LEARNING MULTIPLICATIVE-TIME OPERATORS FOR LONG-HORIZON PDE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon prediction of dissipative PDEs challenges learned fixed-step solvers: solutions change scale while rollout length grows with physical time. We formulate prediction as learning a multiplicative-time evolution operator. For equations with known scaling covariance, normalising space and amplitude makes a fixed ratio of physical times correspond to an autonomous transition of the full state. Our multiplicative-time renormalisation-group evolution method (MTRG) learns this transition from early-time trajectories and composes it with conserved-mass correction to reach distant horizons at logarithmic depth. The basic method fixes the ratio; a ratio-conditioned extension represents a range of intervals. We establish the exact composition law and derive a conditional error recurrence separating approximation, discretisation and propagation. Tests on diffusion and Burgers equations evaluate accuracy across distant horizons and the contributions of conservation and composition consistency. Comparisons with analytic profiles identify regimes where evolving the full state remains useful beyond the supervision window, before the limiting shape becomes the better predictor. MTRG turns known scaling structure into a reusable learned transition for this finite-time evolution.

1 INTRODUCTION

An early segment of a dissipative trajectory may reveal structure that persists much later, yet using a learned solver at long times can remain expensive. A fixed-step model needs approximately $(T - t_0)/\Delta t$ applications to reach T. As the solution spreads and decays, the model also encounters spatial and amplitude scales increasingly different from those seen during training. Improving local prediction accuracy leaves the rollout length and this change of scale untouched.

Scaling laws provide a natural alternative clock. They describe how space, amplitude and time change together; self-similar coordinates use that relation to keep a spreading solution on a fixed scale. A fixed interval in the associated log-time coordinate $\tau = \log t$ multiplies physical time by a fixed factor. Geometrically spaced learned steppers already exploit the shorter composition depth (Liu et al., 2022). For a scaling-covariant equation, state normalisation provides a further property: the exact operator for each such step is independent of absolute time.

This property defines our learning problem. Let w_t denote the full normalised state. We learn the map

$$w_t \longmapsto w_{\mu t}$$

and apply it repeatedly as physical time increases geometrically. The learning target is finite-time evolution of the field, including its approach to the limiting profile. An analytic profile estimator answers a complementary question: how much of a late-time solution can be predicted from the attractor alone?

We call the method multiplicative-time renormalisation-group evolution, or MTRG. Its basic form, MTRG-Fixed, learns a residual state-to-state map at a fixed ratio and enforces the supplied conserved mass after each step. MTRG-Variable adds a ratio input and a composition-consistency loss to represent a range of evolution intervals. Both use the same fixed inference ratio in our evaluation. The variants approximate the same normalised evolution operator using different training procedures.

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

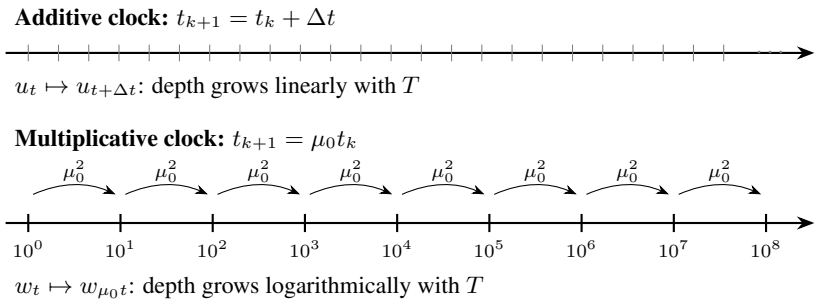

Figure 1: A change of clock changes rollout depth. In normalised coordinates, the same finite-ratio operator advances each step. Each lower arc represents two applications at $\mu_0 = \sqrt{10}$. MTRG-Fixed and MTRG-Variable share this inference schedule.

The operator structure determines the learning target and composition depth. Prediction accuracy depends on its approximation along the rollout. We establish the exact autonomy and composition identities, then derive a conditional recurrence separating approximation error, grid/reference mismatch and error propagation. The experiments compare physical- and multiplicative-time evolution, test repeated prediction across equations, and examine conservation and composition consistency. Analytic-profile controls locate the horizons where a learned transition captures more of the evolving state than its limiting shape. Together, these analyses connect classical dynamic rescaling to a full-state operator trained on early-time trajectories and reused on a logarithmic clock.

2 Related work

Learning dynamics over long horizons. Neural operators represent PDE solution maps through Fourier, branch–trunk or attention-based architectures (Li et al., 2021; Lu et al., 2021; Kovachki et al., 2023; Cao, 2021; Hao et al., 2023), with approximation theory characterising their expressive power (Kovachki et al., 2021; Lanthaler et al., 2022; Lanthaler & Stuart, 2026). Graph-based models extend learned simulation to meshes and particles (Pfaff et al., 2021; Sanchez-Gonzalez et al., 2020; Brandstetter et al., 2022b). Repeated prediction introduces distribution shift and error accumulation, studied in imitation learning (Ross & Bagnell, 2010), model-based reinforcement learning (Asadi et al., 2018) and PDE emulation (McCabe et al., 2023; Li et al., 2022). Rollout training (Um et al., 2020; List et al., 2025; Kochkov et al., 2021), input perturbation and temporal coarsening (Stachenfeld et al., 2022), and prediction refinement or learned correction (Lippe et al., 2023; Wei et al., 2025) improve the learned update. These choices concern the fidelity of individual updates and their behaviour under composition. The time representation sets a separate requirement: how many updates are needed to reach the requested physical horizon? MTRG changes the duration of that update: after state normalisation, its physical duration grows with the current time while its operator remains the same.

Time parameterization and composition. Neural ODEs integrate a learned vector field, while neural flows parameterise time-dependent maps (Chen et al., 2018; Biloš et al., 2021). Conditional neural fields admit arbitrary-time queries (Hagnberger et al., 2024), and flow-matched operators learn continuous-time PDE dynamics (Hou et al., 2026). Deep-OSG trains interval-conditioned operators with a semigroup-consistency loss (Chen & Wu, 2023); semigroup defects also serve as a diagnostic, with regularisation showing task-dependent value (Shikhman, 2026). Geometrically spaced steppers reduce composition depth (Liu et al., 2022). Progressive distillation (Salimans & Ho, 2022) and consistency models (Song et al., 2023) likewise reduce sampling evaluations, although the latter enforce a common trajectory endpoint rather than an evolution-semigroup law. MTRG combines geometric time advance with the autonomous full-state operator supplied by PDE scaling. A fixed-ratio map suffices for this construction; ratio conditioning extends the intervals it can represent. The composition law describes the target evolution, while the consistency penalty is a training mechanism for approximating it. This distinction separates operator structure from the empirical value of regularisation.

Scaling and self-similar dynamics. Self-similar variables and renormalisation-group methods describe spreading and decaying solutions on a fixed scale (Barenblatt, 1996; Goldenfeld et al., 1989; Bricmont et al., 1994). They also motivate dynamic rescaling and moving meshes (Berger & Kohn, 1988; Budd et al., 1999; 2009). Learned models incorporate related structure through scale equivariance (Worrall & Welling, 2019; Sosnovik et al., 2020), equivariant dynamics models and operators (Wang et al., 2021; Helwig et al., 2023), and symmetry-based augmentation or pretraining (Brandstetter et al., 2022a; Mialon et al., 2023). Rescaled coordinates also support blow-up profile discovery (Wang et al., 2023) and symmetry-aware physics-informed solvers (Kavousanakis et al., 2026). Closest to our setting, Wang et al. (2026) improve extrapolation for heat-based equations using self-similar variables; their reported models map space–time coordinates to solution values. MTRG instead maps the current normalised *field* to the field after a finite ratio of physical times. Classical rescaling supplies the autonomous target; learning its full-state transition allows each prediction to become the input for the next, including during the approach to the limiting profile. A coordinate-based fit extrapolates its time input, whereas a state-transition model repeatedly advances the evolving field.

3 MULTIPLICATIVE-TIME EVOLUTION

MTRG learns how the full solution changes when time is multiplied by a fixed factor. The construction has two parts: use the equation's scaling law to define an absolute-time-independent target, then approximate that target with a map that can be composed repeatedly.

3.1 AN AUTONOMOUS TARGET FROM SCALING COVARIANCE

Let u solve $\partial_t u = N[u]$ on $\mathbb{R}^d$, with flow Φ_s and spatial dilation $(D_\lambda f)(x) = \lambda^{-p} f(x/\lambda)$.

Assumption 1 (scaling covariance)**.** *There exist $p, q > 0$ such that $\Phi_{\lambda^q s} \circ D_\lambda = D_\lambda \circ \Phi_s$ for all $\lambda > 0$ and $s \geq 0$ on a scaling-invariant class of states on which the flow is well posed.*

The generator form is $N[D_\lambda f] = \lambda^{-q} D_\lambda N[f]$. This relation determines the admissible exponents: heat gives $q = 2$; the porous-medium family gives $q = (m-1)p + 2$, with mass conservation selecting $p = d$; and viscous Burgers gives $p = 1$, $q = 2$. We supply these exponents from the equation and define

$$w_t(y) \;=\; t^{p/q}\, u\Big(t^{1/q} y,\; t\Big)\,. \tag{1}$$

The coordinate y follows the physical spreading scale, while the amplitude rescaling compensates for decay. Both the transient and the limiting shape remain part of the state w_t.

