Stability. To further evaluate long-term stability, we extend the autoregressive prediction horizon to 200 frames and compare the base models, Base+Re with only the flow refiner, and Base+ReAl with additional trajectory-alignment post-training in Fig. 4. As the rollout horizon increases, prediction errors of the base models accumulate continuously, with some models exhibiting severe error blow-up, such as FNO on FSI and DPOT on Foil. For DPOT on Foil, Base+ReAl limits

[2]For Cylinder, we exclude trajectory `3656.h5` due to a reproducible spatiotemporal misalignment in our processing pipeline.

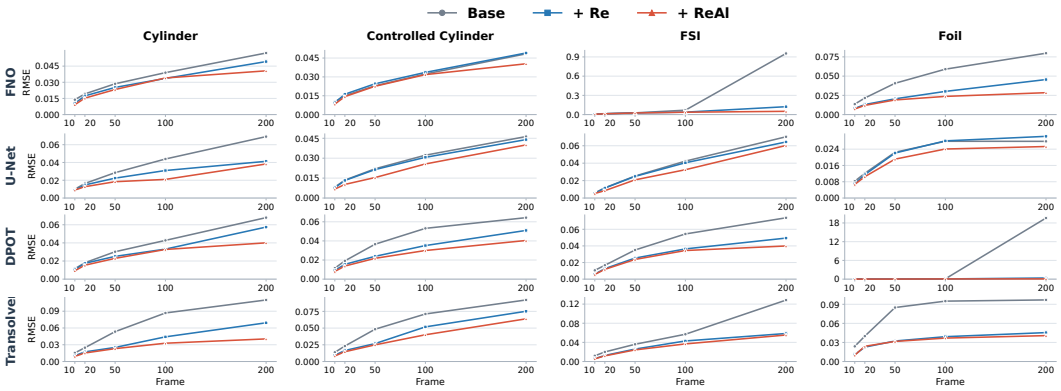


Figure 4: RMSE over long autoregressive rollout frames under the RealPDEBench *Real2Real* setting. Long-horizon error accumulation causes severe degradation for several base models, most notably DPOT on Foil and FNO on FSI. By contrast, Base+ReAl consistently limits error growth and preserves stable predictions over long rollout horizons.

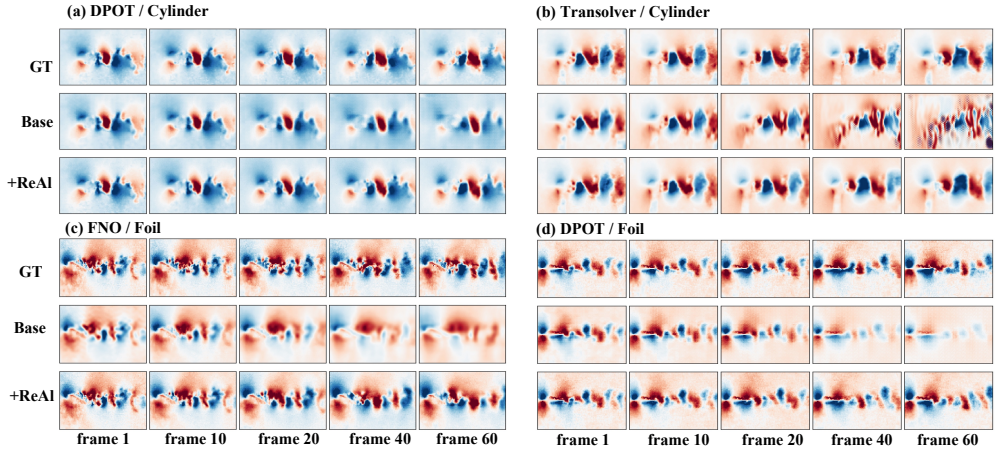


Figure 5: Visualization of long-horizon *Real2Real* rollouts (v channel): Base+**ReAl** (a) restores fine-scale details, (b) suppresses error accumulation, and (c, d) mitigates numerical dissipation to preserve flow energy and wake structures.

200-frame error growth to 10.9×, compared with 44.5× for Base+Re and 1,360× for the base model. Although Base+Re reduces the overall error through fine-scale residual correction, perturbations introduced by stochastic sampling may still be progressively amplified during long-term forecasting. In contrast, Base+ReAl exhibits substantially slower error growth across most models and tasks and effectively suppresses these instabilities. Fig. 5 compares long-horizon predictions. While base predictions become oversmoothed and lose fine-scale vortices, Base+ReAl better preserves these structures throughout the rollout. Additional results are provided in Appendix C.

