SOURCE ANCHORING FOR PHYSICAL CONSISTENCY IN FLOW MATCHING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep generative models are used to solve partial differential equations and model distributions of physical system states, but ensuring that the generated samples satisfy the governing laws remains challenging. Projection-based flow-matching methods enforce physics by correcting the flow from an unconstrained noise distribution. These corrections shift the generated samples away from the distribution of target solutions, especially in high noise regions. To address this limitation, we propose Source Anchoring for Physical Consistency (SAPC), a Functional Flow Matching method that encodes the physical constraints into the source noise before generation begins. We evaluate SAPC on five systems governed by partial differential equations, covering six tasks with linear and non-linear dynamics, and compare results against five baselines and the unconstrained backbone. Anchoring the source reduces the need for large corrections that drive samples onto admissible but off-distribution states, and SAPC reproduces the target distributions most accurately on every evaluated task, while matching the constraint precision of the best projection-based baselines. Ablation experiments show that this gain arises from pairing source projection with a matched training objective that regresses toward the projected source. These results identify the source distribution as a key design choice for physically consistent generative modelling.

1 INTRODUCTION

Partial Differential Equations (PDEs) govern the evolution of many physical phenomena, and solving them is crucial for numerous real-world applications (Wang et al., 2023a; Ji, 2025). Standard numerical methods approximate PDE solutions by discretising the continuous problem onto a spatio-temporal grid, at a cost that increases with finer resolutions or longer simulations, becoming prohibitive in many regimes (Dai et al., 2026b). Machine learning has emerged as an alternative to conventional PDE solvers, with approaches ranging from Physics-Informed Neural Networks (PINNs) to Neural Operators (NOs) (Li et al., 2021; Lu et al., 2021). These methods learn to derive a single solution for a given set of initial conditions (ICs), boundary conditions (BCs), and PDE parameters. In many settings, however, the problem setup is only partially specified, leaving unresolved degrees of freedom and multiple solutions compatible with the available information. Capturing this variability requires representing a range of possible solutions, which a single prediction cannot provide (Dai et al., 2026a). This limitation motivates the use of generative models, which learn a conditional distribution over PDE solutions rather than a single input–output mapping. Sampling from this distribution yields multiple solutions that are consistent with the available information, and captures the variability arising from the incomplete knowledge of the system (Wang et al., 2023b). Among generative approaches, diffusion and flow-based models including Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) and Flow Matching (FM) (Lipman et al., 2023), have become the primary framework for diverse applications (Rombach et al., 2022; Ho et al., 2022; Kong et al., 2020; Chamberlain et al., 2021), given their ability to represent complex and high-dimensional distributions. Applied to PDEs, these models learn to transform noise into candidate solutions by training on a finite set of reference trajectories. However, learning the statistical structure of the training samples does not guarantee that the generated outputs satisfy the governing PDE or its associated physical constraints (Krishnapriyan et al., 2021; Hansen et al., 2024). Ensuring physical consistency therefore remains a key challenge when using diffusion and flow-matching models as samplers of PDE solutions.

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.

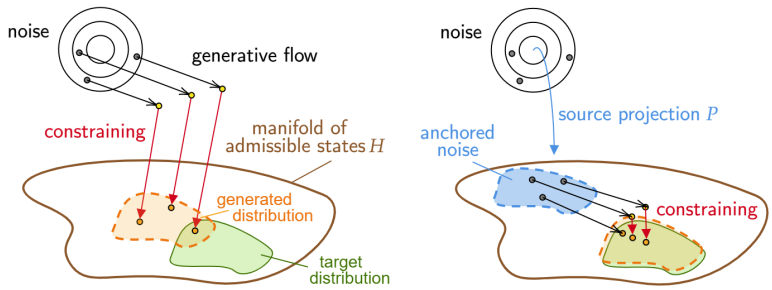


Figure 1: **Left:** existing projection-based flow-matching models correct the diffusion flow onto the manifold of admissible states to enforce physics, and the generated distribution (orange) may drift from the target one (green). **Right:** SAPC projects the noise on the manifold, anchoring it before diffusion starts (blue), mitigating the need for large corrections and reducing the distributional drift.

Different strategies have been proposed to physically constrain diffusion, through training penalties (Bastek et al., 2025; Baldan et al., 2026), architectural design (Beucler et al., 2021), gradient-based guidance (Chung et al., 2023; Huang et al., 2024; Ben-Hamu et al., 2024) and projection (Christopher et al., 2024; Utkarsh et al., 2026; Cheng et al., 2025; Christopher et al., 2026). Within this landscape, projection-based methods have emerged as a promising approach to enforce physical constraints, by mapping intermediate diffusion or final output samples onto the manifold of admissible solutions. These approaches differ in whether projection is incorporated during training or sampling and in where it acts along the generative trajectory. Despite these differences, current projection-based approaches share two main limitations. First, projectors are constructed from the ICs, BCs, and integral conservation laws of the problem, without enforcing the dynamics encoded in the PDE (Hansen et al., 2024; Cheng et al., 2025; Utkarsh et al., 2026; Baldan et al., 2026). The resulting manifold therefore contains all the states that satisfy these constraints, including the true PDE solutions as well as states that are incompatible with the governing dynamics (Rochman Sharabi & Louppe, 2025). Second, generation starts from random noise that does not obey the physical constraints and lies far from the manifold of admissible solutions (Kynkäänniemi et al., 2024; Rojas et al., 2026). When generation starts from an off-manifold source, the learned flow must recover the structure of the target distribution while also overcoming the violation of physical constraints and dynamics. Projection can mitigate these violations, but large corrections may move the samples away from the target distribution while enforcing the prescribed constraints, thus leading to solutions that are physically admissible but off-distribution with respect to the target ones (Figure 1, left).

In this work we propose Source Anchoring for Physical Consistency (SAPC), a Functional Flow Matching (FFM) (Kerrigan et al., 2024) method that projects source samples onto the manifold of admissible solutions $\mathcal{H}$. This keeps the marginals close to $\mathcal{H}$ throughout the flow, limiting the need for large corrective projections and the associated distributional bias (Figure 1, right). The contributions of the work are the following:

- We propose SAPC (Figure 2), a FFM method that anchors source samples onto the constraint manifold $\mathcal{H}$. During training, the network learns to transport anchored source samples to their paired target solutions; at sampling the estimated target solution is additionally projected at every integration step.
- SAPC projects the source at both training and sampling time, and the endpoint estimate at every reverse-time integration step. We show that source anchoring, paired with a training loss on the projected source, is what reduces the distributional bias, while endpoint projection drives constraint violations toward zero.
- We evaluate SAPC on linear and non-linear PDE systems, yielding six tasks: Heat, Reaction–Diffusion, Stokes with initial or boundary conditions, Burgers, and Navier–Stokes. Against five baselines and unconstrained FFM, SAPC reduces distributional bias on every task, while matching the constraint precision of the best projection-based methods.

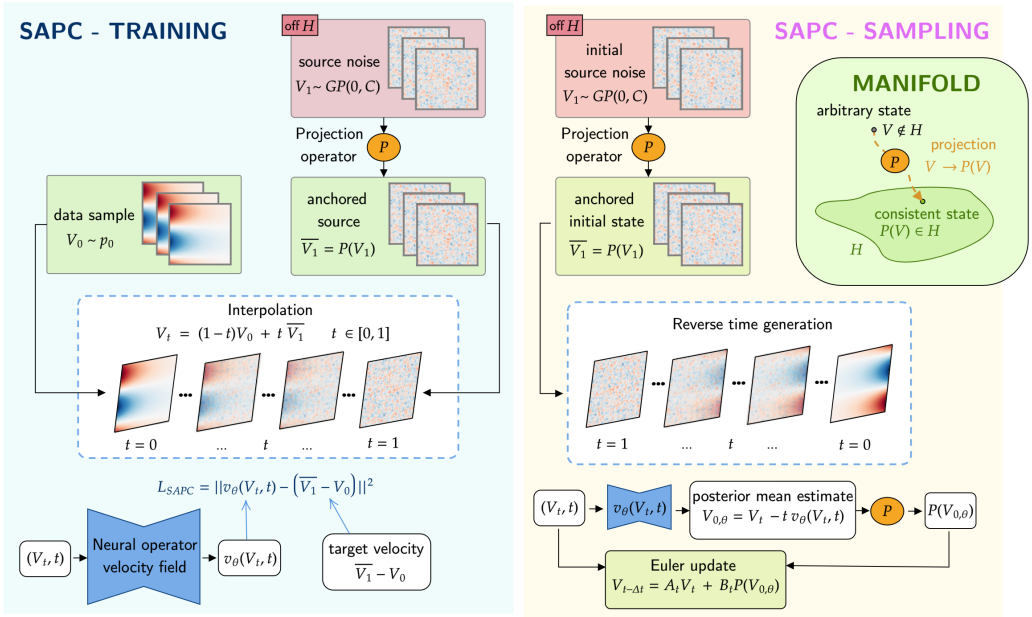


Figure 2: SAPC training (left) and sampling (right). SAPC anchors the source onto the physically-consistent manifold. The posterior-mean estimate is additionally projected at every Euler step.

2 Related work

Different strategies have been proposed to enforce physical knowledge into generative models. Soft constraints augment the training loss with a penalty term proportional to the residual of the governing equation (Li et al., 2024; Bastek et al., 2025; Li et al., 2025; Baldan et al., 2026), whereas hard constraints enforce the physical laws through dedicated output layers (Beucler et al., 2021) or boundary-aware architectures (Wang et al., 2024). Such approaches are tied to the selected system and must be redesigned for each new problem. To improve generalization, a third family of methods enforces the constraint at sampling time, guiding a pre-trained model toward the physically consistent set of solutions. Within this last family, two main categories have emerged. Gradient-based methods steer the sampling trajectory by back-propagating the constraint residual, either by nudging each iterate in the direction that reduces the violation, like classifier guidance (Chung et al., 2023; Huang et al., 2024; Yu et al., 2023), or by optimizing the source point (Ben-Hamu et al., 2024). Projection-based methods instead introduce a projection step of the network's intermediate or denoised estimate onto the manifold of physically admissible states (Hansen et al., 2024; Cheng et al., 2025; Utkarsh et al., 2026; Baldan et al., 2026). Projection does not require calibrating a guidance weight and it avoids back-propagation, reducing computational cost. Existing projection-based methods can be organized along three axes, based on *where* the projection is applied along the diffusion trajectory, whether a *training* phase is included, and *how* the projection is computed with respect to the geometry of the manifold of physically-consistent states. Along the first axis, the projection can act on the posterior-mean estimate (Wang et al., 2023c; Cheng et al., 2025) or on the entire backward diffusion (Christopher et al., 2024). The latter can distort the sampling trajectory when applied at high noise levels (Kynkäänniemi et al., 2024; Rojas et al., 2026). Annealing the projection strength across the diffusion trajectory has been proposed to mitigate this (Narasimhan et al., 2026; Genuist et al., 2026). The projection can also be folded into the training loss (Li et al., 2026; Christopher et al., 2026; Baldan et al., 2026). Whether the projection acts at training or sampling time balances a trade-off between flexibility and consistency. On the one hand, sampling-time projection operates zero-shot on a frozen backbone, and generalises to constraints not seen at training time, but the objective remains that of the unconstrained model and this can degrade sample quality (Christopher et al., 2026). On the other hand, training-time embedding ties the learned weights to the constraint seen during training, reducing generalizability. Finally, when the manifold is a linear subspace, the projector is orthogonal and well-defined; when it is non-linear, the projector is

locally approximated onto the tangent space, as done by PCFM (Utkarsh et al., 2026), which reports improved robustness over ECI (Cheng et al., 2025) on problems with shocks or discontinuities. Despite differences, all existing projection-based methods act on the diffusion trajectory, leaving source samples unconstrained. SAPC anchors the source onto the manifold and folds this into the training loss. This design extends the first axis by introducing a method that projects the source, a setting left unexplored by prior work. Being source projection folded into the training loss, the learned velocity field is aligned with the constrained target distribution. On the third axis, SAPC is independent of the geometry of the constraint manifold and applies to linear and non-linear systems.