For a time ratio $\mu \geq 1$, define

$$R_\mu \;=\; D^{-1}_{\mu^{1/q}} \circ \Phi_{\mu-1}. \tag{2}$$

Here a normalised state is identified with a physical state at unit time, where $w_1 = u(\cdot, 1)$. The operator first evolves this state for $\mu - 1$ units of physical time and then removes the resulting dilation.

Autonomy concerns the transition rule, not the state being evolved. A transient normalised field can continue to change shape, but advancing it by the same ratio always calls for the same exact map. For example, Gaussian mixtures under heat flow change their normalised shape as their components spread and overlap. Learning the transition retains this evolution; replacing the state by a Gaussian attractor would discard it. This distinction makes a full-state operator, rather than a fitted asymptotic profile, the target of MTRG.

Proposition 1 (autonomy and composition)**.** *Under Assumption 1, for every admissible state, every $t > 0$ and every $\mu, \mu_1, \mu_2 \geq 1$,*

$$R_\mu(w_t) = w_{\mu t}, \qquad \textit{and} \qquad R_{\mu_1 \mu_2} = R_{\mu_1} \circ R_{\mu_2}. \tag{3}$$

Proof. Taking $\lambda = t^{1/q}$ in Assumption 1 gives $\Phi_{(\mu-1)t} D_{t^{1/q}} = D_{t^{1/q}} \Phi_{\mu-1}$. Substitute $u(t) = D_{t^{1/q}} w_t$ and apply $D^{-1}_{(\mu t)^{1/q}}$ to obtain the first identity. For composition, covariance gives $\Phi_{\mu_1 - 1} D^{-1}_{\mu_2^{1/q}} = D^{-1}_{\mu_2^{1/q}} \Phi_{\mu_2(\mu_1 - 1)}$. Hence $R_{\mu_1} R_{\mu_2} = D^{-1}_{(\mu_1\mu_2)^{1/q}} \Phi_{\mu_2(\mu_1-1)+\mu_2-1} = R_{\mu_1\mu_2}$ by the flow semigroup property. □

At a fixed ratio, the same operator advances every absolute time. For a sequence of ratios with $T/t_0 = \prod_{k=1}^{K} \mu_k$, their composition reaches T in K applications. A constant ratio requires $K = \log_\mu(T/t_0)$ when the target lies on that geometric sequence; the exact family can reach other targets with a final residual factor in $[1, \mu)$ and total depth $\lceil \log_\mu(T/t_0) \rceil$. Our evaluated horizons use $t_0 = 1$ and $\mu_0 = \sqrt{10}$ throughout, so $K = 2 \log_{10} T$: sixteen applications reach 10^8. The construction uses forward evolution only; $\{R_\mu : \mu \geq 1\}$ is a multiplicative monoid.

3.2 MTRG-Fixed: learn one map and repeat it

MTRG-Fixed approximates R_{μ_0} with a residual U-Net (Ronneberger et al., 2015),

$$\widehat{R}_\theta(w) \;=\; w + s\,\mathcal{N}_\theta(w/s,\; y)\,, \tag{4}$$

where y contains cell-centred coordinate channels. The scalar s is the root-mean-square of the exact w_{t_0} over the training set, held fixed during training and inference. The network takes the current normalised state and coordinates as input; the ratio is fixed by the training task.

For mass-conserving equations we choose $p = d$, since physical mass $M = \int u\,dx$ gives $\int w_t\,dy = t^{(p-d)/q} M$. We enforce the resulting invariant on the grid after every predicted advance. With quadrature Q_h, define

$$\mathcal{P}_{M_h}[z] \;=\; \begin{cases} z\,M_h/Q_h(z), & |Q_h(z)| \geq c_{\mathrm{proj}}\,|M_h| \text{ and } \operatorname{sign} Q_h(z) = \operatorname{sign} M_h, \\ z + \big(M_h - Q_h(z)\big)\,g, & \text{otherwise.} \end{cases} \tag{5}$$

For the nonzero-mass states studied here, the supplied label M_h is the exact conserved mass, $c_{\mathrm{proj}} = 0.05$, and g is a fixed centred Gaussian of width $\max(8, 0.08N)$ cells, normalised to $Q_h(g) = 1$. Both branches enforce $Q_h = M_h$. The additive branch handles small or wrong-sign predicted mass. The multiplicative branch rescales every component: for $z = a\gamma + z_\perp$, with $Q_h(\gamma) = 1$ and $Q_h(z_\perp) = 0$, it gives $M_h\gamma + (M_h/a)z_\perp$. Thus mass correction and shape rescaling are coupled in the implemented advance; Section 5.3 evaluates this advance with and without $\mathcal{P}$.

Training uses the relative squared loss $\ell(a, b) = \sum_i \omega_i (a_i - b_i)^2 / \max(\sum_i \omega_i b_i^2, 10^{-20})$, with uniform grid weights except for the tangent-grid quadrature in fast diffusion. It supervises a short composition against states at successive geometric times:

$$\mathcal{L}_{\mathrm{sup}} = \frac{1}{J} \sum_{j=1}^{J} \ell\big(\widehat{w}_j, w_{\mu_0^j t}\big), \qquad \widehat{w}_j = \mathcal{P}\big[\widehat{R}_\theta(\widehat{w}_{j-1})\big], \qquad \widehat{w}_0 = w_t. \tag{6}$$

The average exposes the network to its own intermediate predictions, with the same mass correction used during projected inference. The training window and dimension-specific unroll settings are recorded in Appendix C.

At inference, start from w_{t_0} and repeat this projected advance K times. The horizon enters through K alone. The resulting normalised field can be mapped back to physical coordinates by inverting equation 1. This defines Fixed: one learned state-to-state map, mass correction, and repeated composition at a prescribed ratio.

3.3 MTRG-Variable: a ratio-conditioned extension

MTRG-Variable represents a family of advances with $\widehat{R}_\theta(w, c) = w + s\,\mathcal{N}_\theta(w/s, c, y)$, where the broadcast conditioning channel is $c = \log \mu$. Training draws μ log-uniformly on $[1.2, \sqrt{10}]$ and uses equation 6 with that sampled ratio. A composition penalty connects different members of the family. For $\mu_a \mu_b \leq \mu_0$,

$$\mathcal{L}_{\mathrm{semi}} = \ell\Big(\widehat{R}_\theta\big(\widehat{R}_\theta(z, \log \mu_a), \log \mu_b\big), \widehat{R}_\theta(z, \log \mu_a + \log \mu_b)\Big)\,, \tag{7}$$

with $\mathcal{P}$ applied to the intermediate state and both outputs whenever training uses projection. Thus the direct prediction supplies the normalising denominator. In one and two dimensions the loss is $\mathcal{L}_{\mathrm{sup}} + \mathcal{L}_{\mathrm{semi}}$, with gradients through both predictions. In 3D, training averages both composition orders against a stop-gradient direct prediction, then averages its supervised and consistency losses. Both variants are evaluated at $\mu_0 = \sqrt{10}$.

The variants also differ in supervision. Fixed uses two unrolled steps in one and two dimensions and one in three; Variable uses up to three. At the nominal two-step Fixed boundary, $T_{\mathrm{train}}/\mu_0^2 = t_0$, the source-time range collapses to t_0, whereas smaller sampled ratios admit later source states. Only Variable includes equation 7. In 3D, Fixed trains with projection and Variable without it. Their comparison therefore evaluates these two training configurations. The component experiment in Section 5.3 separates the contribution of the composition term while retaining the coupled ratio and supervision choices.

Spatial discretisation. The equation supplies (p, q) and the conserved functional; the implementation supplies the normalised grid and inference ratio. Proposition 1 concerns the full-space flow. The learned map usually acts on a finite box in y, whose restriction alone need not determine future interior states when tails or instantaneous propagation matter. We include this truncation effect in the reference mismatch below. For 1D fast diffusion, the tangent-mapped whole-line grid $y = Y\tan(\pi\xi/2)$, with midpoint-uniform ξ, replaces box truncation by tail-resolution and quadrature error. Scaling covariance fixes the target across absolute times; its approximation still depends on which normalised states are represented in the training data.

4 Rollout depth and error

Multiplicative time separates the length of a rollout from the accuracy of each advance. Proposition 1 gives logarithmic depth exactly. If $w_\star$ is a self-similar profile, then $R_\mu w_\star = w_\star$. Reference dynamics in its basin of attraction approach this stationary state. Accuracy of a learned composition depends on its approximation error, the numerical representation, and the sensitivity of the reference advance.

