Semigroup Consistency as a Diagnostic for Learned Physics Simulators

Lennon J. Shikhman [1]

Abstract

Learned physics simulators are often evaluated by one-step or short-horizon prediction error, but these metrics can miss failures in temporal composition and long-horizon rollout. For autonomous, state-complete systems, exact solution maps satisfy a semigroup law: direct evolution over $s + t$ should agree with evolution over s followed by t. We propose normalized semigroup error as a post hoc, model-agnostic diagnostic comparing these direct and composed learned predictions. On one-dimensional heat and Burgers dynamics with time-conditioned ConvNet and FNO baselines, semigroup error is positively associated with rollout degradation, with trajectory-level Spearman correlation $\rho = 0.635$ and 95% CI $[0.621, 0.649]$. Semigroup regularization has mixed effects, supporting semigroup consistency primarily as an evaluation diagnostic rather than a universally beneficial training objective.

1. Introduction

Learned physics simulators provide fast surrogates for PDE-governed systems such as diffusion, fluids, and waves. Neural operators and related models are especially natural for this setting because they learn maps between function spaces rather than finite-dimensional vectors (20; 15). This operator-learning viewpoint has become a standard framework for PDE surrogate modeling (9). Related operator-learning models include multipole and graph constructions (14; 13). Broader neural-operator formulations further develop this direction for scientific simulation (1). Recent work extends these ideas with physics-informed losses and physics-informed operator learning (27; 6). Physics-informed neural operators provide another route to combining operator learning with equation structure (19). Other extensions include transformer-style architectures and graph operators (17; 31). Large-scale learned simulation pipelines further demonstrate the promise of learned surrogates for physical systems (7).

Standard evaluation often emphasizes one-step or short-horizon prediction error. This is useful but insufficient: models can perform well locally while failing under rollout, time-step shift, boundary-condition shift, or broader distribution shift. These issues appear in physics-informed learning and autoregressive PDE benchmarks (10; 8). They also arise in large PDE datasets and out-of-distribution studies of learned surrogates (24; 23). Recent work on neural physics solvers likewise emphasizes out-of-distribution generalization failures (36). Recent failure-mode studies similarly show that in-distribution accuracy can obscure rollout and regime-shift errors (32). A simulator should not merely predict the next frame; it should define a coherent time-evolution rule.

For autonomous well-posed systems, the exact solution maps $\{S_t\}_{t\geq 0}$ satisfy the semigroup law (25; 5):

$$S_{t+s} = S_t \circ S_s, \qquad S_0 = I. \tag{1}$$

Thus, direct evolution by $t + s$ should agree with evolution by s followed by t. Related compositional structure appears in flow-map learning and neural ODEs (4; 26). Data-driven PDE learning in modal space also studies evolution-map structure (37). Similar ideas appear in Koopman modeling and variational Markov models (12; 21). Semigroup-aware learning makes this structure explicit in learned evolution models (3). Similar stability concerns arise in learned dissipative dynamics for chaotic systems (16).

This paper proposes *semigroup consistency* as a diagnostic for learned physics simulators. For a learned model $G_\theta(u, t) \approx S_t u$, we compare the direct prediction $G_\theta(u, s+t)$ with the composed prediction $G_\theta(G_\theta(u, s), t)$. Large disagreement indicates that the learned map is not temporally consistent, even if supervised error is small. This is aligned with concerns that neural operators can learn brittle, regime-specific, or boundary-indexed solution families rather than robust physical evolution rules (33; 22). It also connects to broader concerns in domain generalization (38).

Our claim is that semigroup error complements standard prediction metrics. It can expose rollout instability, time-step extrapolation failure, fixed-step overfitting, and disagree-

[1]Georgia Institute of Technology. Correspondence to: Lennon J. Shikhman <lj@shikhman.net>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

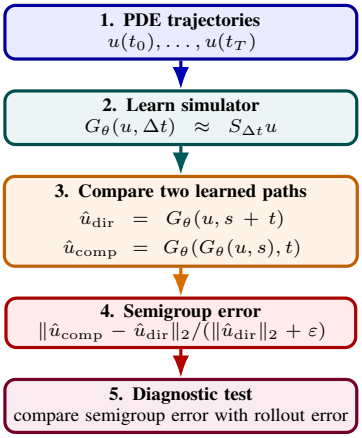

Figure 1. Evaluation pipeline for semigroup consistency. A learned simulator is trained on PDE trajectories, then evaluated by comparing direct and composed learned evolution on held-out states.

ment between direct and autoregressive simulation. Since it requires no architectural change or retraining, it can be applied to neural operators, autoregressive simulators, and continuous-time models (28; 2). It also applies to other learned surrogates used in physical forecasting (11). Empirically, we find that semigroup error is positively associated with rollout degradation on heat and Burgers dynamics, while semigroup regularization has mixed effects and does not yield a reliable aggregate rollout improvement.

Our contributions are:

1. We formulate semigroup consistency as a post hoc, model-agnostic diagnostic for learned physics simulators.

2. We define a normalized semigroup error comparing direct and composed learned evolution without requiring additional ground-truth solves.