3 METHOD

Preliminaries. Many phenomena are governed by PDE systems of the generic form $\mathcal{F}(\mathcal{V}) = 0$, where $\mathcal{V} : \Omega \to \mathbb{R}^d$ is a state of the system defined on the spatio-temporal domain Ω, and $\mathcal{F}$ is a differential operator encoding the physical law. A state $\mathcal{V}$ encodes the spatial configuration of the system and its evolution over time, i.e., the full dynamics, and it is called a *solution* of the PDE when it satisfies the selected law exactly, so that $\mathcal{F}(\mathcal{V}) = 0$. The degree to which a state satisfies $\mathcal{F}(\mathcal{V}) = 0$ defines its *physical consistency*. This condition reduces to a finite set of algebraic checks on $\mathcal{V}$: compliance with the ICs, the BCs, and the integral conservation law $\int_{\Omega'} \mathcal{F}(\mathcal{V})\, dx = 0$ obtained by integrating $\mathcal{F}(\mathcal{V}) = 0$ on subdomains $\Omega' \subseteq \Omega$. To quantify deviations from ICs and BCs, we define a *local* constraint $\mathcal{R}_{\text{loc}}(\mathcal{V}) = \mathcal{V}|_{\Omega_{\text{loc}}} - g$, that measures how much $\mathcal{V}$ deviates from the prescribed IC/BC values g on the subset $\Omega_{\text{loc}} \subset \Omega$ where they apply. To check compliance with the integral conservation law, we also define a *global* residual $\mathcal{R}_{\text{glob}}(\mathcal{V}) = \int_\Omega \mathcal{F}(\mathcal{V})\, dx - c$, that quantifies how much the integral departs from c, with c the value the integral takes on any admissible solution. We stack these into a constraint residual defined as $\mathcal{R}(\mathcal{V}) = [\mathcal{R}_{\text{loc}}(\mathcal{V}),\, \mathcal{R}_{\text{glob}}(\mathcal{V})]$. From $\mathcal{R}(\mathcal{V})$, the manifold $\mathcal{H}$ of physically-consistent states is then defined as:

$$\mathcal{H} \;=\; \left\{\, \mathcal{V} : \Omega \to \mathbb{R}^d \;\middle|\; \mathcal{R}(\mathcal{V}) = 0 \,\right\} \tag{1}$$

Functional Flow Matching and constraint projection. We build our method upon the FFM framework (Kerrigan et al., 2024) (detailed in Appendix A.1). FFM learns a velocity field $v_\theta(\mathcal{V}_t, t)$, where θ denotes the network's trainable parameters, that transports the source noise distribution $p_1 = \mathcal{GP}(0, C)$ at time $t = 1$, defined as a Gaussian process on $L^2(\Omega)$ with zero mean and trace-class covariance operator C, to the target distribution of PDE solutions p_0 at $t = 0$. During training, source $\mathcal{V}_1 \sim p_1$ and target $\mathcal{V}_0 \sim p_0$ samples are connected by the linear interpolant:

$$\mathcal{V}_t(\mathcal{V}_0, \mathcal{V}_1) \;=\; (1-t)\,\mathcal{V}_0 + t\,\mathcal{V}_1, \qquad v_\theta(\mathcal{V}_t \mid \mathcal{V}_0, \mathcal{V}_1) \;=\; \mathcal{V}_1 - \mathcal{V}_0 \tag{2}$$

Standard FFM does not guarantee that generated samples lie on the constraint manifold $\mathcal{H}$, so we enforce $\mathcal{R}(\mathcal{V}) = 0$ through a projection operation. Let $P : L^2(\Omega) \to \mathcal{H}$ denote the metric projection onto $\mathcal{H}$, defined as the optimization problem $P(\mathcal{V}) = \arg\min_{\mathcal{U} \in \mathcal{H}} \|\mathcal{U} - \mathcal{V}\|$. The exact form of P depends on the constraint function $\mathcal{R}(\mathcal{V})$ that defines $\mathcal{H}$. When $\mathcal{R}$ is affine, taking the form $\mathcal{R}(\mathcal{V}) = A\mathcal{V} - b$, being A the linear operator encoding the constraints and b the vector containing their prescribed values, then P is orthogonal to the linear manifold and admits the closed-form $P(\mathcal{V}) = \mathcal{V} - A^\top (AA^\top)^{-1}(A\mathcal{V} - b)$. For non-linear constraints $\mathcal{R}$, a closed-form projection is mathematically intractable. In these cases, P is locally defined onto the tangent space of the curved manifold and approximated numerically using a damped Gauss-Newton algorithm (Utkarsh et al., 2026). The damping parameter ensures convergence, and the updates end once the predefined iteration budget has been reached. Non-linear projection settings are specified in Appendix, Section A.2.

Source Anchoring for Physical Consistency. As shown in Figure 2, SAPC enforces physical consistency on the FFM backbone by projecting two objects onto $\mathcal{H}$: the source draw $\mathcal{V}_1$ at both training and sampling time, and the posterior-mean estimate $\mathcal{V}_{0,\theta}$ at every reverse-time step. During training, the source draw is replaced with its projection onto $\mathcal{H}$, $\bar{\mathcal{V}}_1 = P(\mathcal{V}_1)$, and the FFM is trained to regress the velocity between $\mathcal{V}_0$ and $\bar{\mathcal{V}}_1$:

$$\mathcal{L}_{\text{SAPC}} \;=\; \mathbb{E}_{t,\,\mathcal{V}_0,\,\mathcal{V}_1} \left\| v_\theta(\mathcal{V}_t, t) - \left(\bar{\mathcal{V}}_1 - \mathcal{V}_0\right) \right\|^2, \qquad \mathcal{V}_t \;=\; (1-t)\,\mathcal{V}_0 + t\,\bar{\mathcal{V}}_1 \tag{3}$$

Training is summarised in Algorithm 1. Both endpoints of the diffusion flow lie on $\mathcal{H}$, the target $\mathcal{V}_0$ by construction, the source $\bar{\mathcal{V}}_1$ by projection. The residual along the flow depends on the geometry of $\mathcal{H}$. On the one hand, when $\mathcal{R}$ is affine, it follows from linearity that $\mathcal{R}(\mathcal{V}_t) = (1-t)\,\mathcal{R}(\mathcal{V}_0) +$

$t\,\mathcal{R}(\bar{\mathcal{V}}_1) = 0$ for every $t \in [0,1]$ and the entire diffusion trajectory lies on $\mathcal{H}$. On the other hand, when $\mathcal{R}$ is non-linear, the residual no longer vanishes along the flow but remains bounded:

$$\left\|\mathcal{R}(\mathcal{V}_t)\right\| \;\le\; \tfrac{1}{2}\,t(1-t)\,L\left\|\mathcal{V}_0 - \bar{\mathcal{V}}_1\right\|^2, \qquad L \;=\; \sup_{\mathcal{V}\in[\mathcal{V}_0,\bar{\mathcal{V}}_1]}\left\|\nabla^2\mathcal{R}(\mathcal{V})\right\| \tag{4}$$

The derivation of this last equation is addressed in Appendix A.3. From equation 4, the residual vanishes at both endpoints and grows at most quadratically in between, with a factor $t(1-t)$ that peaks at $t = 1/2$ and decays symmetrically to zero at both ends. In both cases, anchoring the source onto $\mathcal{H}$ keeps the flow near it. As derived in Appendix A.3, the interpolant marginals, i.e., the distributions of $\mathcal{V}_t$ as t varies over $[0,1]$, stay close to $\mathcal{H}$ throughout. Without source anchoring, by contrast, $\mathcal{R}(\mathcal{V}_1) \neq 0$. The residual along the interpolant is $\|\mathcal{R}(\mathcal{V}_t)\| = t\,\|\mathcal{R}(\mathcal{V}_1)\|$ when $\mathcal{H}$ is a linear subspace, and $\|\mathcal{R}(\mathcal{V}_t)\| \le t\,\|\mathcal{R}(\mathcal{V}_1)\| + \mathcal{O}\big(t(1-t)\big)$ when it is curved. In both cases, $\|\mathcal{R}(\mathcal{V}_1)\| \neq 0$ at $t = 1$, so the network has to transport a source that sits a fixed distance off $\mathcal{H}$ onto it during training, rather than refining one that already lies on (or is close to) it.

During sampling, $\mathcal{V}_1$ is again substituted by $\bar{\mathcal{V}}_1 = P(\mathcal{V}_1)$, but projecting the source pins only one endpoint of the flow. The reverse-time iterates $\mathcal{V}_t$ can still drift off $\mathcal{H}$, and the posterior-mean estimate $\mathcal{V}_{0,\theta} = \mathcal{V}_t - t\,v_\theta(\mathcal{V}_t, t)$ inherits any residual bias of the learned velocity field v_θ. During sampling, we therefore replace $\mathcal{V}_{0,\theta}$ with its projection $\bar{\mathcal{V}}_{0,\theta} = P(\mathcal{V}_{0,\theta})$ at each generative step, to steer the update along a constraint-satisfying direction:

$$\mathcal{V}_{t-\Delta t} = A_t\,\mathcal{V}_t + B_t\,\bar{\mathcal{V}}_{0,\theta}, \quad A_t = \frac{t-\Delta t}{t}, \quad B_t = \frac{\Delta t}{t} \tag{5}$$

The full sampling procedure is summarised in Algorithm 2. Endpoint projection is complementary to source anchoring. The latter keeps the training interpolant on or near $\mathcal{H}$ throughout the flow, so the velocity field is regressed on near-consistent states; endpoint projection instead corrects the residual drift that remains along each Euler step. Those sampling-time corrections have been reported to work better when the iterate is already close to the target manifold (Kynkäänniemi et al., 2024; Rojas et al., 2026), as distortions can be introduced when projection is applied at high noise levels.

Algorithm 1 SAPC Training

Require: Data $p_{\mathcal{V}}$, source $\mathcal{GP}(0, C)$, projection P, network v_θ, steps N

for $n = 1$ **to** N **do**
  Sample $\mathcal{V}_0 \sim p_0$, $\mathcal{V}_1 \sim \mathcal{GP}(0, C)$, $t \sim \mathcal{U}[0,1]$
  $\bar{\mathcal{V}}_1 \leftarrow P(\mathcal{V}_1)$
  $\mathcal{V}_t \leftarrow (1-t)\mathcal{V}_0 + t\bar{\mathcal{V}}_1$
  $\mathcal{L} \leftarrow \|v_\theta(\mathcal{V}_t, t) - (\bar{\mathcal{V}}_1 - \mathcal{V}_0)\|^2$
  Update θ by gradient descent on $\mathcal{L}$
end for

Algorithm 2 SAPC Sampling (Euler)

Require: Velocity field v_θ, projection P, steps N

Sample $\mathcal{V}_1 \sim \mathcal{GP}(0, C)$
$\mathcal{V}_1 \leftarrow P(\mathcal{V}_1)$
for $i = N, N-1, \ldots, 1$ **do**
  $\mathcal{V}_{0,\theta} \leftarrow \mathcal{V}_t - t\,v_\theta(\mathcal{V}_t, t)$
  $\mathcal{V}_{t-\Delta t} \leftarrow A_t\,\mathcal{V}_t + B_t\,P(\mathcal{V}_{0,\theta})$
end for
return $\mathcal{V}_0$

4 Empirical evaluation

We evaluate SAPC on five PDE systems from the literature (Cheng et al., 2025; Utkarsh et al., 2026; Christopher et al., 2026), spanning linear and non-linear dynamics and summarised in Table 1: 1D Heat, 1D Reaction-Diffusion (RD), 1D Stokes, 1D Burgers, and 2D Navier-Stokes (NS). Each PDE is paired with a local constraint, defined by an IC, a BC, or a distributional parameter; and a global constraint, involving Linear or Non-Linear Mass Conservation (LMC, NLMC), Non-Linear Energy Conservation (NLEC), or Linear Vorticity Conservation (LVC). The governing equations, the definition of the manifold $\mathcal{H}$ that derives from the chosen constraints, and the sampled parameter ranges are given in Appendix B. For each dataset we generate 6400 training trajectories (10^4 for NS), 1225 validation trajectories, and four disjoint test sets of 1225 trajectories each. The test sets differ only in the local constraint (IC, BC, or distributional parameter) held fixed within the set. To assess stochastic variability, every configuration is drawn under three seeds $\{0,\ 42,\ 205\}$, and all reported metrics are means $\pm$ standard errors across the 3×4 combinations of seed and test sets. We compare SAPC with six competitors that represent the current state of the art in physics-constrained modelling: unconstrained FFM (Kerrigan et al., 2024) as a lower bound; gradient-based D-Flow (Ben-Hamu et al.,

Table 1: Datasets specifications: dimensionality; spatial and temporal domain; spatial and temporal resolution (number of discretization points); local constraint and global conservation law. For the local constraint, either IC/BC or a distributional parameter is held out (like initial vorticity for Navier-Stokes). For the global conservation law either LMC/NLMC, NLEC, or LVC.

Dataset	Dimensionality	Domain	Resolution	Local	Global
Heat	1D	$[0, 2\pi] \times [0, 1]$	100×100	IC	LMC
RD	1D	$[0, 1) \times [0, 1]$	128×100	IC	NLMC
Stokes	1D	$[0, 1] \times [0, 1]$	100×100	IC BC	NLEC
Burgers	1D	$[0, 1) \times [0, 1]$	100×100	IC	LMC
NS	2D	$[0, 1]^2 \times [0, 49]$	$[64, 64] \times 50$	init. vorticity	LVC

2024) (source optimization); soft-constraining PBFM (Baldan et al., 2026) (residual penalty at training); and three sampling-time projection methods, ECI (Cheng et al., 2025), PCFM (Utkarsh et al., 2026), and CAFM (Christopher et al., 2026), with CAFM additionally folding the projection into the training loss. All methods share the same FFM backbone, a Fourier Neural Operator (Li et al., 2021) regressing the conditional velocity v_θ, and the same projector on each dataset. The architecture and other optimization settings are further discussed in Appendix C.1, whilst the implementation of each competitor is detailed in Appendix C.2. For each method we report the pointwise mean and standard deviation for the Mean Square Error (MMSE, SMSE) between generated and reference trajectories. We measure distribution fidelity via the Fréchet Poseidon Distance (FPD), computed on the features extracted from the trajectories by the Poseidon-B encoder (Herde et al., 2024), and the Wasserstein distance W_2, which complements FPD by measuring distributional fidelity pointwise rather than through pooled features. Constraint satisfaction via the constraint error is computed separately on the local and global residuals, $\mathrm{CE}_L = \frac{1}{N}\sum_{n=1}^{N} \|\mathcal{R}_{\mathrm{loc}}(\mathcal{V}_n)\|_2$ and $\mathrm{CE}_G = \frac{1}{N}\sum_{n=1}^{N} \|\mathcal{R}_{\mathrm{glob}}(\mathcal{V}_n)\|_2$, and summarized as CE, computed as the mean of CE_L and CE_G, normalized to the unconstrained FFM ($\mathrm{CE} = 1$ for it). An extended definition of each metric is given in Appendix C.3.

