STABILIZING NEURAL EMULATION OF SPATIOTEMPORAL PHYSICAL DYNAMICS THROUGH DENOISING KOOPMAN LATENT DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Koopman operator theory provides a principled framework for reduced-order modeling of nonlinear spatiotemporal dynamics through linear evolution in an observable space. In practice, however, learned finite-dimensional Koopman representations are typically approximately invariant, causing unresolved dynamics and accumulated closure errors. Recent approaches, motivated by the Mori-Zwanzig formalism, address these unresolved effects through deterministic memory or closure modeling, yet robust extrapolation beyond the training horizon remains challenging. Here, we propose KoopFlow, an efficient prediction–correction framework built on two key insights: First, we identify a non-monotonic relationship between Koopman spectral coherence and predictive accuracy, revealing a tradeoff between long-term information preservation and transient dynamics fidelity. Second, we reinterpret finite-dimensional Koopman error accumulation as a progressive distributional degradation process. Accordingly, KoopFlow treats Koopman predictions as noisy intermediate estimates and leverages a pretrained flow matching model as a probabilistic transport bridge toward the target latent distribution. By coupling structured linear Koopman prediction with generative correction, KoopFlow mitigates accumulated errors while retaining computational efficiency and temporal flexibility. Experiments across diverse nonlinear dynamical systems demonstrate improved long-horizon prediction accuracy and stability over representative baselines, particularly beyond the training horizon.

1 INTRODUCTION

Accurately predicting the spatiotemporal behavior of complex dynamical systems governed by time-dependent partial differential equations (PDEs) is central to scientific computing and engineering, with applications ranging from fluid dynamics Lagemann et al. (2026) and climate forecasting Bi et al. (2023) to electromagnetic wave propagation Cao et al. (2026). Yet high-fidelity numerical simulations become increasingly costly as prediction horizons grow and spatial meshes are refined. Data-driven surrogates represented by neural operators (Lu et al., 2021; Li et al., 2021) offer a promising means of accelerating simulation after offline training. Nevertheless, stable and accurate long-term prediction remains challenging, particularly when extrapolating beyond the training horizon. Moreover, generic operator learning architectures do not necessarily impose explicit dynamical structure on the learned representations, leaving their temporal evolution largely implicit.

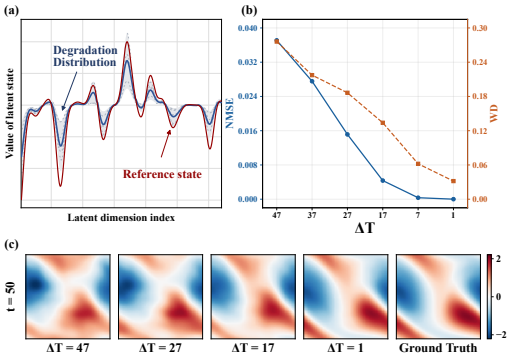

Figure 1: Motivation: (a) Reference latent state (red) and Koopman predictions (blue) evolved from different starting times along the same trajectory to the same target time. (b) NMSE and Wasserstein distance (WD) decrease as the prediction interval ΔT shortens. (c) Ground-truth and predicted fields at $t = 50$ for different ΔT.

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.

In contrast, Koopman operator theory provides a principled framework for modeling dynamical systems by representing nonlinear state evolution through the linear evolution of observables in an appropriate function space (Koopman, 1931; Mezić, 2005; Brunton et al., 2022). Building on this principle, learning-based Koopman operator frameworks have recently attracted increasing attention for reduced-order modeling of spatiotemporal dynamics. Despite this appeal, the practical realization is fundamentally constrained by the gap between the infinite-dimensional Koopman representation and its finite-dimensional approximations. For general nonlinear systems, the existence of a finite-dimensional Koopman-invariant subspace capturing the relevant state dynamics is not guaranteed, whereas practical learning-based models must operate with a finite number of learned observables (Lusch et al., 2018; Xiong et al., 2024). Recent work addresses the quality of these learned representations through information-theoretic regularization, balancing latent linear predictability and expressiveness while mitigating mode collapse (Cheng et al., 2026). Nevertheless, the resulting observable subspace need not be invariant, leaving part of the dynamics unresolved (Brunton et al., 2016; 2022) From the Mori-Zwanzig perspective, restricting high-dimensional dynamics to such a non-closed resolved subspace gives rise to additional memory and unresolved forcing terms beyond the Markovian evolution of the resolved variables (Chorin et al., 2000; Lin et al., 2021). Most existing deep Koopman models, however, assume a closed Markovian latent evolution and therefore do not explicitly account for these memory and forcing terms. The resulting closure errors can accumulate or be amplified during recursive rollouts, degrading long-term prediction stability and accuracy. Recent studies have begun to incorporate explicit memory or closure mechanisms into Koopman models (Gupta et al., 2025; Lu et al., 2026). Yet it remains challenging to achieve computationally tractable closures that accurately model the influence of these unresolved effects, particularly for chaotic and multiscale systems.

In this work, rather than explicitly learning deterministic approximations of these unresolved effects, we model their aggregate influence on finite-dimensional Koopman predictions from a statistical perspective. For exact deterministic dynamics, the semigroup property implies that predictions obtainedfrom different earlier states along the same trajectory should agree at a fixed target Mezić (2005). Finite-dimensional approximations Mauroy et al. (2020) can violate this consistency, producing horizon-dependent deviations, as illustrated in Figure 1. We refer to the empirical distribution of these predictions as a Koopman degradation distribution, which reflects the aggregate influence of unresolved dynamics and finite-dimensional approximation errors on the predicted latent states. From this perspective, long-horizon Koopman prediction can be formulated as a distribution-correction problem, with the goal of reducing the mismatch between predicted and reference latent distributions while preserving trajectory specific information. This insight gives rise to KoopFlow, a prediction–correction framework (Figure 2) that combines efficient Koopman evolution with generative correction. Specifically, (i) we first learn a finite-dimensional Koopman operator over an empirically selected temporal horizon, motivated by our finding that effective Koopman learning requires balancing long-term information preservation with transient dynamics fidelity; and (ii) the resulting prediction is mapped to an intermediate noisy latent state and subsequently denoised by a pretrained flow-matching model. Through this prediction–correction mechanism, KoopFlow mitigates accumulated approximation errors and improves long-horizon predictive stability and accuracy. The main contributions of this paper are summarized as follows:

- We identify a non-monotonic relationship between Koopman spectral coherence and predictive accuracy, showing that effective Koopman learning requires balancing long-term information preservation with transient dynamics fidelity.
- We reinterpret finite-dimensional Koopman error accumulation as a progressive distributional degradation process and propose KoopFlow, an efficient prediction–correction framework that couples structured linear Koopman evolution with generative correction for stable and flexible long-horizon forecasting.
- Extensive experiments across diverse nonlinear dynamical systems demonstrate that KoopFlow consistently improves long-horizon prediction accuracy and stability over representative baselines, particularly beyond the training horizon, while maintaining a favorable accuracy–efficiency trade-off.

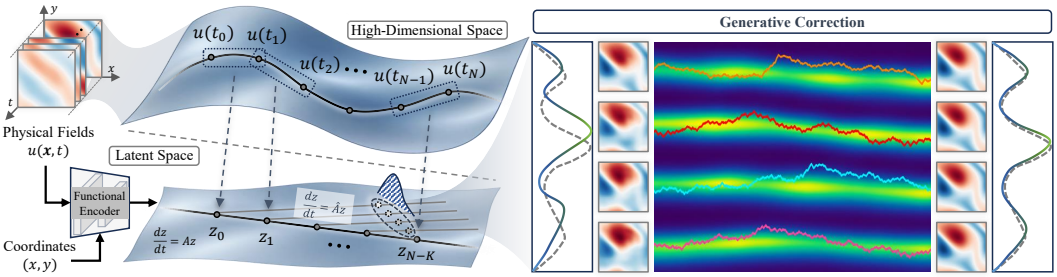

Figure 2: Overview of our proposed KoopFlow. High-dimensional field snapshots are embedded into a latent space via a functional encoder and evolved by a Koopman operator, where finite-dimensional approximation leads to degraded predictions (Left). A lightweight flow matching model transports these states toward the target latent distribution, yielding high-fidelity predictions (Right).

2 Related Work

Koopman Learning. Koopman learning has emerged as an effective paradigm for modeling nonlinear dynamical systems by learning observable representations with approximately linear evolution (Takeishi et al., 2017; Lusch et al., 2018; Azencot et al., 2020). However, finite-dimensional Koopman representations inevitably discard unresolved dynamics, which appear as memory and forcing terms under the Mori-Zwanzig formalism (Mori, 1965; Zwanzig, 1973). Existing methods mainly address this limitations by improving Koopman representations (Haseli & Cortés, 2021; Cheng et al., 2026) or introducing memory and closure corrections (Gupta et al., 2023; Lu et al., 2026). The former reduces projection error but neglects the intrinsic dynamical structure of the underlying physical system, while the latter better captures unresolved dynamics at the cost of additional states or corrective dynamics. Both primarily focus on improving forward evolution. In contrast, we adopt a trajectory consistency view and show that accumulated finite-dimensional Koopman errors induce a progressive, horizon-dependent degradation distribution, shifting the focus from error suppression during propagation to recorrect the degraded latent dynamics.

Generative Modeling. Generative models, particularly diffusion and flow matching models, have recently attracted increasing attention for learning complex PDE dynamics (Huang et al., 2024; Liu et al., 2025; Hou et al., 2026), producing statistically plausible predictions with improved long-horizon stability over deterministic forecasts. However, generative modeling in the high-dimensional physical state space remains computationally expensive, especially at high spatial resolutions (Huang et al., 2024; Kohl et al., 2026; Hou et al., 2026; Holzschuh et al., 2026). Recent studies therefore perform generative modeling in compressed latent spaces (Du et al., 2024; Zhou et al., 2025; Rozet et al., 2025). Despite improved efficiency, their performance remains sensitive to representation quality, while latent compression may compromise fine-scale fidelity or dynamical consistency. Rather than using generative models to directly learn the dynamical evolution, we employ a trained flow matching model to probabilistically correct structured Koopman predictions. This design couples efficient linear Koopman propagation with generative correction, achieving a favorable accuracy–efficiency trade-off for reduced-order modeling.