3. We show that semigroup error is positively associated with rollout degradation and can reveal failures hidden by one-step prediction metrics.

4. We clarify the distinction between semigroup consistency as an evaluation diagnostic and semigroup consistency as a training regularizer.

2. Background and Related Work

2.1. Evolution Equations and Semigroups

Many physical simulators model autonomous, state-complete evolution. Here, autonomous means that the dynamics do not depend explicitly on absolute time, while state-complete means that the current state contains the information needed to determine future evolution. This is closely related to the Markov property, but distinct from autonomy: a system can be Markovian and non-autonomous, or autonomous but non-Markovian if the chosen state omits relevant history. Let $\mathcal{X}$ be a state space and let $S_t : \mathcal{X} \to \mathcal{X}$ map an initial state u_0 to its state after elapsed time t. Under standard well-posedness assumptions, autonomous solution maps form a one-parameter semigroup (25; 5),

$$S_{t+s} = S_t \circ S_s, \qquad S_0 = I. \tag{2}$$

This identity states that evolving for s and then t must match evolving directly for $s + t$. Thus, the semigroup view is not only functional-analytic; it is a structural condition for coherent prediction across temporal subdivisions, linking local differential descriptions to global evolution maps through the semigroup generator (25; 5).

This operator-level view also appears in data-driven dynamics. Flow-map learning directly approximates evolution maps from trajectories (26; 37). Neural ODEs define continuous-time predictions through learned differential equations (4). Koopman and variational Markov methods study learned evolution through composable operators (12; 21). Learned dissipative dynamics similarly emphasize long-time stability and composition effects (16). Deep-OSG is closest to our setting because it explicitly studies operator learning with semigroup structure and temporal partition consistency (3). Prior work uses semigroup structure to design learned evolution models (3); we instead study semigroup consistency as a post hoc diagnostic for rollout and time-step failures.

2.2. Learned Physics Simulators

Learned physics simulators replace or accelerate numerical solves with data-driven surrogates. Neural operators are central because they learn maps between function spaces rather than fixed finite-dimensional discretizations (20; 15). General neural operator theory formalizes this function-space perspective for PDEs (9; 1). DeepONet introduced branch–trunk operator approximation (20), while the Fourier Neural Operator uses global spectral mixing and is a standard baseline for parametric PDEs (15). Later variants extend operator learning through graph and multipole constructions (13; 14). Convolutional, transformer, and finite-regularity constructions further broaden neural-operator architectures (29; 17; 31). Physics-informed methods incorporate equation residuals or physical structure into training (27; 6). Physics-informed neural operators extend this idea to operator learning (19; 18).

In fluid and physical simulation, learned models support surrogate CFD and flow reconstruction (28; 2). They also support machine-learning-accelerated CFD and turbulence emulation (7; 34). Weather prediction provides another large-scale setting for learned physical surrogates (11). These

uses make long-horizon coherence important: a model may perform well at a fixed training interval but fail under rollout, unseen time steps, or shifted physical parameters. Time-conditioned and variable-time models address this by learning $G_\theta(u, \Delta t) \approx S_{\Delta t} u$ rather than a single fixed-step map (26; 37). Semigroup-aware operator learning also directly targets consistency across temporal partitions (3). This setting makes semigroup consistency directly testable by comparing direct prediction at $s + t$ with composed prediction through s and t.

2.3. Reliability Beyond One-Step Error

One-step or short-horizon prediction error is necessary but incomplete. Benchmarks such as PDEBench and APEBench broaden evaluation to rollout, resolution transfer, parameter shift, and perturbations (35; 8). The Well further expands large-scale physics simulation datasets for machine learning (24). Recent robustness studies likewise show that in-distribution accuracy may not predict behavior under changes in PDE family, parameters, boundary or terminal conditions, resolution, or rollout horizon (38; 30). Additional studies emphasize out-of-distribution generalization failures in neural PDE and physics solvers (23; 36). Boundary and terminal conditions are particularly important because varying them can turn one operator-learning problem into a family of condition-indexed operators (22).

These failures motivate structural diagnostics beyond pointwise accuracy. Known issues include physics-informed optimization pathologies (10). Neural operators can also exhibit spectral or discretization sensitivity (9; 1). Autoregressive rollout can introduce additional instability (16; 8). Semigroup error targets a complementary failure mode: disagreement between direct and composed evolution. Unlike supervised error, it does not require ground-truth solves at every composed intermediate state; unlike conservation or energy diagnostics, it is model-agnostic across autonomous evolution equations. It is not sufficient for correctness, since a wrong model can be internally consistent, but it is a lightweight test of whether a learned simulator behaves like a coherent time-evolution map. We therefore report semigroup consistency alongside one-step error, rollout error, and physics-specific checks.

