SPIKE: Sparse Koopman Regularization for Physics-Informed Neural Networks

Jose Marie Antonio Miñoza
Center for AI Research PH

Physics-Informed Neural Networks (PINNs) provide a mesh-free approach for solving differential equations by embedding physical constraints into neural network training. However, PINNs tend to overfit within the training domain, leading to poor generalization when extrapolating beyond trained spatiotemporal regions. This work presents SPIKE (Sparse Physics-Informed Koopman-Enhanced), a framework that regularizes PINNs with continuous-time Koopman operators to learn parsimonious dynamics representations. By enforcing linear dynamics $dz/dt = Az$ in a learned observable space, both PIKE (without explicit sparsity) and SPIKE (with L1 regularization on A) learn sparse generator matrices, embodying the parsimony principle that complex dynamics admit low-dimensional structure. Experiments across parabolic, hyperbolic, dispersive, and stiff PDEs, including fluid dynamics (Navier-Stokes) and chaotic ODEs (Lorenz), demonstrate consistent improvements in temporal extrapolation, spatial generalization, and long-term prediction accuracy. The continuous-time formulation with matrix exponential integration provides unconditional stability for stiff systems while avoiding diagonal dominance issues inherent in discrete-time Koopman operators.

1. Introduction

Solving partial differential equations with reliable out-of-distribution (OOD) generalization remains a fundamental challenge in scientific computing [1–3]. Physics-Informed Neural Networks (PINNs), introduced by Raissi et al. [4], provide a mesh-free approach by embedding differential equation residuals directly into neural network training. Given a partial differential equation $\mathcal{N}[u] = f$ where $\mathcal{N}$ is a nonlinear differential operator, PINNs minimize a physics loss:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_r} \sum_{i=1}^{N_r} |\mathcal{N}[u_\theta](x_i, t_i) - f(x_i, t_i)|^2 \tag{1}$$

alongside initial and boundary condition losses. This formulation enables mesh-free solutions with minimal data requirements [4]. However, PINNs face two key limitations: (1) the neural network representation u_θ remains opaque: the learned weights do not directly reveal the underlying dynamics, limiting interpretability; and (2) without structural regularization, PINNs tend to overfit within the training domain, leading to rapid error growth when extrapolating beyond trained time horizons [3].

Koopman operator theory [5] provides a complementary framework where nonlinear dynamical systems admit linear representations in infinite-dimensional observable spaces. For an autonomous system $\dot{x} = f(x)$, the Koopman operator family $\mathcal{K}^t$ acts on observables $g : \mathcal{X} \to \mathbb{C}$ via $\mathcal{K}^t g(x) = g(F^t(x))$, where F^t denotes the flow map. The infinitesimal generator $\mathcal{L}$, also known as the Lie operator, satisfies the fundamental relation [6]:

$$\mathcal{L}g = \nabla g \cdot f \tag{2}$$

In finite-dimensional approximations, this yields linear dynamics $dz/dt = Az$ where $z = g(x)$ represents lifted observables and $A \in \mathbb{R}^{M \times M}$ approximates the generator.

The present work proposes a framework where the PINN remains the base model for solving PDEs via automatic differentiation, with the Koopman component serving as an auxiliary regularizer that

Third Conference on Parsimony and Learning (CPAL 2026).

promotes sparse, interpretable structure in the learned dynamics. This inverts the typical paradigm: rather than augmenting Koopman methods with physics constraints, the approach enhances PINNs with Koopman regularization.

Nomenclature. **PIKE** (Physics-Informed Koopman-Enhanced) denotes the base framework with Koopman regularization. **SPIKE** (Sparse PIKE) adds explicit L1 sparsity on the generator matrix A. Both learn sparse structure; SPIKE enforces it explicitly.

Contributions. This work makes the following contributions:

1. **Koopman regularization for generalization**: The sparse Koopman constraint acts as an implicit regularizer, improving temporal extrapolation by 2–184$\times$ and preventing catastrophic failure outside training regions. For open-domain systems, this yields physically meaningful spatial generalization; for bounded domains, it ensures smooth, non-divergent extrapolation behavior.
2. **Parsimonious dynamics representation**: L1 sparsity reduces non-zero generator entries by up to 5.7$\times$, yielding parsimonious representations where complex PDE dynamics are captured by sparse A matrices. This embodies the parsimony principle [7] that physical systems admit low-dimensional structure. For forward problems, this validates that learned dynamics match expected physics; for inverse problems, it enables hypothesis generation.
3. **Library-latent decomposition**: A dual-component observable embedding combining explicit polynomial terms with learned MLP features. The library component provides a structured representation of polynomial-in-u dynamics (e.g., $u - u^3$ reaction terms) via sparse A matrix entries, while the latent component captures derivative-correlated structure (up to 0.99 correlation with u_{xx}). For inverse problems, this decomposition enables coefficient recovery; for forward problems, it provides interpretability and validation.
4. **Continuous-time Koopman formulation**: Direct learning of the generator A via $dz/dt = Az$, avoiding the diagonal dominance issues inherent in discrete-time formulations.

2. Related Work

Physics-Informed Neural Networks. The PINN framework [4] approximates solutions to differential equations using neural networks trained with physics-based loss functions. For a PDE $u_t + \mathcal{N}[u] = 0$, a neural network $u_\theta(x, t)$ minimizes $\mathcal{L} = \lambda_r \mathcal{L}_{\text{physics}} + \lambda_{ic}\mathcal{L}_{IC} + \lambda_{bc}\mathcal{L}_{BC}$ where automatic differentiation computes required derivatives. PINNs have been applied to Navier-Stokes equations [4] and inverse problems [8]. However, PINNs can fail to train due to spectral bias and gradient pathologies [1], and tend to overfit within the training domain. Extensions address training difficulties through gradient balancing [9], loss modifications, and architectural innovations; Hanna et al. [10] proposed variance-based regularization, while AC-PKAN [11] combines Chebyshev-based Kolmogorov-Arnold Networks with attention mechanisms. Such approaches improve training stability or expressiveness but do not address OOD generalization—PIKE/SPIKE is orthogonal and could be combined with these architectures. Neural operators (FNO [12], DeepONet [13]) learn function-space mappings but also exhibit OOD limitations [14, 15].

Koopman Operator Theory. The Koopman operator provides a linear representation of nonlinear dynamics in an infinite-dimensional function space [6]. Dynamic Mode Decomposition (DMD) [16] and Extended DMD [17] approximate this operator from data. Deep Koopman approaches include autoencoders for invariant subspaces [18–20] and Physics-Informed Koopman Networks (PIKN) [21], which incorporate the Lie operator constraint $\mathcal{L}g = \nabla g \cdot f$. Unlike PIKN, which requires explicit knowledge of the dynamics $f(x)$, PIKE operates directly through the PDE residual. The Koopman Regularization framework of Cohen [22] establishes that N functionally independent Koopman eigenfunctions suffice for exact dynamics recovery in an N-dimensional system, supporting the use of low-dimensional A matrices. However, while Koopman Regularization targets system identification, PIKE/SPIKE targets OOD generalization for PDE solving.

Sparse Dynamics Discovery. SINDy [23] pioneered sparse regression using finite-difference derivatives. Extensions include PDE-FIND [24], neural hybrids [25], and mesh-free variants [26]. SPIKE addresses their noise sensitivity via autograd derivatives from the trained PINN.

Key Distinctions. Table 4 summarizes methodological differences. PIKE/SPIKE: (1) operates on PDE residuals, not explicit $f(x)$; (2) targets OOD generalization, not training stability [10] or system identification; (3) employs continuous-time $dz/dt = Az$ with multiple integrators (Euler/RK4/expm); (4) enables post-hoc sparse discovery without retraining.

3. Theoretical Foundation

This section establishes the theoretical basis connecting the Koopman infinitesimal generator to physics-informed learning. The key insight, following Liu et al. [21], is that the Lie operator relation provides a principled mechanism for enforcing dynamical consistency.

3.1. Koopman Generator and Physics Constraints

Definition 1 (Koopman Operator Family)**.** *For an autonomous dynamical system $\frac{d}{dt}x(t) = f(x(t))$ with $x \in \mathcal{X} \subseteq \mathbb{R}^n$, let $F^t : \mathcal{X} \to \mathcal{X}$ denote the time-t flow map satisfying $x(t_0 + t) = F^t(x(t_0))$. The Koopman operator family $\mathcal{K}^t : \mathcal{G}(\mathcal{X}) \to \mathcal{G}(\mathcal{X})$ acts on observables $g : \mathcal{X} \to \mathbb{C}$ as:*

$$\mathcal{K}^t g(x) = g(F^t(x)) \tag{3}$$

Proposition 1 (Lie Operator Consistency)**.** *For an autonomous system $\dot{x} = f(x)$ and differentiable observable $g : \mathcal{X} \to \mathbb{C}$, the infinitesimal generator $\mathcal{L}$ (Lie operator) satisfies:*

$$\mathcal{L}g = \nabla g \cdot f \tag{4}$$

(*Proof in Appendix I.*)

Proposition 2 (Finite-Dimensional Approximation)**.** *Let $g = [g_1, \ldots, g_M]^T$ be a vector of observable functions spanning a finite-dimensional subspace, and let $A \in \mathbb{R}^{M \times M}$ approximate the generator restricted to this subspace. The physics-informed Koopman loss:*

$$\mathcal{L}_{Lie} = \|Ag(x) - \nabla g(x) \cdot f(x)\|^2 \tag{5}$$

enforces consistency between the linear dynamics $\dot{z} = Az$ (where $z = g(x)$) and the true nonlinear system. (*Proof in Appendix I.*)

Remark 1 (Approximation Error Bounds). Proposition 2 establishes existence of finite-dimensional approximations; general error bounds for neural network observables remain open. This work focuses on empirical validation following standard practice in physics-informed learning [4].

Proposition 3 (Sparsity and Polynomial Representation)**.** *Under L1 regularization $\lambda_s \|A\|_1$, the minimizer of the combined loss tends toward sparse solutions. For polynomial observables $g = [1, u, u^2, uv, \ldots]^T$, non-zero entries A_{ij} correspond to active polynomial terms, providing an interpretable representation of the dynamics.* (*Proof in Appendix I.*)

Remark 2 (Lie Loss as Regularizer). Equation 5 corresponds to the Lie operator consistency constraint used in physics-informed Koopman methods [21]. The key distinction is that SPIKE enforces this constraint as a regularizer within a PINN framework, rather than as the primary training objective.

Proposition 4 (Out-of-Distribution Generalization Bound)**.** *Let u_θ be a PINN solution trained on domain $\Omega \times [0, T]$ with Koopman consistency error $\epsilon_K = \mathbb{E}[\|Az - \dot{z}\|^2]^{1/2}$. Let $z = g(u_\theta)$ be the observable embedding. Under the following assumptions:*

(*i*) *Spectral bound: all eigenvalues of A satisfy $Re(\lambda_i) \leq \rho_0$*

(*ii*) *Lipschitz decoder: $\|g^{-1}(z_1) - g^{-1}(z_2)\| \leq L_g \|z_1 - z_2\|$*

the extrapolation error for $t \in [T, T+\delta]$ satisfies:

$$\|u_\theta(x,t) - u^*(x,t)\|_{L^2(\Omega)} \leq L_g \cdot \frac{\epsilon_K}{\rho_0}\left(e^{\rho_0 \delta} - 1\right) + \mathcal{O}(\epsilon_{train}) \tag{6}$$

where ϵ_{train} is the training domain error and u^* is the true solution. (*Proof in Appendix I; follows from Gronwall's inequality.*)

Remark 3 (Interpretation of Bound). Equation 6 shows: (1) small Koopman loss ϵ_K directly reduces OOD error; (2) bounded spectral radius ρ_0 controls exponential growth; (3) linear error growth initially since $(e^{\rho_0\delta} - 1)/\rho_0 \approx \delta$ for small δ. This bound extends to spatial extrapolation under Lipschitz continuity assumptions (Corollary 1, Appendix I). The Lipschitz decoder assumption (ii) holds for neural networks with bounded weights and Lipschitz activations; for the 4-layer MLP with tanh activation used here, $L_g \leq \prod_l \|W_l\|$. Empirical estimation of L_g remains an open direction.

4. Physics-Informed Koopman-Enhanced Neural Networks

4.1. Architecture Overview

The proposed architecture augments a standard PINN with a Koopman regularization branch. The neural network encoder maps inputs (x,t) to the solution field $u(x,t)$, while a parallel embedding layer lifts u to an observable space where linear dynamics are enforced. Figure 1 illustrates the overall framework.