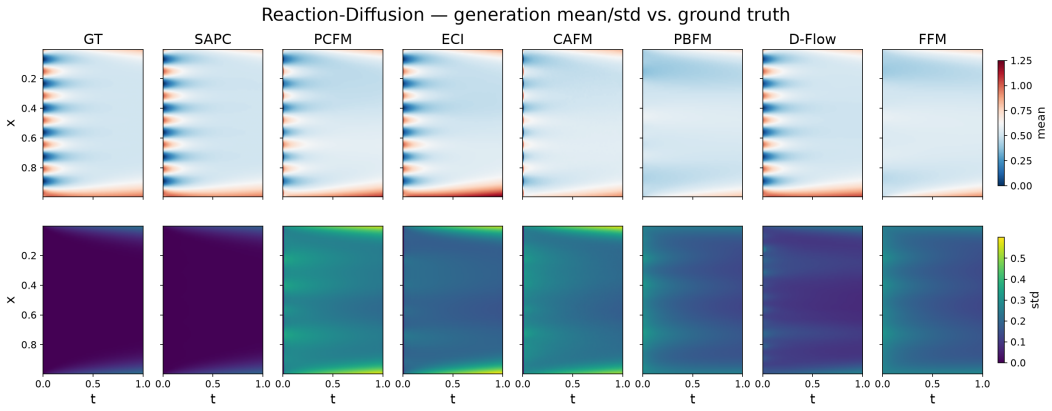


Figure 3: Pointwise mean (top) and standard deviation (bottom) over generated test RD trajectories.

Distributional fidelity. Table 2 shows that SAPC achieves the closest distributional match to the reference, both on the first two marginal moments (MMSE, SMSE) and on FPD, on all the datasets presented in this work. These results suggest that anchoring the source distribution to $\mathcal{H}$ helps the learned flow recover the statistical structure of the reference distribution more faithfully. By contrast, competing methods leave the source noise unconstrained, requiring the flow to bridge the mismatch between physically inconsistent source samples and admissible target solutions. Resolving this mismatch requires large corrective projections that distort the generated distribution, leading to lower distributional metrics. Figure 3 shows that SAPC best reproduces the pointwise mean (top) and standard deviation (bottom) of the ground-truth trajectories (GT, first column) for the RD dataset

Table 2: Generation on PDEs with linear and non-linear dynamics. Each cell reports the mean m and standard error s across seeds and held-out test sets in the compact form $m(s)p$, meaning $(m \pm s) \times 10^p$. For example, $2.2(0.8) - 5$ denotes $(2.2 \pm 0.8) \times 10^{-5}$. Lower values indicate better performance, with best result in bold and second best underlined. Cell shading ranges from dark green (lowest mean) to white (highest mean) on a logarithmic scale, with normalization applied within each row.

	Metric	SAPC	PCFM	ECI	CAFM	PBFM	D-Flow	FFM
Heat	MMSE ↓	**2.2(0.8)−05**	1.2(0.2)−02	1.9(0.6)−02	2.2(0.6)−02	5.8(1.3)−02	8.3(6.0)−02	5.7(1.3)−02
	SMSE ↓	**2.3(0.4)−05**	6.4(0.5)−02	7.7(3.0)−02	1.2(0.3)−01	3.8(0.1)−02	1.5(1.2)+01	3.7(0.1)−02
	FPD ↓	**4.6(1.3)−03**	5.7(0.9)+00	7.7(2.4)+00	1.1(0.3)+01	2.8(0.3)+00	1.5(0.2)+01	2.7(0.3)+00
	CE ↓	**2.0(0.0)−06**	2.0(0.0)−06	2.6(0.1)−06	2.2(0.1)−06	6.5(0.3)−01	2.2(0.9)+00	1
RD	MMSE ↓	**5.8(0.9)−05**	1.9(0.6)−02	5.8(1.6)−03	2.3(0.7)−02	4.1(1.1)−02	5.6(4.0)−02	4.2(1.2)−02
	SMSE ↓	**1.9(0.2)−05**	9.4(1.5)−02	3.4(0.2)−02	5.5(0.1)−02	3.4(0.0)−02	4.8(4.4)+01	3.2(0.1)−02
	FPD ↓	**5.9(2.0)−01**	1.8(0.3)+02	1.4(0.2)+02	1.6(0.2)+02	1.2(0.3)+02	4.1(0.6)+01	1.2(0.3)+02
	CE ↓	**3.3(0.2)−06**	3.3(0.2)−06	3.6(0.2)−06	3.3(0.2)−06	6.9(0.2)−01	3.1(2.7)+01	1
Stokes IC	MMSE ↓	**3.0(1.2)−03**	1.1(0.1)−02	9.8(2.5)−03	1.5(0.2)−02	1.4(0.4)−02	6.3(3.2)−03	1.5(0.4)−02
	SMSE ↓	**1.1(0.4)−03**	2.7(0.3)−02	1.2(0.2)−02	2.7(0.3)−02	1.9(0.2)−02	8.7(7.4)−01	1.9(0.2)−02
	FPD ↓	**2.8(1.2)−01**	3.6(0.4)+00	2.0(0.2)+00	3.7(0.5)+00	7.8(2.1)+00	3.8(1.6)+00	7.9(2.5)+00
	CE ↓	6.1(3.3)−02	6.2(0.9)−01	1.8(0.5)−01	**5.0(1.6)−02**	6.9(0.5)−01	1.0(0.8)+01	1
Stokes BC	MMSE ↓	**1.4(0.1)−03**	1.4(0.3)−02	7.4(1.4)−03	4.4(0.3)−02	5.3(0.8)−02	1.1(0.7)−01	5.3(0.8)−02
	SMSE ↓	**3.8(0.4)−03**	7.5(4.7)−01	7.3(0.8)−03	2.1(0.1)−02	3.5(0.1)−02	1.9(1.6)+01	3.2(0.0)−02
	FPD ↓	**1.7(0.2)+00**	4.7(1.5)+00	2.5(0.9)+00	9.6(1.0)+00	2.2(0.4)+00	1.8(0.3)+01	2.1(0.5)+00
	CE ↓	3.6(0.4)−02	4.6(0.4)−01	2.7(0.4)−02	**1.1(0.2)−06**	6.7(0.2)−01	2.4(2.1)+02	1
Burgers	MMSE ↓	**6.4(1.8)−05**	3.0(0.4)−02	1.5(0.2)−02	2.0(0.2)−02	8.8(1.3)−02	1.5(0.2)−03	9.0(1.2)−02
	SMSE ↓	**5.5(1.1)−05**	1.6(0.3)−01	1.8(0.6)−01	7.6(0.3)−02	4.9(0.1)−02	1.6(0.5)−02	5.0(0.1)−02
	FPD ↓	**1.5(0.4)−02**	1.1(0.3)+01	1.2(0.5)+01	3.4(0.2)+00	2.8(0.4)+00	1.7(0.3)+00	2.8(0.4)+00
	CE ↓	1.7(0.2)−06	1.6(0.0)−06	2.0(0.2)−06	**1.4(0.0)−06**	6.4(0.1)−01	3.4(0.4)−01	1
NS	MMSE ↓	**5.3(0.6)−02**	2.2(0.0)−01	4.4(0.1)−01	2.2(0.1)−01	1.8(0.1)−01	2.3(1.7)−01	1.8(0.1)−01
	SMSE ↓	**3.1(0.4)−02**	1.7(0.1)−01	3.1(0.1)−01	1.7(0.1)−01	6.3(0.3)−02	1.5(1.4)+01	6.9(0.3)−02
	FPD ↓	**1.1(0.1)+00**	3.0(0.1)+00	3.2(0.1)+00	2.8(0.1)+00	2.9(0.1)+00	5.6(1.8)+00	2.7(0.1)+00
	CE ↓	7.3(0.1)−07	**7.0(0.0)−07**	7.2(0.1)−07	7.0(0.1)−07	8.5(0.4)−01	1.2(0.3)+00	1

(other datasets in Appendix E). SAPC improves distributional accuracy on every task, although the magnitude of the gain is dataset-dependent. For each distributional metric and relative to the second-best method, SAPC reduces errors by approximately one to three orders of magnitude on Heat, RD, and Burgers, with smaller gains on the Stokes tasks and 2D NS (Table 2). This trend reflects how strongly the constraint $\mathcal{R}(\mathcal{V}) = 0$ restricts the range of possible solutions in each dataset. For Heat, the IC and mass conservation leave only the diffusivity unspecified within the set of PDE solutions, making the choice of the constraint-enforcement mechanism a key factor in the final result. For NS, fixing the initial vorticity and conserving total vorticity constrains only a limited part of the dynamics, leaving much of the two-dimensional turbulent evolution unspecified by these constraints. All methods must therefore learn the remaining dynamical structure from data, limiting the improvement that constraint enforcement alone can provide. As the constraint no longer determines most of the solution, the choice of enforcement mechanism becomes less critical, and the difference between SAPC and competing methods narrows down.

Constraint errors. Figure 4 compares all methods across the datasets, showing the Wasserstein distance W_2 against the constraint error (CE), normalised by the unconstrained FFM baseline. Both axes use logarithmic scales, and marker colours indicate inference time per sample. SAPC achieves the lowest W_2 across all datasets, extending the distributional improvements observed in MMSE, SMSE, and FPD (Table 2; additional results in Appendix E, Table 7). Two patterns emerge among the competing methods. On the one hand, PCFM, ECI, and CAFM use the same sampling-time projector as SAPC and achieve comparable CE, on the order of 10^{-6}–10^{-7}, on Heat, RD, Burgers,

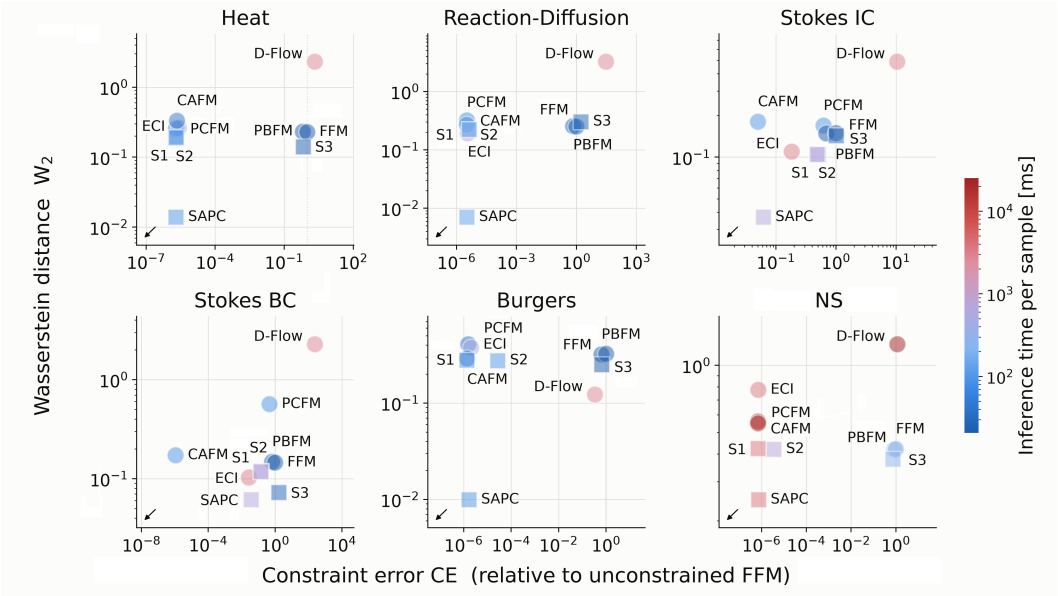


Figure 4: Wasserstein distance W_2 against the constraint error CE (both on log scales), with marker colour encoding wall-clock inference time per sample. Arrows point to the bottom-left corner, where both metrics are jointly minimised. Squares are SAPC and its ablated variants (S1, S2, and S3).

and NS. However, their W_2 values are one to two orders of magnitude higher on Heat, RD and Burgers, and still two to three times higher on NS, showing that comparable constraint satisfaction does not imply comparable agreement with the reference distribution. On the other hand, PBFM and D-Flow remain broadly comparable to unconstrained FFM in both CE and W_2, indicating that neither training-time penalties (PBFM) nor inference-time guidance (D-Flow) achieve the same level of constraint satisfaction and distributional fidelity as SAPC. Stokes BC is the only 1D benchmark where the competing methods show less separation in distributional accuracy. Here, the local constraint specifies the boundary at $x = 0$ rather than the initial spatial profile at $t = 0$. Because the Stokes field decays exponentially with distance, fixing the boundary at $x = 0$ constrains a substantial part of this variability, reducing the differences in distributional accuracy among methods. Stokes BC is also the one task where SAPC does not match the best competitor on CE. Table 7 (Appendix E) shows the gap is concentrated in the global constraint error. Nonetheless, this residual lies at the floor of the discretised energy balance, i.e., the error measured on the exact solutions. Finally, regarding performance, SAPC has an inference cost comparable to that of the other methods on the 1D benchmarks, as indicated by the marker colours in Figure 4. On NS, projection-based methods require comparable inference times, with most of the computational cost arising from projection over the three-dimensional space–time domain. A detailed analysis is provided in Appendix C.4.

