NIGO: Lyapunov-Stable Continuous-Time Neural Operators for Long-Horizon PDE Dynamics

Ashish Musale [1] **Ajay Waghumbare** [1] **Vinduja T** [1] **Upasna Singh** [1]

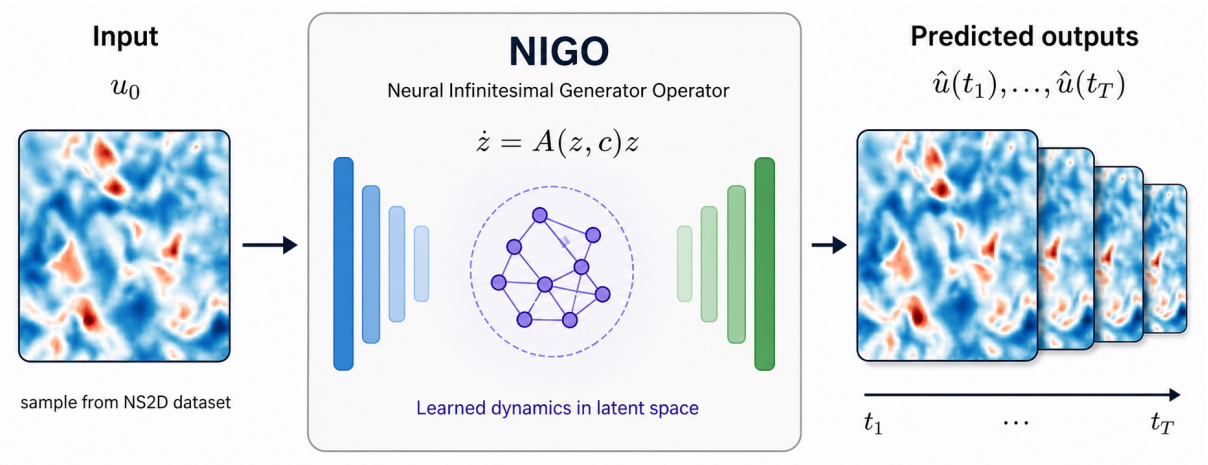


Figure 1. Overview of NIGO. Given an input PDE state, NIGO learns stable continuous-time latent dynamics and produces long-horizon predictions. The example shown uses a sample from the NS2D dataset.

Abstract

Neural operators have become a promising framework for learning solution operators of partial differential equations, yet their long-horizon use remains limited by the stability of autoregressive rollout. In many spatiotemporal settings, learned time-advancement maps must be composed repeatedly, causing small local errors to accumulate into trajectory drift, spectral artifacts, or non-physical energy growth. We propose **NIGO**, a continuous-time neural operator that replaces unconstrained discrete rollout maps with a dissipative infinitesimal generator in latent space. Given an initial PDE state, NIGO encodes the field into a compact latent representation and evolves it analytically through a matrix exponential. The generator is parameterized as a skew-symmetric transport component plus a negative-semidefinite dissipative component, which enforces non-increasing latent energy by construction. A physics-inference network predicts the generator coefficients from the initial state and conditioning variables, while temporal refinement and Lipschitz-controlled decoding map stable latent trajectories back to physical fields. We evaluate NIGO on Burgers, cylinder flow, Kuramoto–Sivashinsky, shallow-water, and incompressible Navier–Stokes systems, covering decaying, periodic, and chaotic regimes. Across extended rollouts, NIGO avoids the blowup and freezing modes observed in standard neural-operator and physics-informed baselines. Ablations show that the dissipative generator is necessary for stable long-horizon evolution, and mechanistic analyses reveal learned structure consistent with viscous scaling, operator locality, and advection–dissipation balance. These results indicate that hard-constrained continuous-time latent dynamics provide a principled foundation for stable neural operators for PDE forecasting. Code is available at `https://github.com/meural-operator/nigo.git`.

1. Introduction

Neural operators have changed the way surrogate models are built for partial differential equations (PDEs). Instead of learning maps between fixed finite-dimensional arrays, they learn maps between function spaces, allowing a single

[1] DIAT, Pune, India. Correspondence to: Ashish Musale <ashish.rsch@gmail.com>.

Accepted at the AI for Physics (AI4Physics) Workshop at ICML 2026, Seoul, South Korea.

model to approximate families of solution operators across varying initial conditions, coefficients, forcing terms, geometries, and discretizations (Li et al., 2020; Lu et al., 2021; Kovachki et al., 2023). This perspective has led to a broad range of architectures, including Fourier neural operators, DeepONets, U-shaped neural operators, message-passing solvers, mesh-agnostic solvers, and attention-based operator learners (Li et al., 2020; Lu et al., 2021; Rahman et al., 2022; Brandstetter et al., 2022; Boussif et al., 2022; Cao, 2021; Li et al., 2022). These models have demonstrated strong accuracy and efficiency on scientific benchmarks, often giving large speedups over classical solvers.

The long-horizon setting exposes a different failure mode. Many neural PDE surrogates are deployed as learned time-advancement maps: a model predicts the next state, its prediction is fed back as input, and the process is repeated. This autoregressive use is computationally convenient, but small local errors can accumulate into trajectory drift, spectral distortion, excessive dissipation, frozen dynamics, or non-physical growth. Prior work has identified uncontrolled error growth as a central source of instability in autoregressive neural operators (McCabe et al., 2023), and benchmark studies such as PDEBench show that time-dependent scientific ML tasks remain challenging for existing models (Takamoto et al., 2022). Existing stabilization strategies, including rollout training, training noise, filtering, or hybrid numerical–learning designs, improve robustness in important cases but do not generally make stability an intrinsic property of the learned temporal operator.

One way to impose physical structure is through the training objective. Physics-informed neural networks and physics-informed neural operators incorporate PDE residuals, boundary conditions, or other governing constraints as loss terms (Raissi et al., 2019; Wang et al., 2021b; Li et al., 2024; Karniadakis et al., 2021). These methods can improve data efficiency and physical consistency, but their constraints are typically soft penalties and remain subject to optimization pathologies, gradient imbalance, and temporal error accumulation (Krishnapriyan et al., 2021; Wang et al., 2021a; Feng et al., 2025). Another approach is to build structure into the architecture itself. Hamiltonian, Lagrangian, and symplectic neural models show that hard constraints can improve learning for conservative systems (Greydanus et al., 2019; Cranmer et al., 2020; David & Méhats, 2023; Zhong et al., 2019). However, many PDEs of interest, especially viscous and turbulent flows, are not purely conservative: they combine transport, energy transfer across scales, and dissipation. For such systems, exact conservation is not the right inductive bias.

We propose **NIGO** (Neural Infinitesimal Generator Operator), a continuous-time neural operator for stable long-horizon PDE forecasting. NIGO replaces unconstrained discrete rollout maps with a structured latent infinitesimal generator. Given an initial PDE state, the model encodes the field into a compact latent representation and evolves it analytically by matrix-exponential dynamics. The generator is parameterized as

$$A = \alpha(K - K^\top) - \beta R^\top R, \tag{1}$$

where the skew-symmetric component represents transport-like, energy-preserving dynamics and the negative-semidefinite component represents dissipation. This construction enforces non-increasing latent energy by design. Stability is therefore not introduced as a rollout penalty or a post hoc regularizer; it is built into the temporal operator.

NIGO separates two issues that are often conflated in neural PDE forecasting. The first is *structural stability*: whether the learned dynamical core can generate unbounded energy growth over long horizons. NIGO addresses this through its dissipative generator parameterization. The second is *predictive fidelity*: whether the stable latent trajectory corresponds to the correct physical solution. NIGO addresses this through a conditioned encoder, a physics-inference network that predicts basis coefficients and advection–dissipation scalings, a temporal refiner for multi-scale corrections, and a Lipschitz-controlled decoder in the bounded variant. This design combines the efficiency of latent operator learning with a hard stability constraint on continuous-time evolution.