3 Methodology

Our methodology proceeds as: (1) the problem formulation (section 3.1) and continuous-time Koopman evolution (section 3.2); (2) a physics-consistency analysis of finite-dimensional Koopman dynamics, leading to a horizon-dependent degradation distribution (section 3.3); (3) a theoretical guarantee for generative recovery from the degraded distribution to the accurate latent state (section 3.4); (4) the resulting KoopFlow framework built upon these theoretical insights (section 3.5).

3.1 Problem Formulation

We consider a nonlinear spatiotemporal dynamical system. Let $\Omega \subset \mathbb{R}^d$ be a spatial domain and $u(t, x) \in \mathbb{R}^c$ denote the system state at time t and location $x \in \Omega$. The dynamics are governed by a

parameterized nonlinear PDE:

$$\begin{aligned} \partial_t u(t,x) &= \mathcal{F}[u(t);\mu](x), \quad && x \in \Omega,\ t > 0, \\ u(0,x) &= u_0(x), && x \in \Omega, \\ \mathcal{B}[u(t);\mu](x) &= 0, && x \in \partial\Omega,\ t > 0, \end{aligned} \tag{1}$$

where $\mathcal{F}$ denotes the nonlinear differential operator, μ represents the physical parameters, $\mathcal{B}$ specifies the boundary conditions, and u_0 is the initial state. Since the underlying evolution is generally unavailable in closed form, we assume access only to discrete trajectory observations $u_n = u(n\Delta t)$ sampled at intervals Δt. Given the recent K observations $u_{n-K+1:n}$, together with μ and $\mathcal{B}$, our goal is to predict the future state u_{n+h} at a horizon $h\Delta t$, where $h \geq 1$. To model this dynamical system efficiently, we next introduce a continuous-time Koopman formulation.

3.2 Continuous-Time Koopman Learning

We learn a latent Koopman representation in which nonlinear spatiotemporal dynamics are approximated by continuous-time linear evolution, enabling flexible prediction horizons. A functional encoder E_θ maps historical observations, potentially sampled at irregular spatial locations and varying resolutions, to a fixed-dimensional latent state $z_n = E_\theta(u_{n-K+1:n}, \mu, \mathcal{B}) \in \mathbb{R}^m$. A paired functional decoder D_θ reconstructs the physical field from the latent state. We parameterize the latent dynamics using a learned finite-dimensional approximation of the Koopman generator, $A \in \mathbb{R}^{m\times m}$:

$$\frac{\mathrm{d}z}{\mathrm{d}t} = Az. \tag{2}$$

The corresponding latent evolution operator is solved by:

$$\mathcal{K}_A^\tau = \exp(\tau A), \qquad \tau \geq 0. \tag{3}$$

This operator satisfies the semigroup property $\mathcal{K}^{\tau_1+\tau_2} = \mathcal{K}^{\tau_1}\mathcal{K}^{\tau_2}$ and allows direct propagation to arbitrary prescribed lead time (Mezić, 2005). Accordingly, for a forecasting horizon $h\Delta t$, the predicted latent state and the corresponding physical fields are given by:

$$\widetilde{z}_{n+h} = \mathcal{K}^{h\Delta t} z_n = \exp(h\Delta t A) z_n, \qquad \widetilde{u}_{n+h} = D_\theta(\widetilde{z}_{n+h}). \tag{4}$$

3.3 Koopman Evolution as Progressive Degradation

Although continuous-time deep Koopman models permit direct prediction at prescribed lead times, finite-dimensional approximations inevitably incur closure and estimation errors. The Mori-Zwanzig formalism describes the influence of unresolved dynamics through memory and orthogonal-dynamics terms (Mori, 1965). Here, we examine how approximation errors propagate and produce discrepancies among predictions initialized differently along the same latent trajectory. By combining error accumulation analysis with the consistency of deterministic dynamics, we characterize the resulting degradation distribution, motivating the subsequent generative correction.

Let $z_{0:T} = \{z_0, z_1, \ldots, z_T\}$ denote a reference latent trajectory obtained by encoding the observed dynamics. For a target future state z_K, all historical states along the same trajectory should evolve to an identical state under the exact Koopman operator:

$$z_K = \mathcal{K}^{K-i} z_i, \quad i = 0, \ldots, K-1. \tag{5}$$

However, practical Koopman models rely on finite-dimensional approximations of the infinite-dimensional Koopman operator. Let $\widehat{\mathcal{K}}$ denote the learned Koopman operator and $\mathcal{K}$ the exact Koopman operator. Then with the mismatch modeled as the operator perturbation $\Delta\mathcal{K} = \widehat{\mathcal{K}} - \mathcal{K}$, we analyze the accumulation of Koopman approximation errors:

Lemma 1 (Accumulation of Koopman Approximation Error) *For any historical state z_i, define the n-step rollout deviation under the perturbed Koopman operator as $e_n^{(i)} = \widehat{\mathcal{K}}^n z_i - \mathcal{K}^n z_i$. As $\|\Delta\mathcal{K}\| \to 0$, the accumulated error can be approximated as*

$$e_n^{(i)} = \sum_{j=0}^{n-1} \mathcal{K}^{n-1-j} \Delta\mathcal{K} \mathcal{K}^j z_i + \mathcal{O}(\|\Delta\mathcal{K}\|^2). \tag{6}$$

The proof is provided in Appendix A.1. Lemma 1 establishes that finite-dimensional Koopman errors accumulate under repeated evolution and vary with the propagation horizon. The operator perturbation formulation can also be interpreted as a continuous-time closure error, where the discrepancy between the true dynamics and the finite-dimensional Koopman representation accumulates along the latent trajectory. We next characterize this horizon-dependent growth.

Proposition 1 (Horizon-dependent Koopman Rollout Error) *Assume that the Koopman operator is bounded as $\|\mathcal{K}\| \leq \rho$, and the finite-dimensional approximation error satisfies $\|\Delta\mathcal{K}\| \leq \epsilon$. Then, the n-step rollout deviation introduced by the finite-dimensional Koopman approximation satisfies*

$$\|e_n^{(i)}\| \leq n\rho^{n-1}\epsilon\|z_i\| + \mathcal{O}(\epsilon^2). \tag{7}$$

The proof is provided in Appendix A.2. This result characterizes the horizon-dependent accumulation of Koopman approximation error, showing that longer rollouts generally introduce larger deviation due to repeated error propagation.

Motivated by the horizon-dependent error accumulation, we interpret finite-dimensional Koopman predictions through trajectory consistency. *For deterministic dynamics, historical states along the same trajectory must evolve to the same target state at a fixed future time.* Given a target state z_K, the exact and approximate Koopman evolutions satisfy

$$z_K = \mathcal{K}^{K-i} z_i, \qquad \widehat{z}_K^{(i)} := \widehat{\mathcal{K}}^{K-i} z_i = z_K + e_{K-i}^{(i)}, \quad i = 0, \ldots, K-1. \tag{8}$$

Thus, finite-dimensional approximation breaks the trajectory consistency of the exact dynamics, with different historical origins producing horizon-dependent deviations around the same target state. We therefore regard $\{\widehat{z}_K^{(i)}\}_{i=0}^{K-1}$ as degraded realizations of z_K, which naturally induces the following degradation distribution.

Definition 1 (Koopman-induced Degradation Distribution) *Consider a target latent state z_K and the historical state index $I \sim p(I)$. The degraded prediction generated from the sampled historical state is given by $Z_K^{\deg} = \widehat{\mathcal{K}}^{K-I} z_I$. The resulting degradation distribution induced by finite-dimensional Koopman evolution is*

$$q_{\deg}(\tilde{z}_K | z_K) = \sum_{i=0}^{K-1} p(I=i)\delta\left(\tilde{z}_K - \widehat{\mathcal{K}}^{K-i} z_i\right), \tag{9}$$

where $\delta(\cdot)$ denotes the Dirac distribution. This distribution is induced by the accumulated Koopman approximation error rather than externally injected perturbations.

To obtain a tractable probabilistic characterization of the degradation distribution, we model the accumulated Koopman error as a continuous random perturbation. Under uniform sampling of historical states, this yields the following Gaussian-mixture form.

Proposition 2 (Gaussian Characterization of Koopman Degradation) *Assume that (i) the historical index is uniformly sampled, $p(I=i) = 1/K$ for $i = 0, \ldots, K-1$, and (ii) the accumulated Koopman rollout error follows $e_n^{(i)} \sim \mathcal{N}(0, \Sigma_n)$, with $\Sigma_{n+1} \succeq \Sigma_n$. Then the Koopman-induced degradation distribution admits the horizon-dependent Gaussian-mixture form*

$$q_{\deg}(\tilde{z}_K | z_K) = \frac{1}{K} \sum_{i=0}^{K-1} \mathcal{N}(\tilde{z}_K; z_K, \Sigma_{K-i}). \tag{10}$$

The proof is provided in Appendix A.3. This result reveals that finite-dimensional Koopman evolution induces a progressive degradation, where the randomness in this distribution arises from sampling the prediction origin, rather than from externally injected noise. Its displacement from the reference state and its spread describe prediction bias and origin-dependent inconsistency, respectively. Therefore, instead of directly correcting deterministic rollout errors, we formulate the long-term prediction problem as recovering the original latent trajectory from the induced degradation distribution, which motivates the proposed flow-based correction. The correction aims to reduce prediction errors while preserving trajectory-specific information.

3.4 Generative Correction

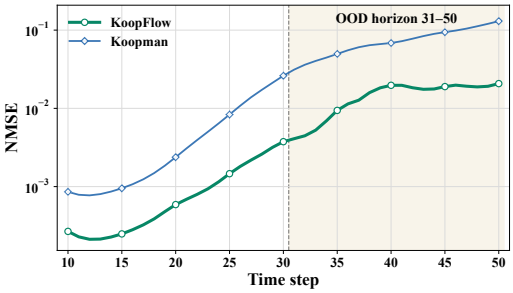

Figure 3: Long-horizon error accumulation for Koopman prediction with and without generative correction.