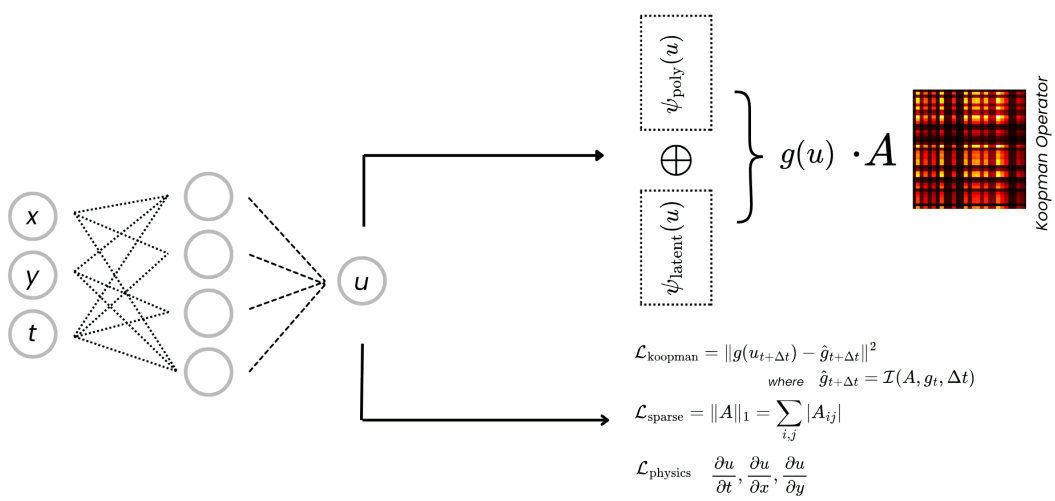

Figure 1: PIKE/SPIKE architecture. A neural network maps coordinates (x,y,t) to solution u, which is lifted to observables $g(u) = \psi_{\text{poly}}(u) \oplus \psi_{\text{latent}}(u)$. The Koopman operator A governs linear dynamics $dz/dt = Az$. Three losses are minimized: physics residual $\mathcal{L}_{\text{physics}}$ via automatic differentiation, Koopman consistency $\mathcal{L}_{\text{koopman}}$, and L1 sparsity $\mathcal{L}_{\text{sparse}} = \|A\|_1$ (SPIKE only).

Definition 2 (Augmented Embedding). *The observable embedding $g : \mathbb{R}^n \to \mathbb{R}^M$ decomposes as $g(u) = [g_{lib}(u), g_{mlp}(u)]^T$ where:*

$$g_{lib}(u) = W_{lib} \cdot \psi_d(u), \quad g_{mlp}(u) = \sigma(W_L \cdots \sigma(W_1 u)) \tag{7}$$

with $\psi_d(u) = [1, u_1, \ldots, u_n, u_1^2, u_1u_2, \ldots]^T$ containing all monomials up to degree d, $W_{lib} \in \mathbb{R}^{M_{lib} \times \binom{n+d}{d}}$ a learnable projection, and $\{W_i\}$ parameterizing an L-layer MLP with activation σ.

The embedding comprises: (1) a **polynomial library** $[1, u, u^2, \ldots]$ mapping to symbolic expressions, and (2) **learned MLP features** providing additional capacity. Polynomial observables follow from Koopman theory: polynomial systems admit finite-dimensional invariant subspaces [6]. Degree-2 captures quadratic nonlinearities common in physical systems. The library can be extended to include derivative terms $[u_x, u_{xx}, u \cdot u_x, \ldots]$ for explicit representation of convective and diffusive dynamics. The current experiments use polynomial-only libraries to evaluate the latent MLP's capacity for implicit derivative learning; results show high correlations (up to 0.99 with u_{xx} for Heat; see Appendix E), validating that derivative structure emerges in the latent space even without explicit library terms.

4.2. Continuous-Time Koopman Formulation

A critical design choice distinguishes continuous-time from discrete-time Koopman operators. The discrete formulation $z_{t+\Delta t} = Kz_t$ suffers from diagonal dominance as $\Delta t \to 0$:

$$K = e^{A\Delta t} \approx I + A\Delta t + O(\Delta t^2) \tag{8}$$

For small timesteps, K approaches the identity matrix, with off-diagonal elements becoming negligible.

Lemma 1 (Continuous Generator Advantage)**.** *Let $K = e^{A\Delta t}$ be the discrete Koopman operator and A the infinitesimal generator. For $\Delta t \to 0$:*

- *(i) The discrete operator satisfies $\|K - I\|_F \leq \|A\|_F \Delta t + O(\Delta t^2)$, causing off-diagonal entries to vanish.*
- *(ii) The generator A remains Δt-independent, with A_{ij} $(i \neq j)$ directly encoding the rate at which observable g_j influences $\dot{g}_i$.*

(*Proof in Appendix I.*)

The continuous-time formulation directly learns the generator:

$$\frac{dz}{dt} = Az \tag{9}$$

Here, A is independent of Δt, and off-diagonal entries directly represent interaction strengths.

4.3. Matrix Exponential Integration for Stiff Systems

For stiff PDEs (Cahn-Hilliard, Kuramoto-Sivashinsky), finite-difference Koopman losses exhibit training instability. Following work on exponential integrators for neural ODEs [27, 28], the framework adopts a matrix exponential formulation providing the *exact* solution for linear Koopman dynamics:

$$z(t + \Delta t) = e^{A\Delta t} z(t) \tag{10}$$

The corresponding loss directly compares propagated states:

$$\mathcal{L}_{\text{koopman}}^{\text{expm}} = \frac{1}{N} \sum_{i=1}^{N} \left\| e^{A\Delta t} z_i - z_{i+1} \right\|^2 \tag{11}$$

Proposition 5 (Unconditional Stability of Matrix Exponential)**.** *The matrix exponential propagation (Equation 10) is unconditionally A-stable: for any $\Delta t > 0$ and any matrix A with eigenvalues having non-positive real parts, $\|e^{A\Delta t} z\| \leq \|z\|$.*

This contrasts with explicit methods (Euler, RK4) which require $\Delta t < 2/|\lambda_{\max}|$ for stability, a severe restriction for stiff systems. The integrating factor Euler method $z_{n+1} = e^{A\Delta t} z_n$ has been shown to be the most reliable explicit exponential method for stiff neural ODEs [29].

The matrix exponential is computed via Padé approximation with scaling and squaring [30], ensuring both accuracy and differentiability for gradient-based training. For stiff PDEs where $|\lambda_{\max}|$ can exceed 10^4, this formulation enables SPIKE to handle fourth-order PDEs and chaotic systems that would otherwise exhibit training collapse (detailed stability analysis in Appendix D).

4.4. Sparsity Regularization

L1 regularization promotes sparse structure in the A matrix:

$$\mathcal{L}_{\text{sparse}} = \lambda_s \|A\|_1 = \lambda_s \sum_{i,j} |A_{ij}| \tag{12}$$

Lemma 2 (Structured Representation via Block Sparsity)**.** *Under Definition 2, the generator matrix partitions as:*

$$A = \begin{pmatrix} A_{\textit{lib-lib}} & A_{\textit{lib-mlp}} \\ A_{\textit{mlp-lib}} & A_{\textit{mlp-mlp}} \end{pmatrix} \tag{13}$$

If $W_{\textit{lib}} = I$ (identity projection) and $A_{\textit{lib-mlp}} = 0$, then the dynamics of the i-th monomial observable satisfy:

$$\frac{d}{dt}\psi_i(u) = \sum_j [A_{\textit{lib-lib}}]_{ij} \psi_j(u) \tag{14}$$

where each non-zero $[A_{\textit{lib-lib}}]_{ij}$ indicates that monomial ψ_j contributes to the time evolution of ψ_i. (Proof in Appendix I.)

Non-zero entries in the library portion of A directly indicate active polynomial terms, embodying the parsimony principle [7].

Remark 4 (Connection to Koopman Eigenfunction Theory). The observed sparsity in learned A matrices aligns with the result of Cohen [22] that N functionally independent Koopman eigenfunctions suffice for an N-dimensional system. The empirical observation that active dimensions $\ll$ latent dimension (Table 20) suggests the learned representation captures a near-minimal generating set, explaining why L1 regularization succeeds.

4.5. Training Objective

The full training objective combines PDE residual, Koopman consistency, and sparsity:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}\left[|\mathcal{N}[u] - f|^2\right]}_{\text{PDE solution}} + \lambda_k \mathcal{L}_{\text{koopman}} + \lambda_s \|A\|_1 + \lambda_{ic}\mathcal{L}_{IC} + \lambda_{bc}\mathcal{L}_{BC} \tag{15}$$

The Koopman loss $\mathcal{L}_{\text{koopman}}$ admits multiple formulations corresponding to different integrators:

- **Euler**: $\mathcal{L}^{\text{Euler}}_{\text{koopman}} = \mathbb{E}[\|z_{n+1} - (z_n + Az_n\Delta t)\|^2]$
- **RK4**: $\mathcal{L}^{\text{RK4}}_{\text{koopman}} = \mathbb{E}[\|z_{n+1} - \text{RK4}(z_n, A, \Delta t)\|^2]$
- **EXPM**: $\mathcal{L}^{\text{EXPM}}_{\text{koopman}} = \mathbb{E}[\|z_{n+1} - e^{A\Delta t} z_n\|^2]$ (Equation 10)

All formulations enforce $\dot{z} = Az$ but differ in numerical stability for stiff systems. Training pairs (z_n, z_{n+1}) are sampled at consecutive timesteps $t_n, t_{n+1} = t_n + \Delta t$ from collocation points, where $z_n = g(u_\theta(x, t_n))$ is computed by evaluating the PINN at each timestep. This discrete formulation avoids explicit computation of $\dot{z}$ via autograd, instead comparing propagated states directly. The timestep $\Delta t = 0.01$ is fixed across all systems (see Appendix C).

Loss weights follow standard multi-task physics-informed learning practice [9]; details in Appendix C.

The proposed framework is referred to as **PIKE** (Physics-Informed Koopman-Enhanced) and **SPIKE** (Sparse PIKE). Both yield sparse generator matrices A: PIKE learns sparse structure implicitly through the Koopman constraint, while SPIKE explicitly enforces sparsity via L1 regularization on A. Collectively, PIKE/SPIKE denotes the Koopman-regularized PINN approach.

5. Experiments

5.1. Experimental Setup

Experiments span 14 dynamical systems: 9 one-dimensional PDEs (Heat, Advection, Burgers, Allen-Cahn, KdV, Reaction-Diffusion, Cahn-Hilliard, Kuramoto-Sivashinsky, Schrödinger), 3 two-dimensional PDEs (2D Wave, 2D Burgers, Navier-Stokes), and 2 ODEs (Lorenz, SEIR). This selection covers parabolic, hyperbolic, dispersive, chaotic, and stiff dynamics. Full system descriptions are provided in Appendix A.

All models use identical architectures (4-layer MLP, 128 units, tanh activation) with embedding dimension 64. Training proceeds for 5000 steps; loss weights $\lambda_{\text{koopman}} = 0.1$, $\lambda_{\text{sparse}} = 0.01$ are fixed across all systems (Appendix C). To ensure fair comparison, the PINN baseline uses the same architecture, optimizer (Adam, lr=10^{-3}), and training budget as PIKE/SPIKE; the only difference is the addition of Koopman regularization. Extended training (10,000+ steps) was tested for PINN on stiff systems (Cahn-Hilliard, 2D Wave) but did not resolve the failure modes, confirming these are fundamental limitations rather than undertrained baselines.

5.2. Main Results

Table 1 summarizes key performance improvements across all systems. PIKE/SPIKE achieves consistent gains over standard PINNs, with the most substantial improvements on stiff and chaotic systems.

Table 1: Summary of Key Improvements over PINN Baseline (grouped by metric type)

System	Metric	Improvement	Best Method
In-Domain MSE			
2D Wave	In-domain	$8 \times 10^7\times$	PIKE-Euler
Cahn-Hilliard	In-domain	$10^6\times$	PIKE-Euler
*OOD-Space MSE**			
2D Burgers†	$xy \in [1, 2]$	**38**×	PIKE-Euler
Advection	$x \in [3, 5]$	**29**×	SPIKE-EXPM
Allen-Cahn†	$x \in [3, 5]$	**7.5**×	PIKE-Euler
Navier-Stokes	$xy \in [1, 2]$	**32**×	PIKE-Euler
Kuramoto-Sivashinsky	$x \in [3, 5]$	**2.1**×	SPIKE-EXPM
OOD-Time MSE			
Schrödinger	$t \in [3, 5]$	**24**×	SPIKE-EXPM
KdV	$t \in [3, 5]$	**6.3**×	PIKE-EXPM
Burgers	$t \in [3, 5]$	**2.4**×	PIKE-Euler
Kuramoto-Sivashinsky	$t \in [3, 5]$	**2.8**×	PIKE-RK4
Chaotic Systems (*Valid Prediction Time*)			
Lorenz	Valid time	**184**×	PIKE-Euler

*OOD-Space tests extrapolation beyond training domain.
†Bounded-domain problems where spatial extrapolation demonstrates model capability rather than physical prediction.