Let $G_{\mu,h} = \mathcal{P}_{M_h} \circ R_{\mu,h}$ denote the projected reference grid map and $\widehat{G}_{\theta,\mu}$ its learned counterpart. Use a fixed norm $\|\cdot\|$ on grid states.

Proposition 2 (conditional rollout bound)**.** *Fix a set Ω containing the reference and learned rollouts, and suppose that on Ω (i) $\sup_{z\in\Omega}\|\widehat{G}_{\theta,\mu_k}(z) - G_{\mu_k,h}(z)\| \le \varepsilon_k$, (ii) $G_{\mu_k,h}$ is Lipschitz with constant ρ_k, and (iii) $\eta_k = \|G_{\mu_k,h}(w_{k-1}^{\mathrm{ref}}) - w_k^{\mathrm{ref}}\|$ is the one-step reference mismatch. Then with $e_k = \|\widehat{w}_k - w_k^{\mathrm{ref}}\|$,*

$$e_K \;\le\; e_0 \prod_{\ell=1}^{K} \rho_\ell \;+\; \sum_{j=1}^{K} (\varepsilon_j + \eta_j) \prod_{\ell=j+1}^{K} \rho_\ell. \tag{8}$$

The triangle inequality gives $e_k \le \varepsilon_k + \rho_k e_{k-1} + \eta_k$; iterating yields equation 8. The factors ρ_k describe how existing errors propagate, while ε_k and η_k introduce new errors at each advance. The bound permits amplification as well as contraction. This recurrence separates error injection from propagation; the experiments measure their combined effect along learned rollouts.

The role of the clock can be seen directly in a nonexpansive special case. If $\rho_k \le 1$ and $\varepsilon_k + \eta_k \le \delta$ throughout the rollout, then $e_K \le e_0 + K\delta = e_0 + \delta \log_{\mu_0}(T/t_0)$ on the evaluated geometric sequence. Logarithmic depth limits how many new approximation errors are introduced; contraction would additionally suppress errors already present. For expansive advances, the products in equation 8 remain essential. Thus the representation determines the depth, while the learned map and its visited states determine the accuracy of that composition.

Composition consistency provides a complementary observable for Variable: two factorisations of the same total ratio agree for R_μ by equation 3, and their learned disagreement is the quantity targeted by equation 7. Section 5.3 tests its contribution to prediction. Appendix B connects the exact operator to weighted-norm stability and the asymptotic theory of diffusion equations.

5 Experiments

The experiments test the complete evolution scheme, compare repeated prediction with learned and analytic alternatives, and examine how conservation and training affect the propagated field.

Table 1: Additive- and multiplicative-time prediction on Burgers. Entries are mean relative L^2 error $\pm$ standard deviation; parentheses give network calls per trajectory. Architecture and training settings are shared; spatial grids differ (Section 5.1).

Method	$T = 10^2$	$T = 10^3$	$T = 10^4$
MTRG-Variable	$\mathbf{3.47e{-2}}$ $\pm$1.65e$-$2 (4)	$\mathbf{4.26e{-2}}$ $\pm$2.13e$-$2 (6)	$\mathbf{4.66e{-2}}$ $\pm$2.39e$-$2 (8)
Physical time, composed	9.28e$-$1 $\pm$1.39e0 (13)	3.87e5 $\pm$1.02e6 (125)	1.84e6 $\pm$3.99e6 (1250)
Physical time, one call	2.89e0 $\pm$2.17e0 (1)	2.37e1 $\pm$2.68e1 (1)	3.22e1 $\pm$4.45e1 (1)

Protocol. We evaluate 1D viscous Burgers, 1D porous medium ($m = 2$), 1D fast diffusion ($m = 1/2$), 2D heat, 2D radial porous medium ($m = 2$), and 3D heat on a 32^3 grid. Training uses the window $1 \le t \le 10$; evaluation uses $T \in \{10^2, \dots, 10^8\}$. Both MTRG variants advance at $\mu_0 = \sqrt{10}$, requiring four to sixteen applications per trajectory. We train seven seeds. For seed s, the error is the average per-trajectory relative L^2 error on the evaluation grid,

$$E_s(T) = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\widehat{w}_{s,T}^{(i)} - w_T^{(i)}\|_{2,h}}{\|w_T^{(i)}\|_{2,h}}, \qquad \|v\|_{2,h}^2 = Q_h[v^2].$$

The quadrature weights account for the nonuniform fast-diffusion grid. All aggregate results report arithmetic mean $\overline{E}(T)$ and sample standard deviation across seeds. All reported seed-level errors are finite and enter these summaries. The primary evaluation applies mass correction after each advance. The 3D study uses four training and eight evaluation trajectories; the other cases use 1024 and 64. Field illustrations fix training seed 0 and held-out sample 0 before reconstruction, using $T = 10^8$. Appendix C gives the configurations.

The relative metric has the same continuum interpretation before and after rescaling. For a normalised domain Ω and its physical counterpart $T^{1/q}\Omega$, the change of variables in equation 1 gives

$$\frac{\|\widehat{u}_T - u_T\|_{L^2(T^{1/q}\Omega)}}{\|u_T\|_{L^2(T^{1/q}\Omega)}} = \frac{\|\widehat{w}_T - w_T\|_{L^2(\Omega)}}{\|w_T\|_{L^2(\Omega)}}.$$

The amplitude and volume factors cancel. A nearly flat relative-error curve therefore describes normalised-field accuracy relative to the reference, rather than merely the decay of its physical amplitude. The discrete metric approximates this comparison on the chosen grid.

The benchmark comparison contains five learning mechanisms evaluated with mass correction: Semigroup, Continuous-generator, Conditional-query, Refinement and Pushforward. Each uses the same normalised state representation and U-Net backbone. Appendix C.1 specifies the adaptations from their source methods. All benchmarks use analytic reference families, allowing accurate evaluation at distant physical times. Learned models receive the conserved-mass label and state. The profile oracle uses the analytic attractor family and estimates mass from the discretised input.

5.1 Reaching distant horizons

Table 1 compares multiplicative-time evolution with a physical-time variable-lag semigroup model on Burgers. The physical-time model learns $u(t) \mapsto u(t + \Delta t)$ for Δt drawn log-uniformly from $[1, 8]$ and reaches each target using the smallest number of equal steps within that range. MTRG-Variable is trained for this comparison separately from the benchmark study in Section 5.2. Both use the same U-Net tensor shapes, 552,769 parameters, optimiser, 5000 updates, batch size, consistency weight and training window. Their grids have 32,768 physical points and 256 normalised points. The comparison evaluates the two evolution schemes together with their spatial representations.

Across the common evaluated horizons, the physical-time model's error grows with its rollout, while the multiplicative model retains a similar error level. The call count grows from 13 to 1250 for physical-time evolution and from four to eight for multiplicative evolution. Measured rollout time includes projection but excludes reference construction, common-grid resampling and metric computation; it also reflects the different spatial-grid costs (Table 2). The one-call physical-time model

Table 2: Burgers rollout time including mass correction, in milliseconds: mean ± standard deviation. Timing excludes reference construction, interpolation and metric evaluation. Physical and normalised grids contain 32,768 and 256 points, respectively.

Method	$T = 10^2$	$T = 10^3$	$T = 10^4$
MTRG-Variable	2.94e−1 ±7.94e−2	4.33e−1 ±7.91e−2	5.71e−1 ±7.89e−2
Physical time, composed	8.49e0 ±7.51e−1	7.89e1 ±8.95e−2	7.89e2 ±5.82e−1
Physical time, one call	6.36e−1 ±9.24e−4	6.37e−1 ±1.13e−3	6.37e−1 ±6.73e−4

Table 3: Mean relative L^2 error ± sample SD at $T = 10^8$. Learned rollouts use mass correction; bold marks the lowest learned mean. The analytic control uses its known limiting profile and input-estimated mass (zero SD); dash: Burgers not evaluated. Full horizons are in Appendix A.