Frequency Information Recovery. To examine how each module corrects autoregressive trajectories, we decompose the prediction error in the radial frequency domain and evaluate the lowest 20% ($f < 20$) and highest 20% ($f \geq 80$) frequency regions. Here, R1, R5, and R10 denote the first, fifth, and tenth autoregressive rollout blocks, corresponding to rollout horizons of 10, 50, and 100 future frames, respectively. For frequency region b, we use the ground-truth-normalized error energy $\frac{E_{\mathrm{err},b}}{E_{\mathrm{GT},b}} \times 100\%$, where lower values indicate better frequency information recovery. As shown in Fig. 6 and Table 2, the Base+Re correction range expands with the rollout. At R1, it reduces high- and low-frequency errors by 15.76% and 1.63%, respectively. At R10, these improvements reach 28.22% and 32.72%, indicating that Base+Re gradually extends from fine-scale to multiscale trajectory correction. In contrast, Base+ReAl consistently targets high-frequency errors: its additional

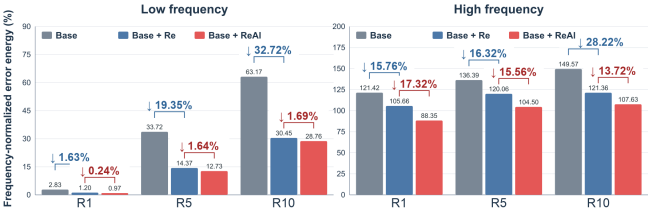


Figure 6: Comparison of frequency-normalized error energy across rollout horizons. Lower is better.

Table 2: Numerical frequency-normalized error energy. Lower is better.

Frequency	Method	R1	R5	R10
Low	Base	2.83	33.72	63.17
	Base + Re	1.20	14.37	30.45
	Base + ReAl	**0.97**	**12.73**	**28.76**
High	Base	121.42	136.39	149.57
	Base + Re	105.66	120.06	121.36
	Base + ReAl	**88.35**	**104.50**	**107.63**

Table 3: Reward ablation results on Controlled Cylinder. h_r denotes the rollout block used to compute the training reward. The best results are shown in **bold**, and the second-best results are underlined.

Method	Reward	h_r	Evaluation R1					Evaluation R5					Evaluation R10				
			RMSE ↓	Rel. L_2 ↓	fRMSE ↓	FE ↓	KE ↓	RMSE ↓	Rel. L_2 ↓	fRMSE ↓	FE ↓	KE ↓	RMSE ↓	Rel. L_2 ↓	fRMSE ↓	FE ↓	KE ↓
Base	–	–	0.0100	0.0750	0.0012	21.5557	3.77e-5	0.0227	0.1771	0.0040	46.0869	5.57e-5	0.0320	0.2603	0.0065	64.3384	6.28e-5
+ Re	–	–	0.0093	0.0673	0.0011	26.3517	3.39e-5	0.0243	0.1830	0.0043	49.6342	5.63e-5	0.0331	0.2492	0.0067	65.5248	6.33e-5
+ ReAl	RMSE	1	**0.0082**	**0.0602**	**0.0010**	**20.0997**	**3.12e-5**	0.0223	0.1694	0.0039	**35.1164**	5.49e-5	0.0314	0.2358	0.0063	59.6282	6.20e-5
+ ReAl	RMSE	5	0.0084	0.0619	0.0010	20.5995	3.22e-5	**0.0200**	**0.1560**	**0.0032**	35.4562	5.38e-5	0.0273	0.2070	0.0054	**55.0176**	5.91e-5
+ ReAl	RMSE	10	0.0087	0.0637	0.0011	20.1948	3.31e-5	0.0210	0.1605	0.0037	35.7731	5.45e-5	**0.0269**	**0.2052**	**0.0053**	55.2919	5.91e-5
+ ReAl	KE	5	0.0084	0.0618	0.0010	20.7217	3.21e-5	0.0208	0.1585	0.0036	35.4098	**5.37e-5**	0.0273	0.2072	0.0054	55.3497	**5.90e-5**
+ ReAl	FE	5	0.0085	0.0624	0.0010	20.3108	3.24e-5	0.0210	0.1602	0.0037	35.2623	5.39e-5	0.0275	0.2095	0.0054	56.0622	5.92e-5

improvements at R1, R5, and R10 are 17.32%, 15.56%, and 13.72%, while the low-frequency gains remain only 0.24%, 1.64%, and 1.69%. Thus, ReAl continuously restores fine-scale information lost during autoregressive prediction.

4.2 Ablation Study

Ablation of Reward Horizon and Metric. We investigate how the rollout block used to compute the training reward, $h_r \in \{1, 5, 10\}$, and the choice of reward metric affect rollout performance. A clear horizon-matching effect emerges: $h_r = 1$, 5, and 10 achieve the best RMSE at R1, R5, and R10, respectively, while remaining competitive elsewhere. Longer reward horizons improve recursive stability. RMSE provides the most consistent gains in RMSE, Rel. L_2, and fRMSE, while KE yields the lowest kinetic-energy error. FE optimization does not consistently minimize FE, indicating that physics-specific rewards need not improve their corresponding metrics. We therefore use RMSE with $h_r = 5$ by default, which best balances short- and long-horizon performance.