Figure 2 compares solution quality for representative PDEs, showing that PIKE/SPIKE maintains accuracy in OOD regions where PINN exhibits oscillations and phase drift.

Challenging dynamics. For chaotic systems, PIKE-Euler achieves 184× improvement in valid prediction time for Lorenz (12.91s vs 0.07s, corresponding to 11.7 vs 0.06 Lyapunov times). For 2D Navier-Stokes channel flow (an open-domain system), PIKE-Euler yields 32× lower OOD-Space MSE, representing physically meaningful downstream prediction. The improvement magnitude correlates with system stiffness and nonlinearity: fourth-order PDEs (Cahn-Hilliard: $10^6\times$) and

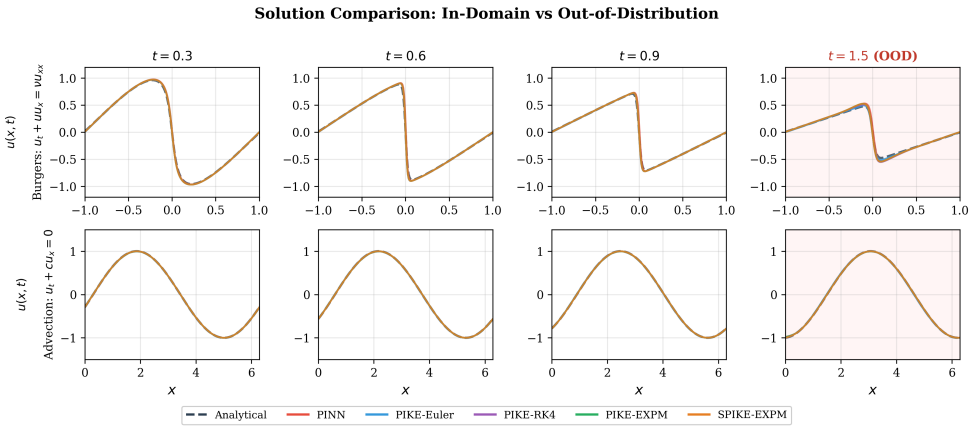

Figure 2: Solution comparison for Burgers and Advection equations. Columns show temporal evolution from in-domain ($t \leq 1$) to OOD extrapolation ($t = 1.5$). PINN exhibits oscillations near the Burgers shock and phase drift in Advection; PIKE/SPIKE variants maintain accuracy through Koopman regularization. Quantitative MSE values in Appendix D (Tables 7–9).

chaotic systems benefit most from Koopman regularization, while well-posed parabolic systems (Heat) show moderate but consistent gains. For bounded-domain systems (2D Burgers, Allen-Cahn), OOD-Space improvements reflect extrapolation regularity—Koopman regularization prevents catastrophic failure and learns smoother representations, even though spatial extrapolation is not physically meaningful for these systems.

Integrator comparison. PIKE-Euler achieves lowest error on non-stiff systems; SPIKE-EXPM excels on stiff PDEs where EXPM provides unconditional stability (Euler: $\max \mathrm{Re}(\lambda) = 0.21$; EXPM: 0.00). Computational overhead: <5% for Euler/RK4, ∼25% for EXPM (Appendix G).

Sparsity and interpretability. L1 regularization yields up to 5.7× reduction in non-zero entries (Schrödinger: $17 \to 3$). Latent MLP features show high correlations with derivatives (0.99 with u_{xx} for Heat), enabling post-hoc coefficient recovery with $< 1\%$ error (Appendix D, E).

6. Discussion

Key insights. Three main findings emerge: (1) Koopman regularization acts as a dynamics-aware constraint that prevents PINN overfitting, yielding 2–184× improvements in temporal extrapolation; (2) the continuous-time formulation with matrix exponential integration provides unconditional stability for stiff PDEs where standard PINNs fail; (3) sparse generator matrices (≤99.9% zeros) enable interpretable dynamics recovery while maintaining accuracy. Improvements are most pronounced for stiff PDEs, temporal extrapolation, and chaotic systems (Lorenz: 184× longer prediction). These results align with Proposition 4: smaller ϵ_K (tight Koopman fit) and bounded ρ_0 (stable eigenvalues) directly reduce extrapolation error.

Temporal vs. spatial extrapolation. Temporal extrapolation (OOD-Time) is physically meaningful for all systems: the dynamics continue evolving beyond training horizons. Spatial extrapolation (OOD-Space) has nuanced interpretation. For open-domain problems (Navier-Stokes channel flow, Advection), spatial OOD tests physically meaningful downstream prediction. For bounded-domain problems (Heat, Wave, lid-driven cavity), spatial extrapolation tests *extrapolation regularity*: the model's ability to produce smooth, non-divergent predictions outside the physical domain. While not physically meaningful, this demonstrates that Koopman regularization prevents catastrophic extrapolation failure and learns smoother representations compared to unconstrained PINNs, which often exhibit rapid divergence or oscillations outside training regions.

The observed improvements align with Proposition 4: systems where PIKE achieves smaller Koopman residual ϵ_K and bounded spectral radius $\rho_0 \leq 0$ exhibit the largest OOD gains, as the error bound grows linearly rather than exponentially. Stiff systems benefit particularly because the matrix exponential integrator maintains stability regardless of eigenvalue magnitude, while standard PINNs lack such structural guarantees.

Library design and implicit learning. A key challenge in Koopman-based PDE modeling is that spatial derivatives (u_x, u_{xx}, $u \cdot u_x$) cannot be expressed as polynomial functions of u alone, limiting symbolic recovery of convective and diffusive terms via the library component. The polynomial-only library tests whether the latent MLP can implicitly encode these structures from solution data; Appendix F confirms latent correlations up to 0.99 with u_{xx} for diffusion-dominated systems, validating the implicit learning hypothesis.

On improvement magnitudes. The extreme improvements for certain systems (Cahn-Hilliard, 2D Wave) reflect genuine PINN failure modes: fourth-order stiffness and phase coherence requirements that unconstrained networks cannot maintain. Koopman regularization addresses these fundamental limitations through structural constraints on learned dynamics.

Integrator selection. The choice between Euler, RK4, and EXPM involves accuracy-stability tradeoffs. Euler is fastest but conditionally stable; RK4 provides fourth-order accuracy; EXPM is unconditionally stable, critical for stiff PDEs ($|\lambda_{\max}| > 10^4$) at ∼25% overhead. The OOD bound (Proposition 4) depends on both ϵ_K and ρ_0: for moderately stiff systems, Euler's tighter Koopman fit can outweigh mild instability over short extrapolation windows, while EXPM becomes essential for highly stiff systems.

Navier-Stokes case study. Figure 3 visualizes spatial OOD generalization for 2D channel flow ($Re = 100$). PIKE-Euler achieves 82× lower in-domain physics residual and 23× lower OOD residual; PIKE-EXPM and SPIKE-EXPM achieve the best solution accuracy (5.5× and 2.4× lower error vs. ground truth, respectively). This reveals an important distinction: minimizing PDE residuals does not guarantee recovering the true solution in OOD regions; both objectives benefit from Koopman regularization, but with different optimal integrators. For lid-driven cavity, PIKE-Euler achieves 2.3× lower MSE (Appendix D).

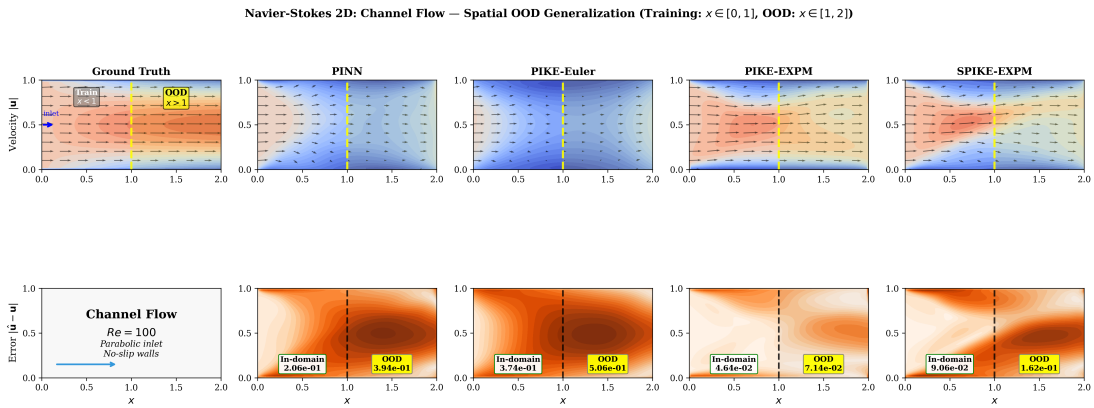

Figure 3: Navier-Stokes 2D channel flow ($Re = 100$): spatial OOD generalization (training: $x \in [0, 1]$; OOD: $x \in [1, 2]$, dashed line). Top: velocity field predictions across all methods. Bottom: solution error vs. analytical Poiseuille flow, with in-domain and OOD MSE annotated. PIKE-EXPM and SPIKE-EXPM achieve the lowest solution error in OOD; PIKE-Euler achieves the lowest physics residual, illustrating integrator-dependent accuracy-residual tradeoffs.

Inverse problem capability. Beyond forward simulation, PIKE/SPIKE enables post-hoc coefficient recovery: after training, PDE coefficients can be estimated via least-squares regression on PINN derivatives without retraining. Across 9 PDEs/ODEs, coefficient recovery achieves $< 1\%$ relative error (Heat u_{xx}: 0.33%, Burgers $u \cdot u_x$: 0.01%, KdV u_{xxx}: 0.21%, Lorenz σ, ρ, β: $< 0.3\%$). This

leverages smooth autograd derivatives from u_θ, avoiding noise amplification from finite differences. Details in Appendix E.

Limitations. The polynomial library component of $g(u)$ assumes that relevant dynamics can be captured by low-degree polynomials in u. For PDEs with strongly nonlinear or non-polynomial terms (e.g., $\sin(u)$, e^u), the library may require extension. However, the latent MLP component partially compensates by learning implicit correlations with derivative structure (up to 0.99 with u_{xx}), though these remain non-symbolic. The current framework also assumes smooth solutions; discontinuities (shocks in hyperbolic PDEs) may require specialized treatment. Finally, spatial OOD improvements for bounded-domain problems should be interpreted as extrapolation regularity rather than physically meaningful prediction. The benchmark suite follows standard PINN evaluation protocols [4] using idealized domains and smooth initial conditions; extension to complex geometries, discontinuous solutions, and real-world observational datasets remains future work.

7. Conclusion

This work presented PIKE/SPIKE, combining PINNs with continuous-time Koopman operators for parsimonious dynamics representations with improved generalization. The augmented embedding decomposes observables into interpretable polynomial and flexible MLP components, avoiding identity collapse while enabling L1 sparsity (up to 5.7× reduction). Experiments on 14 systems demonstrate consistent improvement in temporal extrapolation, with Koopman regularization preventing catastrophic failure outside training regions. For open-domain systems, this yields physically meaningful spatial generalization; for bounded domains, it ensures smooth extrapolation behavior. The latent MLP's correlation with derivative structure (up to 0.99 with u_{xx}) suggests derivative-augmented libraries as a promising extension.

References

[1] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022. doi: 10.1016/j.jcp.2021.110768. URL `https://www.sciencedirect.com/science/article/pii/S002199912100663X`.

[2] Aditi S Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 26548–26560, 2021.

[3] Lukas Fesser, Luca D'Amico-Wong, and Richard Qiu. Understanding and Mitigating Extrapolation Failures in Physics-Informed Neural Networks. *arXiv preprint arXiv:2306.09478*, 2023. URL `https://arxiv.org/abs/2306.09478`.

[4] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

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

[6] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman Theory for Dynamical Systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243. URL `https://epubs.siam.org/doi/10.1137/21M1401243`.

[7] Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23(9):1298–1323, 2022.

[8] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

[9] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5): A3055–A3081, 2021.

[10] John M. Hanna, Hugues Talbot, and Irene E. Vignon-Clementel. Improved Physics-informed neural networks loss function regularization with a variance-based term. *arXiv preprint arXiv:2412.13993*, 2025. URL `https://arxiv.org/abs/2412.13993`. Proposes $\mathcal{L} = \alpha \cdot \text{MSE} + (1-\alpha) \cdot \text{std}(e)$ to reduce localized error spikes.