	1D Burgers	1D porous medium	1D fast diffusion	2D heat	2D porous medium	3D heat
MTRG-Fixed	**3.82e−2** ±7.03e−3	**1.63e−2** ±5.88e−3	4.01e−1 ±2.04e−1	**5.34e−3** ±7.78e−4	**9.34e−3** ±1.21e−3	4.98e−1 ±3.52e−1
MTRG-Variable	1.05e−1 ±1.06e−1	3.21e−2 ±2.36e−2	**3.30e−1** ±1.13e−1	8.19e−3 ±5.79e−3	1.45e−2 ±3.13e−3	3.53e−1 ±3.58e−1
Semigroup	5.40e−1 ±1.22e0	1.77e−2 ±6.36e−3	1.57e0 ±1.84e0	5.84e−3 ±1.44e−3	1.26e−2 ±3.92e−3	4.13e−1 ±3.97e−1
Continuous-generator	9.65e−1 ±5.00e−1	4.69e−1 ±3.95e−1	4.59e1 ±9.98e1	4.78e−2 ±3.65e−2	9.96e−1 ±3.92e−1	4.49e−1 ±2.78e−1
Conditional-query	1.05e0 ±2.41e−1	2.51e−1 ±1.05e−1	1.70e0 ±1.69e0	5.38e−1 ±2.63e−1	7.33e−1 ±1.70e−1	3.56e−1 ±9.38e−2
Refinement	1.46e−1 ±5.33e−2	1.23e−1 ±5.77e−2	4.76e1 ±1.22e2	1.77e−1 ±7.34e−2	1.91e−1 ±2.48e−1	**2.33e−1** ±6.53e−2
Pushforward	1.75e1 ±4.56e1	6.56e−2 ±2.16e−2	5.73e2 ±1.31e3	2.23e−1 ±2.79e−1	4.00e−2 ±1.47e−2	1.22e0 ±9.88e−1
Profile Oracle (analytic control)	—	2.36e−5 ±0	7.43e−8 ±0	1.74e−5 ±0	7.95e−3 ±0	5.03e−3 ±0

extrapolates its lag beyond training and has larger errors than the multiplicative model. Repeated finite-ratio transitions provide an alternative to both long bounded-lag rollouts and a single out-of-range lag query.

5.2 Repeated prediction across equations

At $T = 10^8$, Fixed has the lowest mean learned error on Burgers, 1D porous medium, 2D heat and 2D porous medium (Table 3). Variable leads on fast diffusion, while Refinement leads on 3D heat. The relative accuracy of Fixed and Variable therefore depends on the equation.

Figure 2 shows how errors develop over the full horizon range. Fixed changes little after the initial transients on 2D heat and 2D porous medium. Fast diffusion and 3D heat have larger across-seed variability.

Within the shared normalised representation, Semigroup is close to Fixed on the two-dimensional diffusion problems, while direct horizon queries and the more expensive generator and refinement updates do not give a uniform improvement over Fixed.

Evolution before the limiting profile. The analytic control identifies when the limiting profile becomes the more accurate predictor. Fixed has lower mean error at the sampled horizons 10^2–10^7 on 2D porous medium, at 10^2 and 10^3 on 2D heat, and at 10^2 on 1D porous medium. At the subsequent sampled horizons the oracle has lower error; it also leads throughout the evaluated range on fast diffusion and 3D heat. Thus a distant physical time need not make the limiting profile an

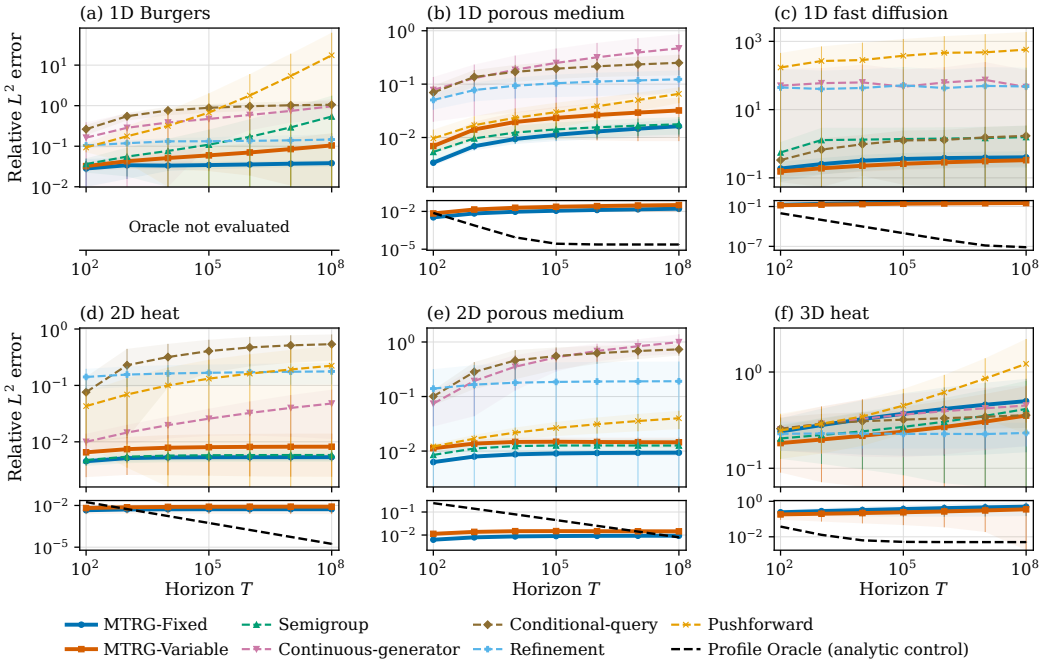

Figure 2: Long-horizon prediction and the analytic-profile control. Lines show arithmetic means; bars and shading show one sample standard deviation under the protocol in Section 5. Learned rollouts use mass correction. Lower strips compare Fixed, Variable and the dashed analytic oracle on a separate logarithmic range. Bounds below each positive axis limit are clipped. Scales differ across benchmarks. Full values appear in Appendix A.

adequate substitute for the evolving state. The 2D porous-medium results show a sustained interval where learning that evolution is useful. At 10^8, the oracle is more accurate than every learned method on its five evaluated benchmarks.

Spatial error. Figure 3 compares signed errors on the same held-out trajectory for Fixed, Variable and Semigroup. In these examples, heat errors are distributed smoothly around the profile, while porous-medium errors concentrate near its compact-support boundary. The common error scale within each row makes their spatial structure and amplitude directly comparable. Opposite-signed errors remain even when the conserved mass is satisfied: the integral constraint fixes total mass, while the learned transition determines its spatial distribution.

5.3 Training the reusable map

The component study applies each projection setting consistently during training and evaluation (Table 4). At 10^8, mass correction reduces Fixed's mean error from 0.0293 to 0.00465 on 2D heat and from 0.0510 to 0.0103 on 2D porous medium, factors of 6.3 and 5.0. Adding composition consistency improves Variable with projection, while Fixed has the lowest mean in this study. Conservation and consistency both improve the learned advance; the single-ratio model already gives accurate repeated prediction on these two benchmarks.

Fixed and Variable also differ in unroll depth and source-time coverage, with a training-projection difference in 3D. Their comparison concerns the complete training configurations, while the consistency comparison holds the other Variable settings fixed.

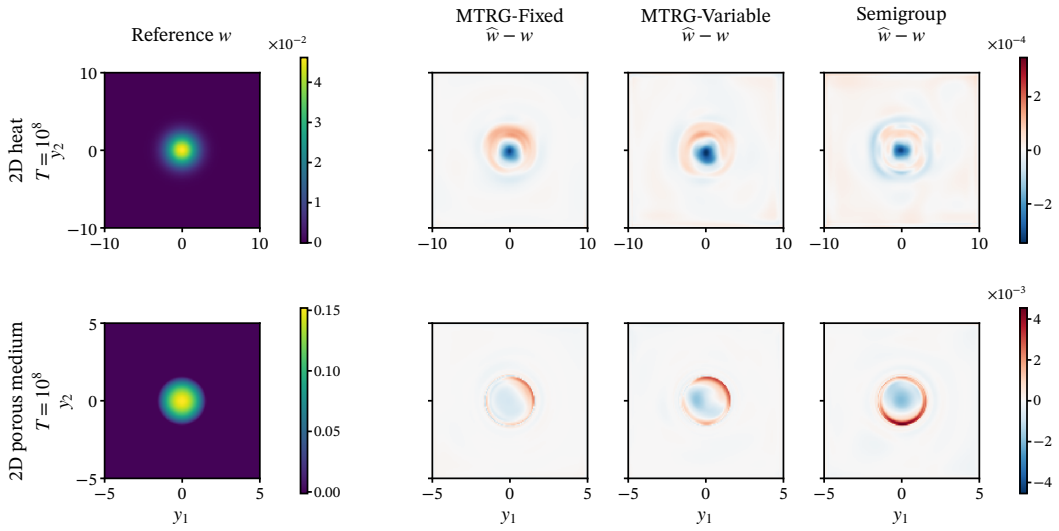

Figure 3: Long-time spatial errors on two-dimensional diffusion problems. For preselected held-out trajectories, the reference field and signed errors are shown in normalised coordinates at $T = 10^8$. Methods share the grid and mass correction. Each row has a common symmetric error scale. Porous-medium fields are cell averages; heat fields are point evaluations. These individual examples complement the aggregate mean-and-standard-deviation results.

Table 4: Component comparisons: mean relative L^2 error $\pm$ sample SD; bold marks the lowest mean. Projection on/off applies to both training and evaluation. The first four configurations are separately trained; the final row uses the MTRG-Variable model evaluated in Table 3.