Building on the Koopman-induced degradation distribution, our goal is to correct a degraded prediction $\tilde{z}_K$ toward its reference latent state z_K while preserving the trajectory-specific information carried by Koopman evolution. We approach this distribution correction using a pre-trained generative prior over reference latent states. Rather than assuming that Koopman approximation errors follow the forward process of a generative model, we map the predicted state to an intermediate noisy representation and then exploit the learned generative dynamics to correct it. This decouples the prediction degradation induced by approximate Koopman evolution from the auxiliary generative path used for correction. Let $\{Z_s\}_{s\in[0,1]}$ denote a continuous path connecting the reference latent state and the degraded prediction, with $Z_0 = z_K$ and $Z_1 = \tilde{z}_K$. Figure 3 shows that generative correction (i.e., KoopFlow) substantially reduces long-horizon error accumulation relative to Koopman prediction, with increasingly pronounced improvements beyond the training horizon.

Theorem 1 (Existence of Koopman-induced Reverse Process) *Given the Koopman-induced degradation distribution $q_{\mathrm{deg}}(\tilde{z}_K|z_K)$ in Definition 1, there exists a continuous stochastic process $\{Z_s\}_{s\in[0,1]}$ satisfying*

$$Z_0 = z_K, \qquad Z_1 = \tilde{z}_K, \quad \tilde{z}_K \sim q_{\mathrm{deg}}(\cdot|z_K). \tag{11}$$

Moreover, its time-reversed process $\bar{Z}_s := Z_{1-s}$ defines a continuous reverse process from the degraded state to the accurate latent state, satisfying $\bar{Z}_0 = \tilde{z}_K$ and $\bar{Z}_1 = z_K$.

The proof is provided in Appendix A.4. Theorem 1 establishes the existence of a reverse stochastic process for recovering accurate latent states from Koopman-induced degradation, motivating the generative correction introduced below:

$$q_{\mathrm{deg}}(\tilde{z}_K|z_K) \to p_\theta(z_K|\tilde{z}_K), \tag{12}$$

where Koopman evolution provides an efficient prediction prior and the generative model performs distribution-level correction. In the following, we further characterize this process under the Gaussian degradation model derived in Proposition 2.

Remark 1 (Gaussian Form) *Under the Gaussian degradation in Proposition 2, the stochastic process in Theorem 1 can be interpreted as a continuous uncertainty accumulation process. Specifically, the discrete covariance sequence $\{\Sigma_n\}$ can be embedded into a monotonic schedule $\Sigma(s)$ with $\Sigma(s_1) \preceq \Sigma(s_2)$ for $s_1 < s_2$, leading to a stochastic trajectory $Z_s = z_K + \epsilon_s$, where $\epsilon_s \sim \mathcal{N}(0, \Sigma(s))$. The boundary conditions satisfy $Z_0 = z_K$ and $Z_1 \sim q_{\mathrm{deg}}(\cdot|z_K)$.*

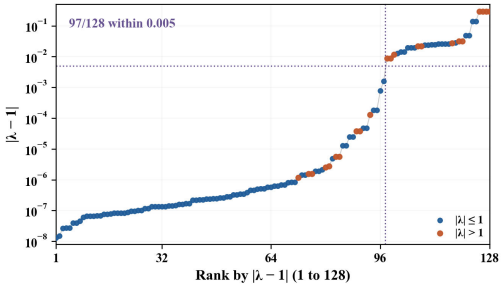

Figure 4: Koopman eigenvalue deviations from 1 at the empirically selected training horizon.

This interpretation reveals that the Koopman-induced degradation process shares a similar structure with diffusion-like stochastic processes, where the uncertainty gradually increases from accurate latent states to degraded predictions.

3.5 Framework of KoopFlow

Building on the above statistical insight, we develop KoopFlow, a prediction–correction framework that is implemented through a two-stage pipeline.

Training of Koopman Dynamics. We first jointly train the encoder E_θ, decoder D_θ, and finite-dimensional Koopman generator A. Given $z_n = E_\theta(u_{n-K+1:n}, \mu, \mathcal{B})$, we obtain the h-step Koop-

man prediction $\widetilde{z}_{n,h} = \exp(h\Delta t A) z_n$. The optimization objective $\mathcal{L}_{\text{Stage I}}$ is defined as follows:

$$\mathcal{L}_{\text{I}} = \mathbb{E}_n \left[\lambda_{\text{rec}} \| D_\phi(z_n) - u_n \|_2^2 + \frac{1}{L} \sum_{h=1}^{L} \left(\lambda_{\text{lat}} \| \widetilde{z}_{n,h} - z_{n+h} \|_2^2 + \lambda_{\text{pred}} \| D_\phi(\widetilde{z}_{n,h}) - u_{n+h} \|_2^2 \right) \right], \tag{13}$$

where L denotes the training horizon, *i.e.*, the number of future steps used for multi-step supervision, while λ_{lat} and λ_{pred} balance latent consistency and physical space prediction. We empirically find that a moderate L performs best, whereas overly short or long horizons degrade performance. As shown in Figure 4, an appropriate training horizon keeps the eigenvalues of the learned Koopman generator A close to 1, ensuring stable long-horizon inference. At inference, $\mathcal{K}^\tau = \exp(\tau A)$ enables direct prediction at arbitrary horizons τ without recursive rollout or restriction to Δt.

Training of Generative Correction. We leverage a learned generative prior to correct the latent states produced by Koopman evolution. Instead of explicitly modeling the forward degradation process, we learn the reverse transport through a latent trajectory prior learned by flow matching. According to Theorem 1, finite-dimensional Koopman evolution induces a stochastic degradation process, where degraded predictions can be viewed as corrupted states generated from the underlying trajectory distribution.

To construct this generative prior, the trained encoder E_θ is frozen and used to map ground-truth physical snapshots into the latent space via $z = E_\theta(u)$. Aggregating these latent representations defines a latent distribution p_z. To model this prior, continuous normalizing flows are parameterized via a time-dependent velocity field v_ϕ, which defines the continuous-time dynamical system $\frac{\mathrm{d}z_s}{\mathrm{d}s} = v_\phi(z_s, s)$. Once trained, this system enables generating valid latent states following p_z by integrating from standard Gaussian noise $\xi \sim \mathcal{N}(0, I_m)$ at $s = 0$ to $s = 1$. To learn this velocity field via flow matching, we independently sample $z \sim p_z$, $\xi \sim \mathcal{N}(0, I_m)$, and flow time $s \sim \mathcal{U}(0, 1)$ to construct the linear probability path $z_s = (1 - s)\xi + sz$. Differentiating this path with respect to s yields the target time-independent velocity $z - \xi$, which is used to optimize v_ϕ via the Stage II objective:

$$\mathcal{L}_{\text{II}} = \mathbb{E}_{z,\xi,s} \left[\| v_\phi(z_s, s) - (z - \xi) \|_2^2 \right]. \tag{14}$$

Given a degraded Koopman prediction $\widetilde{z}_K \sim q_{\text{deg}}(\cdot \mid z_K)$, we initialize the reverse transport from $\widetilde{z}_K$ at an intermediate rectification depth $s_r \in (0, 1)$ of the forward flow, providing a data-dependent alternative to arbitrary noise initialization:

$$\tilde{z}_K = \exp((K - i)\Delta t A) z_i, \tag{15}$$

where z_i is the initial latent state. Because the ODE dynamics are deterministic, directly integrating forward would merely reproduce the uncorrected state. To escape this trajectory, we leverage the score function of the marginal distribution $p_s(z)$. For $s < 1$, the score can be evaluated directly from the velocity field without requiring an auxiliary network:

$$g_\phi(z, s) = \frac{s v_\phi(z, s) - z}{1 - s} \approx \nabla_z \log p_s(z). \tag{16}$$

Utilizing this local ascent direction, we apply a score-guided stochastic correction to drive the state toward regions of higher probability density on the latent manifold. After this stochastic refinement, the state is integrated forward along the learned flow back to $s = 1$. We denote the final flow-rectified latent state as $\widehat{z}_K$, which is then mapped back to the physical domain as:

$$\widehat{u}_K = D_\theta(\widehat{z}_K). \tag{17}$$

For long-term rollouts, this rectification procedure is applied periodically. Between correction checkpoints, intermediate states are propagated purely via the continuous Koopman dynamics.

The two-stage framework decouples efficient prediction from trajectory recovery, with Koopman evolution enabling fast arbitrary-time forecasting and the stochastic bridge transporting degraded predictions toward the accurate latent trajectory distribution.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate KoopFlow on four benchmarks: Navier–Stokes (NS), reaction–diffusion (RD), 2DHeat, and 2D electromagnetic scattering (EM2D). Detailed dataset configurations are provided in Appendix B. **Evaluation Metrics.** We assess forecasting performance using the normalized

Table 1: Average NMSE on NS, RD, 2DHeat, and 2DEM . For NS, RD, and 2DHeat, **Train** denotes forecast time steps within the training-time range and **OOD** denotes time steps beyond the training-time range. (**Bold**: best; Underline: second best).

Method	Navier–Stokes		Reaction–Diffusion		2DHeat		2DEM	
	Train	OOD	Train	OOD	Train	OOD	Train	OOD
DeepONet (Lu et al., 2021)	2.632e-02	2.305e-01	5.396e-01	1.570e-01	2.089e-01	5.495e-02	1.219e+00	8.525e-01
FNO (Li et al., 2021)	8.096e-03	1.154e-01	**6.612e-04**	7.777e-03	**3.297e-03**	1.292e-02	1.136e-01	4.374e-01
KNO (Xiong et al., 2024)	3.254e-03	1.940e+00	2.727e-03	2.685e-03	2.828e-02	1.451e-02	7.870e+00	1.255e+01
LoLA (Rozet et al., 2025)	4.648e-02	1.300e-01	2.043e-01	2.356e-02	2.590e-01	6.877e-02	9.523e-02	3.231e-01
InfoKoop (Cheng et al., 2026)	2.976e-01	7.082e-01	5.499e-01	1.582e-01	6.170e-01	9.768e-02	1.145e+00	5.972e-01
CFO (Hou et al., 2026)	5.313e-02	3.388e-01	2.460e-03	1.818e-01	3.923e-02	2.024e-01	2.252e+00	1.808e+00
MERLIN (Lu et al., 2026)	2.097e-02	1.626e-01	6.533e-01	2.967e-01	3.811e-01	4.803e-02	5.002e-01	6.847e-01
KoopFlow (Ours)	**2.026e-03**	**5.572e-02**	1.806e-03	**2.424e-03**	3.521e-03	**1.174e-05**	**9.107e-03**	**2.898e-02**

mean squared error (NMSE) and report the eigenvector condition number $\kappa(V)$ to quantify the conditioning of the learned Koopman eigenbasis (Drmač & Mezić, 2026), where V denotes the matrix of normalized eigenvectors of the latent Koopman generator. More details are provided in the Appendix. **Baselines.** We compare KoopFlow with several representative baseline methods, including neural operators: DeepONet (Lu et al., 2021) and FNO (Li et al., 2021); Koopman learning approaches: KNO (Xiong et al., 2024), InfoKoop (Cheng et al., 2026), and MERLIN (Lu et al., 2026); and generative models: LoLA (Rozet et al., 2025), CFO (Hou et al., 2026).