Ablation study. Figure 4 also presents three ablated variants of SAPC (named S1, S2, and S3; square markers), designed to assess which part of SAPC contributes to the observed improvements. S1 and S2 project the endpoint estimate at every reverse step without source anchoring; S2 incorporates projection into the training loss, whereas S1 does not. S3 anchors the source only at sampling time on a backbone trained on unconstrained noise. Across all datasets, S1 and S2 achieve CE comparable to PCFM, ECI, and CAFM, but show little improvement in W_2 over the unconstrained backbone. Endpoint projection alone therefore enforces admissibility without resolving distributional bias, while incorporating it into training provides no measurable improvement. S3 yields results similar to FFM and PBFM, with the highest CE among the ablated variants and no improvement in W_2, reflecting the mismatch between the unconstrained training source and the anchored sampling source. Neither component alone reproduces the improvements achieved by SAPC, indicating that reducing both CE and W_2 requires combining source anchoring with a training objective that accounts for the anchored source distribution. Full ablation results are provided in Appendix D.

Trajectory analysis. To examine how source anchoring and endpoint projection jointly influence generation in SAPC, we track three quantities at each sampling step. First, we assess whether the

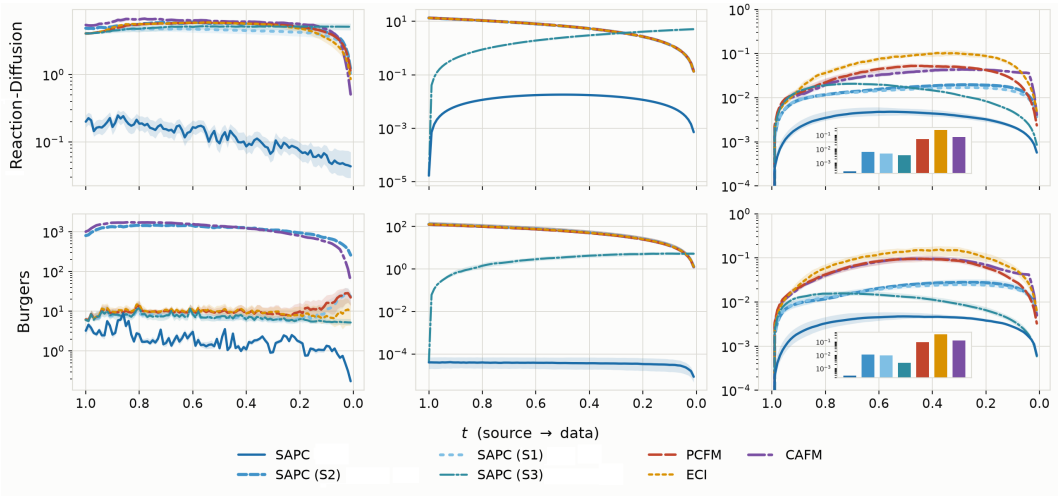


Figure 5: Trajectory analysis (non-linear RD, top, and linear Burgers, bottom). **Left:** residual of endpoint estimate, $\|\mathcal{R}(\mathcal{V}_{0,\theta})\|_2$, before projection. **Middle:** residual $\|\mathcal{R}(\mathcal{V}_t)\|_2$ till last reverse step. **Right:** deviation Δ of the trajectory from the FM interpolant; the inset bars report KE.

network predicts an admissible target by computing the constraint residual of the posterior-mean estimate before projection as a function of time, $|\mathcal{R}(\mathcal{V}_{0,\theta})|_2$. Second, we measure how strongly each iterate violates the constraints defining $\mathcal{H}$, through the residual $|\mathcal{R}(\mathcal{V}_t)|_2$. Third, we quantify the deviation of the flow from the interpolant connecting the source $\mathcal{V}_1$ to the target $\mathcal{V}_0$ as $\Delta = |\mathcal{V}_t - [(1-t)\mathcal{V}_0 + t\mathcal{V}_1]|_2 / |\mathcal{V}_0 - \mathcal{V}_1|_2$, normalising by the source–target distance to enable comparison across datasets. As an integral measure, we also report the excess kinetic energy over the full sampling trajectory, defined as the difference between $\mathrm{KE} = \sum_t |\mathcal{V}_t - \mathcal{V}_{t-\Delta t}|_2^2 / \Delta t$ and the interpolant energy $\mathrm{KE}_{\mathrm{int}} = |\mathcal{V}_0 - \mathcal{V}_1|_2^2$. Figure 5 compares SAPC, its ablated variants (S1–S3), and the projection-based competitors on RD (top) and Burgers (bottom). The curves show the median across generated samples at each sampling step, using a single random seed and data split. The left column shows that SAPC reduces the constraint residual of $\mathcal{V}_{0,\theta}$ by approximately one order of magnitude on RD and two on Burgers relative to unanchored methods throughout sampling. This indicates that the network learns to predict nearly admissible states, leaving only small corrections to the posterior-mean estimate during inference. The middle column provides empirical support for Equation 4. On Burgers, where $\mathcal{R}$ is affine, SAPC maintains a constraint residual on the order of 10^{-5}–10^{-4} throughout sampling. The iterates $\mathcal{V}_t$ therefore lie on the constraint manifold, being those residuals rounding errors of the sampling loop rather than departures from $\mathcal{H}$. On RD, the residual follows the profile predicted by Equation 4, peaking near $t = 1/2$ and decreasing symmetrically towards both endpoints. Finally, the right column shows that SAPC reduces excess KE by two to three orders of magnitude relative to competing methods. On both RD and Burgers, the sampling trajectory remains close to the interpolant connecting the source and target endpoints, whereas competing methods deviate further from it and rely on the sampling-time projector to correct this effect.

5 Conclusion

We introduced Source Anchoring for Physical Consistency (SAPC), a Functional Flow Matching method that projects the source of the flow onto the manifold of physically admissible states, improving distributional fidelity over projection-based, gradient-based, and soft-constraint competitors by one to three orders of magnitude on Heat, Reaction–Diffusion and Burgers, and by a smaller margin on the Stokes tasks and Navier–Stokes, while matching the constraint precision of the best projection-based baselines. The ablation identifies the pairing of source projection with a matched training objective as the mechanism driving these distributional gains. Anchoring the source keeps the flow close to the manifold, so generation refines a near-consistent sample rather than reshaping a noisy one, avoiding the large corrections that projection-based methods apply to restore admissibility and that can drive samples onto admissible but off-distribution states.

AI USE STATEMENT

In this work, we used generative AI tools for create or modify scientific figures or images, create or edit software code, and edit a research paper to improve readability. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, design or provide feedback on research methodology or experiments, interpret results and support qualitative and thematic data analysis. Assist with translation, generate synthetic data sets or clean and reformat dataset are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Code for data generation, training, and sampling of SAPC, its ablation variants, and all baselines is available at `https://anonymous.4open.science/r/SAPC`.

REFERENCES

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Physics vs distributions: Pareto optimal flow matching with physics constraints. In International Conference on Learning Representations, volume 2026, pp. 101215–101243, 2026.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In International Conference on Learning Representations, volume 2025, pp. 3360–3385, 2025.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. Proceedings of Machine Learning Research, 235:3462 – 3483, 2024.

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.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In International Conference on Machine Learning, pp. 1407–1418. PMLR, 2021.

Chaoran Cheng, Boran Han, Danielle Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W Mahoney, and Bernie Wang. Gradient-free generation for hard-constrained systems. In International Conference on Learning Representations, volume 2025, pp. 100510–100539, 2025.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. Advances in Neural Information Processing Systems, 37:89307–89333, 2024.

Jacob K Christopher, James E Warner, and Ferdinando Fioretto. Constraint-aware flow matching: Decision aligned end-to-end training for constrained sampling. arXiv preprint arXiv:2605.12754, 2026.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In International Conference on Learning Representations, 2023.

Yilong Dai, Shengyu Chen, Xiaowei Jia, and Runlong Yu. Flow learners for pdes: Toward a physics-to-physics paradigm for scientific computing. In Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 13164–13169, 2026a.

Yilong Dai, Shengyu Chen, Ziyi Wang, Xiaowei Jia, Yiqun Xie, Vipin Kumar, and Runlong Yu. Learning pde solvers with physics and data: A unifying view of physics-informed neural networks and neural operators. arXiv preprint arXiv:2601.14517, 2026b.

Wilfried Genuist, Éric Savin, Filippo Gatti, and Didier Clouteau. Divergence-free diffusion models for incompressible fluid flows. arXiv preprint arXiv:2601.19368, 2026.

Derek Hansen, Danielle C. Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W. Mahoney. Learning physical models that can respect conservation laws. Physica D: Nonlinear Phenomena, 457:133952, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. Advances in Neural Information Processing Systems, 37:72525–72624, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Advances in Neural Information Processing Systems, 35:8633–8646, 2022.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. Advances in Neural Information Processing Systems, 37: 130291–130323, 2024.

Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. Continuum, 12:1, 2025.

G. Kerrigan, G. Migliorini, and P. Smyth. Functional flow matching. In Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, PMLR, volume 238, pp. 3934–3942, 2024.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. CoRR, abs/2009.09761, 2020.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. Advances in Neural Information Processing Systems, 34:26548–26560, 2021.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. Advances in Neural Information Processing Systems, 37:122458–122483, 2024.

Tianyi Li, Michele Buzzicotti, Fabio Bonaccorso, and Luca Biferale. Physics-constrained diffusion model for synthesis of 3d turbulent data. arXiv preprint arXiv:2603.12834, 2026.

Zeyu Li, Hongkun Dou, Shen Fang, Wang Han, Yue Deng, and Lijun Yang. Physics-aligned field reconstruction with diffusion bridge. In International Conference on Learning Representations, volume 2025, pp. 58864–58900, 2025.

Zongyi Li, Nikola Borislavov 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, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. ACM/IMS Journal of Data Science, 1(3):1–27, 2024.

Yaron Lipman, {Ricky T.Q.} Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In International Conference on Learning Representations, 2023.

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.

Sai Shankar Narasimhan, Shubhankar Agarwal, Litu Rout, Sanjay Shakkottai, and Sandeep Chinchali. Constrained posterior sampling: Time series generation with hard constraints. Advances in Neural Information Processing Systems, 38:107923–107963, 2026.

Omer Rochman Sharabi and Gilles Louppe. Enforcing governing equation constraints in neural pde solvers via training-free projections. In Machine Learning and the Physical Sciences Workshop (NeurIPS 2025), 2025.

Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuhta Takida, Yuki Mitsufuji, and Molei Tao. Improving classifier-free guidance in masked diffusion: Low-dim theoretical insights with high-dim impact. In International Conference on Learning Representations, volume 2026, pp. 33722–33761, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10674–10685. ieee, 2022.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. Advances in Neural Information Processing Systems, 38:160217–160252, 2026.

Haixin Wang, Jiaxin Li, Anubhav Dwivedi, Kentaro Hara, and Tailin Wu. Beno: Boundary-embedded neural operators for elliptic pdes. In International Conference on Learning Representations, volume 2024, pp. 8308–8329, 2024.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. Nature, 620(7972):47–60, 2023a.

Ting Wang, Petr Plechac, and Jaroslaw Knap. Generative diffusion learning for parametric partial differential equations. arXiv preprint arXiv:2305.14703, 2023b.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In International Conference on Learning Representations, 2023c.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 23117–23127. IEEE, 2023.

Supplementary Material

A SAPC CONSTRAINING

A.1 FUNCTIONAL FLOW MATCHING

Flow Matching (FM) (Lipman et al., 2023) is a generative framework that learns a time-dependent vector field $u_t : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ defining a continuous time-dependent diffeomorphism called the flow $\psi_t : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ via the following ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\psi_t(x) \;=\; u_t(\psi_t(x)), \qquad t \in [0,1] \tag{6}$$

that transports a source noise distribution p_1 at $t = 1$ to the target distribution p_0 at $t = 0$ along a prescribed path of marginals p_t satisfying the continuity equation $\partial_t p_t + \nabla \cdot (p_t u_t) = 0$. The vanilla FM objective regresses a learnable vector field v_θ onto the ground truth marginal vector field u_t that generates the path p_t:

$$\mathcal{L}_{\mathrm{FM}} \;=\; \mathbb{E}_{t,x_t\sim p_t(x)}\left[\left\|v_\theta(x_t,t) - u_t(x_t)\right\|^2\right] \tag{7}$$

Generally u_t is not available in closed form, making equation 7 intractable. When conditioned on a pair of endpoints (x_0, x_1), however, the optimal-transport path is the linear interpolant $x_t = (1-t)x_1 + tx_0$, along which the conditional vector field is constant, $u_t(x_t \mid x_0, x_1) = x_0 - x_1$. Regressing v_θ onto this conditional velocity yields the same gradients in θ as equation 7 (Lipman et al., 2023), giving the tractable conditional FM training objective:

$$\mathcal{L}_{\mathrm{CFM}} \;=\; \mathbb{E}_{t,x_0,x_1}\left[\left\|v_\theta(x_t,t) - (x_0 - x_1)\right\|^2\right] \tag{8}$$

with $t \sim \mathcal{U}[0,1]$, $x_0 \sim p_0$, and $x_1 \sim p_1$. Generation reduces to integrating equation 6 backward in t from a source draw $x_1 \sim p_1$ using the learned vector field v_θ.