We evaluate NIGO on Burgers, cylinder flow, Kuramoto–Sivashinsky, shallow-water, and incompressible Navier–Stokes dynamics, spanning decaying, periodic, and chaotic regimes. Across extended rollouts, NIGO avoids the blowup and freezing modes observed in standard neural-operator and physics-informed baselines. Ablations show that the dissipative generator is essential for stable long-horizon prediction. Interpretability analyses further show learned operator structure consistent with viscous spectral scaling, operator locality, state-invariant dissipation, and regime-dependent advection–dissipation balance.

Our contributions are:

- **A continuous-time neural operator for stable PDE forecasting.** NIGO replaces autoregressive composition of unconstrained discrete update maps with matrix-exponential evolution under a learned infinitesimal generator.
- **A dissipative latent generator with Lyapunov stability by construction.** The generator combines a skew-symmetric transport component with a negative-semidefinite dissipative component, enforcing non-increasing latent energy while allowing nontrivial dissipative dynamics.

- **A conditioned architecture linking stable latent evolution to physical predictions.** NIGO uses physics-inference, temporal refinement, and Lipschitz-controlled decoding to map stable latent trajectories back to physical fields.

- **Long-horizon evaluation with ablations and interpretability analysis.** We evaluate across multiple PDE families and attractor regimes, showing that the generator structure is necessary for stability and that the learned operators recover physically interpretable patterns.

2. Related Work

Neural operators for PDE surrogate modeling. Neural operators learn mappings between function spaces and provide fast surrogates for families of PDE solutions. FNO parameterizes integral kernels in Fourier space and demonstrated efficient learning on Burgers, Darcy flow, and Navier–Stokes benchmarks (Li et al., 2020). DeepONet uses branch and trunk networks to approximate nonlinear operators under the universal approximation theorem for operators (Lu et al., 2021). The general neural-operator framework formalizes these models as discretization-invariant operator approximators (Kovachki et al., 2023). Subsequent architectures improve scalability and geometric flexibility through U-shaped designs, message passing, mesh-agnostic representations, and attention-based operator learning (Rahman et al., 2022; Brandstetter et al., 2022; Boussif et al., 2022; Cao, 2021; Li et al., 2022). These methods are highly effective for short-horizon or fixed-window prediction, but long-horizon deployment often requires repeated temporal composition.

Long-horizon rollout stability. Autoregressive rollout instability is a central challenge for neural PDE solvers. When a learned time-stepper is repeatedly composed, small local errors can induce distribution shift, high-frequency artifacts, trajectory drift, or eventual divergence. McCabe et al. analyze this issue for autoregressive neural operators and propose design principles for reducing instability-inducing operations (McCabe et al., 2023). PDEBench provides standardized time-dependent scientific ML benchmarks and shows that many PDE tasks remain difficult for existing models (Takamoto et al., 2022). Learned turbulence and fluid simulators address long-rollout stability through training noise, temporal downsampling, loss constraints, or hybrid learned–numerical components (Stachenfeld et al., 2021; Kochkov et al., 2021). NIGO takes a complementary route: rather than stabilizing a discrete rollout map, it replaces rollout with continuous-time latent evolution under a dissipative generator.

Physics-informed and structure-preserving constraints. Physics-informed learning incorporates governing equations through PDE residuals, boundary conditions, or operator-level constraints (Raissi et al., 2019; Wang et al., 2021b; Li et al., 2024; Karniadakis et al., 2021; Goswami et al., 2023; Cuomo et al., 2022). These approaches improve physical consistency, but their constraints are usually soft optimization penalties and can suffer from ill-conditioning, gradient imbalance, and temporal error accumulation (Krishnapriyan et al., 2021; Wang et al., 2021a; Feng et al., 2025). Structure-preserving neural dynamics impose hard architectural constraints, as in Hamiltonian, Lagrangian, and symplectic models (Greydanus et al., 2019; Cranmer et al., 2020; David & Méhats, 2023; Zhong et al., 2019). These methods are well matched to conservative systems, whereas dissipative PDEs require a different constraint. NIGO adopts the hard-constraint philosophy, but replaces exact conservation with a transport–dissipation decomposition that enforces non-increasing latent energy.

Latent and continuous-time dynamics. Latent operator methods reduce the cost of high-dimensional PDE prediction by learning compact representations in which operators are easier to approximate (Kontolati et al., 2024; Wang & Wang, 2024; Karumuri et al., 2025; Lai et al., 2025). Koopman-inspired approaches seek latent coordinates where nonlinear dynamics become approximately linear, enabling efficient time advancement and improved interpretability (Lusch et al., 2018; Yu et al., 2025; Gupta et al., 2025). Continuous-time models, including neural ODEs, neural controlled differential equations, liquid time-constant networks, and stable learned dynamics, show that neural temporal models can be formulated and constrained through differential equations (Massaroli et al., 2020; Kidger et al., 2020; Hasani et al., 2021; Kolter & Manek, 2019). NIGO combines these directions: it learns a latent neural operator, but constrains the infinitesimal generator itself so that its continuous-time flow is dissipative by construction.

3. Method

NIGO is a continuous-time neural operator for long-horizon PDE forecasting. Following the standard operator-learning viewpoint, the goal is not only to map one discretized array to another, but to approximate a time-indexed solution operator that can be evaluated at requested future times. NIGO realizes this operator by encoding the initial field into a compact latent state, constructing a dissipative infinitesimal generator conditioned on the input regime, evolving the latent state analytically by a matrix exponential, and decoding the resulting trajectory back to physical space. This section first defines the forecasting operator, then formalizes the difference between conventional finite-step querying and the generator-based querying used by NIGO.

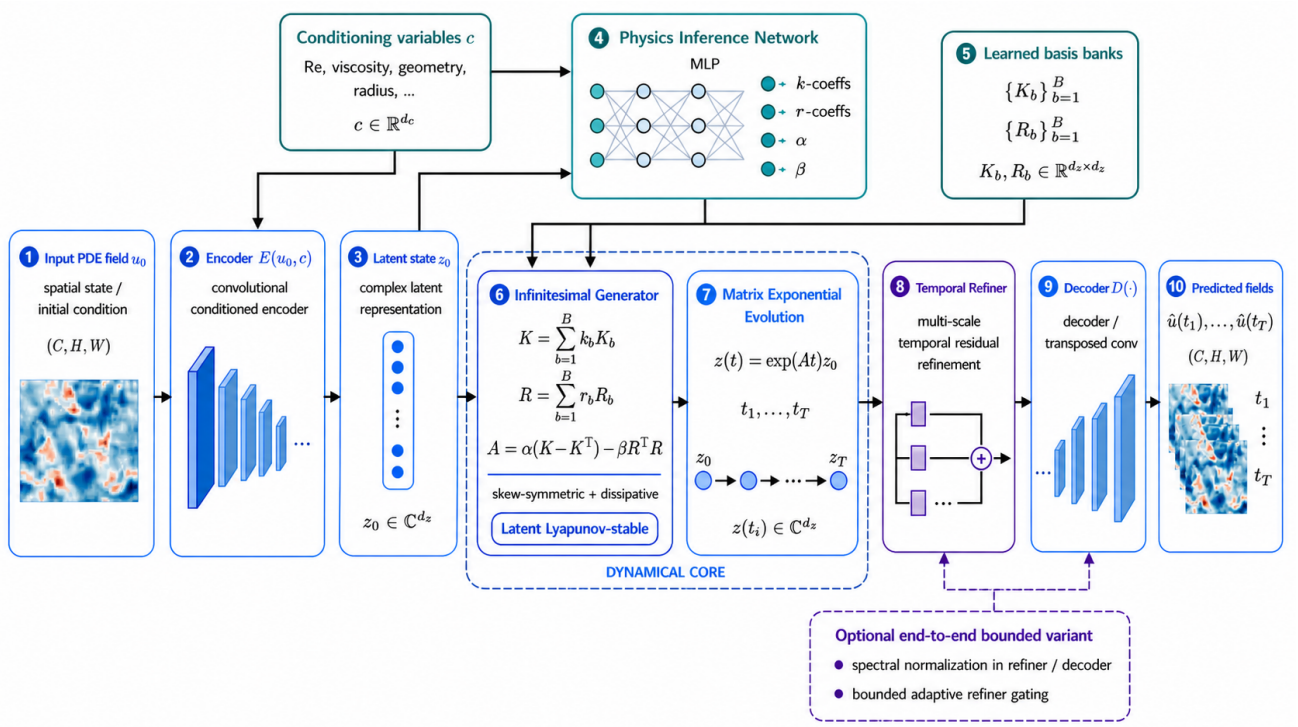


Figure 2. **NIGO architecture.** NIGO encodes an input PDE field u_0 and conditioning variables c into a complex latent state $z_0 \in \mathbb{C}^{d_z}$. A physics-inference network constructs the dissipative generator $A = \alpha(K - K^\top) - \beta R^\top R$, whose matrix-exponential flow gives Lyapunov-stable latent trajectories. A temporal refiner and decoder map these trajectories to predicted fields $\hat{u}(t_1), \ldots, \hat{u}(t_T)$.