4.2 Experimental Results

Long-horizon forecasting performance. Table 1 compares KoopFlow against representative baselines on NS, RD, 2DHeat and 2DEM. Within the training-time range, KoopFlow remains highly competitive, achieving the best performance on NS and the second-best performance on RD and 2DHeat, while attaining the lowest all-horizon error on 2DEM. More importantly, KoopFlow consistently achieves the lowest NMSE across all four benchmarks in the temporal-OOD regime. Compared with the strongest baseline on each benchmark, KoopFlow reduces the OOD error by 51.7%, 9.7%, 99.9%, and 91.0% on NS, RD, 2DHeat, and 2DEM. These results indicate that KoopFlow maintains strong predictive performance within the observed temporal range while exhibiting substantially improved temporal extrapolation beyond the training horizon. Figure 5 further supports this observation: KoopFlow produces forecasts that better preserve the large-scale vorticity structures, particularly at the final time $T = 50$.

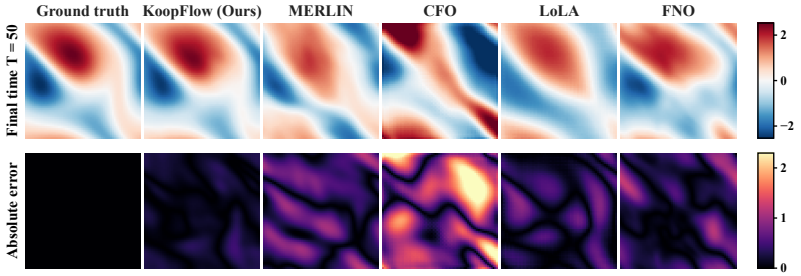

Figure 5: Qualitative comparison of KoopFlow with other baselines on NS at the final time.

Effect of correction times. Table 2 shows that recursive correction consistently outperforms direct correction in long-horizon forecasting. Compared with the direct strategy, all recursive variants reduce NMSE by more than 64%, with the best configuration achieving a 69.1% reduction. This improvement arises because continuous-time Koopman evolution allows corrections to be inserted at intermediate times, shortening the uncorrected propagation horizon and suppressing error accumulation. Accordingly, different numbers of corrections N correspond to different uncorrected propagation intervals and thus different degradation levels, providing a controllable trade-off between inference efficiency and accuracy. The variation in NMSE across N supports our theoretical result that longer uncorrected Koopman evolution leads to more severe degradation, with different horizons inducing distinct degradation levels that collectively form the degradation distribution. Moreover, the recursive settings exhibit stable correction-loss convergence, indicating that the generative model can recover latent states across multiple degradation levels. In contrast, direct prediction ($N = 1$) undergoes the longest uncorrected evolution and therefore the most severe degradation, making a single final correction substantially more difficult.

Table 2: Ablation of correction times

Correction strategy	N	NMSE ↓	Reduction ↑
Direct	1	5.408e-1	–
Recursive	3	1.814e-1	66.5%
Recursive	5	1.923e-1	64.4%
Recursive	7	1.670e-1	**69.1%**

Table 3: Ablation of Koopman training horizon

Horizon	$\kappa(V)$	Koopman NMSE	KoopFlow NMSE
$L = 2$	236.6	4.690e-1	5.523e-2
$L = 7$	144.8	1.406e-1	2.647e-2
$\mathbf{L = 12}$	88.1	**8.296e-2**	**2.489e-2**
$L = 17$	**77.1**	8.564e-2	2.976e-2

Effect of the koopman training horizon. Table 3 shows that increasing the Koopman training horizon progressively improves the conditioning of the learned eigenbasis, but does not lead to monotonically better forecasting accuracy. The reported NMSE is averaged over the entire forecasting interval from time steps 4 to 50. As L increases from 2 to 7 and 12, $\kappa(V)$ decreases from 236.6 to 144.8 and 88.1, while the prediction errors of both the Koopman backbone and KoopFlow decrease substantially. The best forecasting performance is obtained at $L = 12$. Increasing the horizon further to $L = 17$ reduces $\kappa(V)$ to 77.1, but slightly increases the prediction error. This observation suggests that better-conditioned Koopman representations alone do not necessarily translate into improved forecasting accuracy, and that a moderate training horizon provides a better balance between latent dynamical conditioning and predictive fidelity. Importantly, KoopFlow consistently improves the prediction accuracy of the underlying Koopman backbone across all training horizons. Figure 6 further shows progressively reduced eigenvector overlap as L increases.

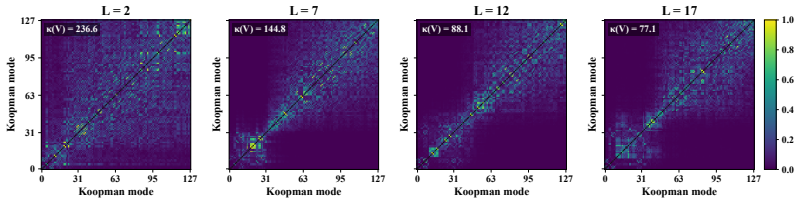

Figure 6: Eigenvector overlap of Koopman representations under different training horizons on NS

Effect of rectification depth s_r. Table 4 studies the effect of the Koopman propagation window L and the rectification starting point s_r, where $s_r = 0$ corresponds to pure noise and $s_r = 1$ to clean data. As L increases, comparable recovery increasingly favors noisier starting points (smaller s_r), while the overall error also grows substantially. This horizon-dependent shift empirically supports our theory that different Koopman propagation horizons induce distinct degradation levels, which collectively form a horizon-dependent degradation distribution, with longer horizons corresponding to more severe degradation.

Table 4: Ablation of rectification depth.

s_r	L=1	L=2	L=4	L=8
0.6	2.350e-02	5.240e-02	7.929e-02	2.755e-01
0.7	2.759e-02	5.784e-02	7.979e-02	2.746e-01
0.8	5.509e-02	7.115e-02	8.486e-02	2.776e-01

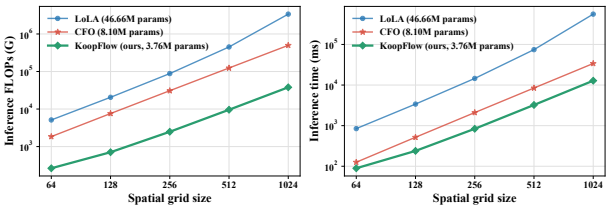

Figure 7: Comparison of inference FLOPs and runtime.

Inference efficiency. KoopFlow, CFO, and LoLA are iterative generative models, enabling a direct comparison of their computational costs. As shown in Figure 7, KoopFlow has $2.2\times$ fewer parameters than CFO and $12.4\times$ fewer than LoLA. At the 64×64 spatial grid size, it requires $7.0\times$ and $19.4\times$ less computation and is $1.4\times$ and $9.5\times$ faster than CFO and LoLA. Moreover, these efficiency gains become increasingly pronounced as the spatial grid size grows, demonstrating that KoopFlow is a more efficient and more stable generative model.

5 Conclusion

We propose **KoopFlow**, a probabilistic Koopman learning framework for modeling the accumulated effects of finite-dimensional approximation and closure errors. Our key theoretical insight is to reinterpret finite-dimensional Koopman error accumulation as a horizon-dependent degradation. Based on this view, KoopFlow couples efficient linear Koopman evolution with generative correction to recover degraded latent dynamics, effectively mitigating error accumulation during long-term rollouts. Experiments across diverse dynamical systems demonstrate improved extrapolation accuracy and stability while maintaining a favorable accuracy–efficiency trade-off.

AI USE STATEMENT

During the preparetion of this manuscript, we used generative AI tools to assist with editing and figure preparation. These tools were used to improve the clarity, readability, grammar, and consistency of the manuscript, we did not use generative AI tools for research ideation, methodological development, experimental design and implementation, data analysis, or interpretation of experimental results. All AI-assisted text and figures were reviewed and verified by the authors. All scientific contributions, including the research ideation, methodological design, experimental analysis, and interpretation of results, were conducted solely by the authors, who take full responsibility for the content of this work.

REFERENCES

Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International conference on machine learning*, pp. 475–485. PMLR, 2020.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2):e0150171, 2016. doi: 10.1371/journal.pone.0150171.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022.

Huilin Cao, Haoran Sun, Jingwei Liang, Maokun Li, and Yuxiang Ren. Physics-augmented graph transformer for multi-step time-domain maxwell's equations in electromagnetic scattering. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 10532–10542, 2026.

Xiaoyuan Cheng, Wenxuan Yuan, Yiming Yang, Yuanzhao Zhang, Sibo Cheng, Yi He, and Zhuo Sun. Information shapes koopman representation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Szh0ELyQxL.

Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction and the mori–zwanzig representation of irreversible processes. *Proceedings of the National Academy of Sciences*, 97(7): 2968–2973, 2000. doi: 10.1073/pnas.97.7.2968.

Zlatko Drmač and Igor Mezić. A data driven koopman–schur decomposition for computational analysis of nonlinear dynamics. *SIAM Journal on Scientific Computing*, 48(3):A1430–A1461, 2026. doi: 10.1137/24M1672584.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15(1):10416, 2024.

Priyam Gupta, Peter J Schmid, Denis Sipp, Taraneh Sayadi, and Georgios Rigas. Mori-zwanzig latent space koopman closure for nonlinear autoencoder. *arXiv preprint arXiv:2310.10745*, 2023.