Functional Flow Matching (FFM) (Kerrigan et al., 2024) extends FM to function space, so that the transported objects are the PDE solutions $\mathcal{V} : \Omega \to \mathbb{R}$, with Ω the spatio-temporal domain of the system. Specifically, FFM learns a time-dependent vector field operator $v_t : \mathcal{U} \times [0,1] \to \mathcal{U}$ over a real separable Hilbert space $\mathcal{U}$ of functions $\mathcal{V}$, that defines a time-dependent diffeomorphism $\psi_t : \mathcal{U} \times [0,1] \to \mathcal{U}$, called the flow, via the ODE:

$$\partial_t \psi_t(\mathcal{V}) \;=\; v_t(\psi_t(\mathcal{V})), \qquad t \in [0,1] \tag{9}$$

The source noise distribution p_1 is a fixed noise measure over $\mathcal{U}$, induced by a Gaussian process with trace-class covariance operator C built from a Matérn kernel. The flow transports p_1 at $t = 1$ to the target state distribution $p_0(\mathcal{V})$ at $t = 0$. As in FM, the vanilla FFM objective is intractable. Under regularity conditions on p_0 and p_1 (Kerrigan et al., 2024), the same conditional construction yields a tractable objective along the optimal-transport path of measures, where the ground truth states are linearly interpolated with source draws as $\mathcal{V}_t = (1-t)\mathcal{V}_1 + t\mathcal{V}_0$:

$$\mathcal{L}_{\mathrm{FFM}} \;=\; \mathbb{E}_{t,\mathcal{V}_0,\mathcal{V}_1}\left[\left\|v_\theta(\mathcal{V}_t,t) - (\mathcal{V}_0 - \mathcal{V}_1)\right\|^2\right] \tag{10}$$

with $t \sim \mathcal{U}[0,1]$, $\mathcal{V}_0 \sim p_0$, $\mathcal{V}_1 \sim p_1$, and the norm the standard $L^2(\Omega)$ norm for square-integrable functions. Generation reduces to integrating equation 9 backward in t from a source draw $\mathcal{V}_1 \sim p_1$ using the learned vector field operator v_θ.

A.2 PROJECTION

For each benchmark, $\mathcal{H}$ embeds two constraints: a *local* constraint pinning IC/BC (or initial vorticity for Navier-Stokes); and a *global* constraint enforcing an integral conservation law along the whole trajectory. Their residuals are concatenated into a single $\mathcal{R}(\mathcal{V}) = [\mathcal{R}_{\mathrm{loc}}(\mathcal{V}),\, \mathcal{R}_{\mathrm{glob}}(\mathcal{V})]$. The projector P realizing the metric projection onto $\mathcal{H}$ then takes one of two following forms, depending on whether the stacked residual $\mathcal{R}(\mathcal{V})$ is affine.

Linear form. When both blocks $\mathcal{R}_{\mathrm{loc}}(\mathcal{V})$ and $\mathcal{R}_{\mathrm{glob}}(\mathcal{V})$ are affine, $\mathcal{R}(\mathcal{V}) = A\mathcal{V} - b$, so P reduces to the orthogonal projector:

$$P(\mathcal{V}) \;=\; \mathcal{V} - A^\top (AA^\top)^{-1}(A\mathcal{V} - b) \tag{11}$$

This is the case for the Heat, Burgers, and Navier-Stokes datasets. The Gram matrix $AA^\top$ is factorized once per benchmark via a Cholesky decomposition, cached, and reused across all projections. We fall back to a pivoted LU factorization when Cholesky fails, and to a pseudo-inverse when $AA^\top$ is rank-deficient. The factorisation and solve are carried out in `float64`.

Non-linear form. If either block $\mathcal{R}_{\text{loc}}(\mathcal{V})$ or $\mathcal{R}_{\text{glob}}(\mathcal{V})$ is non-linear, the concatenated residual $\mathcal{R}(\mathcal{V})$ is projected by Gauss-Newton. Starting from the pre-projection point $\mathcal{V}_0$, the update:

$$\mathcal{V}_{k+1} \;=\; \mathcal{V}_0 \;-\; J_k^\top \big(J_k J_k^\top + \mu I\big)^{-1} \big(h(\mathcal{V}_k) - J_k(\mathcal{V}_k - \mathcal{V}_0)\big), \qquad J_k = \nabla \mathcal{R}(\mathcal{V}_k) \tag{12}$$

is iterated for at most n_{iter} steps, with Tikhonov damping $\mu = 10^{-8}$ and early stopping when $\|\mathcal{R}(\mathcal{V}_k)\|_\infty < 10^{-6}$. We set $n_{\text{iter}} = 3$ on Reaction-Diffusion and $n_{\text{iter}} = 10$ on Stokes. This implementation follows the one reported by Utkarsh et al. (2026). A single iteration ($n_{\text{iter}} = 1$) recovers the tangent-space linearised projection at $\mathcal{V}_0$ used by ECI (Cheng et al., 2025).

A.3 Residual bounds for source anchoring

Residual along the interpolant. Let $\mathcal{V}_0, \bar{\mathcal{V}}_1 \in \mathcal{H}$, so that $\mathcal{R}(\mathcal{V}_0) = \mathcal{R}(\bar{\mathcal{V}}_1) = 0$, and let $\mathcal{V}_t = (1-t)\mathcal{V}_0 + t\bar{\mathcal{V}}_1$ for $t \in [0,1]$. When $\mathcal{R}$ is affine, $\mathcal{R}(\mathcal{V}) = A\mathcal{V} - b$, and linearity gives:

$$\mathcal{R}(\mathcal{V}_t) = (1-t)\,\mathcal{R}(\mathcal{V}_0) + t\,\mathcal{R}(\bar{\mathcal{V}}_1) = 0, \qquad \forall\, t \in [0,1] \tag{13}$$

so the entire flow trajectory lies on $\mathcal{H}$. When $\mathcal{R}$ is C^2 and non-linear, we define $\Delta = \bar{\mathcal{V}}_1 - \mathcal{V}_0$ and $\phi(t) = \mathcal{R}(\mathcal{V}_t) = \mathcal{R}(\mathcal{V}_0 + t\Delta)$, so that $\phi(0) = \phi(1) = 0$. Differentiating twice along the segment yields[1]:

$$\phi''(t) \;=\; \Delta^\top \nabla^2 \mathcal{R}(\mathcal{V}_t)\,\Delta, \qquad \|\phi''(t)\| \;\le\; L\,\|\Delta\|^2, \qquad L \;=\; \sup_{\mathcal{V} \in [\mathcal{V}_0, \bar{\mathcal{V}}_1]} \big\|\nabla^2 \mathcal{R}(\mathcal{V})\big\| \tag{14}$$

Taylor's formula with integral remainder at $t = 0$ is $\phi(t) = t\,\phi'(0) + \int_0^t (t-s)\,\phi''(s)\,ds$. Evaluating at $t = 1$ and using $\phi(1) = 0$ gives $\phi'(0) = -\int_0^1 (1-s)\,\phi''(s)\,ds$, that substituted back gives the Green's-function representation:

$$\phi(t) \;=\; -\int_0^1 G(t,s)\,\phi''(s)\,ds, \qquad G(t,s) \;=\; \begin{cases} s(1-t), & 0 \le s \le t \\ t(1-s), & t \le s \le 1 \end{cases} \tag{15}$$

of the two-point boundary problem $-\phi'' = f$ with homogeneous Dirichlet endpoints. Combining $\|\phi''(t)\| \le L\|\Delta\|^2$ with $\int_0^1 G(t,s)\,ds = \frac{1}{2}t(1-t)$ yields

$$\big\|\mathcal{R}(\mathcal{V}_t)\big\| \;\le\; \tfrac{1}{2}\,t(1-t)\,L\,\big\|\mathcal{V}_0 - \bar{\mathcal{V}}_1\big\|^2 \tag{16}$$

which is the bound stated in equation 4.

Bounding the interpolant marginals. From equation 16, a line $(1-t)\mathcal{V}_0 + t\bar{\mathcal{V}}_1$ drifts off $\mathcal{H}$ by at most $\frac{1}{2}t(1-t)L\|\mathcal{V}_0 - \bar{\mathcal{V}}_1\|^2$. The claim we make in the main text is about the interpolant marginals p_t, i.e. the distributions of $\mathcal{V}_t = (1-t)\mathcal{V}_0 + t\bar{\mathcal{V}}_1$ as t varies over $[0,1]$. Bridging the two requires taking expectations over $(\mathcal{V}_0, \bar{\mathcal{V}}_1)$. The endpoint gap has finite second moment. Indeed,

$$\mathbb{E}\big\|\mathcal{V}_0 - \bar{\mathcal{V}}_1\big\|^2 \;\le\; 2\big(\mathbb{E}\|\mathcal{V}_0\|^2 + \mathbb{E}\|\bar{\mathcal{V}}_1\|^2\big) \;=:\; M \;<\; \infty \tag{17}$$

where $\mathbb{E}\|\mathcal{V}_0\|^2 < \infty$ holds by assumption on $p_{\mathcal{V}}$ (because the solutions of all PDE datasets considered in this study are bounded on the compact spatio-temporal domain Ω), and $\mathbb{E}\|\bar{\mathcal{V}}_1\|^2 < \infty$ is inherited from $\mathcal{GP}(0, C)$ via the projection[2]. Taking expectations of equation 16 therefore yields the marginal residual bound

$$\mathbb{E}_{\mathcal{V}_t \sim p_t}\big\|\mathcal{R}(\mathcal{V}_t)\big\| \;\le\; \tfrac{1}{2}\,t(1-t)\,L\,M \tag{18}$$

From Markov's inequality it follows, for every $\varepsilon > 0$:

$$\Pr\big(\|\mathcal{R}(\mathcal{V}_t)\| \le \varepsilon\big) = 1 - \Pr\big(\|\mathcal{R}(\mathcal{V}_t)\| > \varepsilon\big) \ge 1 - \frac{t(1-t)\,L\,M}{2\,\varepsilon} \;\xrightarrow[t \to 0,\,1]{}\; 1 \tag{19}$$

Most samples drawn from p_t therefore have a residual $\|\mathcal{R}(\mathcal{V}_t)\|$ below the threshold ε, with only a small fraction violating it; that fraction shrinks as t approaches either endpoint of the flow.

[1] The constant L is finite because the segment $[\mathcal{V}_0, \bar{\mathcal{V}}_1] = \{\mathcal{V}_0 + t\Delta : t \in [0,1]\}$ is compact in $L^2(\Omega)$ and $\nabla^2\mathcal{R}$ is continuous on it by the C^2 assumption, so the upper limit exists.

[2] P preserves finite second moments: 1-Lipschitz in the affine case, locally Lipschitz in the C^2 non-linear case by the implicit function theorem applied to equation 12. Together with $\mathbb{E}\|\mathcal{V}_1\|^2 = \mathrm{tr}(C) < \infty$ this gives $\mathbb{E}\|\bar{\mathcal{V}}_1\|^2 < \infty$.

B Dataset generation and definition of constraint manifolds

This section provides a detailed description of the datasets used in this work and their generation. To monitor sensitivity to the constraint enforced at test time, we generate, for each dataset, four held-out test sets of 1225 trajectories each, differing only in the quantity held fixed within the set: the phase ϕ of the IC on Heat; the IC on RD and on Burgers; the initial vorticity on NS; the decay rate k on Stokes IC, and the oscillation frequency ω on Stokes BC. The remaining parameter is redrawn independently per trajectory within the set (the diffusivity α on Heat, the viscosity ν on Burgers, the forcing phase on NS, ω on Stokes IC and k on Stokes BC) so each test set is a distribution of solutions under a single fixed constraint rather than a single trajectory. See the corresponding subsections for more details. Each split (training, validation, and the four held-out test sets) is drawn from an independent random stream.

B.1 Heat Equation

We consider the one-dimensional heat equation with periodic boundary conditions:

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2}, \quad x \in [0, 2\pi], \quad t \in [0, 1] \tag{20}$$

$$u(x,0) = \sin(x+\phi), \quad u(0,t) = u(2\pi,t) \tag{21}$$

The equation describes the distribution of heat $u(x,t)$ along a ring with non-uniform initial temperature $u(x,0)$. As time passes, the heat redistributes and smooths out by internal conduction, with closed-form solution $u(x,t) = e^{-\alpha t}\sin(x+\phi)$.

To train and validate the models, we generate 6400 and 1225 train and validation trajectories respectively, with the diffusion coefficient and phase sampled uniformly and independently for each trajectory, $\alpha \sim U[1,5]$ and $\phi \sim U[0,\pi]$, following the same setting as (Cheng et al., 2025; Utkarsh et al., 2026). To evaluate the models, we generate four held-out test sets of 1225 trajectories each. Within a test set the phase is fixed to one of $\phi \in \{\pi/4,\ \pi/2,\ 3\pi/4,\ \pi\}$, so that all of its trajectories share the same initial condition $u_{\text{IC}}(x,0) = \sin(x+\phi)$, while the diffusivity $\alpha \sim U[1,5]$ is redrawn independently for each trajectory. Each test set is therefore a distribution over solutions under a **fixed IC**, and the test sets differ only in which IC is held fixed (Figure 6).