5 Conclusion

In this paper, we propose **Re**fine and **Al**ign (**ReAl**), a model-agnostic post-training framework for long-horizon PDE forecasting. **ReAl** retains an MSE-trained base model to capture stable dominant dynamics and uses a lightweight conditional Flow Refiner to generatively correct prediction residuals, recovering fine-scale high-frequency structures that are often lost under MSE training. We further introduce trajectory-alignment post-training, which optimizes the Refiner using trajectory-level rewards from multiple stochastic autoregressive rollouts to align local corrections with long-term dynamics and suppress accumulated sampling perturbations. We systematically evaluate **ReAl** on RealPDEBench using UNet, FNO, DPOT-S, and Transolver. **ReAl** reduces RMSE by 33.9% on average for 1-st step predictions and 45.4% by 20-th step, which effectively mitigates error accumulation, trajectory divergence, and numerical dissipation over long autoregressive rollouts. Frequency analysis shows that the Refiner gradually extends to multiscale correction, while **ReAl** mainly provides additional improvements in high-frequency errors. These results demonstrate that **ReAl** serves as a general, lightweight, and plug-and-play post-training module that improves fine-scale detail recovery and long-horizon prediction stability for diverse PDE surrogates under real-world data distributions and simulation-to-reality shifts.

AI USAGE STATEMENT

In this work, we used generative AI tools to assist with language polishing, translation, dataset cleaning and reformatting, qualitative and thematic data analysis, and the interpretation of experimental results. We did not use generative AI to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide key elements of or write mathematical proofs, propose or refine hypotheses, design or provide feedback on research methodologies or experiments, or implement the proposed methods. We additionally used generative AI for brainstorming, information retrieval, improving the readability of the manuscript, and identifying potentially relevant literature. All AI-assisted text and translations were reviewed by the authors, references were verified against their original sources, and data processing, analyses, and interpretations were independently checked against the source data, implementation, and experimental results. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

References

Theodore L Bergman. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

John Crank. *The mathematics of diffusion*. Oxford university press, 1979.

Yilong Dai, Yiming Sun, Yiheng Chen, Shengyu Chen, Xiaowei Jia, and Runlong Yu. Flowrefiner: Flow matching-based iterative refinement for 3d turbulent flow simulation. *arXiv preprint arXiv:2604.17149*, 2026.

Peter Alan Davidson. *Turbulence: an introduction for scientists and engineers*. Oxford university press, 2004.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Stathi Fotiadis, Noah D Brenowitz, Tomas Geffner, Yair Cohen, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Adaptive flow matching for resolving small-scale physics. In *Forty-second International Conference on Machine Learning*, 2025.

Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed deep neural operator networks. In *Machine learning in modeling and simulation: methods and applications*, pp. 219–254. Springer, 2023.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peiyan Hu, Rui Wang, Xiang Zheng, Tao Zhang, Haodong Feng, Ruiqi Feng, Long Wei, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Wavelet diffusion neural operator. In *International Conference on Learning Representations*, volume 2025, pp. 12291–12333, 2025.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, et al. Realpdebench: A benchmark for complex physical systems with real-world data. *International Conference on Learning Representations*, 2026.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2003.

Ruoxi Jiang, Xiao Zhang, Karan Jakhar, Peter Y Lu, Pedram Hassanzadeh, Michael Maire, and Rebecca Willett. Hierarchical implicit neural emulators. *Advances in Neural Information Processing Systems*, 38:73718–73751, 2026.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *Neural Networks*, pp. 108641, 2026.

Zhong-Hao Li, Chao-Yu Liu, and Qian Zhang. Di-bilps: Denoising induced bidirectional latent-pde-solver under sparse observations. *ArXiv*, abs/2605.13790, 2026. URL https://api.semanticscholar.org/CorpusID:288254894.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations (ICLR)*, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations*, 2023.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *ArXiv*, Advances in Neural Information Processing Systems, 2025. URL https://api.semanticscholar.org/CorpusID:278394068.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Li Luo. Pdespectralrefiner: Achieving more accurate long rollouts with spectral adjustment. In *2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT)*, pp. 2097–2100. IEEE, 2025.

KW Morton and DF Mayers. Numerical solution of partial differential equations. *Cambridge University*, 512, 1994.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

Julian Schwinger, Lester L Deraad, Kimball Milton, Wu-yang Tsai, and Joyce Norton. *Classical electrodynamics*. Hachette UK, 1998.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for pde simulations. In *Advances in Neural Information Processing Systems*, 2024.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. *Advances in Neural Information Processing Systems*, 38:160217–160252, 2026.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional pdes with latent spectral models. *International Conference on Machine Learning*, 2023.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Animashree Anandkumar. Guided diffusion sampling on function spaces with applications to pdes. *Advances in Neural Information Processing Systems*, 38:127057–127094, 2026.

Seungwoo Yoo, Juil Koo, Daehyeon Choi, and Minhyuk Sung. Diffusionrollout: Uncertainty-aware rollout planning in long-horizon pde solving. *Transactions on Machine Learning Research*, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.