Priyam Gupta, Peter J. Schmid, Denis Sipp, Taraneh Sayadi, and Georgios Rigas. Mori–zwanzig latent space koopman closure for nonlinear autoencoder. *Proceedings of the Royal Society A*, 481 (2313):20240259, 2025. doi: 10.1098/rspa.2024.0259.

Masih Haseli and Jorge Cortés. Learning koopman eigenfunctions and invariant subspaces from data: Symmetric subspace decomposition. *IEEE Transactions on Automatic Control*, 67(7):3442–3457, 2021.

Benjamin Holzschuh, Georg Kohl, Florian Redinger, and Nils Thuerey. P3d: Highly scalable 3d neural surrogates for physics simulations with global context. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xianglong Hou, Xinquan Huang, and Paris Perdikaris. CFO: Learning continuous-time PDE dynamics via flow-matched neural operators. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IQhaeSzyup.

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.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. APEBench: A benchmark for autoregressive neural emulators of pdes. *Advances in Neural Information Processing Systems*, 37:120252–120310, 2024.

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

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Christian Lagemann, Sajeda Mokbel, Miro Gondrum, Mario Rüttgers, Yuning Wang, Pol Suárez, Ludger Paehler, Deniz A Bezgin, Aaron B Buhendwa, Jared L Callaham, et al. The hydrogym reinforcement learning platform for fluid dynamics. *Nature*, pp. 1–8, 2026.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Yen Ting Lin, Yifeng Tian, Daniel Livescu, and Marian Anghel. Data-driven learning for the mori–zwanzig formalism: A generalization of the koopman learning framework. *SIAM Journal on Applied Dynamical Systems*, 20(4):2558–2601, 2021. doi: 10.1137/21M1401759.

Chuan Liu, Chunshu Wu, Mingkai Chen, James Liang, Ang Li, Michael Huang, Chuang Ren, Yingnian Wu, Dongfang Liu, Tong Geng, et al. Diff-pic: Revolutionizing particle-in-cell nuclear fusion simulation with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 71526–71539, 2025.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Wanfeng Lu, He Ma, Wei Lin, and Qunxi Zhu. Interpretable functional koopman learning with non-markovian closure for spatiotemporal systems. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lquDiBCgNJ.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018. doi: 10.1038/s41467-018-07210-0.

Mauroy, Igor Mezić, and Yoshihiko Susuki. *Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer International Publishing, 2020.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

Hazime Mori. Transport, collective motion, and brownian motion. *Progress of theoretical physics*, 33(3):423–455, 1965.

François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=xoNrbfbekM.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Wei Xiong, Xiaomeng Huang, Ziyang Zhang, Ruixuan Deng, Pei Sun, and Yang Tian. Koopman neural operator as a mesh-free solver of non-linear partial differential equations. *Journal of Computational Physics*, 513:113194, 2024. doi: 10.1016/j.jcp.2024.113194.

Anthony Zhou, Zijie Li, Michael Schneier, John Buchanan, and Amir Barati Farimani. Text2pde: Latent diffusion models for accessible physics simulation. In *International Conference on Learning Representations*, volume 2025, pp. 46344–46373, 2025.

Robert Zwanzig. Nonlinear generalized langevin equations. *Journal of Statistical Physics*, 9(3): 215–220, 1973.

A Theoretical Proofs

A.1 Proof of Lamma 1

Lemma 1 (Accumulation of Koopman Approximation Error). *For an arbitrary historical state z_i, define the n-step rollout deviation under the perturbed Koopman operator as*

$$e_n^{(i)} = \widehat{\mathcal{K}}^n z_i - \mathcal{K}^n z_i.$$

When $\|\Delta\mathcal{K}\|$ is sufficiently small, the accumulated error can be approximated as

$$e_n^{(i)} = \sum_{j=0}^{n-1} \mathcal{K}^{n-1-j} \Delta\mathcal{K} \mathcal{K}^j z_i + \mathcal{O}(\|\Delta\mathcal{K}\|^2).$$

Proof of Lemma 1. Let

$$\widehat{\mathcal{K}} = \mathcal{K} + \Delta\mathcal{K}.$$

Define

$$E_n = (\mathcal{K} + \Delta\mathcal{K})^n - \mathcal{K}^n.$$

Then,

$$\begin{aligned} E_n &= (\mathcal{K} + \Delta\mathcal{K})^{n-1}(\mathcal{K} + \Delta\mathcal{K}) - \mathcal{K}^n \\ &= E_{n-1}\mathcal{K} + (\mathcal{K} + \Delta\mathcal{K})^{n-1}\Delta\mathcal{K}. \end{aligned}$$

Applying this recursion iteratively and retaining first-order terms of $\Delta\mathcal{K}$ gives

$$E_n = \sum_{j=0}^{n-1} \mathcal{K}^{n-1-j} \Delta\mathcal{K} \mathcal{K}^j + O(\|\Delta\mathcal{K}\|^2).$$

Multiplying both sides by z_i yields

$$e_n^{(i)} = E_n z_i = \sum_{j=0}^{n-1} \mathcal{K}^{n-1-j} \Delta\mathcal{K} \mathcal{K}^j z_i + \mathcal{O}(\|\Delta\mathcal{K}\|^2),$$

which proves the lemma. □

A.2 Proof of Proposition 1

Proposition 1 (Horizon-dependent Koopman Rollout Error). *Assume that the Koopman operator is bounded as $\|\mathcal{K}\| \leq \rho$, and the finite-dimensional approximation error satisfies $\|\Delta\mathcal{K}\| \leq \epsilon$. Then, the n-step rollout deviation introduced by the finite-dimensional Koopman approximation satisfies*

$$\|e_n^{(i)}\| \leq n\rho^{n-1}\epsilon\|z_i\| + \mathcal{O}(\epsilon^2).$$

Proof of Proposition 1. According to Lemma 1, the n-step rollout deviation can be expressed as

$$e_n^{(i)} = \sum_{j=0}^{n-1} \mathcal{K}^{n-1-j} \Delta\mathcal{K} \mathcal{K}^j z_i + \mathcal{O}(\|\Delta\mathcal{K}\|^2).$$

Taking the norm on both sides and applying the sub-multiplicative property of operator norms, we obtain

$$\begin{aligned}\|e_n^{(i)}\| &\le \sum_{j=0}^{n-1} \left\|\mathcal{K}^{n-1-j}\Delta\mathcal{K}\mathcal{K}^j z_i\right\| + \mathcal{O}(\|\Delta\mathcal{K}\|^2) \\ &\le \sum_{j=0}^{n-1} \|\mathcal{K}\|^{n-1-j}\|\Delta\mathcal{K}\|\|\mathcal{K}\|^j\|z_i\| + \mathcal{O}(\epsilon^2).\end{aligned}$$

By the boundedness assumption $\|\mathcal{K}\| \le \rho$ and $\|\Delta\mathcal{K}\| \le \epsilon$, we have

$$\begin{aligned}\|e_n^{(i)}\| &\le \sum_{j=0}^{n-1} \rho^{n-1-j}\epsilon\rho^j\|z_i\| + \mathcal{O}(\epsilon^2) \\ &= n\rho^{n-1}\epsilon\|z_i\| + \mathcal{O}(\epsilon^2),\end{aligned}$$

which completes the proof. □

A.3 Proof of Proposition 2.

Proposition 2 (Gaussian Characterization of Koopman Degradation). *Assume that 1) the historical state index is uniformly sampled from the available trajectory segment $p(I = i) = \frac{1}{K}, (i = 0, \dots, K-1)$, and 2) the accumulated Koopman rollout error follows a zero-mean Gaussian distribution $e_n^{(i)} \sim \mathcal{N}(0, \Sigma_n)$ where the covariance increases with the propagation horizon ($\Sigma_{n+1} \succeq \Sigma_n$). Then, the Koopman-induced degradation distribution reduces to the following horizon-dependent Gaussian mixture:*

$$q_{\text{deg}}(\tilde{z}_K|z_K) = \frac{1}{K}\sum_{i=0}^{K-1} \mathcal{N}(\tilde{z}_K; z_K, \Sigma_{K-i}).$$

Proof of Proposition 2. For the historical state z_i, the corresponding rollout horizon to the target state z_K is $K - i$. According to Lemma 1, the degraded prediction can be expressed as

$$\widehat{z}_K^{(i)} = z_K + e_{K-i}^{(i)}.$$

By assumption, the accumulated Koopman rollout error follows

$$e_{K-i}^{(i)} \sim \mathcal{N}(0, \Sigma_{K-i}).$$

Since adding a deterministic vector shifts the mean of a Gaussian distribution without changing its covariance, we have

$$\widehat{z}_K^{(i)} = z_K + e_{K-i}^{(i)} \sim \mathcal{N}(z_K, \Sigma_{K-i}).$$

Furthermore, let the historical index I be uniformly sampled from the trajectory segment, i.e.,

$$p(I = i) = \frac{1}{K}, \quad i = 0, \dots, K-1.$$

The degradation distribution induced by the finite-dimensional Koopman rollout is therefore given by the marginal distribution over all possible historical initial states:

$$\begin{aligned}q_{\text{deg}}(\tilde{z}_K|z_K) &= \sum_{i=0}^{K-1} p(I = i)p(\tilde{z}_K|z_K, I = i) \\ &= \frac{1}{K}\sum_{i=0}^{K-1} \mathcal{N}(\tilde{z}_K; z_K, \Sigma_{K-i}).\end{aligned}$$

This completes the proof. □

A.4 Proof of Theorem 1

Theorem 1 (Existence of Koopman-induced Stochastic Bridge). *Given the Koopman-induced degradation distribution $q_{\deg}(\tilde{z}_K|z_K)$ characterized in Definition 1, there exists a continuous stochastic bridge process $\{Z_s\}_{s\in[0,1]}$ connecting the accurate latent state and the degraded prediction, satisfying*

$$Z_0 = z_K, \qquad Z_1 = \tilde{z}_K, \quad \tilde{z}_K \sim q_{\deg}(\cdot|z_K).$$

Moreover, its time-reversed process $\bar{Z}_s := Z_{1-s}$ defines a continuous reverse bridge satisfying $\bar{Z}_0 = \tilde{z}_K$ and $\bar{Z}_1 = z_K$.