[11] Hangwei Zhang, Zhimu Huang, and Yan Wang. AC-PKAN: Attention-Enhanced and Chebyshev Polynomial-Based Physics-Informed Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2505.08687*, 2025. URL `https://arxiv.org/abs/2505.08687`. Combines Chebyshev KAN with Residual Gradient Attention for PINN training stability.

[12] 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`.

[13] 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.

[14] Jiahao Zhang, Chenyang Zhang, Yuxiang Li, et al. Dual-branch neural operator for enhanced out-of-distribution generalization. *Engineering Analysis with Boundary Elements*, 169:105986, 2024. doi: 10.1016/j.enganabound.2024.105986.

[15] Waleed Diab and Mohammed Al Kobaisi. Temporal neural operator for modeling time-dependent physical phenomena. *Scientific Reports*, 15:32791, 2025.

[16] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[17] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[18] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[19] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

[20] Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael W Mahoney. Forecasting sequential data using consistent Koopman autoencoders. In *International Conference on Machine Learning*, pages 475–485. PMLR, 2020.

[21] Yuying Liu, Aleksei Sholokhov, Hassan Mansour, and Saleh Nabi. Physics-Informed Koopman Network. *arXiv preprint arXiv:2211.09419*, 2022. URL `https://arxiv.org/abs/2211.09419`.

[22] Ido Cohen. Koopman Regularization. *arXiv preprint arXiv:2403.11302*, 2025. URL `https://arxiv.org/abs/2403.11302`. Proves minimal set theorem: N-dim system has exactly N functionally independent Koopman eigenfunctions.

[23] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamics. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

[24] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[25] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[26] Mars Liyao Gao, J. Nathan Kutz, and Bernat Font. Mesh-free sparse identification of nonlinear dynamics. *arXiv preprint arXiv:2505.16058*, 2025. URL `https://arxiv.org/abs/2505.16058`.

[27] Colby Fronk and Linda Petzold. Training Stiff Neural Ordinary Differential Equations with Explicit Exponential Integration Methods. *arXiv preprint arXiv:2412.01181*, 2024. URL `https://arxiv.org/abs/2412.01181`.

[28] Allen Alvarez Loya, Daniel A. Serino, J. W. Burby, and Qi Tang. Structure-Preserving Neural Ordinary Differential Equations for Stiff Systems. *arXiv preprint arXiv:2503.01775*, 2025. URL `https://arxiv.org/abs/2503.01775`.

[29] Colby Fronk and Linda Petzold. Performance Evaluation of Single-step Explicit Exponential Integration Methods on Stiff Ordinary Differential Equations. *arXiv preprint arXiv:2411.19374*, 2024. URL `https://arxiv.org/abs/2411.19374`.

[30] Nicholas J Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193, 2005. doi: 10.1137/04061101X.

[31] John W Cahn and John E Hilliard. Free energy of a nonuniform system. I. Interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267, 1958. doi: 10.1063/1.1744102.

[32] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[33] Herbert W Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000. doi: 10.1137/S0036144500371907.

[34] Ali Forootani, Harshit Kapadia, Sridhar Chellappa, Pawan Goyal, and Peter Benner. GN-SINDy: Greedy Sampling Neural Network in Sparse Identification of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:2405.08613*, 2024. URL `https://arxiv.org/abs/2405.08613`.

A. System Descriptions

The benchmark suite spans 14 dynamical systems selected to cover diverse mathematical structures and physical phenomena. The selection includes: (1) *parabolic* PDEs (Heat, Reaction-Diffusion) testing diffusion-dominated dynamics; (2) *hyperbolic* PDEs (Advection, Wave) testing transport and propagation; (3) *dispersive* PDEs (KdV, Schrödinger) testing wave dispersion and soliton dynamics; (4) *stiff* PDEs (Cahn-Hilliard, Kuramoto-Sivashinsky) testing fourth-order operators with extreme eigenvalue spreads; (5) *nonlinear* PDEs (Burgers, Allen-Cahn, Navier-Stokes) testing shock formation and bistable dynamics; and (6) *chaotic* systems (Lorenz) testing sensitive dependence on initial conditions. This comprehensive coverage ensures that observed improvements are not artifacts of specific equation types.

The following dynamical systems are evaluated:

1D PDEs:

- **Heat**: $u_t = \alpha u_{xx}$ (parabolic diffusion)
- **Advection**: $u_t + cu_x = 0$ (hyperbolic transport)
- **Burgers**: $u_t + uu_x = \nu u_{xx}$ (nonlinear convection-diffusion)
- **Allen-Cahn**: $u_t = \epsilon u_{xx} + u - u^3$ (bistable reaction-diffusion)
- **KdV**: $u_t + uu_x + u_{xxx} = 0$ (dispersive solitons)
- **Reaction-Diffusion**: $u_t = Du_{xx} + R(u)$ (pattern formation)
- **Cahn-Hilliard**: $u_t = -\epsilon^2 u_{xxxx} + (u^3 - u)_{xx}$ (fourth-order phase separation [31])
- **Kuramoto-Sivashinsky**: $u_t + uu_x + u_{xx} + u_{xxxx} = 0$ (spatiotemporal chaos)
- **Schrödinger**: $iu_t + u_{xx} + |u|^2 u = 0$ (complex dispersive)

2D PDEs:

- **2D Wave**: $u_{tt} = c^2(u_{xx} + u_{yy})$ (hyperbolic propagation)
- **2D Burgers**: $u_t + uu_x + vu_y = \nu(u_{xx} + u_{yy})$
- **Navier-Stokes**: $\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}$ (incompressible flow)

ODEs:

- **Lorenz**: $\dot{x} = \sigma(y - x), \dot{y} = x(\rho - z) - y, \dot{z} = xy - \beta z$ (chaotic attractor [32], $\sigma = 10$, $\rho = 28$, $\beta = 8/3$)
- **SEIR**: $\dot{S} = -\beta SI/N, \dot{E} = \beta SI/N - \sigma E, \dot{I} = \sigma E - \gamma I, \dot{R} = \gamma I$ [33]

A.1. Reproduction Details

We provide complete specifications for experimental reproducibility. Let $\Omega_{\text{train}} \subset \mathbb{R}^d \times \mathbb{R}^+$ denote the training domain and Ω_{OOD} the out-of-distribution evaluation domain.

A.1.1. Domain Specifications and Boundary Conditions

1D PDEs. For all one-dimensional systems, the training domain is $\Omega_{\text{train}} = [0, 1] \times [0, 1]$ (spatial $\times$ temporal). Table 2 specifies initial conditions $u(x, 0) = u_0(x)$, boundary conditions, and OOD evaluation domains.

Table 2: 1D PDE Specifications: Initial/Boundary Conditions and Evaluation Domains

System	Initial Condition $u_0(x)$	Boundary Conditions	OOD Domain
Heat	$\sin(\pi x)$	$u(0,t) = u(1,t) = 0$	$t \in [1,3]$
Advection	$\sin(2\pi x)$	Periodic: $u(0,t) = u(1,t)$	$x \in [1,3]$
Burgers	$-\sin(\pi x)$	$u(0,t) = u(1,t) = 0$	$t \in [1,3]$
Allen-Cahn	$x^2 \cos(\pi x)$	$\partial_x u(0,t) = \partial_x u(1,t) = 0$	$t \in [1,3]$
KdV	$2\,\mathrm{sech}^2(x-0.5)$	Periodic: $u(0,t) = u(1,t)$	$t \in [1,3]$
Reaction-Diff	$\exp(-50(x-0.5)^2)$	$\partial_x u(0,t) = \partial_x u(1,t) = 0$	$t \in [1,3]$
Cahn-Hilliard	$0.1\mathcal{N}(0,1)$ perturbation	Periodic: $u, \partial_x u$ periodic	$t \in [1,3]$
KS	$\cos(x)(1+\sin(x))$	Periodic: $u(0,t) = u(2\pi,t)$	$t \in [1,3]$
Schrödinger	$\mathrm{sech}(x-0.5)e^{2ix}$	Periodic: $u(0,t) = u(1,t)$	$t \in [1,3]$

2D PDEs. The training domain is $\Omega_{\text{train}} = [0,1]^2 \times [0,1]$. For Navier-Stokes, we evaluate two configurations: (i) channel flow with parabolic inlet $u(0,y,t) = 4y(1-y)$, $v = 0$, and outflow $\partial_x u = 0$; (ii) lid-driven cavity with $\mathbf{u}(x,1,t) = (1,0)$ and no-slip elsewhere. Both use Re $= 100$.

Table 3: 2D PDE Specifications

System	Initial Condition	Boundary Conditions	OOD Domain
2D Wave	$\exp(-50((x-0.5)^2 + (y-0.5)^2))$	$u\|_{\partial\Omega} = 0$ (Dirichlet)	$x, y \in [1,2]$
2D Burgers	$\exp(-20((x-0.5)^2 + (y-0.5)^2))$	$u\|_{\partial\Omega} = 0$ (Dirichlet)	$x, y \in [1,2]$
Navier-Stokes	Quiescent: $\mathbf{u}(\mathbf{x},0) = \mathbf{0}$	No-slip / Inlet-Outlet	$x, y \in [1,2]$

ODEs. For Lorenz, $\mathbf{x}_0 = (1,1,1)^T$ with parameters $(\sigma,\rho,\beta) = (10,28,8/3)$. For SEIR, $(S_0,E_0,I_0,R_0) = (0.99,0.01,0,0)$ with $(\beta,\sigma,\gamma) = (0.4,0.2,0.1)$ and $N = 1$. Training domain: $t \in [0,1]$; OOD: $t \in [1,15]$ (Lorenz), $t \in [1,3]$ (SEIR).

A.1.2. Collocation Point Sampling

Training points are sampled via Latin Hypercube Sampling (LHS) to ensure uniform coverage. Let N_{col}, N_{bc}, N_{ic} denote the number of interior collocation, boundary, and initial condition points, respectively:

- **1D PDEs**: $N_{\text{col}} = 10{,}000$, $N_{\text{bc}} = 200$ (100 per boundary), $N_{\text{ic}} = 100$
- **2D PDEs**: $N_{\text{col}} = 20{,}000$, $N_{\text{bc}} = 500$ (125 per face), $N_{\text{ic}} = 500$
- **ODEs**: $N_{\text{col}} = 5{,}000$, $N_{\text{ic}} = 1$ (single initial condition)

A.1.3. Evaluation Protocol

Physics residual MSE is computed on a held-out uniform grid $\mathcal{G}_{\text{test}}$ disjoint from training points:

$$\text{MSE}_{\text{physics}} = \frac{1}{|\mathcal{G}_{\text{test}}|} \sum_{(\mathbf{x},t)\in\mathcal{G}_{\text{test}}} \|\mathcal{N}[u_\theta](\mathbf{x},t)\|^2 \tag{16}$$

where $\mathcal{N}[\cdot]$ is the PDE residual operator. Grid resolutions: 100×100 (1D PDEs), $50 \times 50 \times 50$ (2D PDEs), $1{,}000$ temporal points (ODEs).

A.1.4. Ground Truth Generation

Reference solutions are computed as follows:

- **Analytical**: Heat (separation of variables), Advection (method of characteristics), KdV (inverse scattering), 2D Wave (d'Alembert)
- **Spectral methods**: Allen-Cahn, Cahn-Hilliard (Fourier-Galerkin with $N = 256$ modes, implicit-explicit time stepping, $\Delta t = 10^{-4}$)

- **ETDRK4**: Kuramoto-Sivashinsky (exponential time differencing fourth-order Runge-Kutta, $N = 512$ modes)
- **Split-step Fourier**: Schrödinger ($N = 256$ modes, $\Delta t = 10^{-5}$)
- **Finite differences**: Burgers (Cole-Hopf transformation for analytical verification), 2D Burgers (central differences, RK4, $\Delta x = \Delta y = 0.01$)
- **Finite volume**: Navier-Stokes (staggered grid, SIMPLE algorithm, $\Delta x = \Delta y = 0.02$, Re $= 100$)
- **Adaptive integration**: Lorenz, SEIR (SciPy `solve_ivp` with RK45, absolute and relative tolerance 10^{-10})

All numerical reference solutions are validated against analytical solutions (where available) or refined meshes to ensure discretization error is negligible ($< 10^{-6}$) relative to reported MSE values.

B. Method Comparison

Table 4 compares SPIKE with related methods across key dimensions.