	2D heat		2D porous medium	
Configuration	$T = 10^2$	$T = 10^8$	$T = 10^2$	$T = 10^8$
Fixed, projection off	7.12e−3 ±2.09e−3	2.93e−2 ±9.14e−3	9.27e−3 ±6.33e−4	5.10e−2 ±1.06e−2
Fixed, projection on	**4.01e−3** ±8.59e−4	**4.65e−3** ±1.15e−3	**6.40e−3** ±6.11e−4	**1.03e−2** ±1.35e−3
Variable, projection off, no consistency	1.68e−2 ±8.09e−3	9.15e−2 ±5.73e−2	1.81e−2 ±2.47e−3	8.14e−2 ±2.65e−2
Variable, projection on, no consistency	1.17e−2 ±5.78e−3	1.64e−2 ±9.50e−3	1.29e−2 ±1.46e−3	1.77e−2 ±2.28e−3
Variable, projection on, with consistency	6.52e−3 ±4.15e−3	8.19e−3 ±5.79e−3	1.13e−2 ±1.96e−3	1.45e−2 ±3.13e−3

6 Conclusion

MTRG learns a full-state transition over a ratio of physical times. Scaling covariance makes the exact target autonomous after normalisation; repeated composition then gives logarithmic rollout depth. Fixed implements this construction with one learned map and conserved-mass correction; Variable extends it to a ratio-conditioned family. The error bound separates this reduction in depth from the accuracy of the learned transitions. Experiments identify finite-time intervals on porous-medium and heat equations where Fixed is more accurate than the analytic-profile control, before that control becomes the better predictor. The component study shows that mass correction improves repeated prediction on the two-dimensional problems. Learning the full-state transition thus captures evolution beyond the supervision window, including transients that the limiting profile does not resolve.

AI USE STATEMENT

We used AI tools for manuscript editing, feedback on experimental design, and computational workflows, including experiment automation and result analysis. We checked AI-assisted code through tests and verified reported results against experiment records. The authors take responsibility for the final manuscript and results.

ETHICS STATEMENT

This work studies numerical prediction for mathematical PDE models using synthetic reference trajectories. It involves no human participants or personal data. The principal application risk is using a learned long-horizon prediction beyond its validated equation, initial-condition or resolution range. Safety-critical applications require independent physical validation and error assessment.

REPRODUCIBILITY STATEMENT

Section 3 specifies the operator, projection and training losses; Section 4 states the assumptions and derives the rollout bound. Section 5 defines the evaluation protocol and aggregation rule. Appendix A reports the complete horizon tables, Appendix B gives analytical context, and Appendix C describes grids, data families, optimisation and comparison mechanisms. The accompanying reporting archive contains the seed-level statistics, figure arrays, and scripts that regenerate the numerical tables and plots.

REFERENCES

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 264–273, 2018.

Dominique Bakry and Michel Émery. Diffusions hypercontractives. *Séminaire de probabilités*, 19: 177–206, 1985.

Grigory Isaakovich Barenblatt. *Scaling, Self-similarity, and Intermediate Asymptotics*. Cambridge University Press, 1996. doi: 10.1017/CBO9781107050242.

Marsha Berger and Robert V. Kohn. A rescaling algorithm for the numerical calculation of blowing-up solutions. *Communications on Pure and Applied Mathematics*, 41(6):841–863, 1988.

Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, 126(9):098302, 2021. doi: 10.1103/PhysRevLett.126.098302.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. In *Advances in Neural Information Processing Systems 34*, volume 34, 2021.

Adrien Blanchet, Matteo Bonforte, Jean Dolbeault, Gabriele Grillo, and Juan Luis Vázquez. Asymptotics of the fast diffusion equation via entropy estimates. *Archive for Rational Mechanics and Analysis*, 191(2):347–385, 2009. doi: 10.1007/s00205-008-0155-z.

Johannes Brandstetter, Max Welling, and Daniel E. Worrall. Lie point symmetry data augmentation for neural pde solvers. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022), PMLR vol. 162*, volume 162, pp. 2241–2256, 2022a.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022b.

Jean Bricmont, Antti Kupiainen, and Gang Lin. Renormalization group and asymptotics of solutions of nonlinear parabolic equations. *Communications on Pure and Applied Mathematics*, 47(6):893–922, 1994. doi: 10.1002/cpa.3160470606.

Chris J. Budd, G. J. Collins, W. Z. Huang, and R. D. Russell. Self-similar numerical solutions of the porous-medium equation using moving mesh methods. *Philosophical Transactions of the Royal Society A*, 357(1754):1047–1077, 1999.

Chris J. Budd, Weizhang Huang, and Robert D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009. doi: 10.1017/S0962492906400015.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In *Advances in Neural Information Processing Systems 34*, 2021.

J. A. Carrillo and G. Toscani. Asymptotic L^1-decay of solutions of the porous medium equation to self-similarity. *Indiana University Mathematics Journal*, 49(1):113–142, 2000. doi: 10.1512/iumj.2000.49.1756.

Junfeng Chen and Kailiang Wu. Deep-OSG: Deep learning of operators in semigroup. *Journal of Computational Physics*, 493:112498, 2023. doi: 10.1016/j.jcp.2023.112498.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*, volume 31, 2018.

Herman Chernoff. A note on an inequality involving the normal distribution. *The Annals of Probability*, 9(3):533–535, 1981. doi: 10.1214/aop/1176994428.

Manuel Del Pino and Jean Dolbeault. Best constants for gagliardo-nirenberg inequalities and applications to nonlinear diffusions. *Journal de Mathematiques Pures et Appliquees*, 81(9):847–875, 2002. doi: 10.1016/S0021-7824(02)01266-7.

Jochen Denzler and Robert J. McCann. Fast diffusion to self-similarity: complete spectrum, long-time asymptotics, and numerology. *Archive for Rational Mechanics and Analysis*, 175(3):301–342, 2005. doi: 10.1007/s00205-004-0336-3.

Thierry Gallay and C. Eugene Wayne. Invariant manifolds and the long-time asymptotics of the Navier–Stokes and vorticity equations on $\mathbb{R}^2$. *Archive for Rational Mechanics and Analysis*, 163 (3):209–258, 2002. doi: 10.1007/s002050200200.

Abdelilah Gmira and Laurent Véron. Large time behaviour of the solutions of a semilinear parabolic equation in $\mathbb{R}^N$. *Journal of Differential Equations*, 53(2):258–276, 1984.

Nigel Goldenfeld, Olivier Martin, and Y. Oono. Intermediate asymptotics and renormalization group theory. *Journal of Scientific Computing*, 4(4):355–372, 1989. doi: 10.1007/BF01060993.

Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *Transactions on Machine Learning Research*, 2023.

Jan Hagnberger, Marimuthu Kalimuthu, Daniel Musekamp, and Mathias Niepert. Vectorized conditional neural fields: A framework for solving time-dependent parametric partial differential equations. In *International Conference on Machine Learning*, 2024.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics, vol. 31, Springer-Verlag, Berlin/Heidelberg, 2006. doi: 10.1007/3-540-30666-8.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 12556–12569, 2023.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Xianglong Hou, Xinquan Huang, and Paris Perdikaris. CFO: Learning continuous-time PDE dynamics via flow-matched neural operators. In *International Conference on Learning Representations*, 2026.

Michail E. Kavousanakis, Gianluca Fabiani, Anastasia S. Georgiou, Constantinos Siettos, Panayotis G. Kevrekidis, and Ioannis G. Kevrekidis. Going with the flow to solve for symmetry-driven PDE dynamics with physics-informed neural networks. *Nature Communications*, 17:9420, 2026.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Samuel Lanthaler and Andrew M. Stuart. The parametric complexity of operator learning. *IMA Journal of Numerical Analysis*, 46(2):647–712, 2026. doi: 10.1093/imanum/draf028.

Samuel Lanthaler, Siddhartha Mishra, and George Em Karniadakis. Error estimates for deeponets: a deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022. doi: 10.1093/imatrm/tnac001.

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

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. In *Advances in Neural Information Processing Systems 35*, 2022. doi: 10.52202/068431-1220.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, 2023.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in unrolled training of neural physics simulators on transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433:117441, 2025. doi: 10.1016/j.cma.2024.117441.

Yuying Liu, J. Nathan Kutz, and Steven L. Brunton. Hierarchical deep learning of multiscale differential equation time-steppers. *Philosophical Transactions of the Royal Society A*, 380(2229): 20210200, 2022.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators. *Transactions on Machine Learning Research*, 2023.

Grégoire Mialon, Quentin Garrido, Hannah Lawrence, Danyal Rehman, Yann LeCun, and Bobak T. Kiani. Self-supervised learning with lie symmetries for partial differential equations. In *Advances in Neural Information Processing Systems 36*, 2023.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The

well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Jack Richter-Powell, Yaron Lipman, and Ricky T. Q. Chen. Neural conservation laws: A divergence-free perspective. In *Advances in Neural Information Processing Systems 35*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, Lecture Notes in Computer Science, Springer*, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR*, volume 9, pp. 661–668, 2010.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 8459–8468, 2020.