Proof of Theorem 1. The Koopman-induced degradation distribution defines the joint distribution

$$p(z_K, \tilde{z}_K) = p(z_K) q_{\deg}(\tilde{z}_K|z_K),$$

which induces a coupling γ between the accurate and degraded latent distributions. Let

$$(z_0, z_1) \sim \gamma, \qquad z_0 = z_K, \quad z_1 = \tilde{z}_K.$$

Since both endpoints lie in the same latent vector space, we construct the interpolation

$$Z_s = (1-s)z_0 + sz_1, \qquad s \in [0,1].$$

For each coupled pair (z_0, z_1), Z_s is continuous in s and satisfies

$$Z_0 = z_K, \qquad Z_1 = \tilde{z}_K \sim q_{\deg}(\cdot|z_K),$$

establishing the existence of a continuous stochastic bridge between the accurate and degraded latent states.

Now define its time reversal by

$$\bar{Z}_s := Z_{1-s}, \qquad s \in [0,1].$$

Substituting the interpolation above gives

$$\bar{Z}_s = sz_0 + (1-s)z_1 = (1-s)\tilde{z}_K + sz_K.$$

Since Z_s is continuous, $\bar{Z}_s$ is also continuous, with boundary conditions

$$\bar{Z}_0 = \tilde{z}_K, \qquad \bar{Z}_1 = z_K.$$

Therefore, $\{\bar{Z}_s\}_{s\in[0,1]}$ defines a continuous reverse bridge from the Koopman-induced degraded state to the accurate latent state. □

B Dataset Details

Navier–Stokes (NS). Following Li et al. (2021), we consider the two-dimensional incompressible Navier–Stokes equations in vorticity form on the periodic unit torus $\Omega = (0,1)^2$:

$$\partial_t \omega + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega + f, \qquad \nabla \cdot \mathbf{u} = 0, \tag{18}$$

where $\omega = \nabla \times \mathbf{u}$ denotes the vorticity, $\mathbf{u}$ is the velocity field, and the viscosity is fixed to $\nu = 10^{-3}$. The external forcing is

$$f(x_1, x_2) = 0.1\left[\sin\left(2\pi(x_1 + x_2)\right) + \cos\left(2\pi(x_1 + x_2)\right)\right]. \tag{19}$$

The initial vorticity is sampled from the Gaussian random field

$$\omega_0 \sim \mathcal{N}\left(0,\, 7^{3/2}(-\Delta + 49I)^{-2.5}\right), \tag{20}$$

with periodic boundary conditions.

We use the publicly released `NavierStokes_V1e-3_N5000_T50` dataset from Li et al. (2021), which contains 5,000 trajectories on a 64×64 spatial grid.For all experiments, we use the first 1,000 trajectories, randomly split with seed 777 into 800 training trajectories and 200 test trajectories.

Reaction–diffusion (RD). We consider a two-channel periodic reaction–diffusion system on $\Omega = [0, 2\pi)^2$:

$$\begin{aligned} \partial_t u &= \Delta u + u - u^3 - v + 0.01, \\ \partial_t v &= \Delta v + \beta(u - v), \end{aligned} \tag{21}$$

where u and v denote the two evolving fields, and β controls their reaction coupling. The initial conditions are sampled from random truncated Fourier series with periodic boundary conditions. The dataset contains 200 trajectories on a 64×64 spatial grid, with 40 trajectories for each $\beta \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. The 200 trajectories in the training file. An independent set of 30 trajectories is used for testing.

2DHeat. Following Koehler et al. (2024), we consider the two-dimensional heat equation on the periodic unit square $\Omega = [0, 1)^2$:

$$\partial_t h = \nu \Delta h, \tag{22}$$

where h denotes the scalar temperature field and ν is the trajectory-wise constant diffusivity. No external heat source is applied. The initial temperature fields are sampled from random Fourier fields with a cutoff wavenumber of five, zero mean, and unit maximum absolute amplitude, with periodic boundary conditions.

We use trajectories generated with the diffusion scenario of APEBench (Koehler et al., 2024), where each trajectory is simulated on a 64×64 periodic spatial grid with a temporal interval of $\Delta t = 0.1$. The dataset contains 250 training trajectories and 60 validation trajectories. Evaluation is performed on 360 independent test trajectories, comprising 150 trajectories at seen diffusivities, 90 at interpolation diffusivities, and 120 at extrapolation diffusivities.

2D electromagnetic scattering(2DEM). We consider two-dimensional electromagnetic scattering by dielectric objects under transverse-magnetic (TM_z) polarization, governed by

$$\begin{aligned} \frac{\partial E_z}{\partial y} &= -\mu_0 \frac{\partial H_x}{\partial t}, \\ \frac{\partial E_z}{\partial x} &= \mu_0 \frac{\partial H_y}{\partial t}, \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \varepsilon_0 \varepsilon_r \frac{\partial E_z}{\partial t} + J_z, \end{aligned} \tag{23}$$

where E_z is the out-of-plane electric field and H_x and H_y are the in-plane magnetic fields. The medium is lossless and nonmagnetic. The spatially varying relative permittivity ε_r determines the geometry and material property of the dielectric scatterer: the background is vacuum with $\varepsilon_r = 1$, while the scatterer has relative permittivity selected from $\{6, 6.5, 7, 7.5, 8\}$ and one of four predefined geometries: ellipses, triangles, quadrilaterals, and tri-lobed structures. Second-order Mur absorbing boundary conditions are imposed along the four outer boundaries.

The electromagnetic fields are excited by a Gaussian-modulated sinusoidal point source.

$$J_z(t) = -\cos(\omega_c t) \exp\left[-\frac{4\pi(t - t_0)^2}{\tau^2}\right], \tag{24}$$

where the center frequency ω_c is randomly sampled between 4×10^{15} and $6 \times 10^{15}\,\mathrm{rad/s}$, the temporal delay t_0 is randomly sampled between 1.8 and $2.2\,\mathrm{fs}$, and the pulse width is set to $\tau = t_0/0.8$. The source location is also varied across trajectories.

The dataset contains 200 trajectories generated using the FDTD method on a 128×128 Yee grid over an approximately $2.39\,\mu\mathrm{m} \times 2.39\,\mu\mathrm{m}$ domain, with $\Delta x = \Delta y \approx 18.84\,\mathrm{nm}$ and $\Delta t \approx 0.044\,\mathrm{fs}$. We use the post-excitation source-free evolution of the electromagnetic fields for forecasting.

C Evaluation Metrics

Normalized Mean Squared Error (NMSE). We evaluate forecasting accuracy using the normalized mean squared error (NMSE) employed in our main comparison table. For each physical-field

Table 5: Training configurations on NS, 2DHeat, 2DEM2D, and RD.

Category	Parameter	NS	2DHeat	RD	2DEM
Koopman stage					
	Epochs	800	1,000	800	800
	Batch size	32	32	32	32
	Optimizer	Adam	Adam	Adam	Adam
	Learning rate	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-4}
	Weight decay	1×10^{-4}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	Training-time range	1–30	1–51	1–51	1–30
	History-state length	3	3	3	3
	Prediction horizon	12	12	12	12
	λ_{rec}	1	1	1	1
	λ_{lat}	0.05	0.01	0.01	0.01
	λ_{phy}	0.1	1	1	1
KoopFlow stage					
	Epochs	500	500	500	500
	Batch size	64	64	64	64
	Optimizer	Adam	Adam	Adam	Adam
	Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
	Weight decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}

channel c, let μ_c and σ_c denote the mean and standard deviation estimated exclusively from the permitted training trajectories and training-time steps. These statistics are fixed during validation and testing. The normalized ground-truth and predicted fields are defined as

$$\widetilde{\mathbf{u}}_{n,t,c} = \frac{\mathbf{u}_{n,t,c} - \mu_c}{\sigma_c}, \qquad \widehat{\widetilde{\mathbf{u}}}_{n,t,c} = \frac{\widehat{\mathbf{u}}_{n,t,c} - \mu_c}{\sigma_c}, \tag{25}$$

where n, t, and c index test trajectories, forecast time steps, and physical-field channels.

For a set of forecast time steps $\mathcal{T}$, the reported NMSE is the mean squared error in this fixed normalized field space:

$$\mathrm{NMSE}(\mathcal{T}) = \frac{1}{N|\mathcal{T}|C|\Omega|} \sum_{n=1}^{N} \sum_{t\in\mathcal{T}} \sum_{c=1}^{C} \left\| \widehat{\widetilde{\mathbf{u}}}_{n,t,c} - \widetilde{\mathbf{u}}_{n,t,c} \right\|_2^2, \tag{26}$$

where N is the number of test trajectories, C is the number of physical-field channels, and $|\Omega|$ is the number of spatial grid points.

Eigenvector-Matrix Condition Number. Let $A \in \mathbb{R}^{m\times m}$ denote the learned latent Koopman generator. Assuming that A is diagonalizable, we write

$$A = V\Lambda V^{-1}, \tag{27}$$

where $V = [v_1, \ldots, v_m]$ contains the right eigenvectors of A. Since eigenvectors are defined only up to an arbitrary nonzero scaling, we normalize each eigenvector to unit Euclidean norm,

$$v_i \leftarrow \frac{v_i}{\|v_i\|_2}. \tag{28}$$

We then compute

$$\kappa_2(V) = \|V\|_2 \|V^{-1}\|_2 = \frac{\sigma_{\max}(V)}{\sigma_{\min}(V)}, \tag{29}$$

where $\sigma_{\max}(V)$ and $\sigma_{\min}(V)$ denote the largest and smallest singular values of V, respectively.

D Training and evaluation details

Common protocol. All KoopFlow experiments use random seed 777 and are conducted on NVIDIA GeForce RTX 4090 GPUs. For all datasets, the first three observed snapshots are used

as the historical context, from which the subsequent trajectory is predicted autonomously without access to future ground-truth states. This choice follows the history-length ablation reported by MERLIN (Lu et al., 2026). We use the same three-frame input protocol across all methods to ensure a consistent information budget. Channel-wise normalization statistics are estimated exclusively from the permitted training trajectories and are then fixed for validation and testing. The Koopman stage learns an encoder, a decoder, and a latent evolution operator. In the KoopFlow stage, all Koopman components are frozen, and a conditional flow model is trained to correct the coarse latent trajectory produced by the frozen operator. Detailed dataset-specific training configurations are summarized in Table 5.