3. Problem Setup

3.1. Physical Systems

We study autonomous PDEs with solution maps satisfying the semigroup law in Equation (2). The experiments use two lightweight one-dimensional systems: the heat equation and viscous Burgers equation. The heat equation,

$$\partial_t u = \nu \Delta u, \tag{3}$$

provides a stable dissipative test case. The viscous Burgers equation,

$$\partial_t u + u \partial_x u = \nu \partial_{xx} u, \tag{4}$$

adds nonlinear transport while remaining computationally simple. Both systems are evaluated with periodic boundary conditions so that the experiment isolates temporal composition rather than boundary-condition effects.

3.2. Learned Simulator

Let $S_t u$ denote the reference solution after elapsed time t. We study time-conditioned learned simulators

$$G_\theta(u, \Delta t) \approx S_{\Delta t} u, \tag{5}$$

where G_θ is instantiated as either a time-conditioned convolutional network or a compact Fourier Neural Operator. The same learned map can be queried at different time increments, which makes it possible to compare direct evolution $G_\theta(u, s + t)$ with composed evolution $G_\theta(G_\theta(u, s), t)$.

3.3. Training Objective

The base objective is supervised prediction against reference trajectories:

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}_{(u,t)} \left[\| G_\theta(u, t) - S_t u \|_2^2 \right]. \tag{6}$$

This measures direct accuracy but does not enforce temporal composition. For the semigroup-regularized ablation, we add

$$\begin{aligned} \mathcal{L}_{\text{sg}}(\theta) = \mathbb{E}_{(u,s,t)} \big[\| & G_\theta(G_\theta(u, s), t) \\ & - G_\theta(u, s + t) \|_2^2 \big], \end{aligned} \tag{7}$$

and train with

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{pred}}(\theta) + \lambda_{\text{sg}} \mathcal{L}_{\text{sg}}(\theta). \tag{8}$$

The regularized variant is not the main contribution; it is included to distinguish semigroup consistency as an evaluation diagnostic from semigroup consistency as a training objective.

4. Semigroup Consistency Diagnostic

4.1. Definition

For an autonomous, state-complete system, temporal evolution is not an arbitrary collection of input-output maps: the maps must compose according to the semigroup law. This gives a simple theoretical test for learned simulators. If $G_\theta(u, t)$ approximates $S_t u$, then the two learned paths

$$G_\theta(u, s + t) \qquad \text{and} \qquad G_\theta(G_\theta(u, s), t)$$

should agree. Their discrepancy measures whether the learned simulator behaves like a coherent time-evolution map.

Definition 1 (Semigroup error)**.** *Given a learned simulator $G_\theta : \mathcal{X} \times \mathbb{R}_+ \to \mathcal{X}$, define*

$$\mathcal{E}_{\mathrm{sg}}(u; s, t) = \frac{\|G_\theta(G_\theta(u, s), t) - G_\theta(u, s + t)\|_2}{\|G_\theta(u, s + t)\|_2 + \varepsilon}. \quad (9)$$

The normalization makes the metric comparable across states and times with different magnitudes. The small constant $\varepsilon > 0$ prevents instability when the predicted state norm is close to zero, which can occur in dissipative systems. A small value of $\mathcal{E}_{\mathrm{sg}}$ does not prove physical correctness, but a large value indicates that the learned model violates a necessary composition property of autonomous evolution.

4.2. Diagnostic Protocol

Given a trained simulator, we sample held-out states u and evaluation time pairs (s, t) such that $s, t > 0$ and $s + t$ lies within the trajectory horizon. For each pair, we compute the direct prediction

$$\widehat{u}_{\mathrm{dir}} = G_\theta(u, s + t)$$

and the composed prediction

$$\widehat{u}_{\mathrm{comp}} = G_\theta(G_\theta(u, s), t).$$

The diagnostic is the normalized discrepancy between these two predictions, averaged over states, time pairs, random seeds, systems, and model classes.

We evaluate two regimes. Seen time pairs use increments similar to those used during training. Unseen time pairs use larger or less common decompositions. This distinction matters because a model may learn accurate local interpolation without learning a consistent family of time-indexed maps. In that case, semigroup error should remain small on familiar increments but increase under unseen temporal decompositions.

We use semigroup error in two ways. First, as a post hoc diagnostic, it measures whether a trained simulator is internally consistent. Second, as an association test, it evaluates whether composition inconsistency is statistically linked to downstream failures such as rollout drift or time-step extrapolation error. We therefore compare $\mathcal{E}_{\mathrm{sg}}$ with one-step error and long-horizon rollout error, and report correlations between per-trajectory semigroup error and rollout degradation.

4.3. Failure Modes

Semigroup error is designed to flag failure modes suggested by the structure of autonomous evolution. The guiding principle is that if a learned model does not approximately satisfy a necessary composition law, then its repeated use as a simulator is suspect even when its local prediction error is small.

Fixed-step overfitting. A model trained mainly at one time increment can learn a good approximation to $S_{\Delta t}$ without learning the full family $\{S_t\}_{t \geq 0}$. Theoretical consistency requires $S_{2\Delta t} = S_{\Delta t} \circ S_{\Delta t}$, and more generally $S_{s+t} = S_t \circ S_s$. If $G_\theta(u, 2\Delta t)$ and $G_\theta(G_\theta(u, \Delta t), \Delta t)$ disagree, the model has likely learned time-step-specific interpolation rather than a coherent evolution rule.