3.1. Operator-Learning Problem

Let $u(t) \in \mathbb{R}^{C \times H \times W}$ denote a discretized PDE state at time t, with C physical channels and spatial grid size (H, W). For one-dimensional systems, we use the same notation with one spatial dimension suppressed. Each trajectory may also be associated with conditioning variables $c \in \mathbb{R}^{d_c}$, such as viscosity, Reynolds number, geometry parameters, forcing, or boundary-condition descriptors. Given an initial state $u_0 = u(0)$ and conditioning variables c, we seek to approximate the time-conditioned solution operator

$$\mathcal{G}^\star : (u_0, c, t) \mapsto u(t), \tag{2}$$

and evaluate it on a set of forecast times $\{t_i\}_{i=1}^{T}$:

$$\hat{u}_{1:T} = \big(\hat{u}(t_1), \ldots, \hat{u}(t_T)\big) \approx \big(u(t_1), \ldots, u(t_T)\big).$$

This formulation separates the learned representation of the initial condition from the mechanism used to query future times. In autoregressive neural operators, the query mechanism is usually repeated application of a finite-step map. In NIGO, future times are queried from a continuous latent trajectory generated by an infinitesimal operator.

3.2. From Finite-Step Querying to Generator-Based Querying

A conventional learned time-stepper approximates a fixed-step transition map

$$\Phi_\theta^{\Delta t} : (u_n, c) \mapsto u_{n+1}, \tag{3}$$

where $u_n \approx u(t_n)$ and $t_{n+1} = t_n + \Delta t$. A k-step prediction is then obtained by composition,

$$\hat{u}(t_{n+k}) = \big(\Phi_\theta^{\Delta t}\big)^k (u_n, c). \tag{4}$$

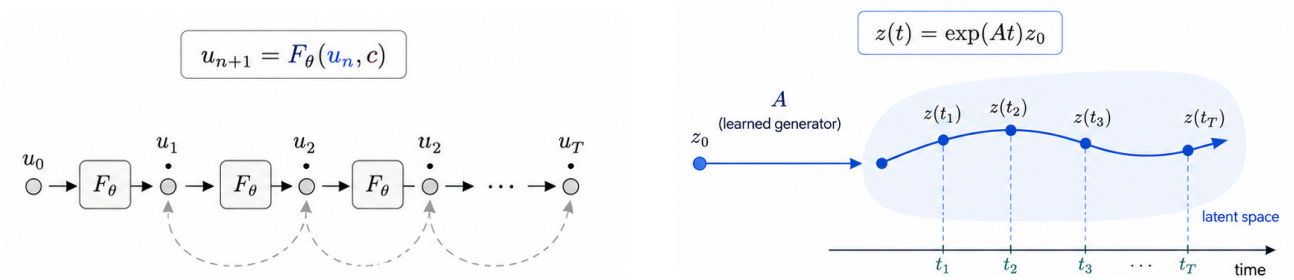
This is a discrete querying strategy: each later query depends on previous model-generated states. Consequently, local one-step errors can be repeatedly reintroduced into the input stream, which is a common source of long-horizon drift, excessive damping, frozen dynamics, or non-physical energy growth.

NIGO instead parameterizes a time-conditioned operator through a latent infinitesimal generator. It first encodes the initial condition and regime,

$$z_0 = E_\theta(u_0, c), \tag{5}$$

then constructs a generator $A = G_\phi(z_0, c)$ and evaluates future states by

$$z(t) = \exp(At) z_0, \qquad \hat{u}(t) = D_\psi(\mathcal{R}_\omega(z(t))). \tag{6}$$



(a) Autoregressive finite-step querying

(b) Generator-based continuous querying

Figure 3. **Finite-step rollout versus generator-based continuous querying.** A standard learned time-stepper obtains long-horizon predictions by repeatedly composing a finite-step map $\Phi_\theta^{\Delta t}$, so prediction errors can be fed back into subsequent queries. NIGO constructs a latent infinitesimal generator A from the initial state and conditioning variables, then queries future times along the single continuous trajectory $z(t) = \exp(At)z_0$.

The same generator-induced trajectory is queried at all requested times, rather than repeatedly applying an unconstrained transition map. Figure 3 illustrates this distinction: standard neural-operator rollout uses repeated finite-step composition, while NIGO queries a continuous latent flow induced by a structured generator.

3.3. NIGO Architecture

The complete architecture is shown in Figure 2. NIGO consists of four components: an encoder E_θ, a physics-inference network g_ϕ, a matrix-exponential latent evolution module, and a temporal refiner–decoder pair $(\mathcal{R}_\omega, D_\psi)$.

The encoder maps the initial field and conditioning variables to a complex latent state,

$$z_0 = E_\theta(u_0, c) \in \mathbb{C}^{d_z}, \tag{7}$$

where $z_0 = z_0^{\mathrm{re}} + i z_0^{\mathrm{im}}$ and $z_0^{\mathrm{re}}, z_0^{\mathrm{im}} \in \mathbb{R}^{d_z}$. Conditioning the encoder allows the latent representation to depend on both the observed field and the physical regime.

The physics-inference network predicts the coefficients of the latent generator,

$$(k, r, \alpha, \beta) = g_\phi(z_0, c), \tag{8}$$

where $k, r \in \mathbb{R}^B$ are basis coefficients and α, β scale transport and dissipation. Given the generator A, NIGO defines the latent dynamics by

$$\dot{z}(t) = Az(t), \qquad z(0) = z_0, \tag{9}$$

and evaluates the solution at requested forecast times as

$$z(t_i) = \exp(At_i)z_0, \qquad i = 1, \ldots, T. \tag{10}$$

The resulting sequence $Z = (z(t_1), \ldots, z(t_T))$ is refined and decoded:

$$\widetilde{Z} = \mathcal{R}_\omega(Z), \qquad \hat{u}(t_i) = D_\psi(\widetilde{z}(t_i)). \tag{11}$$

The refiner supplies multi-scale temporal corrections to the matrix-exponential trajectory, while the decoder maps refined latent states to grid-valued physical fields. In our grid-based experiments, D_ψ is implemented as a transposed-convolutional decoder.

3.4. Dissipative Infinitesimal Generator

The central object in NIGO is the structured infinitesimal generator. Let

$$\{K_b\}_{b=1}^B, \qquad \{R_b\}_{b=1}^B, \qquad K_b, R_b \in \mathbb{R}^{d_z \times d_z}$$

be learned real-valued basis banks. The coefficients inferred by g_ϕ form

$$K = \sum_{b=1}^B k_b K_b, \qquad R = \sum_{b=1}^B r_b R_b. \tag{12}$$

The generator is then parameterized as

$$A = \alpha(K - K^\top) - \beta R^\top R. \tag{13}$$

Throughout, K and R are real-valued matrix combinations, while z may be complex-valued. The skew-symmetric term $K - K^\top$ represents transport-like, energy-preserving dynamics. The term $-R^\top R$ is negative semidefinite and represents dissipation when $\beta \geq 0$. Thus, the basis banks provide expressivity, the physics-inference network adapts the generator to the state and conditioning variables, and the skew-plus-dissipative form prevents the symmetric part of A from injecting positive latent energy.