Lennon J. Shikhman. Semigroup consistency as a diagnostic for learned physics simulators. In *ICML 2026 Workshop on AI for Physics*, 2026. URL https://openreview.net/forum?id=MeAFOZnrvM. arXiv:2605.26324.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 32211–32252, 2023.

Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020.

Kimberly Stachenfeld, Drummond B. Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned coarse models for efficient turbulence simulation. In *International Conference on Learning Representations*, 2022.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track*, 2022.

Kiwon Um, Robert Brand, Yun (Raymond) Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In *Advances in Neural Information Processing Systems 33*, 2020.

Juan Luis Vázquez. *Smoothing and Decay Estimates for Nonlinear Diffusion Equations: Equations of Porous Medium Type*. Oxford University Press, 2006. doi: 10.1093/acprof:oso/9780199202973.001.0001.

Juan Luis Vázquez. *The Porous Medium Equation: Mathematical Theory*. Oxford University Press, 2007. doi: 10.1093/acprof:oso/9780198569039.001.0001.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*, 2021.

Shihao Wang, Qipeng Qian, and Jingquan Wang. Learning heat-based equations in self-similar variables. In *International Conference on Learning Representations*, 2026.

Yongji Wang, Ching-Yao Lai, Javier Gómez-Serrano, and Tristan Buckmaster. Asymptotic self-similar blow-up profile for three-dimensional axisymmetric Euler equations using neural networks. *Physical Review Letters*, 130(24):244002, 2023. doi: 10.1103/PhysRevLett.130.244002.

Hao Wei, Aleksandra Franz, Björn List, and Nils Thuerey. Inc: An indirect neural corrector for auto-regressive hybrid pde solvers. In *Advances in Neural Information Processing Systems 38*, 2025.

Daniel E. Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In *Advances in Neural Information Processing Systems*, 2019.

A FULL PREDICTION RESULTS

All learned entries report the arithmetic mean and sample standard deviation of seed-level errors, using the protocol in Section 5. Each seed-level error is an average over evaluation trajectories. Boldface identifies the smallest mean in a column. The source CSV and table-generation script accompany this paper.

The learned-method comparisons use mass correction during evaluation. The analytic control is reported separately on its five evaluated benchmarks.

Table 5: Relative L^2 error on 1D Burgers with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	$\mathbf{2.82e{-2}}$ $\pm 2.05e{-2}$	$\mathbf{3.40e{-2}}$ $\pm 1.83e{-2}$	$\mathbf{3.33e{-2}}$ $\pm 8.86e{-3}$	$\mathbf{3.43e{-2}}$ $\pm 7.53e{-3}$	$\mathbf{3.55e{-2}}$ $\pm 6.79e{-3}$	$\mathbf{3.70e{-2}}$ $\pm 7.05e{-3}$	$\mathbf{3.82e{-2}}$ $\pm 7.03e{-3}$
MTRG-Variable	$3.23e{-2}$ $\pm 7.57e{-3}$	$4.27e{-2}$ $\pm 1.53e{-2}$	$5.11e{-2}$ $\pm 2.41e{-2}$	$5.96e{-2}$ $\pm 3.52e{-2}$	$7.01e{-2}$ $\pm 5.13e{-2}$	$8.50e{-2}$ $\pm 7.44e{-2}$	$1.05e{-1}$ $\pm 1.06e{-1}$
Semigroup	$3.63e{-2}$ $\pm 1.40e{-2}$	$5.55e{-2}$ $\pm 3.76e{-2}$	$7.66e{-2}$ $\pm 7.66e{-2}$	$1.09e{-1}$ $\pm 1.48e{-1}$	$1.71e{-1}$ $\pm 2.93e{-1}$	$2.92e{-1}$ $\pm 5.93e{-1}$	$5.40e{-1}$ $\pm 1.22e0$
Continuous-generator	$1.63e{-1}$ $\pm 2.27e{-1}$	$2.85e{-1}$ $\pm 3.68e{-1}$	$3.84e{-1}$ $\pm 4.03e{-1}$	$4.70e{-1}$ $\pm 3.63e{-1}$	$5.91e{-1}$ $\pm 3.43e{-1}$	$7.43e{-1}$ $\pm 3.62e{-1}$	$9.65e{-1}$ $\pm 5.00e{-1}$
Conditional-query	$2.63e{-1}$ $\pm 7.49e{-2}$	$5.49e{-1}$ $\pm 1.19e{-1}$	$7.61e{-1}$ $\pm 1.84e{-1}$	$8.86e{-1}$ $\pm 2.12e{-1}$	$9.69e{-1}$ $\pm 2.31e{-1}$	$1.02e0$ $\pm 2.42e{-1}$	$1.05e0$ $\pm 2.41e{-1}$
Refinement	$1.05e{-1}$ $\pm 1.63e{-2}$	$1.19e{-1}$ $\pm 2.17e{-2}$	$1.32e{-1}$ $\pm 2.61e{-2}$	$1.33e{-1}$ $\pm 3.24e{-2}$	$1.36e{-1}$ $\pm 4.36e{-2}$	$1.40e{-1}$ $\pm 4.54e{-2}$	$1.46e{-1}$ $\pm 5.33e{-2}$
Pushforward	$9.33e{-2}$ $\pm 2.24e{-2}$	$1.76e{-1}$ $\pm 1.36e{-1}$	$3.16e{-1}$ $\pm 4.35e{-1}$	$6.75e{-1}$ $\pm 1.34e0$	$1.78e0$ $\pm 4.24e0$	$5.39e0$ $\pm 1.37e1$	$1.75e1$ $\pm 4.56e1$

Table 6: Relative L^2 error on 1D porous medium with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	$\mathbf{3.37e{-3}}$ $\pm 4.16e{-4}$	$\mathbf{6.87e{-3}}$ $\pm 1.12e{-3}$	$\mathbf{9.38e{-3}}$ $\pm 1.84e{-3}$	$\mathbf{1.13e{-2}}$ $\pm 2.72e{-3}$	$\mathbf{1.30e{-2}}$ $\pm 3.73e{-3}$	$\mathbf{1.46e{-2}}$ $\pm 4.80e{-3}$	$\mathbf{1.63e{-2}}$ $\pm 5.88e{-3}$
MTRG-Variable	$6.91e{-3}$ $\pm 1.78e{-3}$	$1.41e{-2}$ $\pm 5.77e{-3}$	$1.95e{-2}$ $\pm 1.01e{-2}$	$2.33e{-2}$ $\pm 1.41e{-2}$	$2.64e{-2}$ $\pm 1.76e{-2}$	$2.93e{-2}$ $\pm 2.08e{-2}$	$3.21e{-2}$ $\pm 2.36e{-2}$
Semigroup	$5.36e{-3}$ $\pm 1.74e{-3}$	$9.61e{-3}$ $\pm 3.24e{-3}$	$1.24e{-2}$ $\pm 4.26e{-3}$	$1.42e{-2}$ $\pm 4.97e{-3}$	$1.55e{-2}$ $\pm 5.52e{-3}$	$1.67e{-2}$ $\pm 5.97e{-3}$	$1.77e{-2}$ $\pm 6.36e{-3}$
Continuous-generator	$7.79e{-2}$ $\pm 5.82e{-2}$	$1.29e{-1}$ $\pm 1.04e{-1}$	$1.86e{-1}$ $\pm 1.57e{-1}$	$2.49e{-1}$ $\pm 2.14e{-1}$	$3.18e{-1}$ $\pm 2.74e{-1}$	$3.91e{-1}$ $\pm 3.35e{-1}$	$4.69e{-1}$ $\pm 3.95e{-1}$
Conditional-query	$6.95e{-2}$ $\pm 2.40e{-2}$	$1.36e{-1}$ $\pm 4.75e{-2}$	$1.70e{-1}$ $\pm 5.98e{-2}$	$1.95e{-1}$ $\pm 6.93e{-2}$	$2.16e{-1}$ $\pm 7.99e{-2}$	$2.34e{-1}$ $\pm 9.27e{-2}$	$2.51e{-1}$ $\pm 1.05e{-1}$
Refinement	$5.05e{-2}$ $\pm 1.73e{-2}$	$7.74e{-2}$ $\pm 2.99e{-2}$	$9.34e{-2}$ $\pm 3.96e{-2}$	$1.04e{-1}$ $\pm 4.69e{-2}$	$1.12e{-1}$ $\pm 5.22e{-2}$	$1.17e{-1}$ $\pm 5.41e{-2}$	$1.23e{-1}$ $\pm 5.77e{-2}$
Pushforward	$9.65e{-3}$ $\pm 4.40e{-3}$	$1.70e{-2}$ $\pm 8.06e{-3}$	$2.34e{-2}$ $\pm 1.17e{-2}$	$2.99e{-2}$ $\pm 1.50e{-2}$	$3.84e{-2}$ $\pm 1.72e{-2}$	$5.03e{-2}$ $\pm 1.90e{-2}$	$6.56e{-2}$ $\pm 2.16e{-2}$