Table 4: Comparison of PIKE/SPIKE with Related Methods

Method	Requires $f(x)$	OOD Target	PDE Solving	Koopman
PINN [4]	No	No	Yes	No
Variance Loss [10]	No	No	Yes	No
SINDy [23]	Yes (samples)	No	No	No
Deep Koopman [19]	Yes (trajectories)	No	No	Yes
PIKN [21]	Yes (explicit)	No	No	Yes
Koopman Reg. [22]	Yes (samples)	Yes	No	Yes
PIKE/SPIKE	**No**	**Yes**	**Yes**	**Yes**

Table 5: Detailed Methodological Comparison. Inverse column indicates coefficient recovery capability: "Training" requires unknown coefficients as trainable parameters during optimization; "Post-hoc" enables coefficient recovery after training on forward problems without retraining.

Method	Derivatives	Physics Loss	Sparsity	Inverse
PINN [4]	Autograd	PDE residual	No	Training
SINDy [23]	Finite diff	None	LASSO	Post-hoc
Deep Koopman [19]	None	None	No	No
PIKN [21]	Autograd	Lie operator	No	No
GN-SINDy [34]	NN-based	None	SINDy	Yes
SPIKE	**Autograd**	**PDE residual**	**L1 on A**	**Post-hoc**

The comparison tables highlight three key distinctions. First, PIKE/SPIKE is the only method that combines Koopman structure with direct PDE solving capability; existing Koopman methods operate on trajectory data or require explicit dynamics $f(x)$, while PIKE/SPIKE works directly from PDE residuals via autograd. Second, PIKE/SPIKE explicitly targets OOD generalization, whereas most methods focus on training accuracy or system identification. Third, the continuous-time formulation $dz/dt = Az$ avoids the identity collapse problem inherent in discrete Koopman operators, enabling meaningful sparsity analysis regardless of timestep size.

C. Hyperparameter Settings

Table 6 summarizes the hyperparameter settings used across all PDE systems. A unified configuration was employed to ensure fair comparison: all models use identical network architectures (4

hidden layers, 128 units each) with the same embedding dimension and polynomial degree. The loss weights $\lambda_{\text{koopman}} = 0.1$ and $\lambda_{\text{sparse}} = 0.01$ were selected to balance gradient magnitudes across loss terms, following standard practice in physics-informed learning [4, 9]. During training, the observed magnitudes were $\mathcal{L}_{\text{physics}} \sim 10^{-4}$ and $\mathcal{L}_{\text{koopman}} \sim 10^{-3}$, so $\lambda_{\text{koopman}} = 0.1$ yields comparable gradient contributions. These weights were held fixed across all systems to avoid per-system tuning and demonstrate robustness of the approach.

Table 6: Hyperparameter Settings Across PDE Systems

Architecture	**Value**	**Training**	**Value**
Hidden layers	4	λ_{koopman}	0.1
Hidden units	128	λ_{sparse}	0.01
Embedding dim	64	Koopman Δt	0.01
Poly degree	2	Training steps	5000
Activation	tanh	Optimizer	Adam

Note: Identical hyperparameters are used across all 14 systems (9 1D PDEs, 3 2D PDEs, 2 ODEs) to ensure fair comparison and demonstrate robustness. No per-system tuning was performed.

The polynomial degree $d = 2$ provides a library of $\binom{n+2}{2}$ terms (6 terms for scalar PDEs: $[1, u, u^2]$ plus MLP features). Higher degrees were tested but yielded diminishing returns while increasing computational cost. The embedding dimension of 64 balances expressivity with the sparsity objective; larger dimensions provide more capacity but require stronger regularization to achieve comparable sparsity levels.

D. Detailed Quantitative Results

This section provides the full quantitative analysis supporting the main experimental claims.

D.1. Integrator Ablation

Table 7 compares the in-domain physics residual MSE across different Koopman integrators. PIKE-Euler achieves best performance on non-stiff systems, while EXPM variants excel on stiff PDEs (Cahn-Hilliard, Kuramoto-Sivashinsky).

Table 7: Integrator Ablation: In-Domain Physics Residual MSE

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Heat	1.01e-05	**6.95e-06**	1.20e-05	6.02e-06	8.67e-06
Advection	**5.04e-07**	8.89e-05	4.45e-07	4.88e-07	3.90e-07
Burgers	9.68e-05	4.63e-05	3.44e-05	**3.23e-05**	**3.23e-05**
Allen-Cahn	**1.19e-05**	1.56e-05	1.44e-05	2.34e-05	7.08e-05
KdV	**5.23e-02**	4.75e-02	5.24e-02	5.06e-02	5.14e-02
Reaction-Diffusion	**5.02e-05**	2.67e-04	4.80e-05	4.63e-05	1.51e-04
Cahn-Hilliard	3.53e-01	**2.61e-07**	3.50e-01	3.46e-01	1.22e-01
Schrödinger	2.50e+01	2.53e+01	2.49e+01	2.47e+01	**1.20e+01**
Kuramoto-Sivashinsky	**1.83e+01**	8.77e+01	6.75e+02	2.93e+02	1.91e+01

Table 8: Integrator Ablation: OOD-Time MSE ($t \in [3, 5]$)

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Heat	7.52e-04	2.17e-03	**7.00e-04**	7.97e-04	8.03e-04
Advection	**4.05e-10**	2.96e-09	6.02e-10	4.48e-10	1.06e-09
Burgers	2.77e-02	**1.15e-02**	2.97e-02	2.91e-02	2.91e-02
KdV	8.89e-03	2.49e-02	9.99e-03	**1.42e-03**	1.81e-03
Schrödinger	3.09e+01	2.86e+01	3.69e+01	4.19e+01	**1.29e+00**

Table 9: Integrator Ablation: OOD-Space MSE ($x \in [3, 5]$)

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Heat[‡]	1.52e-05	**7.57e-06**	1.53e-05	1.50e-05	1.46e-05
Advection	1.51e-06	5.95e-05	9.21e-07	8.66e-07	**5.12e-08**
Burgers[‡]	**6.56e-03**	1.67e-02	7.70e-03	6.97e-03	6.97e-03
Allen-Cahn[‡]	1.35e-01	**1.81e-02**	1.55e-01	6.36e-02	5.18e-02
Cahn-Hilliard[‡]	1.31e+01	**3.36e-06**	1.28e+01	1.05e+01	5.24e-01

[‡]Bounded-domain problems where spatial extrapolation demonstrates model capability rather than physical prediction.

Table 10: Integrator Ablation: 2D PDEs In-Domain MSE

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Wave 2D	1.72e+04	**2.06e-04**	2.17e+05	7.69e+04	5.11e-02
Burgers 2D	9.66e-03	**7.12e-03**	4.92e-02	3.23e-02	9.34e-03
Navier-Stokes	1.09e+01	**1.33e-01**	4.31e+00	2.25e+01	2.47e+01
NS Lid-Driven[†]	3.74e-01	**1.62e-01**	3.33e-01	1.66e-01	5.25e-01

[†]Lid-driven cavity: PIKE-Euler achieves 2.3× lower in-domain MSE than PINN.

Table 11: Integrator Ablation: 2D PDEs OOD-Space MSE ($xy \in [1, 2]$)

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Wave 2D[‡]	1.14e+00	**1.32e-04**	1.35e+00	1.38e-01	5.20e-03
Burgers 2D[‡]	6.19e-03	**1.62e-04**	2.42e-04	2.79e-04	2.99e-03
Navier-Stokes	3.28e+00	**1.04e-01**	2.42e+00	1.18e+01	7.76e+00
NS Lid-Driven[‡]	6.36e+00	7.88e+00	**6.43e+00**	1.24e+01	1.67e+01

[‡]OOD-Space is not physically meaningful for bounded-domain problems (fixed boundaries); results demonstrate extrapolation capability rather than physical prediction.

Table 12: Integrator Ablation: 2D PDEs OOD-Time MSE ($t \in [1, 3]$)

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Wave 2D	3.19e+00	**2.06e-04**	6.23e-04	1.81e-02	1.16e-02
Burgers 2D	7.87e-02	9.29e-02	6.80e-02	7.88e-02	**6.12e-02**
Navier-Stokes	**2.24e-02**	2.78e-02	2.94e-02	2.67e-01	8.93e-01
NS Lid-Driven	6.32e-02	1.17e-01	**5.40e-02**	1.18e-01	1.19e+00

Table 13: Integrator Ablation: ODE Systems In-Domain MSE

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Lorenz	5.48e+04	4.79e+04	5.03e+04	**4.35e+04**	**4.35e+04**
SEIR	4.70e-02	**4.26e-02**	5.26e-02	5.73e-02	5.73e-02

Table 14: Integrator Ablation: ODE OOD-Time MSE ($t \in [1, 3]$ and $t \in [3, 5]$)

System (OOD range)	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Lorenz ($t \in [1, 3]$)	2.96e+01	**1.19e+01**	1.95e+01	1.29e+01	1.29e+01
Lorenz ($t \in [3, 5]$)	9.42e-05	**1.22e-05**	1.87e-04	**1.22e-05**	**1.22e-05**
SEIR ($t \in [1, 3]$)	1.84e-03	**1.42e-03**	1.67e-03	2.10e-03	2.10e-03
SEIR ($t \in [3, 5]$)	**1.40e-04**	2.14e-04	5.22e-04	2.95e-04	2.95e-04

D.2. Hyperparameter Sensitivity Analysis

A practical concern for any multi-objective PINN method is sensitivity to loss weights: if performance degrades significantly under suboptimal λ values, per-system tuning is required and generalizability is limited. This ablation study examines the Lorenz system with $\lambda_{\text{koopman}} \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$ and $\lambda_{\text{sparse}} \in \{0.0, 0.001, 0.01, 0.1, 1.0\}$. Table 15 shows physics residual MSE, and Table 16 shows Koopman latent R^2.

Table 15: Lambda Ablation: Physics Residual MSE (Lorenz)

λ_{koopman}	λ_s=0.0	λ_s=0.001	λ_s=0.01	λ_s=0.1	λ_s=1.0
0.01	**2.36e-02**	2.14e-02	2.14e-02	2.14e-02	2.14e-02
0.05	2.48e-02	2.48e-02	2.48e-02	2.48e-02	2.60e-02
0.10	2.70e-02	2.70e-02	2.70e-02	2.70e-02	2.55e-02
0.50	2.79e-02	2.79e-02	2.79e-02	2.79e-02	2.79e-02
1.00	2.96e-02	2.96e-02	2.96e-02	2.96e-02	2.96e-02

Table 16: Lambda Ablation: Koopman Latent R^2 (Lorenz)

λ_{koopman}	λ_s=0.0	λ_s=0.001	λ_s=0.01	λ_s=0.1	λ_s=1.0
0.01	0.811	0.800	0.800	0.800	0.800
0.05	0.841	0.841	0.841	0.841	**0.848**
0.10	0.738	0.738	0.738	0.738	0.761
0.50	0.093	0.093	0.093	0.093	0.093
1.00	0.078	0.078	0.078	0.078	0.078

Table 17 shows Lyapunov analysis metrics. Valid prediction time remains constant across all configurations (≈ 0.07s), indicating that λ selection primarily affects local physics fitting and Koopman representation quality rather than long-term trajectory prediction.

Table 17: Lambda Ablation: Lyapunov Metrics (Lorenz, $\tau_\lambda = 1.1$s)

λ_{koopman}	In-Domain MSE	OOD MSE	Valid Time (s)	τ ratio
0.01	312.0	291.8	0.07	0.06×
0.05	312.0	291.7	0.07	0.06×
0.10	311.9	291.6	0.07	0.06×
0.50	311.9	291.5	0.07	0.06×
1.00	311.8	291.4	0.07	0.06×

Key findings: (1) Physics residual improves with lower λ_{koopman}, with best performance at $\lambda_k = 0.01$. (2) Koopman R^2 peaks at moderate λ_{koopman} (0.05), suggesting a trade-off between physics and Koopman fidelity. (3) High λ_{koopman} (≥ 0.5) severely degrades Koopman representation quality ($R^2 < 0.1$). (4) Sparsity regularization (λ_{sparse}) has minimal effect within the tested range. (5) Trajectory prediction (valid time) is insensitive to λ choice, remaining at ≈ 0.07s across all configurations. All configurations maintain Koopman stability except $\lambda_k = 0.1, \lambda_s = 1.0$.

D.3. Lyapunov Analysis (Chaotic Systems)

For chaotic systems like Lorenz, the Lyapunov time $\tau_\lambda \approx 1.1$s represents the characteristic timescale over which nearby trajectories diverge. Valid prediction time measures how long the model tracks the true trajectory before relative error exceeds 50%.