3.5. Training Objective and Stability

NIGO is trained by supervised sequence prediction. Given target states $u(t_1), \ldots, u(t_T)$, the default objective is

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{T} \sum_{i=1}^T \|\hat{u}(t_i) - u(t_i)\|_2^2 . \tag{14}$$

Unless otherwise stated, we use the sequence mean-squared error in Eq. (14); benchmark-specific training details are reported in Appendix B. The stability property below follows from the generator parameterization and does not rely on auxiliary losses.

Latent stability. Consider the latent energy $V(z) = \frac{1}{2}\|z\|_2^2$. For $\dot{z} = Az$, using the complex inner product with conjugate transpose z^*,

$$\dot{V}(z) = \mathrm{Re}[z^* A z]\,. \tag{15}$$

Since $K - K^\top$ is real skew-symmetric,

$$\mathrm{Re}\big[z^*(K - K^\top)z\big] = 0.$$

Since $R^\top R$ is real positive semidefinite,

$$\mathrm{Re}\big[z^* R^\top R z\big] = \|Rz\|_2^2 \geq 0.$$

Substituting Eq. (13) therefore gives

$$\dot{V}(z) = -\beta\|Rz\|_2^2 \leq 0, \qquad \beta \geq 0. \tag{16}$$

Hence the latent flow generated by A is Lyapunov-stable: latent energy cannot increase along the matrix-exponential trajectory.

This guarantee is intentionally a statement about the latent dynamical core, not a claim that decoded trajectories are exact PDE solutions. Predictive accuracy depends on the learned representation, temporal refinement, decoding, and training objective. To reduce amplification from latent to physical space, NIGO also supports a bounded variant in which spectral normalization controls the Lipschitz constants of the refiner and decoder, while adaptive gating bounds the residual refinement magnitude. Full derivations for the complex-valued latent representation and the bounded latent-to-physical mapping are provided in Appendix A.

4. Experiments

We evaluate NIGO on five PDE benchmarks spanning distinct long-time regimes: Burgers and shallow-water dynamics from PDEBench, incompressible Navier–Stokes from PDEBench, cylinder flow from CFDBench, and a generated Kuramoto–Sivashinsky dataset. These systems cover dissipative decay, wave-like transients, periodic vortex shedding, and chaotic spatiotemporal dynamics. Unless otherwise stated, all models are evaluated in long-horizon rollout. Baseline time-steppers are rolled out autoregressively, while NIGO constructs a generator from the initial condition and conditioning variables, evolves the latent state by matrix exponential, and decodes the requested future states. Additional implementation details, dataset descriptions, and rollout protocols are provided in Appendices B and C.

Table 1. **Burgers 1000-step rollout.** Baseline results use official PDEBench models/checkpoints, including PINN, FNO1d, and UNet1d (Takamoto et al., 2022). NIGO avoids the blowup and freezing modes observed in these baselines.

Model	Final residual error E_{1000}	Long-horizon behavior
PINN	0.8566	Frozen dynamics
FNO1d	0.6617	Non-physical energy growth
UNet1d	0.2282	Noisy plateau / high-frequency artifacts
NIGO	**0.0021**	Stable dissipative attractor

4.1. Long-Horizon Forecasting

Table 1 reports the 1000-step Burgers comparison against PDEBench baselines. The baselines fail in qualitatively different ways: FNO1d develops non-physical energy growth, UNet1d reaches a noisy high-frequency plateau, and PINN predictions freeze. NIGO remains stable over the full rollout and achieves substantially lower final residual error.

The result illustrates the failure mode targeted by NIGO: low short-window error does not guarantee stable long-horizon behavior when a learned update map is repeatedly composed. By replacing discrete autoregressive rollout with a dissipative continuous-time latent generator, NIGO prevents unbounded latent energy growth and empirically avoids physical-space blowup.

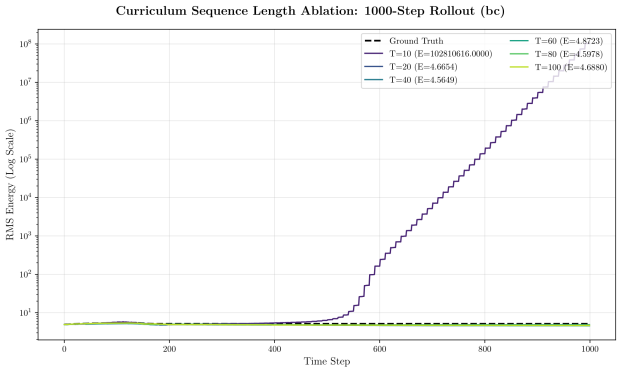
4.2. Stability Diagnostics and Ablations

We next study whether long-horizon stability depends on temporal coverage and generator structure. For cylinder flow, we train models with sequence lengths $T \in \{10, 20, 40, 60, 80, 100\}$ and evaluate each over a 1000-step rollout. Figure 4a shows RMS energy over time: $T = 10$ diverges, while all horizons $T \geq 20$ remain bounded. Table 2a reports the corresponding rollout errors and final energies. This indicates that the architecture requires enough temporal context to identify the vortex-shedding scale, but is not sensitive to a single tuned sequence length once that threshold is reached.

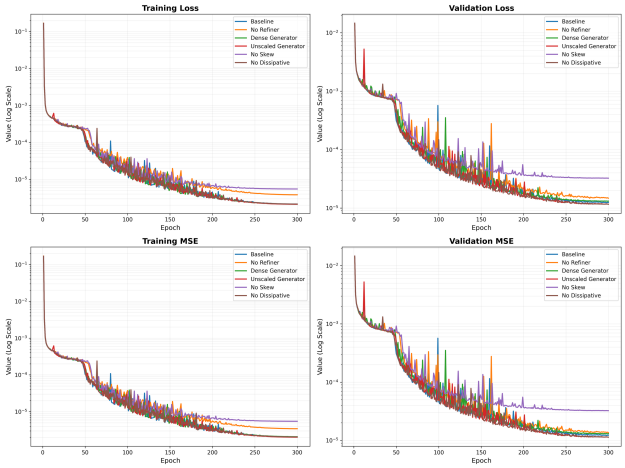
Figure 4b and Table 2b show that supervised training loss alone is not a reliable indicator of long-horizon behavior. Several ablated variants optimize to low short-window error, but structurally weakened generators fail during rollout. In particular, replacing the skew-plus-dissipative generator with a dense unconstrained generator gives short-window fit but long-rollout blowup. Removing dissipation weakens adaptive damping, while removing skew-symmetry limits transport-like dynamics. These results support the decomposition

$$A = \alpha(K - K^\top) - \beta R^\top R$$

as the mechanism responsible for stable and expressive latent evolution. Additional discussion of the sequence-length and component ablations is provided in Appendix D.



(a) Sequence-length sensitivity



(b) Component ablations

Figure 4. **Long-horizon stability diagnostics.** Left: cylinder-flow RMS energy over a 1000-step rollout; $T = 10$ diverges, while $T \geq 20$ remains bounded. Right: training and validation curves for ablation variants, showing that low short-window loss does not by itself imply stable rollout.

Table 2. **Long-horizon stability ablations.** a: cylinder-flow sequence-length ablation over a 1000-step rollout. b: component ablations showing that both transport and dissipation are required for stable long-horizon behavior.