Table 7: Relative L^2 error on 1D fast diffusion with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	1.88e−1 ±1.12e−1	2.54e−1 ±1.29e−1	3.19e−1 ±1.58e−1	3.58e−1 ±1.76e−1	3.81e−1 ±1.88e−1	3.92e−1 ±1.96e−1	4.01e−1 ±2.04e−1
MTRG-Variable	**1.54e−1** ±4.77e−2	**1.91e−1** ±6.26e−2	**2.27e−1** ±7.55e−2	**2.59e−1** ±8.85e−2	**2.86e−1** ±9.92e−2	**3.09e−1** ±1.07e−1	**3.30e−1** ±1.13e−1
Semigroup	5.56e−1 ±5.74e−1	1.28e0 ±1.65e0	1.32e0 ±1.61e0	1.37e0 ±1.61e0	1.42e0 ±1.63e0	1.47e0 ±1.68e0	1.57e0 ±1.84e0
Continuous-generator	5.03e1 ±1.09e2	5.96e1 ±1.05e2	6.26e1 ±1.47e2	4.69e1 ±1.03e2	6.23e1 ±1.13e2	7.43e1 ±1.74e2	4.59e1 ±9.98e1
Conditional-query	3.33e−1 ±2.58e−1	6.72e−1 ±7.95e−1	9.67e−1 ±8.54e−1	1.25e0 ±1.19e0	1.29e0 ±1.25e0	1.54e0 ±1.57e0	1.70e0 ±1.69e0
Refinement	4.45e1 ±1.11e2	4.03e1 ±9.98e1	4.32e1 ±1.08e2	5.15e1 ±1.26e2	4.29e1 ±1.05e2	4.95e1 ±1.27e2	4.76e1 ±1.22e2
Pushforward	1.72e2 ±2.87e2	2.65e2 ±4.47e2	2.84e2 ±6.35e2	3.78e2 ±7.95e2	4.62e2 ±9.60e2	4.77e2 ±1.15e3	5.73e2 ±1.31e3

Table 8: Relative L^2 error on 2D heat with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	**4.52e−3** ±5.06e−4	**5.14e−3** ±6.86e−4	**5.29e−3** ±7.49e−4	**5.33e−3** ±7.70e−4	**5.34e−3** ±7.76e−4	**5.34e−3** ±7.78e−4	**5.34e−3** ±7.78e−4
MTRG-Variable	6.52e−3 ±4.15e−3	7.41e−3 ±4.83e−3	7.86e−3 ±5.31e−3	8.04e−3 ±5.54e−3	8.14e−3 ±5.70e−3	8.18e−3 ±5.77e−3	8.19e−3 ±5.79e−3
Semigroup	4.75e−3 ±1.02e−3	5.45e−3 ±1.27e−3	5.69e−3 ±1.37e−3	5.78e−3 ±1.41e−3	5.82e−3 ±1.43e−3	5.83e−3 ±1.43e−3	5.84e−3 ±1.44e−3
Continuous-generator	1.00e−2 ±3.91e−3	1.46e−2 ±5.45e−3	1.97e−2 ±8.26e−3	2.57e−2 ±1.30e−2	3.25e−2 ±1.96e−2	3.99e−2 ±2.76e−2	4.78e−2 ±3.65e−2
Conditional-query	7.59e−2 ±6.65e−2	2.28e−1 ±2.17e−1	3.18e−1 ±2.27e−1	4.07e−1 ±2.39e−1	4.71e−1 ±2.55e−1	5.14e−1 ±2.62e−1	5.38e−1 ±2.63e−1
Refinement	1.41e−1 ±5.58e−2	1.56e−1 ±6.20e−2	1.63e−1 ±6.56e−2	1.67e−1 ±6.78e−2	1.71e−1 ±6.98e−2	1.75e−1 ±7.20e−2	1.77e−1 ±7.34e−2
Pushforward	4.30e−2 ±3.45e−2	6.90e−2 ±6.56e−2	9.99e−2 ±1.08e−1	1.32e−1 ±1.54e−1	1.62e−1 ±1.95e−1	1.92e−1 ±2.35e−1	2.23e−1 ±2.79e−1

Table 9: Relative L^2 error on 2D porous medium with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	**6.28e−3** ±3.79e−4	**7.93e−3** ±6.64e−4	**8.71e−3** ±8.95e−4	**9.05e−3** ±1.04e−3	**9.21e−3** ±1.12e−3	**9.29e−3** ±1.18e−3	**9.34e−3** ±1.21e−3
MTRG-Variable	1.13e−2 ±1.96e−3	1.37e−2 ±2.56e−3	1.47e−2 ±2.80e−3	1.48e−2 ±2.88e−3	1.47e−2 ±2.97e−3	1.45e−2 ±3.07e−3	1.45e−2 ±3.13e−3
Semigroup	8.50e−3 ±1.23e−3	1.11e−2 ±2.25e−3	1.23e−2 ±2.86e−3	1.27e−2 ±3.21e−3	1.27e−2 ±3.52e−3	1.26e−2 ±3.76e−3	1.26e−2 ±3.92e−3
Continuous-generator	7.41e−2 ±4.58e−2	1.93e−1 ±1.50e−1	3.52e−1 ±2.24e−1	5.24e−1 ±2.36e−1	6.84e−1 ±2.47e−1	8.34e−1 ±2.87e−1	9.96e−1 ±3.92e−1
Conditional-query	1.01e−1 ±3.16e−2	2.83e−1 ±1.46e−1	4.59e−1 ±2.52e−1	5.53e−1 ±2.43e−1	6.24e−1 ±2.26e−1	6.84e−1 ±1.99e−1	7.33e−1 ±1.70e−1
Refinement	1.40e−1 ±1.78e−1	1.65e−1 ±2.11e−1	1.79e−1 ±2.29e−1	1.85e−1 ±2.38e−1	1.89e−1 ±2.43e−1	1.90e−1 ±2.46e−1	1.91e−1 ±2.48e−1
Pushforward	1.21e−2 ±1.36e−3	1.69e−2 ±2.92e−3	2.18e−2 ±5.22e−3	2.65e−2 ±7.72e−3	3.11e−2 ±1.02e−2	3.56e−2 ±1.25e−2	4.00e−2 ±1.47e−2

Table 10: Relative L^2 error on 3D heat with mass correction: mean $\pm$ sample standard deviation. Bold marks the lowest mean in each column.

Method	10^2	10^3	10^4	10^5	10^6	10^7	10^8
MTRG-Fixed	2.41e−1 ±8.91e−2	2.81e−1 ±1.36e−1	3.24e−1 ±1.86e−1	3.68e−1 ±2.34e−1	4.14e−1 ±2.79e−1	4.57e−1 ±3.18e−1	4.98e−1 ±3.52e−1
MTRG-Variable	**1.83e−1** ±9.26e−2	**1.99e−1** ±1.27e−1	**2.18e−1** ±1.61e−1	2.40e−1 ±1.95e−1	2.69e−1 ±2.34e−1	3.05e−1 ±2.87e−1	3.53e−1 ±3.58e−1
Semigroup	2.04e−1 ±7.90e−2	2.22e−1 ±1.11e−1	2.42e−1 ±1.45e−1	2.69e−1 ±1.84e−1	3.04e−1 ±2.35e−1	3.51e−1 ±3.02e−1	4.13e−1 ±3.97e−1
Continuous-generator	2.54e−1 ±1.01e−1	2.90e−1 ±1.49e−1	3.26e−1 ±1.95e−1	3.61e−1 ±2.31e−1	3.92e−1 ±2.53e−1	4.20e−1 ±2.66e−1	4.49e−1 ±2.78e−1
Conditional-query	2.58e−1 ±8.05e−2	2.90e−1 ±7.78e−2	3.08e−1 ±7.53e−2	3.20e−1 ±7.57e−2	3.31e−1 ±7.90e−2	3.43e−1 ±8.50e−2	3.56e−1 ±9.38e−2
Refinement	2.28e−1 ±7.73e−2	2.30e−1 ±7.54e−2	2.30e−1 ±6.97e−2	**2.28e−1** ±6.73e−2	**2.28e−1** ±6.73e−2	**2.27e−1** ±6.69e−2	**2.33e−1** ±6.53e−2
Pushforward	2.47e−1 ±1.22e−1	2.90e−1 ±1.48e−1	3.45e−1 ±1.74e−1	4.48e−1 ±2.21e−1	6.13e−1 ±3.21e−1	8.56e−1 ±5.34e−1	1.22e0 ±9.88e−1

Table 11: Analytic profile controls: relative L^2 error, mean $\pm$ standard deviation. These deterministic controls use the known limiting profile and mass estimated by quadrature from the discretised input; repeated seed evaluations are identical.