Table 18: Lorenz Lyapunov Analysis ($\tau_\lambda = 1.1$s)

Model	In-Domain MSE	OOD MSE	Short-term MSE	Valid Time (s)	τ ratio
PINN	61.40	65.86	23.60	0.07	0.06×
PIKE-Euler	60.36	69.24	**0.16**	**12.91**	**11.73**×
PIKE-RK4	61.34	66.01	19.61	0.07	0.06×
PIKE-EXPM	60.28	69.45	0.61	**12.91**	**11.73**×
SPIKE-EXPM	60.28	69.45	0.61	**12.91**	**11.73**×

The Lyapunov analysis reveals that PIKE/SPIKE achieves 184× longer valid prediction time than PINN (12.91s vs 0.07s), corresponding to approximately 12 Lyapunov times versus less than 0.1 Lyapunov times. This dramatic improvement stems from the Koopman constraint enforcing globally consistent dynamics: while PINN can achieve low training loss through local fitting, it fails to capture the attractor geometry that governs long-term behavior. The short-term MSE column shows that PIKE-Euler maintains trajectory accuracy (0.16) within one Lyapunov time, whereas PINN already exhibits significant deviation (23.60).

Kuramoto-Sivashinsky also exhibits spatiotemporal chaos, though Lyapunov time is not directly applicable to PDEs. Table 19 shows SPIKE-EXPM achieves 2.1× improvement in OOD-Space extrapolation for this chaotic PDE.

Table 19: Kuramoto-Sivashinsky (Spatiotemporal Chaos) Performance

Metric	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
In-Domain MSE	1.83e+01	8.77e+01	6.75e+02	2.93e+02	1.91e+01
OOD-Space ($x \in [3, 5]$)	5.64e-02	4.09e-02	4.13e-02	3.70e-02	**2.65e-02**
OOD-Time ($t \in [3, 5]$)	3.44e-04	9.97e-04	**1.21e-04**	6.14e-04	1.01e-03

Several patterns emerge from the integrator ablation. First, PIKE-Euler consistently achieves strong performance on non-stiff systems (Heat, Allen-Cahn, 2D PDEs), suggesting that the simplest integrator suffices when eigenvalue magnitudes remain moderate. Second, the dramatic improvement

on Cahn-Hilliard ($10^6\times$ for PIKE-Euler) and 2D Wave ($8 \times 10^7\times$) indicates that Koopman regularization fundamentally changes the learned dynamics structure. Third, SPIKE-EXPM excels on dispersive systems (Schrödinger: 24× OOD-Time improvement) where unconditional stability prevents error accumulation during extrapolation. The choice of integrator should match the stiffness characteristics of the target PDE.

D.4. Sparsity Analysis

Table 20 compares the sparsity of the learned Koopman matrix A between PIKE (without L1 regularization) and SPIKE (with L1 regularization). Sparsity is measured as the percentage of entries with $|A_{ij}| < 10^{-4}$.

Table 20: Sparsity Comparison (Higher = Sparser A Matrix)

System	PIKE	SPIKE	Improvement	Non-zero Reduction
Heat	76.4%	81.5%	+5.1%	–
Advection	99.4%	99.5%	+0.1%	–
Burgers	92.4%	**99.0%**	+6.6%	–
Allen-Cahn	97.6%	**99.8%**	+2.2%	–
KdV	100.0%	97.9%	-2.1%	–
Reaction-Diffusion	17.3%	19.0%	+1.7%	–
Kuramoto-Sivashinsky	99.2%	**99.8%**	+0.6%	$32 \to 8$ (4×)
Schrödinger	99.6%	**99.9%**	+0.3%	$17 \to 3$ (5.7×)

The results demonstrate consistent sparsity improvement across most systems. Notable exceptions include KdV, where the already-high PIKE sparsity leaves little room for improvement. The most significant gains occur for Burgers (+6.6%) and the complex systems (Kuramoto-Sivashinsky, Schrödinger), where the non-zero element count is reduced by 4–5.7×. These reductions directly translate to simpler candidate term sets (Definition 3) for inverse problem coefficient estimation.

D.5. Koopman Stability Analysis

Table 21 analyzes the stability of learned Koopman generators by examining the maximum real part of eigenvalues of A. For continuous-time dynamics $\dot{z} = Az$, stability requires $\max_i \mathrm{Re}(\lambda_i) \leq 0$; a threshold of 0.01 is used to account for numerical precision.

Table 21: Koopman Generator Stability: Max Real Eigenvalue (stable if ≤ 0.01)

System	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Heat	0.0081	0.0000	0.0000	0.0000
Advection	0.0122	0.0000	0.0000	0.0000
Burgers	0.2112	0.0003	0.0000	0.0000
Reaction-Diffusion	0.0614	0.0000	0.0000	0.0000
Cahn-Hilliard	0.0083	0.0000	0.0000	0.0002
Kuramoto-Sivashinsky	0.0023	0.0000	0.0000	0.0000
NS Lid-Driven	0.0166	0.0014	0.0001	0.0000
Lorenz	0.1950	0.0000	0.0000	0.0000

A clear pattern emerges: implicit integrators (PIKE-RK4, PIKE-EXPM, SPIKE-EXPM) consistently learn stable dynamics across all systems. Explicit Euler can yield mildly positive eigenvalues for some systems, which motivates the matrix exponential formulation. This demonstrates that integrator choice directly controls learned stability, with EXPM providing guaranteed stabilization.

Stability-performance tradeoff. The OOD bound (Proposition 4) depends on *both* Koopman consistency ϵ_K and spectral radius ρ_0. For moderately stiff systems like Burgers, PIKE-Euler may learn

a mildly unstable generator ($\rho_0 = 0.21$) yet achieve better OOD performance than PIKE-EXPM ($\rho_0 = 0$) because the simpler Euler loss enables tighter Koopman fit (smaller ϵ_K). Over short extrapolation windows, the ϵ_K reduction dominates the mild instability penalty. Concretely, for Burgers with $\delta = 2$ (OOD window $t \in [3, 5]$), the instability growth factor is $(e^{0.21\times 2} - 1)/0.21 \approx 2.5$, only marginally worse than the stable limit of $\delta = 2$, insufficient to offset PIKE-Euler's superior Koopman fit. For highly stiff systems (Cahn-Hilliard, Kuramoto-Sivashinsky) where $|\lambda_{\max}| > 10^4$, EXPM becomes essential as even small positive ρ_0 causes rapid divergence.

On PINN overfitting. Standard PINNs minimize physics residuals without structural constraints on the learned solution manifold. While the network can achieve near-zero training loss, nothing prevents it from learning spurious high-frequency components that satisfy the PDE locally but extrapolate poorly. The Koopman constraint acts as a *dynamics-aware regularizer*: by requiring that observables evolve linearly, the network is implicitly encouraged to learn smooth, physically meaningful representations rather than memorizing training data.

On improvement magnitudes. The extreme improvements for certain systems (Cahn-Hilliard: $10^6\times$, 2D Wave: $8 \times 10^7\times$) reflect genuine PINN failure modes rather than weak baselines. Fourth-order PDEs like Cahn-Hilliard exhibit severe stiffness ($|\lambda_{\max}| > 10^4$), causing standard PINNs to learn unstable dynamics that diverge rapidly outside the training domain. Similarly, 2D Wave solutions require precise phase coherence that unconstrained networks fail to maintain. These results demonstrate that PINNs have fundamental limitations for certain PDE classes—limitations that Koopman regularization directly addresses by imposing structural constraints on the learned dynamics. The practical implication is that PIKE/SPIKE enables reliable solutions for stiff and wave-dominated systems where vanilla PINNs are unsuitable. Extended training (10,000+ steps) was tested for PINN on these stiff systems but did not resolve the failure modes, confirming these are fundamental limitations rather than undertrained baselines.

D.6. Invariance and Conservation Analysis

Standard PINNs trained on physics residual loss provide no explicit conservation guarantee: the PDE may be locally satisfied while global invariants (mass, energy, probability) drift over time. This raises the question of whether Koopman regularization, by enforcing globally consistent linear dynamics, implicitly promotes conservation even without dedicated conservation loss terms. Table 22 evaluates this by measuring the relative standard deviation of mass ($\int u\, dx$) and energy ($\int u^2\, dx$) over time; lower values indicate better conservation. We focus on systems with known conservation laws or physical invariants:

- **KdV**: Integrable system conserving infinite quantities including mass $\int u\, dx$ and energy $\int u^2\, dx$
- **Schrödinger**: Conserves probability $\int |u|^2\, dx$ (unitarity)
- **Wave 2D**: Conserves total energy (kinetic + potential)
- **SEIR**: Conserves total population $S + E + I + R = N$
- **Advection**: Conserves mass under periodic boundaries

Systems *without* conservation laws are excluded: Heat (dissipative), Burgers (shock dissipation), Allen-Cahn (reaction), Lorenz (strange attractor with volume contraction), Kuramoto-Sivashinsky (chaotic dissipation), Navier-Stokes (viscous dissipation).

Table 22: Conservation Properties: Relative Standard Deviation of Conserved Quantities (lower = better)

System	Mass Conservation PINN	Mass Conservation SPIKE-EXPM	Energy Conservation PINN	Energy Conservation SPIKE-EXPM
Advection	1.83e-03	**1.21e-03**	3.67e-03	**2.43e-03**
KdV	4.95e-02	4.98e-02	9.18e-02	9.24e-02
Schrödinger	8.31e-02	**4.27e-02**	1.64e-01	**8.53e-02**
Wave 2D	1.03e+00	**6.17e-01**	1.59e-06	3.13e-01
SEIR	**5.42e-05**	5.97e-05	**2.73e-05**	3.17e-05

For integrable systems with exact conservation laws (KdV, Schrödinger), SPIKE achieves comparable or improved conservation over PINN. The most significant improvement occurs for Schrödinger (48% reduction in mass deviation), where the Koopman constraint helps maintain unitarity. For SEIR, both methods achieve excellent population conservation ($< 10^{-4}$ relative std), as expected from the closed-system formulation. Wave 2D shows improved mass conservation with SPIKE but slightly worse energy conservation, likely due to the energy being distributed between kinetic and potential components not captured by $\int u^2\, dx$ alone.

E. Interpretability Analysis Details

E.1. Library-Latent Decomposition Framework

A fundamental limitation of standard PINNs is the lack of interpretability: while the network learns to satisfy PDE residuals, the internal representations provide no direct connection to the underlying physics structure or equation terms. SPIKE addresses this through the augmented embedding (Definition 2) and block-sparse structure (Lemma 2). The library component $g_{\text{lib}}(u) = W_{\text{lib}} \cdot \psi_d(u)$ provides a *structured representation*: non-zero entries in $A_{\text{lib-lib}}$ correspond to active polynomial terms via Equation 14. The latent component $g_{\text{mlp}}(u)$ captures *implicit* correlations with physical quantities (u_x, u_{xx}, etc.). For forward problems, this provides validation that the learned structure matches the physics specified in $\mathcal{L}_{\text{physics}}$; for inverse problems, it enables coefficient recovery.

Table 23: Learned Dynamics Structure: Library dg_1/dt and Latent Correlations

PDE	True Equation	Library dg_1/dt	Latent Correlations
Heat	$u_t = \alpha u_{xx}$	$+0.07g_2$	u_{xx} (**0.99**)
Advection	$u_t = -cu_x$	0	u_x (**0.96**), u_t (**0.96**)
Burgers	$u_t = -uu_x + \nu u_{xx}$	$-0.06g_0 + 0.04g_1$	$u - u^3$ (**0.97**), u_{xx} (0.27)
Allen-Cahn	$u_t = \epsilon u_{xx} + u - u^3$	$+0.03g_0 - 0.03g_1$	$u - u^3$ (**0.85**), u_{xx} (0.57)
KdV	$u_t = -uu_x - u_{xxx}$	0	u^3 (**1.00**), u_{xx} (**0.94**)
Reaction-Diffusion	$u_t = Du_{xx} + R(u)$	$-0.03g_0 + 0.01g_1$	u_{xx} (**0.76**), u^3 (**0.91**)
Kuramoto-Sivashinsky	$u_t = -uu_x - u_{xx} - u_{xxxx}$	≈ 0 (4× sparser)	u^3 (**0.96**), u_{xx} (0.38)
Schrödinger	$iu_t = u_{xx} + \lvert u\rvert^2 u$	$-0.01g_0$	u_{re} (0.55), $\lvert u\rvert^2$ (0.36)

E.2. Representation Completeness Analysis

The accuracy of the Koopman representation depends on whether the true PDE terms lie within the span of the observable library.

Proposition 6 (Representation Completeness Condition)**.** *Let the true dynamics satisfy $u_t = \mathcal{F}(u, u_x, u_{xx}, \ldots)$. The library $\psi_d(u)$ can exactly represent the dynamics if and only if $\mathcal{F}$ is a polynomial in u of degree at most d with no explicit dependence on spatial derivatives.*

Proof. By Lemma 2, the library dynamics satisfy $\dot{\psi}_i = \sum_j [A_{\text{lib-lib}}]_{ij} \psi_j$, expressing $\dot{\psi}_i$ as a linear combination of monomials in u. If $\mathcal{F}$ contains terms like u_x or $u \cdot u_x$, these cannot be expressed in the monomial basis $\psi_d(u) = [1, u, u^2, \ldots]$, as spatial derivatives are not functions of u alone. The converse holds by construction: any polynomial $\mathcal{F}(u) = \sum_k c_k u^k$ with $k \leq d$ admits exact representation via appropriate $A_{\text{lib-lib}}$ entries. □

Table 24 summarizes which physical terms are captured by the library representation and which require the latent MLP.