(a) Sequence-length ablation

Horizon	1000-step MSE	Final energy	Diverged
$T = 10$	2.0969×10^{15}	very large	yes
$T = 20$	0.5098	4.6654	no
$T = 40$	0.5926	4.5649	no
$T = 60$	**0.4250**	4.8723	no
$T = 80$	0.5548	4.5978	no
$T = 100$	0.5191	4.6880	no

(b) Component ablations

Variant	Stability	Main observation
Full NIGO	Stable	Best accuracy and spectra
w/o dissipation	Marginal	Loses adaptive damping
w/o skew-symmetry	Limited	Weak advective transport
Dense generator	Diverges	Short fit, rollout blowup
Scaled generator	Stable	No explicit α/β split
w/o refiner	Stable	Higher transient error

4.3. Interpreting the Learned Generator

Finally, we ask whether the stability constraint induces physically meaningful structure rather than merely restricting capacity. We reconstruct the learned generator from the basis banks and physics-inference outputs, then analyze eigenvalues, advection–dissipation ratios, basis locality, and state-to-state parameter variation. Table 3 summarizes the main diagnostics. On Burgers, the dissipative eigenvalues follow $|\operatorname{Re}(\lambda_k)| \propto k^{2.12}$, close to the viscous scaling νk^2. On NS2D, the learned generator is approximately balanced with $\alpha/\beta \approx 1.01$, and the dissipative bases are substantially more banded than the skew-symmetric bases. On cylinder flow, α/β decreases from 0.973 to 0.672 as the training horizon increases, indicating stronger long-horizon dissipation. These diagnostics suggest that NIGO learns a stable transport–dissipation decomposition aligned with the qualitative structure of the underlying PDEs.

Overall, the experiments support the main claim: hard-constrained continuous-time latent dynamics improve long-horizon stability while preserving nontrivial PDE behavior across decaying, periodic, and chaotic regimes.

5. Limitations and Future Work

NIGO provides a structural stability guarantee for the learned latent dynamics, but the guarantee has a precise scope. The dissipative generator ensures that latent energy is non-increasing under the matrix-exponential flow; it does not imply that decoded predictions exactly follow the true PDE trajectory. Physical accuracy still depends on the learned encoder, temporal refiner, decoder, and the coverage of the training distribution. The bounded variant reduces latent-to-physical amplification through Lipschitz-controlled decoding and gated refinement, but this is a boundedness statement rather than a convergence guarantee to the true physical attractor.

The current certificate is also isotropic, using $V(z) = \|z\|_2^2$. This choice gives a simple and effective stability condition, but it may be restrictive for systems with strongly anisotropic dynamics, multiple interacting scales, or geometry-dependent stability directions. A natural ex-

Table 3. **Generator interpretation across regimes.** Learned spectra, locality, and advection–dissipation ratios vary consistently with the physical character of each dataset.

Dataset	Diagnostic	Value	Interpretation
Burgers	decay exponent	2.12	close to viscous k^2 scaling
KS	max $\mathrm{Re}(\lambda)$	-6.55×10^{-2}	stable chaotic latent flow
KS	α/β	0.408	strongly dissipative balance
NS2D	α/β	1.014	balanced advection–dissipation
NS2D	$\mathcal{B}(S), \mathcal{B}(N)$	$0.152, 0.590$	localized dissipation
Cylinder $T = 20 \to 100$	α/β	$0.973 \to 0.672$	long-horizon dissipation

tension is to learn anisotropic certificates, such as $V(z) = z^\top P z$, or more general state-dependent Lyapunov functions that encode physically meaningful stable and weakly stable directions.

A second limitation is that the present models are trained on isolated PDE families. Although the physics-inference network separates equation-level scalings from state-dependent coefficients, the experiments do not yet establish zero-shot transfer across fundamentally different equations. Extending NIGO toward cross-PDE neural operators conditioned on physical invariants, such as Reynolds number, viscosity, forcing, and geometry, is an important direction.

Finally, NIGO uses a fixed generator A for each initial condition and conditioning vector. This makes long-horizon evolution efficient and analytically tractable, but can smooth sharp nonlinear transients such as shocks, rapid regime shifts, or strongly intermittent events. Future work should consider state-dependent generators $A(z)$ with contraction-theoretic constraints, preserving stability while increasing expressivity. Extending the framework to fully three-dimensional turbulent flows is another important step, where vortex stretching, helicity, and enstrophy dynamics may require richer stable operator structure.

6. Conclusion

We presented NIGO, a continuous-time neural operator for long-horizon PDE forecasting. Instead of learning an unconstrained discrete update map and applying it autoregressively, NIGO evolves a compact latent state under a structured infinitesimal generator. The generator decomposes latent dynamics into a skew-symmetric transport component and a negative-semidefinite dissipative component, yielding non-increasing latent energy by construction. This provides a simple stability mechanism that is architectural rather than imposed through rollout training or soft regularization.

The experiments show that this structure is useful across qualitatively different PDE regimes. On dissipative, periodic, wave-like, and chaotic systems, NIGO remains stable over extended rollouts while preserving nontrivial dynamics. The ablations indicate that the skew-plus-dissipative generator is essential: unconstrained or weakened variants can fit short prediction windows but fail under long rollout. The interpretability analyses further show that the learned generators recover structure consistent with the underlying physics, including viscous spectral scaling, localized dissipation, state-invariant dissipative parameters, and regime-dependent advection–dissipation balance.

These results suggest that long-horizon neural PDE forecasting should not rely on short-term accuracy alone. Stability must be part of the learned temporal operator. NIGO provides one realization of this principle by combining neural operator learning with a hard-constrained continuous-time latent dynamical system. We view this as a step toward scientific surrogates whose long-time behavior is shaped not only by data, but also by explicit dynamical structure.

References

Boussif, O., Bengio, Y., Benabbou, L., and Assouline, D. Magnet: Mesh agnostic neural pde solver. *Advances in Neural Information Processing Systems*, 35:31972–31985, 2022.

Brandstetter, J., Worrall, D., and Welling, M. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Cao, S. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34: 24924–24940, 2021.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

David, M. and Méhats, F. Symplectic learning for hamiltonian neural networks. *Journal of Computational Physics*, 494:112495, 2023.

Feng, X., Shangguan, H., Tang, T., and Wan, X. Integral regularization pinns for evolution equations. *arXiv preprint arXiv:2503.23729*, 2025.

Goswami, S., Bora, A., Yu, Y., and Karniadakis, G. E. Physics-informed deep neural operator networks. In *Machine learning in modeling and simulation: methods and applications*, pp. 219–254. Springer, 2023.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Gupta, P., Schmid, P., Sipp, D., Sayadi, T., and Rigas, G. Mori–zwanzig latent space koopman closure for nonlinear autoencoder. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481 (2313), 2025.

Hasani, R., Lechner, M., Amini, A., Rus, D., and Grosu, R. Liquid time-constant networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7657–7666, 2021.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Karumuri, S., Graham-Brady, L., and Goswami, S. Physics-informed latent neural operator for real-time predictions of time-dependent parametric pdes. *arXiv preprint arXiv:2501.08428*, 2025.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33: 6696–6707, 2020.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

Kolter, J. Z. and Manek, G. Learning stable deep dynamics models. *Advances in neural information processing systems*, 32, 2019.

Kontolati, K., Goswami, S., Em Karniadakis, G., and Shields, M. D. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, 2024.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Lai, P., Chen, Y., and Xu, H. Dymixop: Guiding neural operator design for pdes from a complex dynamics perspective with local-global-mixing. *arXiv preprint arXiv:2508.13490*, 2025.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

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

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via deeponet based on the

universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Lusch, B., Kutz, J. N., and Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Massaroli, S., Poli, M., Park, J., Yamashita, A., and Asama, H. Dissecting neural odes. *Advances in neural information processing systems*, 33:3952–3963, 2020.

McCabe, M., Harrington, P., Subramanian, S., and Brown, J. Towards stability of autoregressive neural operators. *arXiv preprint arXiv:2306.10619*, 2023.