Rollout drift. Autoregressive simulation repeatedly feeds model outputs back into the model. If direct and composed predictions disagree, then the model's own predictions move it onto states where later updates follow a different effective dynamics. This provides a theoretical reason for high semigroup error to be associated with long-horizon rollout drift.

Excessive damping. In dissipative systems, a learned model may reduce short-horizon error by over-smoothing high-frequency content. Such a model can appear stable, but it may not preserve the correct relationship between short and long time evolution. Semigroup error can detect this when repeated small-step damping does not match direct large-step damping.

Temporal aliasing. A time-conditioned model may encode different effective dynamics at different queried time increments. Then $G_\theta(u, s + t)$ and $G_\theta(G_\theta(u, s), t)$ behave as if they came from different learned simulators. This failure is especially likely under unseen time pairs, where the model must extrapolate its representation of elapsed time.

Boundary of the diagnostic. These failure predictions are one-way. Large semigroup error is evidence of temporal inconsistency, but small semigroup error is not a certificate of correctness. A model can be compositionally consistent and still solve the wrong equation. Semigroup error may reveal this mismatch when repeated small-step smoothing does not agree with direct large-step smoothing, although energy or spectral diagnostics are needed to identify damping errors.

5. Experiments

5.1. Research Questions

We evaluate whether semigroup consistency is a useful diagnostic for learned simulators through five questions:

1. **Correlation:** Does semigroup error correlate with long-horizon rollout error across trajectories and model variants?
2. **Time-composition shift:** Does semigroup error increase on larger unseen composition pairs (s, t)?
3. **Regularization:** Does semigroup regularization reduce semigroup error relative to prediction-only training?

4. **Predictive utility:** Does reduced semigroup error improve rollout error, or does it mainly improve internal consistency?

5. **Beyond one-step error:** Can models with similar one-step error differ materially in semigroup and rollout behavior?

5.2. Datasets and Splits

We use synthetic trajectories from two one-dimensional PDE systems with periodic boundary conditions: the heat equation and viscous Burgers equation. Initial conditions are sampled as random Fourier series with decaying amplitudes. Trajectories are generated on a uniform spatial grid and saved at uniformly spaced times in $[0, 1]$.

The experiments use grid size $N_x = 128$, 21 saved time points, and $128/32/64$ train/validation/test trajectories. The heat equation uses viscosity $\nu = 0.05$, while Burgers uses $\nu = 0.02$. Out-of-distribution tests include unseen larger composition pairs (s, t), a shifted initial-condition spectrum, and a Burgers viscosity shift to $\nu = 0.03$. Splits are performed by trajectory to avoid leakage between training and testing time pairs.

Table 1. Dataset configuration.

System	Grid	Times	Test shifts
Heat, $\nu = 0.05$	128	21	unseen (s, t); IC spectrum
Burgers, $\nu = 0.02$	128	21	unseen (s, t); IC spectrum; $\nu \to 0.03$

5.3. Models

We evaluate two compact neural simulator families. The first is a time-conditioned residual ConvNet with one-dimensional convolutional blocks and sinusoidal time embeddings. The second is a compact one-dimensional Fourier Neural Operator (FNO1D) with truncated spectral convolutions, residual 1×1 paths, and the same time-conditioning mechanism.

For each family, we train a prediction-only baseline and a semigroup-regularized variant:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{sg}} \mathcal{L}_{\text{sg}}, \qquad \lambda_{\text{sg}} = 0.01.$$

5.4. Metrics

All primary errors use normalized relative L^2:

$$\text{relL2}(a, b) = \frac{\|a - b\|_2}{\|b\|_2 + \varepsilon}.$$

We report one-step error, rollout AUC error, final-time rollout error, seen semigroup error, unseen semigroup error, and Spearman correlation between per-trajectory semigroup error and rollout AUC.

Table 2. Model families.

Model	Time handling	Purpose
TC-Conv	$\Delta t/T$ embedding	prediction-only local baseline
TC-Conv + SG	same	semigroup-regularized local model
FNO1D	$\Delta t/T$ embedding	operator-style baseline
FNO1D + SG	same	architecture-agnostic SG test

The semigroup error is

$$\mathcal{E}_{\text{sg}}(u; s, t) = \frac{\|G_\theta(G_\theta(u, s), t) - G_\theta(u, s + t)\|_2}{\|G_\theta(u, s + t)\|_2 + \varepsilon}.$$

Seen semigroup error averages this quantity over training-style time pairs. Unseen semigroup error averages it over larger composition pairs.

All reported means use bootstrap 95% confidence intervals with 2000 resamples. We also report paired bootstrap intervals for baseline versus semigroup-regularized differences, paired Wilcoxon signed-rank tests when available, and Cohen's d effect sizes. The default run uses 5 random seeds.