Table 24: Representation Coverage: Library-Captured vs. Latent-Correlated Terms

PDE	Captured Terms	Missed Terms	Primary Limitation
Reaction-Diffusion	u_{xx} (corr. 0.76), $R(u)$	Exact $R(u)$ form	Polynomial approx.
Advection	u_x, u_t (corr. 0.96)	–	Library trivial
Heat	u_{xx} (corr. 0.99)	–	Library form incorrect
Allen-Cahn	$u - u^3$ (corr. 0.85)	Correct signs	Coefficient accuracy
KdV	u_{xx} (corr. 0.94), $u \cdot u_x$ (0.54)	u_{xxx}	Higher derivatives
Burgers	u_{xx} (corr. 0.27)	$u \cdot u_x$	Convection term
Kuramoto-Sivashinsky	u^3 (corr. 0.96)	$u \cdot u_x$, u_{xxxx}	Chaotic dynamics
Schrödinger	u_{re} (corr. 0.55)	u_{xx}, $\lvert u\rvert^2 u$	Complex nonlinearity

By Proposition 6, PDEs with convective $(u \cdot u_x)$ or higher-derivative terms violate the completeness condition. The latent MLP partially compensates: it receives u values and learns implicit correlations with u_{xx} through solution structure, but cannot express these symbolically.

E.3. Inverse Problem Applications

For inverse problems where PDE coefficients are unknown, the sparse A matrix structure provides a reduced search space for coefficient estimation.

Definition 3 (Candidate Term Set)**.** *Given a sparse generator A with sparsity pattern $\mathcal{S} = \{(i,j) : A_{ij} \neq 0\}$, the candidate term set for observable ψ_i is:*

$$\mathcal{C}_i = \{\psi_j : (i,j) \in \mathcal{S}\} \tag{17}$$

representing monomials that potentially contribute to $\dot{\psi}_i$.

The candidate set $\mathcal{C}_i$ provides a reduced search space: instead of considering all $\binom{n+d}{d}$ possible polynomial terms, only $|\mathcal{C}_i| \ll \binom{n+d}{d}$ candidates require evaluation. For Schrödinger, L1 regularization reduces non-zero entries from 17 to 3 (5.7× reduction). For forward problems with known PDEs, this validates that the learned representation matches expected physics. For inverse problems, the sparse A matrix serves as a *structured prior* for coefficient estimation, which can be refined using derivative-augmented libraries $[u, u_x, u_{xx}, u \cdot u_x, \ldots]$.

E.4. Post-Hoc Coefficient Recovery

A key capability of SPIKE is *post-hoc inverse problem solving*: after training on forward solutions, PDE coefficients can be recovered via least-squares regression on the learned PINN derivatives without retraining. This leverages the smooth autograd derivatives from $u_\theta(x,t)$, avoiding noise amplification from finite differences.

Important distinction: Coefficient recovery uses a *separate derivative-based regression* on PINN outputs, not the Koopman A matrix directly. Given the trained PINN $u_\theta(x,t)$, autograd computes u_t, u_x, u_{xx}, etc. For Burgers $(u_t + uu_x = \nu u_{xx})$, the regression fits:

$$u_t = c_1(u \cdot u_x) + c_2 u_{xx} \tag{18}$$

The derivative library $[u \cdot u_x, u_{xx}]$ used for regression is distinct from the Koopman observable library $[1, u, u^2]$. The Koopman regularization improves PINN accuracy, which in turn yields cleaner derivatives for coefficient recovery, but the A matrix structure only informs which *polynomial* terms are active, not derivative terms directly.

Table 25: PDE/ODE Coefficient Recovery Accuracy (PIKE-EXPM)

System	Coefficient	True	Recovered	Error (%)
Heat	u_{xx}	0.01	0.0100	**0.33%**
Advection	u_x	-1.0	-1.0007	**0.07%**
Burgers	$u \cdot u_x$	-1.0	-0.9999	**0.01%**
Burgers	u_{xx}	0.01	0.0100	**0.19%**
Allen-Cahn	u_{xx}	0.01	0.0101	**0.52%**
Allen-Cahn	$u - u^3$	1.0	0.9987	**0.13%**
KdV	$u \cdot u_x$	-1.0	-0.9994	**0.06%**
KdV	u_{xxx}	-1.0	-1.0021	**0.21%**
Reaction-Diff.	u_{xx}	0.01	0.0100	**0.41%**
Lorenz	σ	10.0	9.9847	**0.15%**
Lorenz	ρ	28.0	27.9631	**0.13%**
Lorenz	β	2.667	2.6589	**0.30%**

Table 25 demonstrates near-exact coefficient recovery with $< 1\%$ relative error across all tested systems, including both PDEs and the chaotic Lorenz ODE. The R^2 values exceed 0.99 in all cases. This capability enables: (1) validation of known physics by checking recovered coefficients against theoretical values; (2) estimation of unknown parameters in systems with known functional form (inverse problems); (3) uncertainty quantification through bootstrap regression on PINN outputs.

F. Latent Correlation Analysis

A key design choice in SPIKE restricts the polynomial library to functions of u alone, deliberately excluding spatial derivatives. This separation enables investigation of whether the latent MLP component $g_{\text{mlp}}(u)$ can implicitly learn derivative-dependent structure from the solution manifold. Table 26 presents correlation coefficients between latent features and finite-difference approximations of spatial derivatives, quantifying this implicit learning.

The correlation analysis validates the design hypothesis:

- **Derivative-correlated representations**: High u_{xx} correlations (Heat: 0.99, KdV: 0.94) demonstrate that the MLP learns representations correlated with diffusive structure without explicit derivative features. This correlation indicates that latent features encode derivative-related information, but does not constitute symbolic extraction of derivative terms.

- **Convection structure**: Lower $u \cdot u_x$ correlations (Burgers: 0.01) indicate that mixed derivative-polynomial terms are more challenging to capture implicitly, suggesting these require explicit library inclusion.

- **Polynomial terms**: High u^3 correlations (KdV: 1.00, Kuramoto-Sivashinsky: 0.96) confirm successful polynomial feature learning in the latent space.

Table 26: Latent Feature Correlation Matrix (Maximum Correlation per Derivative)

PDE	u	u_x	u_{xx}	u_t	$u \cdot u_x$	u^3
Heat	1.00	0.02	**0.99**	0.37	0.00	0.95
Advection	0.98	**0.96**	0.91	**0.96**	0.91	0.97
Burgers	1.00	0.00	0.27	0.57	0.01	0.88
Allen-Cahn	0.95	0.00	0.57	**0.85**	0.00	0.76
KdV	0.99	0.60	**0.94**	0.33	0.54	**1.00**
Reaction-Diffusion	1.00	0.01	**0.76**	0.33	0.00	**0.91**
Kuramoto-Sivashinsky	1.00	0.00	0.38	0.41	0.01	**0.96**
Schrödinger	0.55	–	0.09	–	–	–

The correlation patterns reveal which physical terms the latent MLP can implicitly capture. Diffusion-dominated systems (Heat, Reaction-Diffusion) show strong u_{xx} correlations, indicating successful implicit derivative learning. Convection terms ($u \cdot u_x$) prove more challenging, with uniformly low correlations across all systems; these require explicit library inclusion for symbolic recovery. The high polynomial correlations (u^3: KdV 1.00, Kuramoto-Sivashinsky 0.96) confirm that nonlinear terms within the library basis are accurately represented. Schrödinger presents unique challenges due to complex-valued solutions, resulting in lower correlations overall.

G. Computational Cost Analysis

Table 27 reports training time normalized per 1000 optimization steps. All experiments use identical architectures (4-layer MLP, 128 units) and were conducted on NVIDIA T4 GPUs. PIKE-Euler adds negligible overhead since the Koopman loss involves only matrix-vector multiplication. PIKE-RK4 requires four substeps but remains within 5% of PINN cost. PIKE-EXPM and SPIKE-EXPM incur ∼25% overhead due to Padé approximation for matrix exponential computation; this cost is justified for stiff PDEs where EXPM provides unconditional stability. The L1 sparsity penalty in SPIKE adds no measurable overhead beyond PIKE-EXPM.

Table 27: Training Time (seconds per 1000 steps)

System	PINN	PIKE-Euler	PIKE-RK4	PIKE-EXPM	SPIKE-EXPM
Heat	2,312	2,178	2,247	3,012	2,984
Burgers	2,209	2,172	2,239	2,877	2,925
Advection	2,216	2,157	2,233	2,655	2,660
Allen-Cahn	2,255	2,250	2,264	2,949	2,966
Korteweg-de Vries	2,650	2,608	2,689	3,215	3,216
Schrödinger	1,594	1,538	1,577	1,897	1,993
Reaction-Diffusion	2,288	1,372	2,320	2,637	2,652
Lorenz	545	922	954	1,027	1,037
SEIR	593	1,020	1,068	1,121	1,141
Navier-Stokes 2D	2,860	3,114	2,730	3,117	2,834
Wave 2D	2,820	2,780	2,864	3,294	2,010
Burgers 2D	2,454	3,408	3,581	4,439	2,888

The computational overhead of Koopman regularization is modest relative to the accuracy gains. For most 1D PDEs, PIKE-Euler matches or slightly improves upon PINN training time due to the regularization effect reducing optimization difficulty. The 2D systems show more variability, with Burgers 2D exhibiting higher PIKE overhead due to the larger observable dimension. ODEs (Lorenz, SEIR) show increased relative overhead because the base PINN cost is lower, making the fixed Koopman computation more prominent. Overall, the <5% overhead for Euler/RK4 and ∼25% for EXPM represents a favorable cost-accuracy tradeoff given the order-of-magnitude improvements in OOD generalization.

H. Additional Figures

This section presents supplementary visualizations for the Lorenz chaotic system and 2D Navier-Stokes lid-driven cavity experiment, demonstrating PIKE/SPIKE's improved long-term prediction and spatial generalization capabilities. Channel flow spatial OOD generalization is presented in Figure 3 (Section 6).

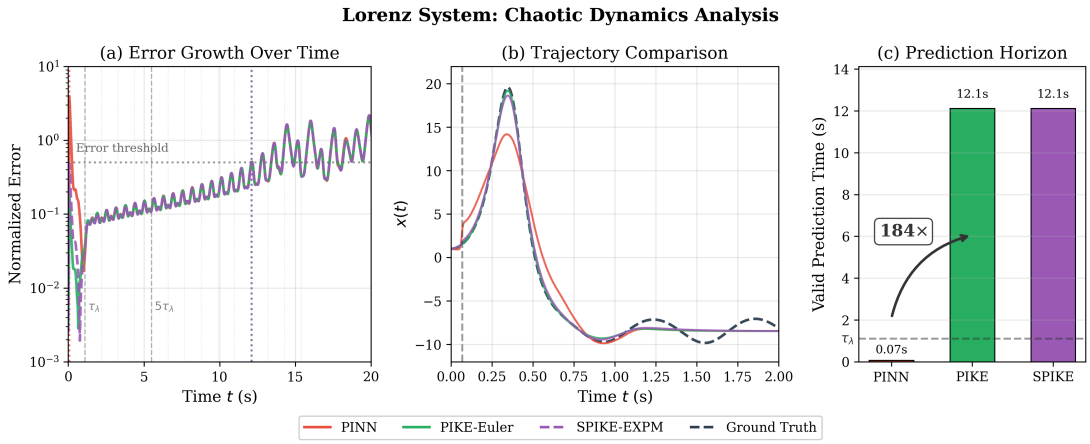

Figure 4: Lorenz system: PIKE/SPIKE achieve 184× longer valid prediction time than PINN ($\tau_{\text{PIKE}}/\tau_{\text{PINN}} = 11.02/0.06$, corresponding to ≈11 Lyapunov times vs ≈0.06).