Evaluation protocol. We evaluate temporal out-of-distribution generalization by separately reporting errors within and beyond the training-time range. For NS and EM2D the Train and OOD intervals correspond to forecast time steps 4–30 and 31–50, respectively. For RD and Heat2D, they correspond to time steps 4–51 and 52–101 . We further evaluate physical-parameter generalization on Heat2D using previously unseen diffusivities, including interpolation within the training parameter range and extrapolation below and above that range. The temporal OOD and physical-parameter generalization results are reported in Tables 1 and 6.

E BENCHMARK PROTOCOL

Baseline training. All baselines use the same trajectory split, visible training-time range,and training-set normalization statistics as KoopFlow. To avoid repetitive descriptions, we detail the training procedure on NS below. The same method-specific objectives and architectures are used on RD,Heat2D,and EM2D .

FNO. FNO (Li et al., 2021) uses four Fourier layers with width 32 and 16 Fourier modes. Each sequence maps three historical observations to the subsequent T time steps. The model is trained by minimizing the MSE over the complete predicted field block. We train for 500 epochs using Adam with batch size 32, learning rate 10^{-3},weight decay 10^{-4}, gradient clipping at 1.0, and StepLR$(25, 0.9)$.

DeepONet. DeepONet (Lu et al., 2021) uses 128 basis functions and a three-layer trunk network with hidden width 128. It is trained on the same sequences as FNO using the full-block field MSE. The optimizer, batch size, learning rate, weight decay, training budget, and learning-rate schedule are also identical to those used for FNO.

KNO. KNO (Xiong et al., 2024) uses an observable width of 32, retains 16 Fourier modes, and applies a linear Koopman operator with power 8. It uses the same NS training sequences and optimization schedule as FNO. Its objective is

$$\mathcal{L}_{\mathrm{KNO}} = 5\mathcal{L}_{\mathrm{pred}} + 0.5\mathcal{L}_{\mathrm{rec}}, \tag{30}$$

MERLIN. MERLIN (Lu et al., 2026) is trained using length-10 sequences contained entirely within NS time steps 1–30. Its Phase–I stage learns the encoder, decoder, and latent dynamics, while the residual-memory component models unresolved latent dynamics. Training uses Adam for 500 epochs with an initial learning rate of 5×10^{-3}, weight decay 10^{-4}, and StepLR$(50, 0.8)$.

LoLA. LoLA (Rozet et al., 2025) first learns a field autoencoder and then trains a latent diffusion model on temporal latent bundles. The autoencoder is trained for 500 epochs with 128 updates per epoch, batch size 64, and learning rate 10^{-4}. The latent diffusion transformer has width 384, depth 8, and temporal bundle length 4. It is trained for 1,024 epochs with 64 updates per epoch, batch size 256, and learning rate 10^{-4}.

CFO. CFO (Hou et al., 2026) uses the official continuous-flow objective and UNet2D architecture. The 800 NS training trajectories over time steps 1–30 are represented using quintic temporal splines. Training uses 5,000 updates, batch size 8, learning rate 2×10^{-4}, and regularization coefficient $\gamma = 10^{-5}$. Validation is performed every 1,000 updates.

Table 6: Results on the Heat2D generalization benchmark.

Method	Overall	Seen	Interpolation	Extrapolation (low)	Extrapolation (high)
DeepONet (Lu et al., 2021)	1.304e-01	1.050e-01	7.402e-02	3.544e-01	5.433e-02
FNO (Li et al., 2021)	8.209e-03	3.147e-03	1.012e-03	3.924e-02	6.230e-04
KNO (Xiong et al., 2024)	2.126e-02	4.652e-03	4.575e-03	1.006e-01	8.441e-03
LoLA (Rozet et al., 2025)	1.619e-01	1.363e-01	8.142e-02	4.726e-01	3.622e-02
InfoKoop (Cheng et al., 2026)	3.521e-01	3.223e-01	2.357e-01	8.010e-01	1.519e-01
CFO (Hou et al., 2026)	1.225e-01	1.147e-01	9.620e-02	2.236e-01	8.011e-02
MERLIN (Lu et al., 2026)	2.112e-01	1.781e-01	1.180e-01	5.673e-01	7.761e-02
KoopFlow	**1.730e-03**	**9.260e-04**	**2.900e-04**	**7.564e-03**	**6.800e-05**

Information-Koopman. Information-Koopman (Cheng et al., 2026) is trained on fixed-length temporal sequences sampled exclusively from the permitted training-time range of each dataset. The sequence length and sampling density are adjusted according to the available training horizon. It is trained for 500 epochs using Adam with batch size 16, learning rate 10^{-3}, and StepLR$(20, 0.8)$. The training objective is

$$\mathcal{L}_{\mathrm{InfoKoop}} = 0.1\mathcal{L}_{\mathrm{rec}} + 2\mathcal{L}_{\mathrm{InfoNCE}} + 0.1\mathcal{L}_{\mathrm{ent}}, \tag{31}$$

with InfoNCE neighborhood size 3, temperature 0.07, and ridge coefficient 0.1.

E.1 Additional Experimental Results and Analysis

Results on physical-parameter generalization. We further evaluate the generalization ability of KoopFlow to unseen physical parameters on Heat2D. As shown in Table 6, KoopFlow achieves the lowest error across all evaluation regimes, including seen parameters, interpolation within the training range, and extrapolation beyond the lower and upper boundaries of the training range. Compared with FNO, the strongest baseline across all five metrics, KoopFlow reduces the overall error by 78.9%, and improves the seen and interpolation errors by 70.6% and 71.3%, respectively. The improvement is even more pronounced under parameter extrapolation, with error reductions of 80.7% and 89.1% for the low- and high-parameter regimes.

The Analysis of Dissipative Dynamics and Late-Time Prediction on Heat2D. We further analyze the particularly low temporal-OOD error observed on Heat2D. Heat2D is governed by a purely dissipative diffusion process without an external heat source. As the system evolves, high-frequency spatial modes are progressively attenuated, and the temperature field becomes increasingly smooth and approaches an equilibrium state. Consequently, the intrinsic complexity of the late-time dynamics decreases, making states at longer horizons easier to predict. Nevertheless, all methods are evaluated under the same normalization and temporal split, and KoopFlow consistently yields the lowest forecasting error in this late-time regime, indicating that it accurately captures the dissipative evolution without substantial long-horizon error accumulation.

Long-horizon rollout stability. Figure 8 further compares the evolution of forecasting errors over time. Beyond the training-time range, KoopFlow exhibits substantially more stable error dynamics than the competing methods. On NS, its error increases after entering the temporal-OOD regime but gradually saturates, rather than continuing to accumulate over the rollout horizon. On RD, KoopFlow maintains an almost constant error level across the training/OOD boundary, while several baselines exhibit abrupt error growth or pronounced fluctuations at later time steps. The stabilization effect is even more evident on 2Dheat, where the KoopFlow error rapidly decreases to a low level and remains nearly unchanged throughout the remaining rollout. In contrast, the errors of several baselines continue to grow with the prediction horizon. These results suggest that KoopFlow not only improves pointwise forecasting accuracy but also suppresses progressive error accumulation, leading to more stable long-horizon rollouts under temporal distribution shift.

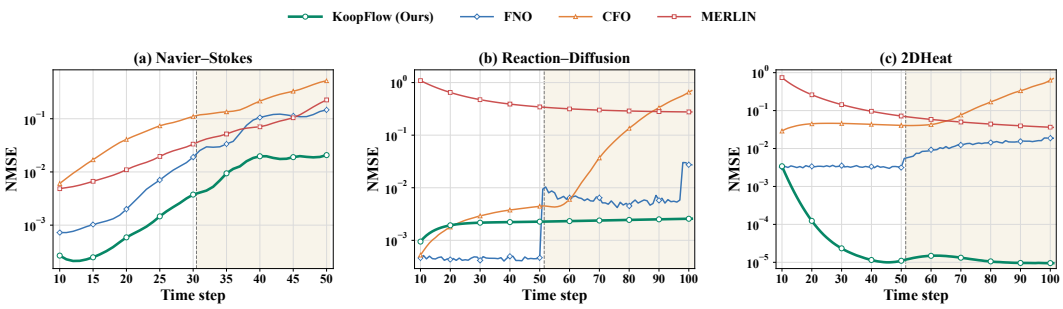

Figure 8: Error comparison of KoopFlow with strong baselines across three benchmarks.

E.2 MORE QUALITATIVE EXAMPLES

This section presents additional qualitative examples comparing Koopflow against the baseline models.