5.5. Main Results

Across all held-out trajectory evaluations, systems, architectures, and training variants, unseen semigroup error is positively associated with rollout AUC error, with global Spearman correlation

$$\rho = 0.635, \qquad 95\% \text{ CI} = [0.621, 0.649].$$

This supports the central diagnostic claim: larger composition inconsistency is associated with worse long-horizon rollout behavior.

Table 3 reports in-distribution means by system, model, and training variant. One-step error and rollout error are not in one-to-one correspondence. In particular, FNO1D variants generally achieve lower one-step error than TC-Conv variants, but semigroup and rollout behavior still vary across systems and regularization settings. Semigroup regularization also does not produce a reliable aggregate improvement: the paired baseline-minus-SG difference in unseen semigroup error is 0.0017 with CI $[-0.0123, 0.0156]$, while the paired difference in rollout AUC is -0.0633 with CI $[-0.2408, 0.1171]$. Overall, the results support semigroup error more strongly as a diagnostic than as a universally beneficial regularizer.

5.6. Ablations

We study two compact ablations. First, we compare semigroup error on seen and unseen composition pairs to test

Table 3. In-distribution quantitative results. Values are means; SG denotes unseen semigroup error.

System	Model	1-step	Rollout	SG	ρ
Burgers	FNO1D	0.066	0.380	0.058	0.158
Burgers	FNO1D+SG	0.062	0.350	0.055	0.247
Burgers	TC-Conv	0.118	0.967	0.136	0.414
Burgers	TC-Conv+SG	0.131	0.764	0.156	0.592
Heat	FNO1D	0.081	0.585	0.093	0.590
Heat	FNO1D+SG	0.079	0.956	0.093	0.513
Heat	TC-Conv	0.145	0.814	0.102	0.142
Heat	TC-Conv+SG	0.146	0.928	0.078	0.260

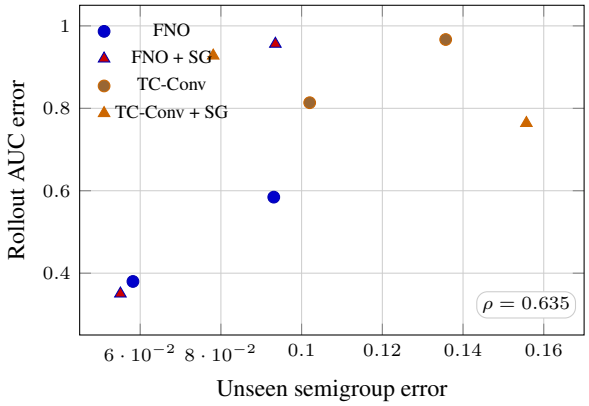

Figure 2. Relationship between unseen semigroup error and rollout AUC error across systems, architectures, and training variants. Each point shows a model-level mean, while the annotated ρ reports the global trajectory-level Spearman correlation computed across all held-out evaluations.

whether temporal inconsistency becomes more visible under time-composition shift. Averaged across all evaluated settings, unseen semigroup error exceeds seen semigroup error by 0.0165, indicating that larger unseen compositions expose additional inconsistency.

Second, we compare prediction-only and semigroup-regularized training to test whether directly minimizing $\mathcal{L}_{\text{sg}}$ improves downstream behavior. In aggregate, this effect is weak. The paired baseline-minus-SG difference in unseen semigroup error is 0.0017 with 95% CI $[-0.0123, 0.0156]$, and the paired baseline-minus-SG difference in rollout AUC is -0.0633 with 95% CI $[-0.2408, 0.1171]$. Both intervals contain zero, so the regularization effect is not statistically reliable at the aggregate level.

These ablations distinguish diagnostic value from regularization value. Semigroup error reveals temporal inconsistency under unseen compositions, but explicitly minimizing the semigroup penalty does not reliably improve rollout behavior in aggregate.

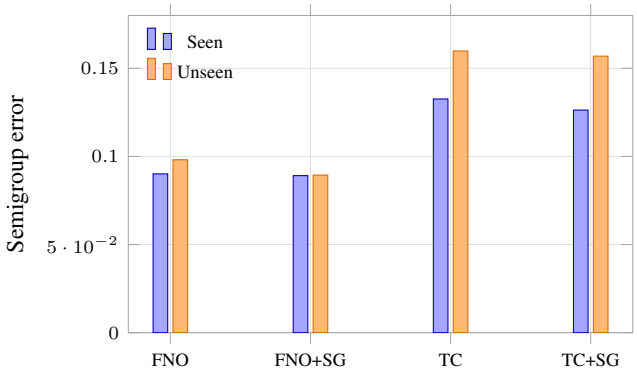

Figure 3. Seen and unseen semigroup error averaged across evaluated regimes. Bars show model-level means. Unseen composition pairs generally increase semigroup error, consistent with time-composition shift.

6. Discussion

The results support semigroup error primarily as a diagnostic for learned physics simulators. Across systems, architectures, and variants, higher unseen semigroup error is associated with worse rollout behavior. This suggests that temporal composition provides information not captured by one-step prediction error alone. A model may approximate short-time targets well while still failing to represent a coherent family of time-indexed evolution maps.