Benchmark	10^2	10^3	10^4	10^5	10^6	10^7	10^8
1D porous medium	7.55e−3 ±0	7.70e−4 ±0	8.71e−5 ±0	2.73e−5 ±0	2.37e−5 ±0	2.36e−5 ±0	2.36e−5 ±0
1D fast diffusion	9.62e−3 ±0	9.73e−4 ±0	9.75e−5 ±0	9.76e−6 ±0	9.90e−7 ±0	1.36e−7 ±0	7.43e−8 ±0
2D heat	1.74e−2 ±0	5.49e−3 ±0	1.74e−3 ±0	5.49e−4 ±0	1.74e−4 ±0	5.49e−5 ±0	1.74e−5 ±0
2D porous medium	2.42e−1 ±0	1.39e−1 ±0	7.90e−2 ±0	4.46e−2 ±0	2.51e−2 ±0	1.41e−2 ±0	7.95e−3 ±0
3D heat	3.77e−2 ±0	1.29e−2 ±0	6.33e−3 ±0	5.17e−3 ±0	5.04e−3 ±0	5.03e−3 ±0	5.03e−3 ±0

B Operator structure and analytical context

Exact heat evolution. For $\partial_t u = \nu \Delta u$, the normalised equation in $\tau = \log t$ is

$$\partial_\tau w = \nu \Delta_y w + \tfrac{1}{2} \nabla_y \cdot (yw).$$

Its unit-mass stationary profile is $\gamma(y) = (4\pi\nu)^{-d/2} \exp(-|y|^2/(4\nu))$. The exact finite-ratio map has eigenvalues $\mu^{-n/2}$ on Hermite modes of total degree n (Gallay & Wayne, 2002). The mass direction has eigenvalue one, while equal-mass perturbations exclude that direction. The neutral invariant and decaying shape modes thus play distinct roles in long-time evolution.

Norm-specific stability. In $H = L^2(\gamma^{-1} dy)$, the Gaussian Poincaré inequality (Chernoff, 1981; Bakry & Émery, 1985) gives the exact heat-flow estimate

$$\|R_\mu f - R_\mu g\|_H \le \mu^{-1/2} \|f - g\|_H$$

for equal-mass states with finite weighted norm. On a bounded grid, passing to an unweighted norm introduces constants determined by the extremal values of γ. The weighted estimate describes the exact dynamics in their natural geometry, whereas Proposition 2 keeps sensitivity in the grid norm explicit.

Nonlinear diffusion. The porous-medium and fast-diffusion scaling exponents follow from covariance and mass conservation (Vázquez, 2007; 2006). Entropy and asymptotic methods characterize approach to self-similarity for porous-medium equations (Carrillo & Toscani, 2000; Del Pino & Dolbeault, 2002) and fast-diffusion equations (Denzler & McCann, 2005; Blanchet et al., 2009) under their respective hypotheses. Related large-time analysis also applies to semilinear parabolic equations (Gmira & Véron, 1984). These results explain why normalised states can approach a stationary profile while physical width and amplitude continue to change. Learning R_μ retains the finite-time path towards that profile rather than replacing the path by its limit.

Projection and reference mismatch. The correction in equation 5 enforces the supplied invariant in both branches. Its multiplicative branch rescales every field component; its additive branch changes the field along a fixed unit-mass shape. Both therefore affect shape error as well as mass error. In Proposition 2, η_k includes quadrature, spatial restriction and invariant-label mismatch between the projected grid advance and the evaluated reference.

C Configurations and comparison mechanisms

Table 12: Benchmark configurations. The similarity grid has N^d cells on $[-Y, Y]^d$, except fast diffusion, which uses $y = Y \tan(\pi\xi/2)$ on midpoint-uniform ξ covering the whole line; there Y is a tangent scale. All equations are posed on the whole line or whole space.

Benchmark	d	(p, q)	N	Y	Train traj.	Eval traj.
Viscous Burgers (ν=0.05)	1	$(1, 2)$	256	5	1024	64
Porous medium (m=2)	1	$(1, 3)$	256	5	1024	64
Fast diffusion (m=1/2)	1	$(1, 3/2)$	2048	2	1024	64
Heat	2	$(2, 2)$	128	10	1024	64
Radial porous medium (m=2)	2	$(2, 4)$	128	5	1024	64
Heat	3	$(3, 2)$	32	5	4	8

Backbone and optimisation. The U-Net uses channel widths $(16, 32, 64, 128, 128)$ with two convolution blocks per level in one and two dimensions, and $(8, 16, 32, 64)$ with two blocks per level in three. AdamW starts at $2 \cdot 10^{-3}$ ($2 \cdot 10^{-4}$ in 3D), with cosine decay to 10^{-5} ($2 \cdot 10^{-6}$ in 3D), weight decay 10^{-4}, gradient clipping at 1.0, 5000 updates and batch size 16 (2 in 3D). Supervision uses early states from $t_0 = 1$ up to $T_{\text{train}} = 10$, including the Fixed two-step endpoint.

Ratios and supervision. Fixed trains at $\mu_0 = \sqrt{10}$ with two supervised steps in one and two dimensions and one in three. Variable draws μ log-uniformly on $[1.2, \sqrt{10}]$, uses up to three supervised steps, and uses the dimension-specific consistency loss in Section 3.3. Both use $\mu_0 = \sqrt{10}$ at inference, with $K = 2\log_{10} T$ applications and equation 5 after each in the primary evaluation. Variable's 3D training omits projection; Fixed's includes it.

Initial data and invariants. The 1D porous-medium and fast-diffusion cases use shifted Barenblatt profiles with log-uniform shifts on $[0.25, 8]$. Heat uses shifted anisotropic Gaussian mixtures. Radial porous medium uses Barenblatt data on the radial grid; Burgers uses whole-line bump mixtures of nonzero mass. Mass correction uses the exact conserved mass supplied with each trajectory. Hard constraints and conservation-aware parameterisations are established structure-preserving strategies in learned models (Beucler et al., 2021; Richter-Powell et al., 2022) and numerical integration (Hairer et al., 2006).

Benchmark focus. PDEBench, PDEArena and The Well provide broad simulation suites (Takamoto et al., 2022; Gupta & Brandstetter, 2023; Ohana et al., 2024), while APEBench supports systematic autoregressive-emulator comparisons (Koehler et al., 2024). Our suite targets scaling-covariant evolution at widely separated physical times using analytic reference families throughout. The 3D study covers heat at one resolution with the smaller dataset shown in Table 12.

C.1 Comparison mechanisms in the normalised representation

The comparison evaluates published learning mechanisms with the shared U-Net, normalisation, sampling and mass correction. Row names identify the implemented mechanism; the following details specify its relation to the source method.

Semigroup (Deep-OSG). The implementation retains the lag-weighted residual $\Phi(w, s) = w + s\mathcal{N}(w, s)$ with $s = \log \mu$, and symmetric consistency over both composition orders (Chen & Wu, 2023). The U-Net replaces the released FCN/FNO. It uses the same multiplicative time, ratio range, number of unrolled training steps and consistency weight as Variable; the remaining differences concern residual parameterisation and loss.

Continuous-generator (CFO). We retain the quintic-Hermite path construction, conditional path parameter, pointwise velocity loss and fixed-step Heun integrator (Hou et al., 2026). The U-Net replaces the released JAX operator, and suppressing absolute-time input gives an autonomous generator in $\log t$. The evaluated configuration uses four network evaluations per macro-step.

Conditional-query (VCNeF). The retained interface is $(w_{t_0}, \log(T/t_0)) \mapsto w_T$, answering each horizon directly (Hagnberger et al., 2024). A parameter-matched U-Net replaces the published attention and modulation blocks; PDE-parameter conditioning is omitted. This contrasts a direct horizon query with repeated state advances.

Refinement (PDE-Refiner). The implementation retains the four-step scheduler, exponential noise schedule, v-prediction target and training with one refinement index sampled per update (Lippe et al., 2023). Three refinement stages and a terminal evaluation give four U-Net evaluations per macro-step. It uses the same checkpoint policy as MTRG, without exponential moving averaging.

Pushforward (MP-PDE). This mechanism applies the unbundled pushforward training correction (Brandstetter et al., 2022b) to the single-ratio U-Net. The evaluated model uses the common convolutional backbone rather than the source method's message-passing network.

Analytic profile control. The profile oracle receives the closed-form attractor family and estimates mass with Q_h from the discretised input. Its additional information is the profile shape; learned methods instead receive the exact mass label. It is deterministic for a given evaluation set and has zero across-seed standard deviation for that fixed set. The analytic-profile comparison covers the five diffusion benchmarks.

C.2 COMPUTE ACCOUNTING

MTRG's recurrence depths are $4, 6, 8, 10, 12, 14, 16$ at $T = 10^2, \ldots, 10^8$. Continuous-generator and Refinement make four network evaluations per macro-step; Pushforward makes two. Conditional-query answers each horizon from the initial state. These counts describe per-trajectory work independently of batching.

The physical-time comparison in Section 5.1 measures rollout and projection per trajectory. It excludes reference construction, common-grid resampling and metric computation. The physical and normalised grids contain $32{,}768$ and 256 points, respectively, so the wall-clock measurements in Table 2 include their different spatial costs.