The projector is defined to map the diffusion trajectory onto the manifold of states consistent with the fixed IC and the **Linear Mass Conservation (LMC)** of the system:

$$\mathcal{H} = \left\{ u : [0,2\pi]\times[0,1] \to \mathbb{R} \;\middle|\; \begin{array}{l} u(x,0) = u_{\text{IC}}(x), \ \ \forall\, x \in [0,2\pi]\,; \\ \int_0^{2\pi} u(x,T)\,dx - \int_0^{2\pi} u(x,0)\,dx = 0, \ \ \forall\, T \in [0,1] \end{array} \right\} \tag{22}$$

being $m(t) = \int_0^{2\pi} u(x,t)\,dx$ the mass of the system.

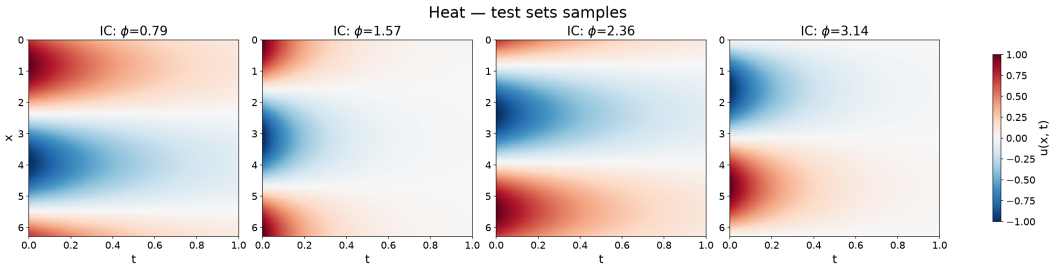


Figure 6: Representative test-set trajectories of the heat benchmark, one per held-out IC. Each panel is drawn from a different test split, pinned to a distinct initial phase $\phi \in \{\pi/4, \pi/2, 3\pi/4, \pi\}$, while the diffusivity α is drawn from the training range $[1,5]$ within each split. The heat solution $u(x,t) = e^{-\alpha t}\sin(x+\phi)$ decays exponentially in time while preserving the spatial profile set by ϕ.

B.2 Reaction-Diffusion (RD) Equation

We consider the non-linear one-dimensional reaction-diffusion equation with Neumann boundary conditions:

$$\frac{\partial u(x,t)}{\partial t} = \nu \frac{\partial^2 u(x,t)}{\partial x^2} + \rho\, u(x,t)\big(1 - u(x,t)\big), \quad x \in [0,1], \quad t \in [0,1] \tag{23}$$

$$-\nu \left.\frac{\partial u(x,t)}{\partial x}\right|_{x=0} = g_L, \qquad -\nu \left.\frac{\partial u(x,t)}{\partial x}\right|_{x=1} = g_R \tag{24}$$

The equation describes the evolution of a population density (or concentration) $u(x,t)$ diffusing along a 1D strip with diffusivity ν, while growing locally according to logistic reaction kinetics $\rho\, u(1-u)$ with growth rate ρ, saturating at the carrying capacity $u = 1$. We set $\nu = 0.005$ and $\rho = 0.01$, as in (Utkarsh et al., 2026). The boundary fluxes at the left and right ends of the domain are specified as g_L and g_R, respectively. The initial condition $u_{\text{IC}}(x,0)$ is sampled from a randomized combination of sinusoidal and localized bump functions. Specifically, each initial condition is drawn as a random superposition of up to three sinusoidal modes, with amplitudes and phases sampled uniformly. With probability 0.1 the profile is additionally localized by a smooth $\tanh$ window, producing bump-like states. Every realization is rescaled to $[0,1]$ range.

To train and validate the models, we generate 6400 and 1225 train and validation trajectories respectively, by combining 80 (or 35) initial conditions u_{IC} with 80 (or 35) boundary conditions, with (g_L, g_R) sampled uniformly, $g_L \sim U[0, 0.05]$ and $g_R \sim U[-0.05, 0]$. To evaluate the models, we generate four held-out test sets of 1225 trajectories each. Every test set pins a u_{IC}, drawn from the same distribution, and pairs it with 1225 independent boundary-flux draws. Each test set is therefore a distribution over solutions sharing one **fixed IC**, and the test sets differ only in which IC is held fixed (Figure 7).

The projector is defined to map the diffusion trajectory onto the manifold of states consistent with the fixed IC and the **Non-Linear Mass Conservation (NLMC)** of the system:

$$\mathcal{H} = \left\{ u : [0,1]\times[0,1] \to \mathbb{R} \;\middle|\; \begin{array}{l} u(x,0) = u_{\text{IC}}(x), \;\; \forall\, x \in [0,1]\,; \\ \int_0^1 \big[u(x,T) - u(x,0)\big]\, dx = \int_0^T \big(g_L - g_R\big)\, ds + \\ \quad +\rho \int_0^T\!\!\int_0^1 u(x,s)\big(1-u(x,s)\big)\, dx\, ds, \;\; \forall\, T \in [0,1] \end{array} \right\} \tag{25}$$

being $m(t) = \int_0^1 u(x,t)\, dx$ the mass of the system. The constraint equation 25 involves the boundary fluxes (g_L, g_R), which are unknown for a field produced by the generative model. At evaluation time we therefore estimate them from the field itself by one-sided finite differences of the Neumann condition, $g_L = -\nu\, \partial_x u|_{x=0} \approx -\nu\,(u_1 - u_0)/\Delta x$ and $g_R = -\nu\, \partial_x u|_{x=1} \approx -\nu\,(u_{N-1} - u_{N-2})/\Delta x$. This estimator is first-order accurate and evaluated one cell away from the boundary, so the resulting residual retains an $O(\Delta x)$ bias: applied to the exact solutions of the test sets it yields a non-zero constraint error, between 1.7×10^{-2} and 1.0×10^{-1} depending on the test set. We therefore read the NLMC constraint error as a *relative* measure of mass-balance violation, comparable across methods, rather than as an absolute deviation from the physical balance.

B.3 Stokes Problem

We consider Stokes' second problem, describing the unsteady flow of a viscous incompressible fluid generated by the oscillatory motion of a flat plate:

$$\frac{\partial u(x,t)}{\partial t} = \nu \frac{\partial^2 u(x,t)}{\partial x^2}, \quad x \in [0,1], \quad t \in [0,1] \tag{26}$$

$$u(x,0) = A e^{-kx} \cos(kx), \qquad u(0,t) = A\cos(\omega t) \tag{27}$$

where $u(x,t)$ is the fluid velocity parallel to the plate, x the distance from the plate, and ν the viscosity. The plate is set into motion at $t = 0$ with sinusoidal velocity of amplitude $A = 2$ and oscillation frequency ω, with $k = \sqrt{\omega/(2\nu)}$.

To train and validate the models, we generate 6400 and 1225 train and validation trajectories respectively, by uniformly and independently sampling $\omega \sim U[2,8]$ and $k \sim U[2,20]$ for each trajectory,

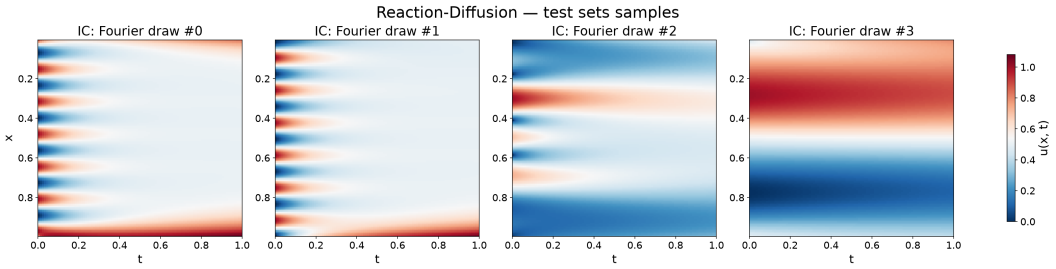


Figure 7: Representative test-set trajectories of the RD benchmark, one per held-out IC. Each panel is drawn from a different test split, pinned to a distinct random low-mode Fourier draw $u_{\text{IC}}(x, 0)$, while the Neumann boundary fluxes (g_L, g_R) vary per sample, drawn from the same training ranges within each split. The solution's dynamics are dominated by diffusive smoothing of the initial fine-scale Fourier structure and by mass injected or removed through the Neumann boundary fluxes.

following the same setting as (Cheng et al., 2025). The two parameters are sampled independently to decouple the IC (governed by k) from the BC (governed by ω); the viscosity is then determined by $\nu = \omega/(2k^2)$. To evaluate the models, we consider two separate Stokes experiments: in the **IC task**, $u_{\text{IC}}(x, 0) = Ae^{-kx}\cos(kx)$ is held out through the decay rate k, while in the **BC task** the oscillating boundary $u_{\text{BC}}(0, t) = A\cos(\omega t)$ is held out through the frequency ω. To this end, we generate four held-out test sets of 1225 trajectories each per task. In the IC task, each test set fixes $k \in \{4, 8, 12, 16\}$, so that all of its trajectories share the same $u_{\text{IC}}(x, 0)$, while $\omega \sim U[2, 8]$ is still redrawn per trajectory (Figure 8, top). In the BC task, each test set fixes $\omega \in \{3, 4.5, 6, 7.5\}$, so that all of its trajectories share the same $u_{\text{BC}}(0, t)$, while $k \sim U[2, 20]$ is redrawn per trajectory (Figure 8, bottom). Each test set is therefore a distribution over solutions under a single fixed constraint.

The projector is defined to map the diffusion trajectory onto the manifold of states consistent with the constraint (either BC or IC) and the **Non-Linear Energy Conservation (NLEC)** of the system. Multiplying equation 26 by u and integrating over the spatial domain gives the energy balance:

$$\mathcal{H} = \left\{ u : [0,1]\times[0,1] \to \mathbb{R} \;\middle|\; \begin{array}{l} u(x,0) = u_{\text{IC}}(x), \;\; \forall\, x \in [0,1] \quad \text{(for IC task)} \\ u(0,t) = A\cos(\omega t), \;\; \forall\, t \in [0,1] \quad \text{(for BC task)} \\ \int_0^1 u^2(x,T)\,dx - \int_0^1 u^2(x,0)\,dx = \\ 2\nu \int_0^T \left[u(1,t)\,\frac{\partial u(1,t)}{\partial x} - u(0,t)\,\frac{\partial u(0,t)}{\partial x} \right] dt + \\ -\,2\nu \int_0^T \int_0^1 \left(\frac{\partial u(x,t)}{\partial x} \right)^2 dx\,dt, \;\; \forall\, T \in [0,1] \end{array} \right\} \tag{28}$$

being $E(t) = \int_0^1 u^2(x, t)\,dx$ the energy of the system, the first integral on the right side of the energy balance equation 28 the net boundary work done on the fluid, and the second the viscous dissipation. Energy balance depends on the viscosity ν, which is unknown for a generated field. During generation, we therefore estimate it per sample by least squares on the governing equation 26, i.e. $\hat{\nu} = \langle \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2} \rangle / \langle \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x^2} \rangle$, with the spatial and temporal derivatives taken by finite differences. On the ground-truth fields $\hat{\nu}$ recovers the true viscosity $\nu = \omega/(2k^2)$ to within 4.7-5.4% on the BC task and 3.7-16.4% on the IC task, the latter degrading with the held-out decay rate k as the field concentrates into an increasingly thin boundary layer. The residual of equation 28 is accumulated over the trajectory, so the first-order errors of the finite-difference derivatives do not cancel: applied to the exact solutions it yields a constraint error between 8.5×10^{-2} and 1.1×10^{-1} across the eight held-out test sets, rather than zero. Substituting the exact ν in place of $\hat{\nu}$ raises it further, to $1.27\text{-}1.40 \times 10^{-1}$, confirming that the accumulation, not the viscosity estimate, sets this floor. The NLEC constraint error should therefore be read as a comparison between methods, all scored with the same residual, and not as an absolute measure of physical energy imbalance.

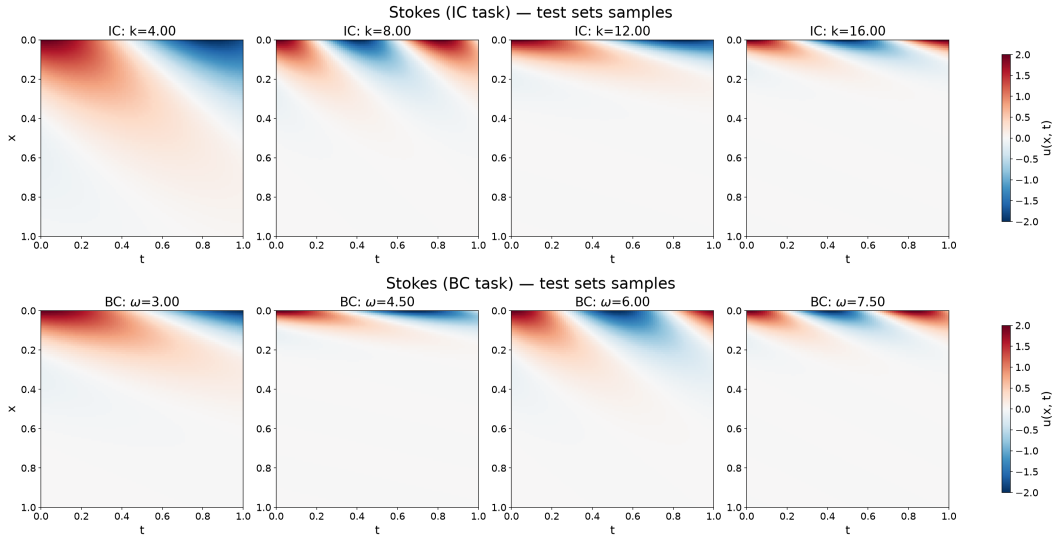


Figure 8: Representative test-set samples for the two Stokes tasks. Both solve the classic Stokes velocity field $u(x,t)$, a wave that decays exponentially away from the oscillating boundary at $x = 0$, with decay rate and spatial wave number both set by k and ω. **IC task (top):** One sample from each of the four held-out test sets, each pinning a distinct decay rate $k \in \{4, 8, 12, 16\}$, while ω varies per sample, drawn from the training range $[2, 8]$. Larger k gives a thinner boundary layer and faster spatial oscillation near $x = 0$. **BC task (bottom):** One sample from each of the four held-out test sets, each pinning a distinct frequency $\omega \in \{3, 4.5, 6, 7.5\}$, while k varies within its training range $[2, 20]$ per sample. Larger ω completes more oscillation periods over $t \in [0, 1]$.