The regularization results are more mixed. Adding $\mathcal{L}_{\text{sg}}$ does not yield a reliable aggregate improvement in either semigroup error or rollout AUC. This distinction is important: semigroup consistency can be useful for evaluation even when directly optimizing a simple semigroup penalty is not sufficient to improve learned dynamics. In this sense, the diagnostic value of semigroup error is better supported here than its use as a universal training objective.

Semigroup error should also be interpreted as a one-way warning signal. Large semigroup error indicates that direct and composed learned evolution disagree, which violates a necessary structural property of autonomous evolution. Small semigroup error, however, does not imply that the learned model is physically correct. A model can be internally consistent while solving the wrong equation, using the wrong dissipation rate, or failing under parameter and boundary shifts. Semigroup consistency is therefore best reported alongside one-step error, rollout error, and physics-specific diagnostics.

7. Limitations

This study uses compact one-dimensional heat and Burgers systems. These settings isolate temporal composition cleanly, but they do not establish that the same diagnostic behaves identically for high-dimensional fluids, turbu-

lent flows, complex geometries, or multi-physics simulators. Extending the evaluation to larger PDE benchmarks is an important next step.

The diagnostic is most appropriate for autonomous, state-complete systems. If the dynamics are non-autonomous, controlled, stochastic, history-dependent, or driven by changing boundary conditions, the one-parameter semigroup law is not the right object without modification. In those cases, the diagnostic should be replaced by an evolution-family or cocycle consistency condition.

Semigroup consistency is not a certificate of correctness. It measures internal temporal coherence of the learned map, not agreement with the true physical dynamics. The normalization in $\mathcal{E}_{\mathrm{sg}}$ can also become delicate for strongly dissipative systems when state norms are small. Finally, for chaotic systems, state-space rollout error may become less meaningful at long horizons, and semigroup diagnostics should be paired with statistical or distributional measures of physical behavior.

8. Conclusion

We proposed semigroup consistency as a post hoc diagnostic for learned physics simulators. The diagnostic compares direct learned evolution over $s + t$ with composed learned evolution over s followed by t, producing a normalized semigroup error that can be evaluated without additional ground-truth solves. Across heat and Burgers experiments, semigroup error is positively associated with rollout degradation, supporting its use as a structural warning signal beyond one-step prediction error. Semigroup regularization has mixed effects and does not produce a reliable aggregate rollout improvement, so the evidence supports semigroup consistency more strongly as an evaluation diagnostic than as a universal training objective. Future work should test this diagnostic on higher-dimensional PDEs, more realistic physics simulators, and richer composition-aware training methods.

Impact Statement

This work aims to improve reliability evaluation for learned physics simulators by identifying temporal inconsistency that may be missed by one-step prediction error. Semigroup error is a lightweight diagnostic, not a certification method: a model can compose consistently while still being physically wrong. In high-stakes scientific or engineering settings, it should be used alongside numerical benchmarks, uncertainty quantification, physical diagnostics, and expert review.

Acknowledgements

Reproducibility. Code to reproduce all experiments is available at https://github.com/lennonshikhman/semigroups-for-physics/.

Computational Resources. The author gratefully acknowledges Dell Technologies, and in particular the Dell Pro Max workstation team, for providing computational resources that supported the experiments in this work. Experiments were conducted on a Dell Pro Max T2 workstation equipped with an Intel Core Ultra 9 285K processor, 128 GB of DDR5 ECC memory, and an NVIDIA RTX PRO 6000 Blackwell GPU. The views and conclusions expressed herein are those of the author and do not necessarily reflect the views of Dell Technologies.

References

[1] Azizzadenesheli, K., Kovachki, N., Li, Z., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6:320–328, 2024. doi: 10.1038/s42254-024-00712-5.

[2] Brunton, S. L., Noack, B. R., and Koumoutsakos, P. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52 (1):477–508, January 2020. ISSN 1545-4479. doi: 10.1146/annurev-fluid-010719-060214. URL http://dx.doi.org/10.1146/annurev-fluid-010719-060214.

[3] Chen, J. and Wu, K. Deep-osg: Deep learning of operators in semigroup. *Journal of Computational Physics*, 493:112498, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112498. URL https://www.sciencedirect.com/science/article/pii/S0021999123005934.

[4] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[5] Evans, L. C. *Partial Differential Equations*. American Mathematical Society, 2 edition, 2010.

[6] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021. doi: 10.1038/s42254-021-00314-5.

[7] 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. doi: 10.1073/pnas.2101784118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2101784118.

[8] Koehler, F., Niedermayr, S., Westermann, R., and Thuerey, N. Apebench: A benchmark for autoregressive neural emulators of pdes, 2024. URL https://arxiv.org/abs/2411.00180.

[9] 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. URL http://jmlr.org/papers/v24/21-1524.html.