Rahman, M. A., Ross, Z. E., and Azizzadenesheli, K. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

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

Stachenfeld, K., Fielding, D. B., Kochkov, D., Cranmer, M., Pfaff, T., Godwin, J., Cui, C., Ho, S., Battaglia, P., and Sanchez-Gonzalez, A. Learned coarse models for efficient turbulence simulation. *arXiv preprint arXiv:2112.15275*, 2021.

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., and Niepert, M. Pdebench: An extensive benchmark for scientific machine learning. *Advances in neural information processing systems*, 35: 1596–1611, 2022.

Wang, S., Teng, Y., and Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021a.

Wang, S., Wang, H., and Perdikaris, P. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40): eabi8605, 2021b.

Wang, T. and Wang, C. Latent neural operator for solving forward and inverse pde problems. *Advances in Neural Information Processing Systems*, 37:33085–33107, 2024.

Yu, R., Herbert, M., Klein, M., and Hodzic, E. Koopman theory-inspired method for learning time advancement operators in unstable flame front evolution. *Physics of Fluids*, 37(2), 2025.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.

A. Additional Theoretical Details

The main text gives the latent Lyapunov argument in compact form. Here we provide the full derivation, clarify the complex-valued latent representation, and state the bounded latent-to-physical variant. These details support the scope of the main guarantee: NIGO enforces stability of the latent dynamical core, while physical prediction accuracy remains learned from data.

A.1. Complex-valued latent representation

NIGO represents the latent state as

$$z = x + iy, \qquad x, y \in \mathbb{R}^{d_z}.$$

For a real-valued generator $A \in \mathbb{R}^{d_z \times d_z}$, the complex system

$$\dot{z}(t) = Az(t)$$

is equivalent to the real block system

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}.$$

The latent norm decomposes as

$$\|z\|_2^2 = \|x\|_2^2 + \|y\|_2^2.$$

Thus, applying the same dissipative generator to the real and imaginary components preserves the latent energy argument used in the main text.

A.2. Full Lyapunov derivation

For each input trajectory, NIGO constructs

$$A = \alpha(K - K^\top) - \beta R^\top R, \qquad \beta \geq 0,$$

where $K, R \in \mathbb{R}^{d_z \times d_z}$. Let

$$V(z) = \frac{1}{2}\|z\|_2^2 = \frac{1}{2} z^* z.$$

Along $\dot{z} = Az$,

$$\dot{V}(z) = \frac{1}{2}\left(\dot{z}^* z + z^* \dot{z}\right) = \frac{1}{2}\left((Az)^* z + z^* Az\right) = \mathrm{Re}[z^* Az].$$

Let $S = K - K^\top$. Since $S^\top = -S$, for any complex z,

$$(z^* Sz)^* = z^* S^\top z = -z^* Sz.$$

Therefore $z^* Sz$ is purely imaginary and

$$\mathrm{Re}\big[z^*(K - K^\top)z\big] = 0.$$

For the dissipative term,

$$\mathrm{Re}\big[z^* R^\top Rz\big] = (Rz)^*(Rz) = \|Rz\|_2^2 \geq 0.$$

Combining the two terms gives

$$\dot{V}(z) = -\beta\|Rz\|_2^2 \leq 0.$$

Consequently, the matrix-exponential trajectory is non-expansive in latent energy:

$$\|z(t)\|_2 = \|\exp(At)z_0\|_2 \leq \|z_0\|_2, \qquad t \geq 0.$$

A.3. Bounded latent-to-physical mapping

The latent stability result controls the continuous-time dynamical core. It does not by itself imply that decoded predictions exactly match the true PDE trajectory. To reduce amplification from latent to physical space, the bounded variant controls the downstream maps.

Let the predicted field be

$$\hat{u}(t) = D_\psi(\mathcal{R}_\omega(\exp(At)E_\theta(u_0, c))).$$

Assume the encoder has Lipschitz constant L_E, the decoder has Lipschitz constant L_D, and the refiner satisfies

$$\|\mathcal{R}_\omega(z) - z\|_2 \leq \epsilon_{\max}\|z\|_2.$$

Since the latent flow is non-expansive,

$$\|\exp(At)E_\theta(u_0, c)\|_2 \leq \|E_\theta(u_0, c)\|_2 \leq L_E\|u_0\|_2.$$

Therefore,

$$\|\hat{u}(t)\|_2 \leq L_D(1 + \epsilon_{\max})L_E\|u_0\|_2.$$

This is a boundedness statement for the physical prediction under the bounded variant, not a convergence proof to the true PDE solution.

B. Architecture and Implementation Details

This section records implementation choices that are useful for reproducibility but too detailed for the main text.

B.1. Model components

NIGO consists of four learned components:

$$E_\theta, \qquad g_\phi, \qquad \mathcal{R}_\omega, \qquad D_\psi.$$

The encoder E_θ maps the input field and conditioning variables to a compact complex latent state. The physics-inference network g_ϕ predicts generator coefficients and transport–dissipation scales. The temporal refiner $\mathcal{R}_\omega$ applies residual corrections to the matrix-exponential latent trajectory. The decoder D_ψ maps refined latent states back to the physical grid.

B.2. Generator construction

The generator is assembled from learned basis banks:

$$K = \sum_{b=1}^{B} k_b K_b, \qquad R = \sum_{b=1}^{B} r_b R_b.$$

The dissipative scale β is constrained to be non-negative so that $-\beta R^\top R$ remains negative semidefinite. In practice, this can be enforced by a positive activation such as softplus. The transport scale α controls the magnitude of the skew-symmetric component.

B.3. Training objective

Unless otherwise stated, NIGO is trained using sequence mean-squared error:

$$\mathcal{L}_{\text{pred}} = \frac{1}{T}\sum_{i=1}^{T} \|\hat{u}(t_i) - u(t_i)\|_2^2.$$

This objective controls predictive fidelity. The latent stability property follows from the generator parameterization and does not rely on additional loss terms.

B.4. Matrix-exponential evaluation

For each input trajectory, the model constructs a generator A and evaluates

$$z(t_i) = \exp(At_i)z_0, \qquad i = 1, \ldots, T.$$

Because A is shared across queried forecast times for a given initial condition, the same structured latent operator defines the full predicted trajectory. The matrix exponential is evaluated using differentiable linear-algebra routines.

C. Dataset and Evaluation Details

The main text summarizes the evaluation protocol. Here we give additional dataset and rollout details without repeating the main-body result tables.

C.1. Datasets

We evaluate on five PDE datasets chosen to cover qualitatively distinct long-time regimes. Table 4 summarizes dataset sources and raw variables, while Table 5 reports preprocessing and rollout settings used in our code.

Burgers. We use the PDEBench Burgers file `1D_Burgers_Sols_Nu0.1.hdf5`. The loader preserves the native one-dimensional spatial grid with 1024 cells and uses a deterministic $90/10$ split over the first 5000 loaded trajectories. Training samples are sliding windows with input u_0 and 20 future states.

Cylinder flow. The cylinder-flow dataset is stored as `case*` folders containing velocity fields `u.npy` and `v.npy`. We use the two velocity channels as the state and condition the generator on $[Re, \text{radius}, \text{inlet velocity}, \text{bc type}]$. Cases are split deterministically into 90% train and 10% validation. The sequence-length ablation trains with $T \in \{10, 20, 40, 60, 80, 100\}$ and evaluates 1000-step rollout.

Kuramoto–Sivashinsky. The KS dataset is generated from ground-truth trajectories by sampling windows of length 768, consisting of 256 context steps and 512 horizon steps. The generated HDF5 file contains 40000 train windows and 10000 test windows, with system length labels $L \in \{56, 64, 72\}$. During NIGO training we load 5000 training windows and 1000 validation windows, apply temporal stride 4, downsample 512 spatial points to 64, and use $L/100$ as the first conditioning coordinate.

Shallow water. We use the PDEBench radial dam-break shallow-water file `2D_rdb_NA_NA.h5`. The loader uses 900 trajectories for training and 100 for validation, each with one scalar field on a 128×128 grid. The model receives the initial field and predicts 20 future steps; conditioning is a zero vector because no additional metadata is used by the current loader.