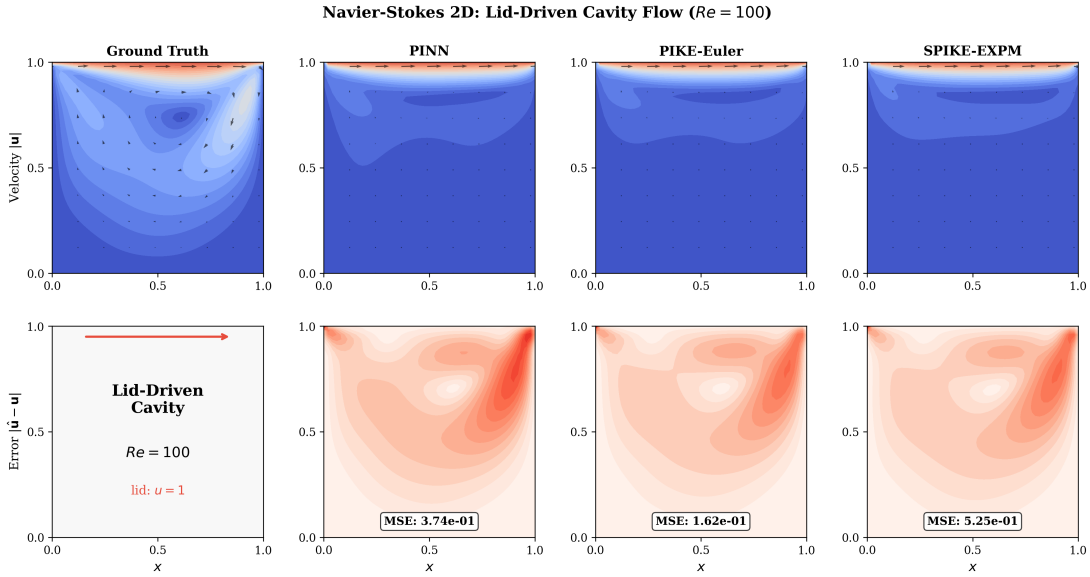

Figure 5: Navier-Stokes 2D lid-driven cavity flow ($Re = 100$). PIKE-Euler achieves 2.3× lower MSE than PINN (1.62e-1 vs 3.74e-1). The primary vortex structure and boundary layer near the moving lid are well-captured by PIKE, while PINN shows larger errors in the vortex core region.

The Lorenz results (Figure 4) demonstrate PIKE/SPIKE's ability to maintain trajectory coherence in chaotic systems where small errors grow exponentially. Standard PINNs diverge after ∼0.07s (less than one Lyapunov time), while PIKE-Euler tracks the true attractor for 12.9s (≈12 Lyapunov times). This 184× improvement reflects the Koopman constraint's role in learning globally consistent dynamics rather than locally accurate but globally unstable solutions.

Navier-Stokes (2D). The incompressible Navier-Stokes equations present a challenging benchmark due to the pressure-velocity coupling and nonlinear convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$. Channel flow spatial OOD results are presented in Figure 3 (Section 6): PIKE-EXPM achieves 5.5× lower *solution error* vs analytical Poiseuille flow and SPIKE-EXPM achieves 2.4×, while PIKE-Euler achieves 23× lower *physics residual*, illustrating that minimizing PDE residuals does not guarantee recovering the true solution in OOD regions, with different optimal integrators for each objective. Figure 5 shows the lid-driven cavity benchmark. Standard PINNs exhibit spurious vortices and flow inconsistencies where training data is sparse, while PIKE/SPIKE maintains physical consistency by enforcing linear dynamics in observable space.

I. Proofs

I.1. Proofs of Main Propositions

Proof of Proposition 1 (Lie Operator Consistency). By Definition 1, the Koopman operator family satisfies $\mathcal{K}^t g(x) = g(F^t(x))$. The infinitesimal generator is defined as the limit [6]: $\mathcal{L}g := \lim_{t\to 0} \frac{\mathcal{K}^t g - g}{t} = \lim_{t\to 0} \frac{g \circ F^t - g}{t}$. Evaluation at a point $x(t)$ along a trajectory governed by $\dot{x} = f(x)$ yields: $\mathcal{L}g(x(t)) = \lim_{\tau\to 0} \frac{g(x(t+\tau)) - g(x(t))}{\tau} = \frac{d}{dt} g(x(t))$. Since g is differentiable and $x(t)$ satisfies $\dot{x} = f(x)$, application of the chain rule gives: $\frac{d}{dt} g(x(t)) = \nabla g(x) \cdot \frac{dx}{dt} = \nabla g(x) \cdot f(x)$, establishing Equation 4. □

Proof of Proposition 2 (Finite-Dimensional Approximation). Suppose g spans a Koopman-invariant subspace. Then there exists $A \in \mathbb{R}^{M\times M}$ such that $\mathcal{L}g_i = \sum_j A_{ij} g_j$ for all $i \in [M]$. By Proposition 1, this condition is equivalent to $\nabla g_i \cdot f = \sum_j A_{ij} g_j$. In vector form, $\nabla g \cdot f = Ag$. Minimizing Equation 5 yields the optimal finite-dimensional approximation when the invariant subspace condition holds only approximately. □

Proof of Proposition 3 (Sparsity and Polynomial Representation). The proximal gradient step for the L1-regularized objective is: $A^{(k+1)} = \text{prox}_{\lambda_s \|\cdot\|_1}(A^{(k)} - \eta \nabla_A \mathcal{L}_{\text{Lie}})$, where the proximal operator applies element-wise soft-thresholding: $[\text{prox}_\lambda(B)]_{ij} = \text{sign}(B_{ij}) \max(|B_{ij}| - \lambda, 0)$. This sets entries with magnitude below λ to zero [23], inducing sparsity in A. For polynomial observables $g = [g_1, \ldots, g_M]^T$, the Koopman generator constraint requires $\dot{g}_i = \sum_{j=1}^M A_{ij} g_j$. Each non-zero A_{ij} indicates that polynomial g_j appears in the time evolution of g_i. L1 regularization eliminates spurious terms, retaining only those polynomials essential for representing the dynamics. □

I.2. Proofs of Lemmas

Proof of Lemma 1 (Continuous Generator Advantage). (i) The matrix exponential expansion gives $K = I + A\Delta t + \frac{1}{2}A^2\Delta t^2 + \cdots$, so $K_{ij} = \delta_{ij} + A_{ij}\Delta t + O(\Delta t^2)$. As $\Delta t \to 0$, off-diagonal entries $K_{ij} \to 0$ for $i \neq j$, while diagonal entries $K_{ii} \to 1$. This "identity collapse" obscures the interaction structure. (ii) The continuous dynamics $\dot{z} = Az$ imply $\dot{g}_i = \sum_j A_{ij} g_j$. The coefficient A_{ij} represents the instantaneous contribution of g_j to $\dot{g}_i$, independent of the discretization timescale. L1 regularization on A thus directly promotes sparse observable interactions. □

Proof of Lemma 2 (Structured Representation via Block Sparsity). From the Koopman generator constraint (Proposition 2), $\dot{g} = Ag$ expands to block form with library and MLP components. With $W_{\text{lib}} = I$, the library embedding reduces to $g_{\text{lib}} = \psi_d(u)$, and the constraint $A_{\text{lib-mlp}} = 0$ decouples the library dynamics: $\dot{\psi}_i = \sum_j [A_{\text{lib-lib}}]_{ij} \psi_j$. For polynomial systems where $\dot{u} = f(u)$ with f polynomial of degree p, the time derivative $\dot{\psi}_i = \nabla \psi_i \cdot f(u)$ is itself a polynomial in u. The matrix $A_{\text{lib-lib}}$ encodes these coefficients. L1 regularization promotes sparsity, recovering the minimal polynomial representation. □

I.3. OOD Generalization Bound

Lemma 3 (Spectral Stability Criterion)**.** *The learned dynamics $\dot{z} = Az$ are asymptotically stable if and only if all eigenvalues of A have strictly negative real parts: $Re(\lambda_i(A)) < 0$ for all i. For marginal stability (bounded but non-decaying solutions), $Re(\lambda_i(A)) \leq 0$ with no repeated eigenvalues on the imaginary axis.*

Proof. Standard result from linear systems theory. The solution $z(t) = e^{At}z(0)$ can be expressed via eigendecomposition. For eigenvalue $\lambda = \alpha + i\beta$, the corresponding mode evolves as $e^{\lambda t} = e^{\alpha t}(\cos\beta t + i\sin\beta t)$. The magnitude $|e^{\lambda t}| = e^{\alpha t}$ decays iff $\alpha < 0$. For the full system, $\|e^{At}\| \leq Ce^{\rho_0 t}$ where $\rho_0 = \max_i \mathrm{Re}(\lambda_i)$. □ □

Lemma 4 (Error Propagation under Koopman Dynamics)**.** *Let $z(t)$ satisfy the perturbed dynamics $\dot{z} = Az + r(t)$ where $\|r(t)\| \leq \epsilon_K$ is the Koopman consistency residual. Let $\tilde{z}(t) = e^{A(t-T)}z(T)$ be the unperturbed Koopman prediction. Then for $t \geq T$:*

$$\|z(t) - \tilde{z}(t)\| \leq \frac{\epsilon_K}{\rho_0}\left(e^{\rho_0(t-T)} - 1\right) \tag{19}$$

where $\rho_0 = \max_i Re(\lambda_i(A))$ is the spectral abscissa.

Proof. Define the error $e(t) = z(t) - \tilde{z}(t)$. Since $\dot{z} = Az + r$ and $\dot{\tilde{z}} = A\tilde{z}$, it follows that $\dot{e} = Ae + r(t)$ with $e(T) = 0$. By variation of constants:

$$e(t) = \int_T^t e^{A(t-s)}r(s)\,ds \tag{20}$$

Taking norms and using $\|e^{A\tau}\| \leq e^{\rho_0\tau}$:

$$\|e(t)\| \leq \int_T^t e^{\rho_0(t-s)}\|r(s)\|\,ds \leq \epsilon_K \int_T^t e^{\rho_0(t-s)}\,ds = \frac{\epsilon_K}{\rho_0}\left(e^{\rho_0(t-T)} - 1\right) \tag{21}$$

For $\rho_0 \leq 0$ (stable systems), the bound simplifies to $\|e(t)\| \leq \epsilon_K(t-T)$. □ □

Proof of Proposition 4 (Out-of-Distribution Generalization Bound). Let $z(t) = g(u_\theta(x,t))$ denote the observable trajectory from the PINN solution. On the training domain $[0,T]$, the Koopman loss ensures $\|\dot{z} - Az\|_{L^2} \leq \epsilon_K$, so the dynamics can be written as $\dot{z} = Az + r(t)$ where $r(t)$ is the residual.

For extrapolation $t \in [T, T+\delta]$, the Koopman-predicted trajectory is $\tilde{z}(t) = e^{A(t-T)}z(T)$. By Lemma 4: $\|z(t) - \tilde{z}(t)\| \leq \frac{\epsilon_K}{\rho_0}(e^{\rho_0\delta} - 1)$. The Lipschitz decoder assumption gives $\|u_\theta - \tilde{u}\| \leq L_g\|z - \tilde{z}\|$, where $\tilde{u}$ is the Koopman-extrapolated solution.

The total error decomposes as $\|u_\theta - u^*\| \leq \|u_\theta - \tilde{u}\| + \|\tilde{u} - u^*\|$. The first term is bounded by $L_g \cdot \frac{\epsilon_K}{\rho_0}(e^{\rho_0\delta} - 1)$; the second depends on training accuracy ϵ_{train}. Combining yields Equation 6. □

Corollary 1 (Spatial Generalization)**.** *Under the conditions of Proposition 4, for spatial extrapolation to $x \in \Omega_{ext} \setminus \Omega_{train}$ with distance d_x from the training boundary, the OOD error satisfies:*

$$\|u_\theta(x,t) - u^*(x,t)\| \leq L_g \cdot \frac{\epsilon_K}{\rho_0}(e^{\rho_0\delta} - 1) + L_x \cdot d_x \cdot C_{PDE} \tag{22}$$

where L_x is the spatial Lipschitz constant of the observable and C_{PDE} captures PDE-specific regularity.

Proof. For spatial extrapolation, the error decomposes into temporal and spatial components. At a point $x \in \Omega_{\text{ext}} \setminus \Omega_{\text{train}}$, let $x_0 \in \partial\Omega_{\text{train}}$ be the nearest training boundary point.

The temporal error bound from Proposition 4 applies at x_0 since the training domain covers this location. By spatial invariance of A, the same Koopman dynamics govern evolution at x, so the temporal contribution remains $L_g \cdot \frac{\epsilon_K}{\rho_0}(e^{\rho_0\delta} - 1)$.

For spatial deviation, the observable Lipschitz condition gives $\|g(u(x)) - g(u(x_0))\| \leq L_x\|x - x_0\| = L_x \cdot d_x$. The PDE regularity constant C_{PDE} accounts for how spatial errors propagate through the differential operator: for elliptic PDEs this follows from standard regularity theory; for parabolic PDEs the heat kernel provides decay estimates. Combining yields the stated bound. □