[10] Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021. doi: 10.48550/arXiv.2109.01050. URL https://arxiv.org/abs/2109.01050.

[11] Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., Miele, A., Kashinath, K., and Anandkumar, A. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, PASC '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701900. doi: 10.1145/3592979.3593412. URL https://doi.org/10.1145/3592979.3593412.

[12] Kutz, J. N., Proctor, J. L., and Brunton, S. L. Koopman theory for partial differential equations, 2016. URL https://arxiv.org/abs/1607.07076.

[13] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Graph kernel network for partial differential equations, 2020. URL https://arxiv.org/abs/2003.03485.

[14] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Multipole graph neural operator for parametric partial differential equations, 2020. URL https://arxiv.org/abs/2006.09535.

[15] Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

[16] Li, Z., Liu-Schiaffini, M., Kovachki, N., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Learning dissipative dynamics in chaotic systems, 2022. URL https://arxiv.org/abs/2106.06898.

[17] Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

[18] 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, 2023. URL https://arxiv.org/abs/2111.03794.

[19] 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, 2023. URL https://arxiv.org/abs/2111.03794.

[20] 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. doi: 10.1038/s42256-021-00302-5. URL https://doi.org/10.1038/s42256-021-00302-5.

[21] Mardt, A., Pasquali, L., Wu, H., and Noé, F. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5, 2018. doi: 10.1038/s41467-017-02388-1.

[22] Mousavi, S., Mishra, S., and Lorenzis, L. D. Imposing boundary conditions on neural operators via learned function extensions, 2026. URL https://arxiv.org/abs/2602.04923.

[23] Nguyen, B. D. and Sandfeld, S. Out-of-distribution generalization of deep-learning surrogates for 2d pde-generated dynamics in the small-data regime, 2026. URL https://arxiv.org/abs/2601.08404.

[24] Ohana, R., McCabe, M., Meyer, L., Morel, R., Agocs, F. J., Beneitez, M., Berger, M., Burkhart, B., Burns, K., Dalziel, S. B., Fielding, D. B., Fortunato, D., Goldberg, J. A., Hirashima, K., Jiang, Y.-F., Kerswell, R. R., Maddu, S., Miller, J., Mukhopadhyay, P., Nixon, S. S., Shen, J., Watteaux, R., Blancard, B. R.-S., Rozet, F., Parker, L. H., Cranmer, M., and Ho, S. The well: a large-scale collection of diverse physics simulations for machine learning, 2025. URL https://arxiv.org/abs/2412.00568.

[25] Pazy, A. *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Applied Mathematical Sciences. Springer New York, 2012. URL https://books.google.com/books?id=DQvpBwAAQBAJ.

[26] Qin, T., Wu, K., and Xiu, D. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395: 620–635, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.06.042. URL https://www.sciencedirect.com/science/article/pii/S0021999119304504.

[27] Raissi, M., Perdikaris, P., and Karniadakis, G. 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. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[28] Raissi, M., Yazdani, A., and Karniadakis, G. E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020. doi: 10.1126/science.aaw4741.

[29] Raonic, B., Molinaro, R., Ryck, T. D., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bezenac, E. Convolutional neural operators for robust and accurate learning of PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MtekhXRP4h.

[30] Setinek, P., Galletti, G., Gross, T., Schnürer, D., Brandstetter, J., and Zellinger, W. SIMSHIFT: A benchmark for adapting neural surrogates to distribution shifts, 2025. URL https://openreview.net/forum?id=Eo4cRmb1yn.

[31] Shih, B., Peyvan, A., Zhang, Z., and Karniadakis, G. E. Transformers as neural operators for solutions of differential equations with finite regularity. *Computer Methods in Applied Mechanics and Engineering*, 434:117560, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117560. URL https://www.sciencedirect.com/science/article/pii/S0045782524008144.

[32] Shikhman, L. Diagnosing failure modes of neural operators across diverse PDE families. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=0S1LWZHQYn.

[33] Shikhman, L. One operator to rule them all? on boundary-indexed operator families in neural PDE solvers. In *AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations*, 2026. URL https://openreview.net/forum?id=lDjWQ9UxRy.

[34] Stachenfeld, K., Fielding, D. B., Kochkov, D., Cranmer, M., Pfaff, T., Godwin, J., Cui, C., Ho, S., Battaglia, P., and Sanchez-Gonzalez, A. Learned simulators for turbulence. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=msRBojTz-Nh.

[35] Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., and Niepert, M. PDEBench: An extensive benchmark for scientific machine learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=dh_MkX0QfrK.

[36] Wei, Z., Ooi, C. C., Wong, J. C., Gupta, A., Chiu, P.-H., and Ong, Y.-S. Out-of-distribution generalization for neural physics solvers, 2026. URL https://arxiv.org/abs/2601.19091.

[37] Wu, K. and Xiu, D. Data-driven deep learning of partial differential equations in modal space. *Journal of Computational Physics*, 408: 109307, 2020. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2020.109307. URL https://www.sciencedirect.com/science/article/pii/S0021999120300814.