B.4 Burgers' Equation

We consider the one-dimensional viscous Burgers' equation with periodic boundary conditions:

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\,\frac{\partial u(x,t)}{\partial x} = \nu\,\frac{\partial^2 u(x,t)}{\partial x^2}, \quad x \in [0,1], \quad t \in [0,1] \tag{29}$$

$$u(x,0) = u_{\text{IC}}(x), \qquad u(0,t) = u(1,t) \tag{30}$$

The equation combines non-linear advection, $u\,\frac{\partial u(x,t)}{\partial x}$, with linear diffusion of strength ν, and is a canonical simplified version of the Navier-Stokes equations modelling the interplay between convection and viscosity. The initial condition $u_{\text{IC}}(x,0)$ is sampled from a randomized superposition of low-frequency Fourier modes.

To train and validate the unconstrained model, we generate 6400 and 1225 train and validation trajectories respectively, with viscosity sampled uniformly, $\nu \sim \mathcal{U}[0.01, 0.05]$, and initial condition realizations drawn independently for each trajectory as randomized superpositions of four low-frequency Fourier modes, rescaled to $[0, 1]$ range.

To evaluate the models, we generate four held-out test sets of 1225 trajectories each. Every test set pins a u_{IC}, drawn from the same Fourier distribution, while the viscosity keeps varying over $\mathcal{U}[0.01, 0.05]$ within the split. Each test set is therefore a distribution over solutions sharing one **fixed IC**, and the test sets differ only in which IC is held fixed (Figure 9).

The projector is defined to map the diffusion trajectory onto the manifold of states consistent with the fixed IC and the **Linear Mass Conservation (LMC)** of the system. Indeed, since the domain is periodic, the non-linear advective flux integrates to zero over $[0, 1]$, so mass is exactly conserved:

$$\mathcal{H} = \left\{ u : [0,1]\times[0,1] \to \mathbb{R} \;\middle|\; \begin{array}{l} u(x,0) = u_{\text{IC}}(x), \;\; \forall\, x \in [0,1] \\ \displaystyle\int_0^1 u(x,T)\,dx - \int_0^1 u(x,0)\,dx = 0, \;\; \forall\, T \in [0,1] \end{array} \right\} \tag{31}$$

being $m(t) = \int_0^1 u(x,t)\,dx$ the mass of the system.

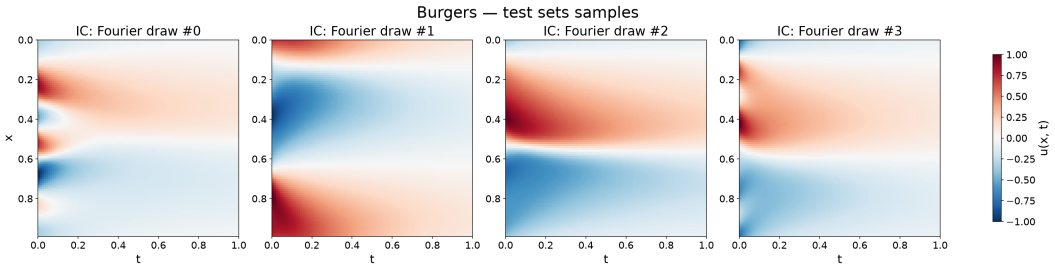


Figure 9: Representative test-set trajectories of the Burgers benchmark. Each panel is drawn from a different test split, pinned to a distinct random low-mode Fourier draw $u_{\mathrm{IC}}(x, 0)$, while the viscosity ν varies per sample, drawn from the training range $[0.01, 0.05]$. Non-linear advection steepens the initial profile within the first few time steps, producing sharp layers whose thickness is set by ν; the field then relaxes into near-uniform plateaus separated by these thin layers from then on.

B.5 Navier-Stokes (NS) Equation

We consider the two-dimensional Navier-Stokes equation for a viscous, incompressible fluid in vorticity form with periodic boundary conditions:

$$\frac{\partial w(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t)\cdot\nabla w(\mathbf{x},t) = \nu\,\Delta w(\mathbf{x},t) + f(\mathbf{x}), \quad \mathbf{x}\in[0,1]^2, \quad t\in[0,49] \tag{32}$$

$$\nabla\cdot\mathbf{u}(\mathbf{x},t) = 0, \qquad w(\mathbf{x},0) = w_0(\mathbf{x}) \tag{33}$$

where $\mathbf{u}(\mathbf{x},t)$ is the velocity field, $w = \nabla\times\mathbf{u}$ is the vorticity, w_0 is the initial vorticity, ν is the viscosity, and $f(\mathbf{x})$ is a fixed forcing term. The incompressibility constraint $\nabla\cdot\mathbf{u} = 0$ enforces conservation of mass, and the domain is treated with periodic boundary conditions in both spatial directions.

As in (Utkarsh et al., 2026), training and validation trajectories are obtained by pairing initial vorticities w_0, sampled from a Gaussian random field, with forcing realizations $f(\mathbf{x}) = 0.1\sqrt{2}\sin(2\pi(x_1+x_2)+\phi)$, $\phi\sim U[0,\pi/2]$, with viscosity fixed to $\nu = 10^{-3}$. Specifically, we draw 100 initial vorticities and 100 forcing phases and form the full 100×100 product for training (10000 trajectories), and likewise 35 initial vorticities and 35 forcing phases for validation ($35\times 35 = 1225$ trajectories). To evaluate generalization to unseen initial conditions, we build four held-out test sets of 1225 trajectories each: every test set is generated from a single initial vorticity w_0, unseen during training, paired with 1225 independently sampled forcing phases $\phi\sim U[0,\pi/2]$; the four test sets thus differ only in their initial vorticity (Figure 10).

The projector is defined to map the diffusion trajectory onto the manifold of states consistent with the linear conservation of total vorticity of the system **(LVC, Linear Vorticity Conservation)**, induced by the periodicity of the domain and the divergence-free velocity field:

$$\mathcal{H} = \left\{ w:[0,1]^2\times[0,49]\to\mathbb{R} \;\middle|\; \begin{array}{l} w(\mathbf{x},0) = w_0(\mathbf{x}), \;\; \forall\,\mathbf{x}\in[0,1]^2 \\ \displaystyle\int_{[0,1]^2} w(\mathbf{x},T)\,d\mathbf{x} - \int_{[0,1]^2} w(\mathbf{x},0)\,d\mathbf{x} = \\ \qquad\qquad \displaystyle = T\int_{[0,1]^2} f(\mathbf{x})\,d\mathbf{x}, \;\; \forall\,T\in[0,49] \end{array} \right\} \tag{34}$$

being $\Omega(t) = \int_{[0,1]^2} w(\mathbf{x},t)\,d\mathbf{x}$ the total vorticity of the system.

C Implementation Details

C.1 Training and Sampling Setup

The velocity field $v_\theta(\mathcal{V}_{t_i}, t_i)$ is parametrized by a Fourier Neural Operator (FNO) (Li et al., 2021), following the setup of ECI (Cheng et al., 2025) and PCFM (Utkarsh et al., 2026), and implemented

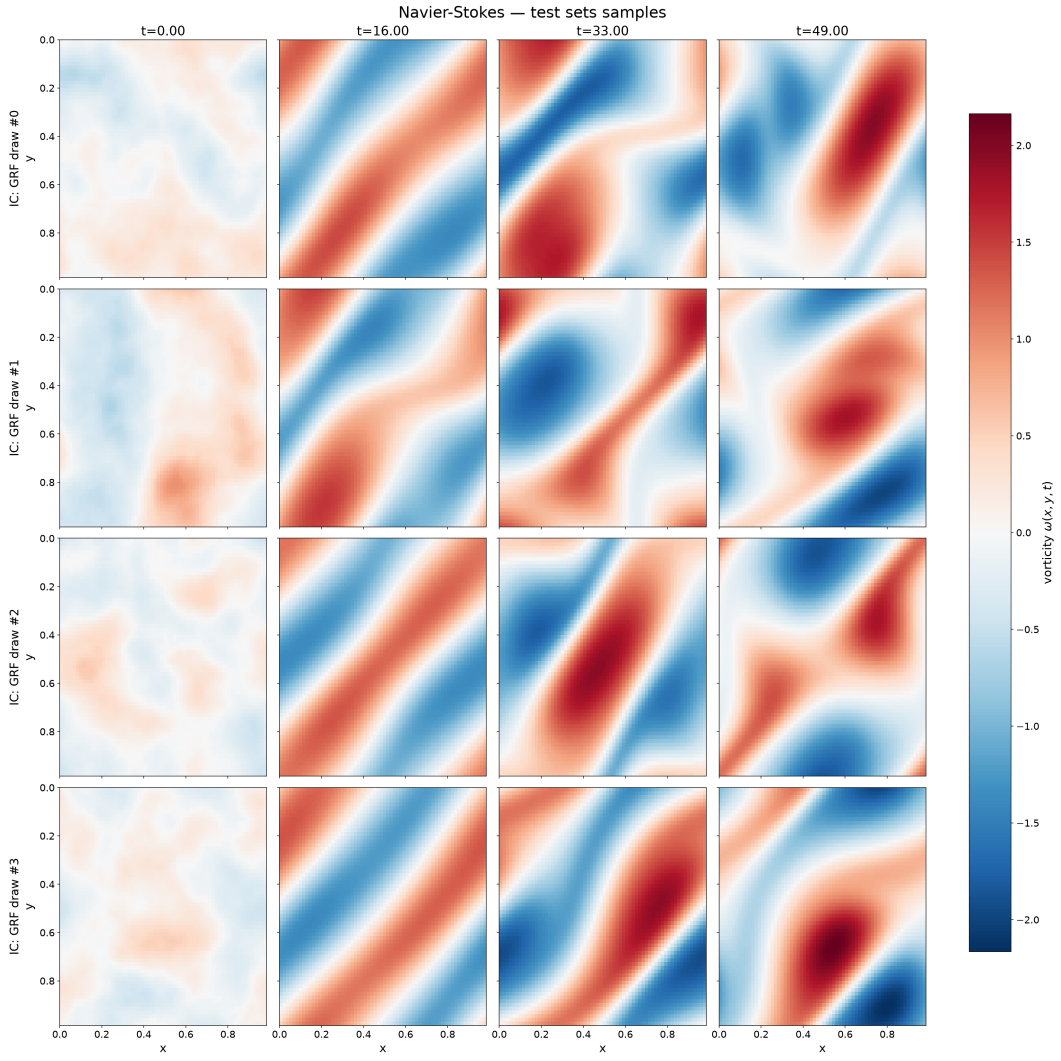


Figure 10: One representative test-set trajectory of the 2D Navier-Stokes benchmark for each of its four held-out IC, shown as a 4×4 grid: each row is a different test split pinned to a distinct unseen initial vorticity field ω_0 drawn from a Gaussian random field; each column is one of four evenly spaced snapshots across the rollout $t \in [0, 49]$. The small-scale initial vorticity is rapidly overtaken by the large-scale pattern imposed by the diagonal forcing mode, with ϕ varying across trajectories.

in PyTorch 2.5.1. The one-dimensional PDE systems are treated as two-dimensional fields over the space-time grid (n_x, n_t) and are modelled by a 2D FNO, while Navier-Stokes is modelled by a 3D FNO over (n_x, n_y, n_t). The FNO input is the concatenation of the current state, a linear positional encoding with one channel for each axis of the field (the physical time included, x, t in 2D; x, y, t in 3D), and a sinusoidal embedding of the flow time. Table 3 summarises the two backbone configurations. Fields enter the network in raw units: no per-channel standardisation, min-max rescaling or normalisation is applied at any stage. The source is drawn from a Gaussian Process with a Matérn kernel (smoothness $\nu = 0.5$, length scale 10^{-3}, unit variance), sampled by a dense Cholesky factorisation on the (n_x, n_t) grids and by circulant FFT synthesis on the larger Navier-Stokes volume, and the flow is integrated over $N = 100$ steps.

The model is trained by minimising the FFM mean-squared-error objective (Eq. 10) with the Adam optimiser ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay) at a base learning rate of 3×10^{-4}, for 2×10^4 optimiser steps, with the batch size of Table 3: 256 on the 2D backbone, reduced to 32 on Navier-Stokes to fit the activation memory of the space-time volume. The learning rate follows a reduce-

Table 3: FNO backbone configurations. 2D FNO for 1D datasets; 3D FNO for 2D datasets.