Incompressible Navier–Stokes. We use the PDEBench incompressible inhomogeneous Navier–Stokes HDF5 format, with velocity fields stored as `velocity` and forcing fields stored as `force`. The loader downsamples velocity fields to 256×256, normalizes them using the fixed range $[-3, 3]$, and conditions the generator on a four-dimensional force fingerprint consisting of the mean and standard deviation of the two forcing channels. The split uses all but the last HDF5 case for training and the last case for validation.

C.2. Rollout protocol and metrics

For autoregressive baselines, future states are produced by repeatedly feeding predictions back as inputs. For NIGO, the initial state and conditioning variables determine z_0 and A, after which all forecast times are obtained from the matrix-exponential latent trajectory. This distinction is central to the evaluation: baseline degradation can arise from repeated composition of a learned discrete map, whereas NIGO evaluates a continuous-time latent flow.

We use sequence mean-squared error as the primary prediction metric:

$$\text{MSE} = \frac{1}{T}\sum_{i=1}^{T} \|\hat{u}(t_i) - u(t_i)\|_2^2.$$

For long-horizon stability, we also track final-time residuals, final energy, RMS energy over rollout, and divergence. A rollout is marked as divergent if numerical blowup or nonphysical energy growth prevents meaningful comparison at the target horizon.

C.3. Baseline protocol

The Burgers comparison uses official PDEBench baselines/checkpoints for PINN, FNO1d, and UNet1d. The purpose of this comparison is not to re-tune each baseline, but to evaluate NIGO against a standardized benchmark implementation under extended rollout.

Table 4. **Dataset summary.** Public benchmark statistics are combined with the dataset variants used in our experiments.

Dataset	Source / generation	State variables	Raw resolution	Samples / trajectories	Regime
Burgers	PDEBench, `1D_Burgers_Sols_Nu0.1.hdf5`	scalar $u(x,t)$	1024 spatial cells	5000 loaded trajectories	dissipative shocks / decay
Cylinder flow	CFDBench-style `bc/case*` folders	velocity (u,v)	native case grid	all available `case*` folders	periodic vortex shedding
Kuramoto–Sivashinsky	generated KS windows	scalar $u(x,t)$	512 raw points; 64 used	40000 train / 10000 test windows	spatiotemporal chaos
Shallow water	PDEBench `2D_rdb_NA_NA.h5`	scalar height field	128×128	900 train / 100 validation trajectories	nonlinear wave transients
NS2D	PDEBench incompressible inhomogeneous NS	velocity (u,v)	downsampled to 256×256	all but last HDF5 case for train; last for validation	chaotic/turbulent flow

Table 5. **NIGO preprocessing and rollout protocol.** This table reports the preprocessing and evaluation settings used by our code.

Dataset	Split	Input context	Train horizon	Long rollout	Conditioning / normalization
Burgers	deterministic 90/10 of 5000 trajectories	u_0	20	1000	zero 4D condition; min–max normalization
Cylinder flow	deterministic 90/10 over `case*` folders	u_0	$10, 20, 40, 60, 80, 100$ in ablation	1000	$[Re, radius, inlet, bc]$; normalized fields/conditions
Kuramoto–Sivashinsky	5000 train / 1000 validation windows loaded	u_0	20	experiment-specific	$L/100$ condition; train min–max normalization
Shallow water	900/100 trajectory split	u_0	20	experiment-specific	zero 4D condition; train min–max normalization
NS2D	all but last HDF5 case for train; last for validation	u_0	20	experiment-specific	force mean/std fingerprint; fixed $[-3, 3]$ normalization

D. Additional Ablation Details

The main text reports the compact sequence-length and component-ablation summary. Here we provide additional interpretation of those ablations without repeating the main-body table or figure.

D.1. Sequence-length sensitivity

The sequence-length study separates architectural stability from temporal coverage. The $T = 10$ cylinder-flow model diverges during 1000-step evaluation, indicating that an extremely short training horizon does not capture the characteristic vortex-shedding scale. Once the horizon reaches $T \geq 20$, the rollouts remain bounded across a broad range of training sequence lengths.

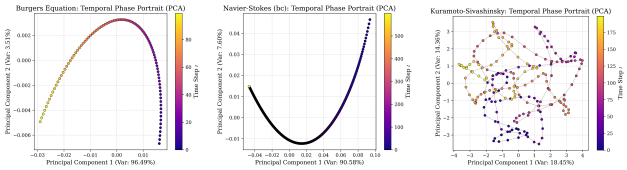
This result should not be interpreted as saying that NIGO is horizon-independent. Rather, it shows that once the training window contains enough temporal information to identify the dominant dynamics, the structured generator supports stable extrapolation beyond the training horizon.

D.2. Component ablations

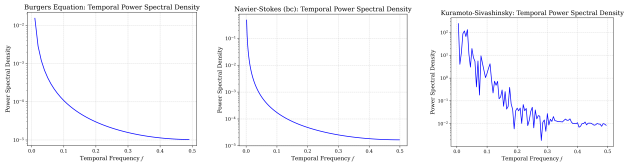
The component ablations test whether long-horizon stability comes from the generator structure or merely from optimization. The dense-generator variant can fit short windows but diverges under long rollout, indicating that short-horizon supervised error is not sufficient to characterize stability. Removing the dissipative component weakens damping, while removing the skew-symmetric component limits transport-like evolution. Removing the refiner preserves stability but increases transient error, indicating that the refiner improves fidelity rather than providing the core stability mechanism.

E. Attractor Diagnostics

The main paper reports scalar errors and stability diagnostics. This section provides additional qualitative analysis of the dynamical regimes through phase portraits and temporal power spectra. These figures are not repeated in the main body.



(a) Burgers phase portrait *(b)* Cylinder-flow phase portrait *(c)* KS phase portrait



(d) Burgers temporal PSD *(e)* Cylinder-flow temporal PSD *(f)* KS temporal PSD

Figure 5. **Attractor diagnostics across regimes.** Phase portraits and temporal spectra distinguish dissipative decay, periodic vortex shedding, and chaotic spatiotemporal behavior.

F. Additional Interpretability Results

The main body reports a compact cross-regime interpretability summary. Here we provide the full diagnostic dashboards for the learned generators.

F.1. Cylinder-flow generator diagnostics

The cylinder-flow horizon analysis shows that the learned generator remains stable across training horizons and that the advection–dissipation ratio decreases as the horizon increases. Figure 6 summarizes the spectra, stability margins, coefficient ratios, and locality statistics across horizons.

F.2. NS2D generator diagnostics

For NS2D, the reconstructed generators have stable spectra, approximately balanced advection–dissipation ratios, and localized dissipative bases. These diagnostics support the interpretation of NIGO as learning a transport–dissipation latent operator rather than a generic black-box transition.