[38] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022. ISSN 1939-3539. doi: 10.1109/tpami.2022.3195549. URL http://dx.doi.org/10.1109/TPAMI.2022.3195549.

A. Numerical Solvers

All reference trajectories are generated before training using deterministic numerical solvers. The experiments use two one-dimensional systems: the heat equation and the viscous Burgers equation, both with periodic boundary conditions. Periodic domains simplify the solver implementation and avoid boundary-condition ambiguity, which is useful because the present work studies temporal composition rather than boundary generalization.

Initial conditions are sampled as random Fourier series with controlled smoothness. For the heat equation, trajectories are generated with viscosity $\nu = 0.05$. For viscous Burgers dynamics, trajectories are generated with viscosity

$\nu = 0.02$, with an additional shifted-viscosity test regime using $\nu = 0.03$. The solvers use periodic finite-difference derivatives and a sufficiently small internal time step relative to the saved trajectory spacing. The implementation checks that generated trajectories remain finite and contain no NaNs or infinities.

In the main experiments, saved trajectories have shape

$$[N, T, X],$$

where N is the number of trajectories, T is the number of saved times, and X is the spatial grid size. We use $X = 128$, $T = 21$, and $128/32/64$ train/validation/test trajectories. Datasets are split by initial condition rather than by sampled time pairs, preventing leakage between training and testing trajectories. Shifted test regimes include unseen time-pair compositions, shifted Burgers viscosity, and shifted initial-condition spectra.

B. Model Details

The experiments use compact one-dimensional learned simulators of the form

$$G_\theta(u, \Delta t) \approx S_{\Delta t} u.$$

The two model families are a time-conditioned convolutional residual network and a compact one-dimensional Fourier Neural Operator (FNO). Both models receive the current state u and a normalized time increment $\Delta t/T$. Time conditioning is implemented using sinusoidal time embeddings injected into the model layers.

The convolutional baseline consists of one-dimensional convolutional residual blocks. The FNO baseline uses spectral convolution layers with a fixed number of retained Fourier modes, followed by pointwise channel mixing. Both architectures use residual prediction, returning the input state plus a learned update.

For each model family, we train two variants. The baseline minimizes only the supervised prediction loss,

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}_{(u,t)} \left[\|G_\theta(u, t) - S_t u\|_2^2 \right].$$

The semigroup-regularized variant minimizes

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{pred}}(\theta) + \lambda_{\text{sg}} \mathcal{L}_{\text{sg}}(\theta),$$

where

$$\mathcal{L}_{\text{sg}}(\theta) = \mathbb{E}_{(u,s,t)} \left[\|G_\theta(G_\theta(u, s), t) - G_\theta(u, s + t)\|_2^2 \right].$$

The main experiments use $\lambda_{\text{sg}} = 0.01$.

All models are trained with AdamW, gradient clipping, and deterministic random seeds where possible. The reported experiments use five seeds and bootstrap 95% confidence intervals with 2000 resamples. The implementation automatically uses CUDA when available and otherwise falls back to CPU.

C. Non-Autonomous Extensions

The semigroup identity used in the main paper applies most directly to autonomous, state-complete systems. Autonomy means that the evolution rule does not depend explicitly on absolute time. State-completeness means that the current state contains the information needed to determine future evolution. Many physical systems are instead non-autonomous or not state-complete: the dynamics may depend explicitly on time, external controls, changing boundary conditions, moving geometry, stochastic forcing, or history-dependent material response. In these cases, the one-parameter semigroup law

$$S_{t+s} = S_t \circ S_s$$

is generally not the correct structure.

For non-autonomous dynamics, the appropriate object is an evolution family. Let $S(t, r)$ denote the map that evolves a state from time r to time t. The composition law becomes

$$S(t, r) = S(t, s) \circ S(s, r), \qquad r \leq s \leq t.$$

A learned non-autonomous simulator should therefore be written as

$$G_\theta(u, r, t) \approx S(t, r)u,$$

or, when controls or forcing are present,

$$G_\theta(u, r, t, c_{[r,t]}) \approx S_c(t, r)u,$$

where $c_{[r,t]}$ denotes the control, forcing, or boundary information over the interval.

The corresponding diagnostic compares direct and composed evolution:

$$\mathcal{E}_{\text{evol}}(u; r, s, t) = \frac{\|G_\theta(G_\theta(u, r, s), s, t) - G_\theta(u, r, t)\|_2}{\|G_\theta(u, r, t)\|_2 + \varepsilon}.$$

This reduces to the semigroup diagnostic in the autonomous case when $S(t, r) = S_{t-r}$. For controlled or externally forced systems, the diagnostic must also ensure that both direct and composed predictions condition on consistent forcing information over the same interval. Otherwise, disagreement may reflect mismatched inputs rather than failure of the learned dynamics.

Thus, semigroup consistency should not be applied blindly to every simulator. It is best viewed as the autonomous case of a broader composition-consistency principle: a learned simulator should agree with itself when the same physical evolution is represented either as one interval or as a composition of subintervals.