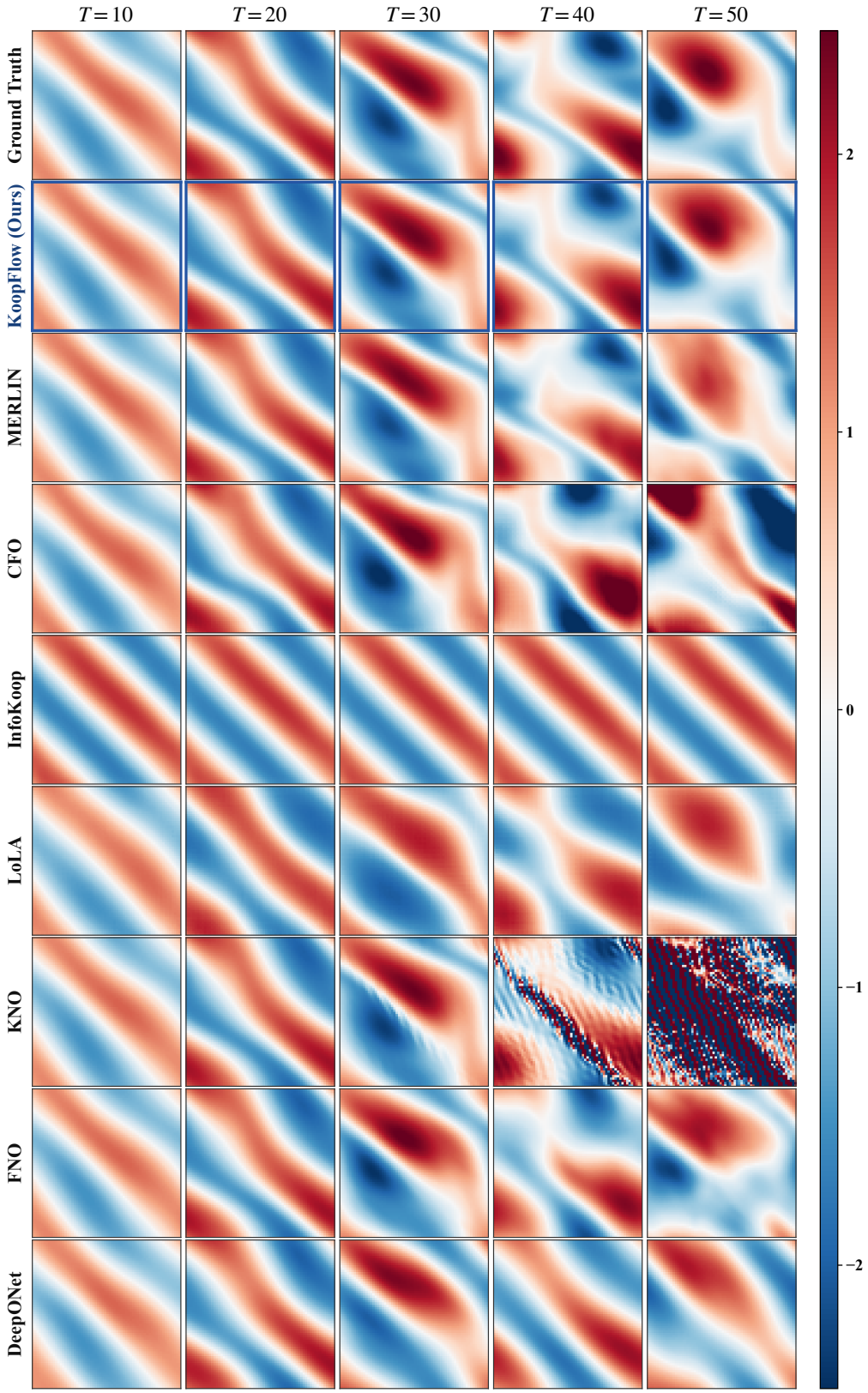

Figure 9: Qualitative comparison of Koopflow with other baselines on Navier-Stokes

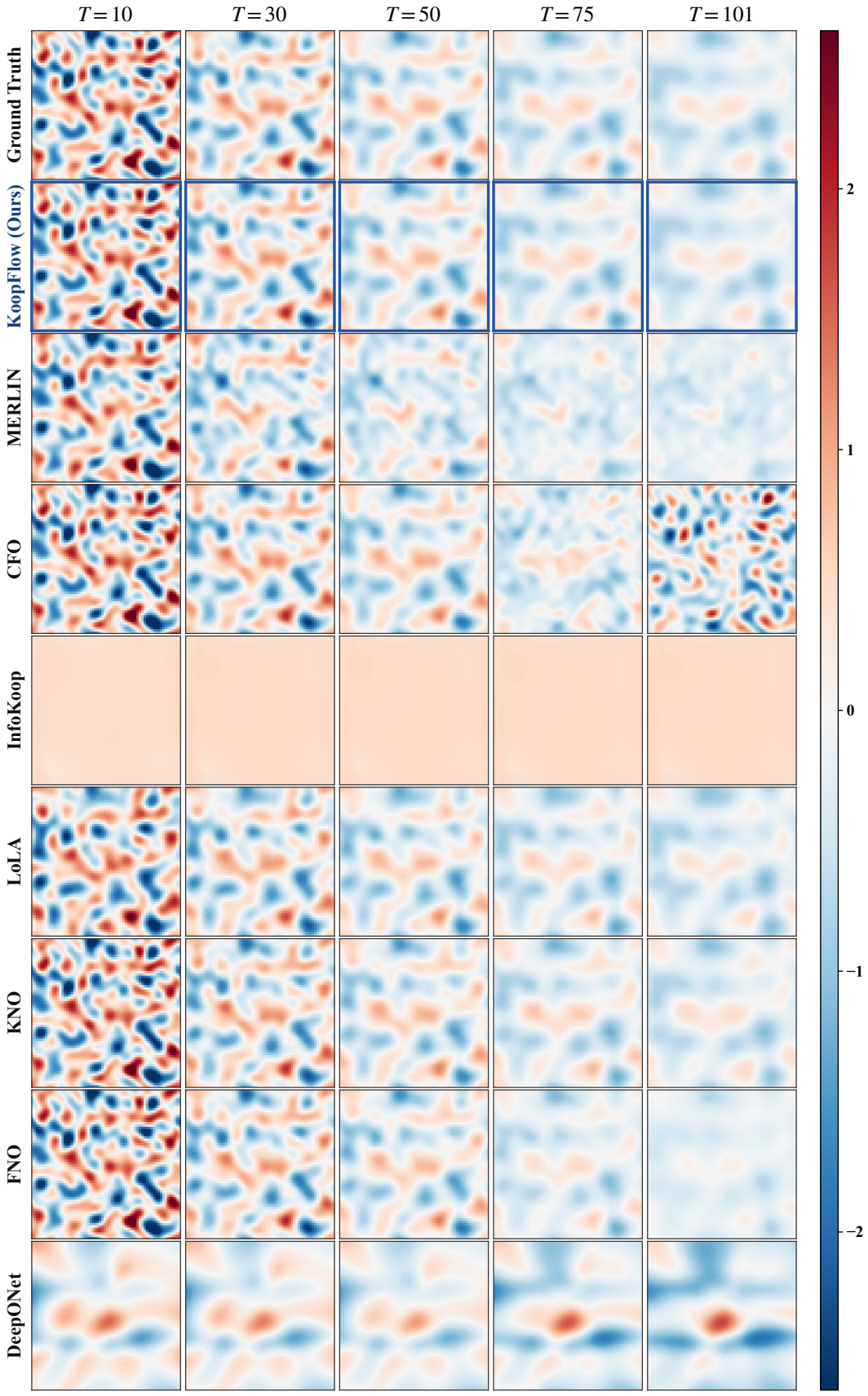

Figure 10: Qualitative comparison of Koopflow with other baselines on Reaction–diffusion

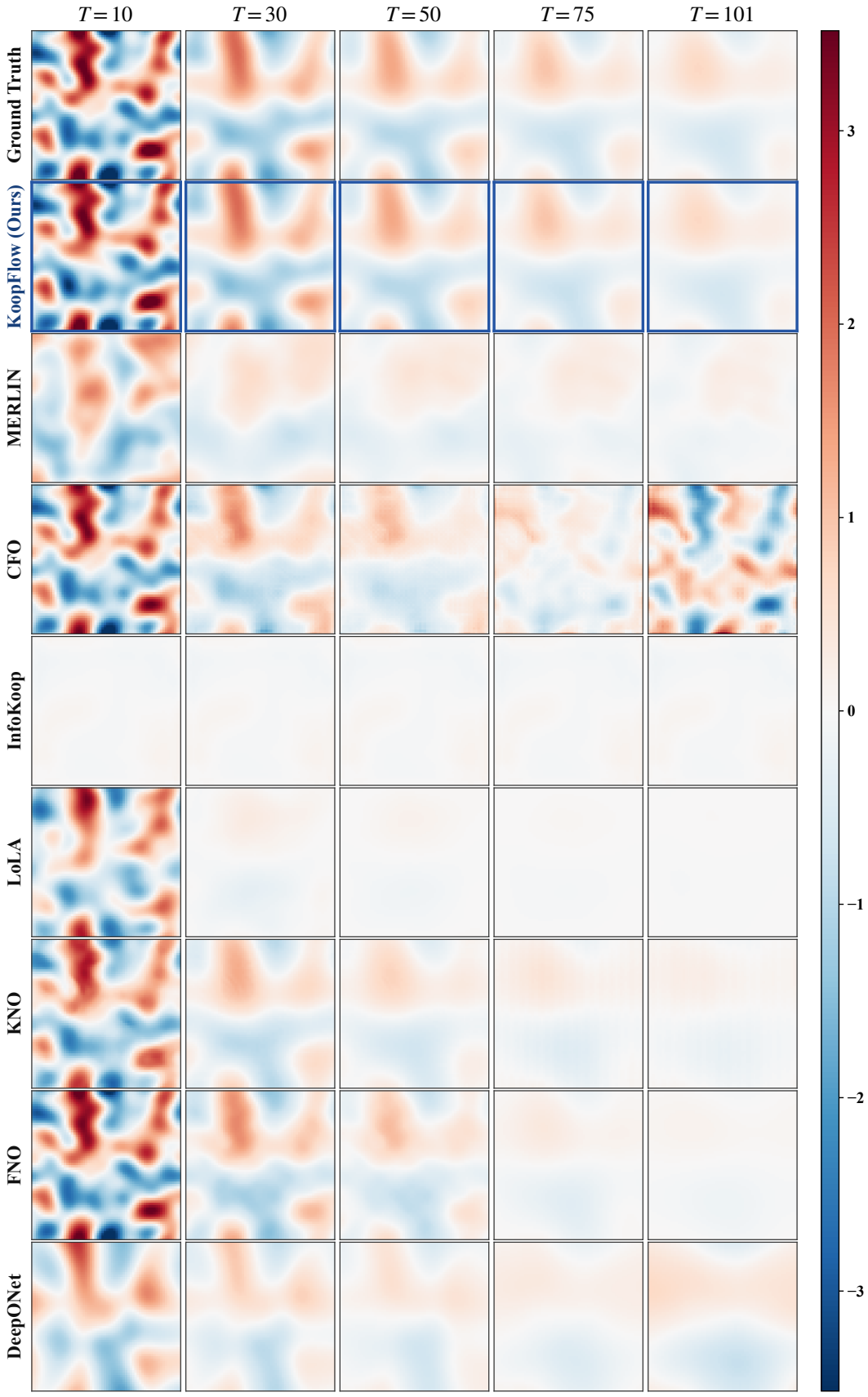

Figure 11: Qualitative comparison of Koopflow with other baselines on Heat2D