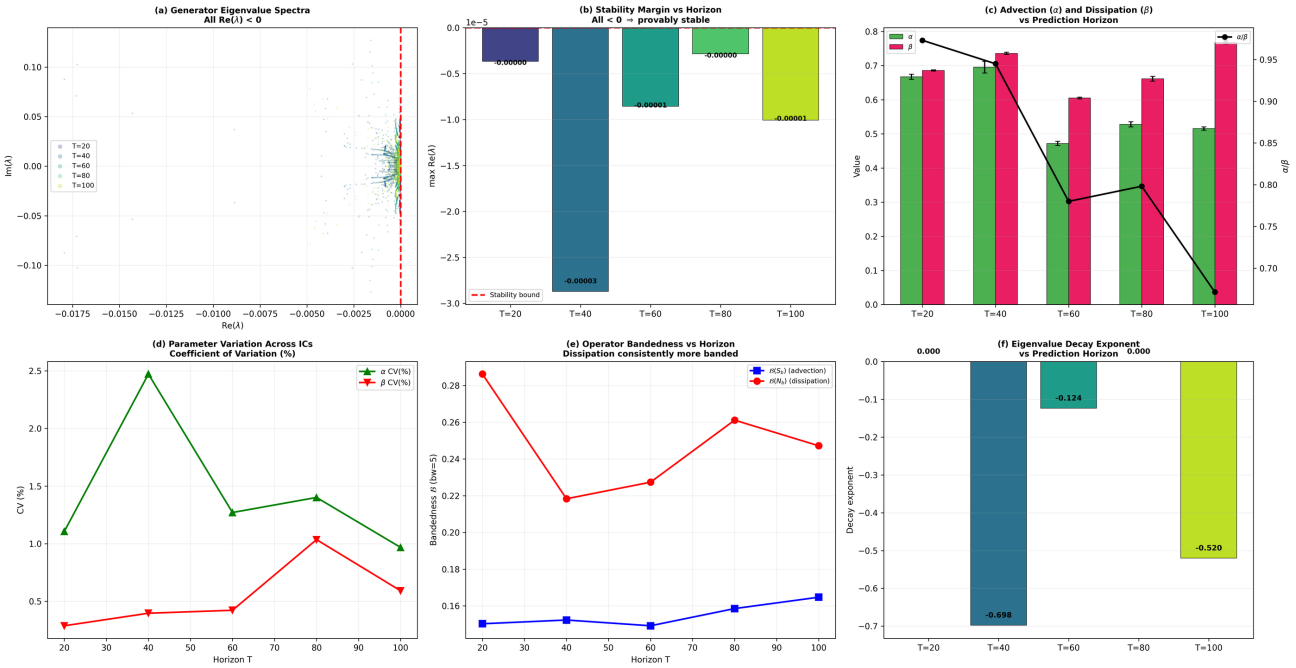


Figure 6. **Cylinder-flow generator diagnostics across horizons.** Eigenvalue spectra, stability margins, advection–dissipation ratios, and basis-locality statistics across training horizons.

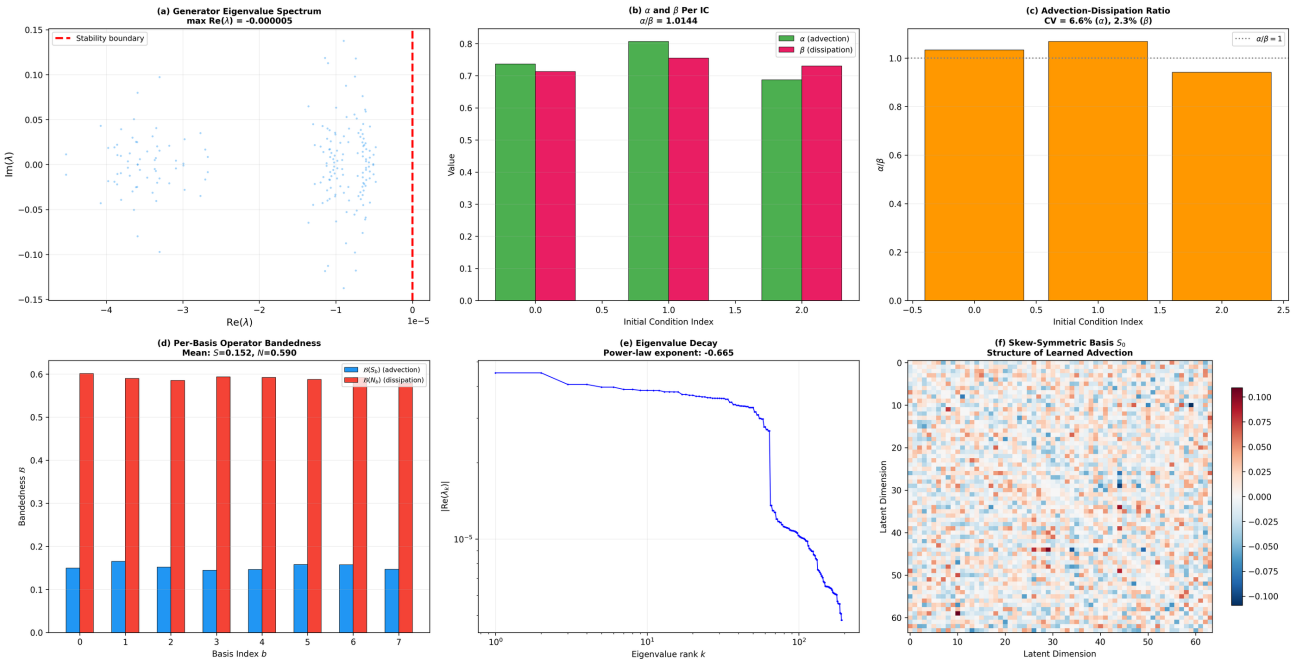


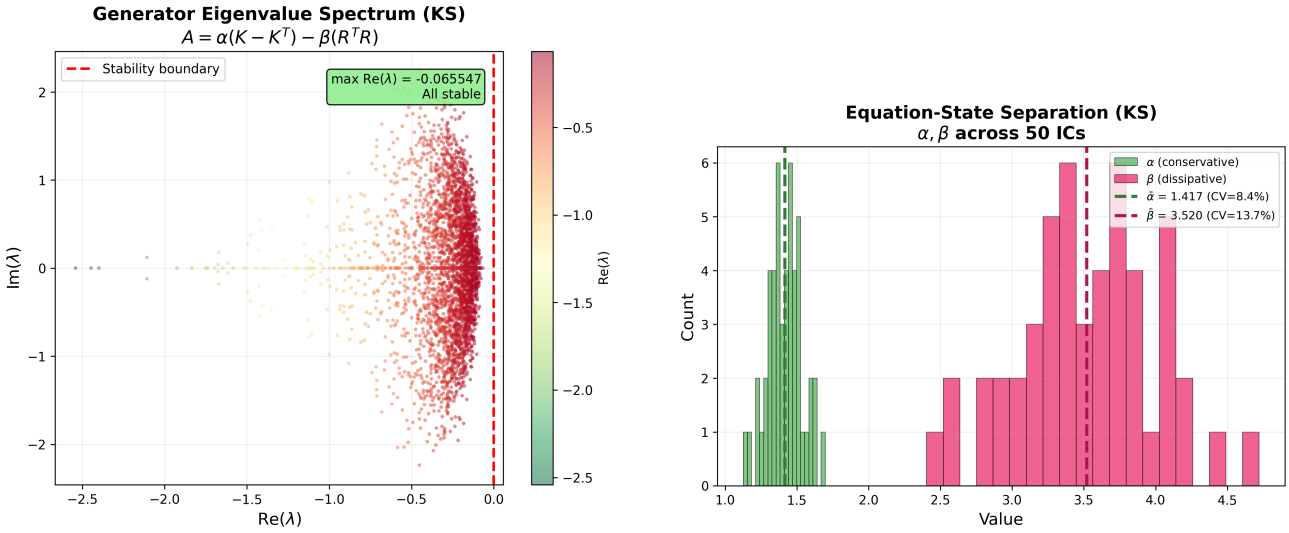
Figure 7. **NS2D generator diagnostics.** The dashboard summarizes spectra, advection–dissipation balance, basis locality, and state variation for reconstructed NS2D generators.

F.3. Kuramoto–Sivashinsky generator diagnostics

The KS generator remains stable while representing a chaotic regime. Its spectrum and advection–dissipation ratio differ from pure diffusion, consistent with the competing destabilizing and stabilizing terms in the KS equation.

F.4. Summary

Across systems, the learned generator exhibits structure aligned with the governing dynamics. Burgers recovers near-viscous spectral scaling, cylinder flow shifts toward stronger dissipation for longer horizons, NS2D shows local-



(a) Eigenvalue spectrum

(b) Advection–dissipation coefficients

Figure 8. **Kuramoto–Sivashinsky generator diagnostics.** Spectral and coefficient analyses for the learned KS generator.

ized dissipative bases, and KS retains a stable generator in a chaotic regime. These results support the view that the stability constraint shapes the learned representation rather than merely reducing model capacity.