	2D FNO	3D FNO
Fourier layers	4	4
Fourier modes per axis	32	16
Hidden channels	64	32
Projection channels	256	128
Time-embedding channels	32	16
Training batch size	256	32
Sampling batch size	32	16

on-plateau schedule (factor 0.5, patience 10 validation steps, floor 10^{-4}) and gradients are clipped to a maximum norm of 100. Validation loss is evaluated every 200 optimiser steps on the held-in validation split and is used to drive the schedule: no exponential moving average of the weights and no validation-based checkpoint selection is performed, so every reported number comes from the final iterate at step 2×10^4. Training runs in `float32` throughout, with no mixed precision, under cuDNN deterministic kernels and per-run seeding of the PyTorch and CUDA generators. SAPC additionally folds the projection of the training source into the loss as described in Section 3; all other hyperparameters are shared with the unconstrained backbone. Each configuration is trained with three random seeds $\{0, 42, 205\}$. Every run occupies one NVIDIA GH200 (96 GB) with four data loader workers. One seed of the unconstrained backbone takes 50-75 minutes; SAPC costs the projection of the training source on top of that, from about 1 hour on Heat and Burgers, where the projector is linear, to approximately 9 hours on Navier-Stokes.

Generation integrates the reverse process backward in t on the uniform grid $t_i = i/N, i = N, \ldots, 1$, with $N = 100$ as in the forward process. The FM objective draws $t \sim \mathcal{U}[0, 1]$ continuously rather than on a grid, so there is no training-time discretisation for sampling to match. Table 4 gives the sampling-time projector settings. On the datasets whose constraint families are affine (Heat, Burgers, Navier-Stokes) the projection is the exact orthogonal projector $P = I - A^\top (AA^\top)^{-1} A$, applied in one solve through a Gram factorisation cached across steps. On RD and the two Stokes tasks the constraint is curved and P is the damped Gauss-Newton iteration of Eq. 12, with Tikhonov damping $\mu = 10^{-8}$, stopped early once $\|h(u_k)\|_\infty < 10^{-6}$. Stokes stacks a linear pin with the non-linear kinetic-energy balance and stalls at $\|h\|_\infty \approx 10^{-4}$ after three iterations, so its cap is raised to ten, at which the residual reaches $\approx 10^{-9}$ and is unchanged at twenty; RD converges within the default three. Following PCFM Utkarsh et al. (2026), the curved cases apply one further full Gauss-Newton projection at $t = 0$ to remove the linearisation residual. These settings are fixed per benchmark a priori and are shared by SAPC and by every sampling-time baseline that uses a projector; no sampling-time hyperparameter is tuned against the reported test splits.

Each benchmark carries four held-out test splits. Per split we generate 1225 trajectories at an evaluation batch size of 32; on Navier-Stokes we generate 100 trajectories at batch size 16, the 3D rollout making the full 1225 prohibitive, and score them against the 1225 reference trajectories of the split. Sampling one (seed, split) pair takes 3-15 minutes on the 2D datasets and ≈ 22 minutes on Navier-Stokes. Every reported number is therefore the mean $\pm$ standard error over the $3 \times 4 = 12$ (seed, split) runs of a cell.

Table 4: Sampling-time projector configuration. Affine families admit an exact one-shot projection; the curved ones use damped Gauss-Newton with an early stop on $\|h\|_\infty$ and a final projection at $t = 0$.

Benchmark	Projector	Max iters.	Tolerance	Damping μ	Final $t = 0$ pass
Heat	affine (exact)	1	-	-	no
Burgers	affine (exact)	1	-	-	no
NS	affine (exact)	1	-	-	no
RD	Gauss-Newton	3	10^{-6}	10^{-8}	yes
Stokes IC / BC	Gauss-Newton	10	10^{-6}	10^{-8}	yes

C.2 Baseline Models

All competitors share the same FFM backbone as our method, so that any performance gap reflects how the physical constraint is enforced, rather than differences in architecture, dataset, projector, or noise. D-Flow, ECI, and PCFM are *zero-shot* (sampling-time only); CAFM and PBFM fold the constraint into training. For this work, we retrain all baselines following the authors' specification.

D-Flow. D-Flow (Ben-Hamu et al., 2024) enforces constraints on samples from a pretrained FFM by optimizing the source noise so that the resulting trajectory's endpoint minimizes a loss, without any change to the underlying network. Because the objective is evaluated only at the endpoint, standard backpropagation through an unrolled Euler solver supplies the gradient of the loss with respect to the source. Following the D-Flow setup of Cheng et al. (2025); Utkarsh et al. (2026), an initial noise sample is drawn from the Gaussian-process source and integrated forward through $n = 100$ Euler steps of the frozen FFM velocity network, yielding an endpoint estimate $\hat{V}_0$. The loss combines two squared residuals: a local term $\|(\hat{\mathcal{V}}_0 - \mathcal{V}_0) \odot M\|^2$ pinning the IC/BC values through a Boolean mask M over the constrained grid points, equivalent to $\|\mathcal{R}_{\text{loc}}(\hat{\mathcal{V}}_0)\|^2$ in Section A.2; and a weighted global term $\lambda\|\mathcal{R}_{\text{glob}}(\hat{\mathcal{V}}_0)\|^2$. We set $\lambda = 1$, mirroring the equal-weight concatenation of local and global residuals used by the projection operator across all competitors. The combined loss is minimized with respect to the source noise by L-BFGS (20 iterations, learning rate 0.1). Once L-BFGS terminates, one final forward Euler integration with the optimized, detached noise produces the returned sample.

ECI. ECI (Cheng et al., 2025) is a training-free, zero-shot sampler for FFM that, at every discretized time step, cycles through three operations: extrapolating the current state to an endpoint estimate $\hat{V}_0$ via the flow, correcting $\hat{V}_0$ by projecting it onto the constraint set, and interpolating between the corrected endpoint and the source sample to obtain the next state. Repeating this cycle several times per outer step enforces constraint satisfaction at the cost of extra network evaluations. We follow the sampling procedure of Cheng et al. (2025), interleaving extrapolation, correction, and interpolation at every Euler step, with $n_{\text{mix}} = 5$ mixing updates per outer step and no periodic noise resampling on all datasets. Unlike the reference implementation, our projector is not restricted to constraints admitting a closed-form oblique projection: on non-linear constraints, we substitute the non-linear projector used by the other competitors.

PCFM. PCFM (Utkarsh et al., 2026) generalizes ECI by replacing its closed-form linear projection with the Gauss-Newton projector (Section A.2) to handle non-linear constraints. We reproduce authors' implementation: $n_{\text{iter}} = 1$ Gauss-Newton step per Euler update (in root-finding mode, damping $\epsilon = 10^{-6}$); optional relaxed-penalty correction disabled; an OT-displacement reverse update in place of ECI's direct re-interpolation; and a final Newton-Schur projection of 3 iterations at $t = 1$. The last two elements (the OT-displacement update and the Newton-Schur pass at $t = 1$) distinguish PCFM from ECI here, since the two competitors already share the same non-linear Gauss-Newton projector in our setup.

PBFM. PBFM (Baldan et al., 2026) incorporates the governing equations in the training loss, leaving inference unmodified. Two mechanisms distinguish it from a naive physics-penalty baseline. First, the FM loss $\mathcal{L}_{\text{FM}}$ and a physical-residual loss $\mathcal{L}_{\mathcal{R}}$ are combined via conflict-free gradient updates rather than a scalar weight (ConFIG, Conflict-Free Inverse Gradients). Denoting their gradients $g_{\text{FM}}, g_{\mathcal{R}}$, the orthogonal component $\mathcal{O}(g_1, g_2) = g_2 - \frac{g_1^\top g_2}{\|g_1\|^2} g_1$, and normalisation $\mathcal{U}(g) = g/\|g\|$, the update direction $g_v = \mathcal{U}[\mathcal{U}(\mathcal{O}(g_{\text{FM}}, g_{\mathcal{R}})) + \mathcal{U}(\mathcal{O}(g_{\mathcal{R}}, g_{\text{FM}}))]$ is rescaled by $g_{\text{update}} = (g_{\text{FM}}^\top g_v + g_{\mathcal{R}}^\top g_v)\, g_v$, so that it never opposes either individual gradient. Second, the residual is evaluated on the endpoint of an n-step Euler unrolling of v_θ from $\mathcal{V}_t$ rather than on the one-step estimate $\hat{\mathcal{V}}_0 = \mathcal{V}_t - t\, v_\theta(\mathcal{V}_t, t)$ (in the convention that data is at $t = 0$). The latter extrapolates linearly using the velocity at the current point and is exact only if v_θ is constant along the trajectory; since the field varies along the flow, the one-step endpoint lies off the clean-data manifold, and the non-linear $\mathcal{R}$ can be dominated by that extrapolation error rather than by genuine constraint violation. Unrolling recomputes v_θ at each intermediate state, following the curvature of the learned ODE and producing an endpoint that mirrors the sampler's inference-time output, so $\mathcal{R}$ measures the physical fidelity of states the model actually generates. Our implementation follows the refer-

ence training procedure. For a sampled $t \sim \mathcal{U}[0,1]$ and interpolant $\mathcal{V}_t$, the endpoint is reached by unrolling the network for $n_{\text{unroll}} \in [1,4]$ Euler steps, with n_{unroll} ramped linearly from 1 to 4 over four training stages (their curriculum, reproduced against optimizer steps rather than epochs). We evaluate $\mathcal{R}(\cdot)$ on the unrolled endpoint and multiply the residual loss by $(1-t)$, which is PBFM's t^p scaling at $p=1$ rewritten in our time convention (data at $t=0$), and $p=1$ is the value they identify as optimal. The FM and residual losses are backpropagated separately and their gradients combined via the ConFIG rule described above. The logit-normal timestep weighting of the FM loss is left off, matching the reference default. At sampling time, PBFM performs plain Euler integration of the trained network with no projection or gradient guidance; we do not enable PBFM's optional stochastic sampler, matching the reference's deterministic default.

CAFM. CAFM (Christopher et al., 2026) targets the training/sampling mismatch of projection-based FFM samplers with a training objective that regresses the projected prediction directly against the true endpoint, $\mathcal{L}_{\text{CAFM}} = \|P\big(z_t + (1-t)\, v_\theta(z_t,t)\big) - z_1\|^2$. The projection P is realized as a differentiable projection layer (as in Utkarsh et al. (2026)) to manage non-linear constraints. We implement CAFM following authors' specifications: sampling $t \sim \mathcal{U}[t_{\min},1]$ with $t_{\min}=0.05$ (mirroring their $t \le 0.95$ under our time convention), forming the interpolant $V_t = (1-t)V_0 + tV_1$ and the network's raw endpoint estimate $\hat{V}_0 = V_t - t\, v_\theta(V_t,t)$, projecting it through the benchmark's differentiable projector to obtain $V_0^{\text{proj}} = P(\hat{V}_0)$, and regressing v_θ toward the resulting projected velocity $v^{\text{proj}} = (V_t - V_0^{\text{proj}})/\max(t, 10^{-3})$ against the ground-truth target $V_1 - V_0$ with an ℓ_2 loss. As in the paper, the loss also adds a small residual regularizer $\lambda\|\mathcal{R}(V_0^{\text{proj}})\|^2$ with $\lambda = 10^{-3}$ on the projected endpoint. Sampling is done using the same hyperparameters as the PCFM baseline, so that any difference in downstream results isolates the effect of the training objective.

C.3 Evaluation metrics

Pointwise moments. To compare two distributions over continuous functions, following Kerrigan et al. (2024) and all successive works, we compute the pixel-wise mean $\bar{\mathcal{V}}(\omega)$ and Bessel-corrected standard deviation $\sigma(\omega)$ of the generated and reference trajectories at each spatial location $\omega \in \Omega$, and report the mean squared errors between the two:

$$\text{MMSE} = \tfrac{1}{|\Omega|}\sum_{\omega\in\Omega}\big(\bar{\mathcal{V}}^{\text{gen}}(\omega) - \bar{\mathcal{V}}^{\text{gt}}(\omega)\big)^2, \qquad \text{SMSE} = \tfrac{1}{|\Omega|}\sum_{\omega\in\Omega}\big(\sigma^{\text{gen}}(\omega) - \sigma^{\text{gt}}(\omega)\big)^2 \tag{35}$$

Lower values indicate that the generated trajectories reproduce the first two marginal moments of the reference distribution more faithfully at every spatial location.

Fréchet Poseidon Distance. FPD is the Fréchet distance between Gaussian fits to the encoder-bottleneck activations of the pretrained PDE foundation model Poseidon-B (Herde et al., 2024):

$$\text{FPD} = \|\mu_{\text{gen}} - \mu_{\text{gt}}\|_2^2 + \text{Tr}\Big(\Sigma_{\text{gen}} + \Sigma_{\text{gt}} - 2\,(\Sigma_{\text{gen}}\Sigma_{\text{gt}})^{1/2}\Big) \tag{36}$$

Here μ and Σ denote the mean and Bessel-corrected covariance of the d-dimensional feature vectors $\phi(\mathcal{V}) \in \mathbb{R}^d$ extracted from the Poseidon encoder bottleneck, computed over the generated and reference batches. As in Cheng et al. (2025), we used the version of Poseidon with 157.7M model parameters, whose hidden activation size is 16×768, which we mean-pool into a 768-dimensional vector for FPD computation. For spatio-temporal datasets, FPD is computed per frame and averaged over time. A lower FPD indicates a closer match in the learned feature space.

Per-pixel marginals. FPD summarises quality of the generated trajectories at the level of aggregate features and can miss discrepancies in the marginal distribution at individual locations. We therefore complement it with the Wasserstein-2 distance W_2 (Baldan et al., 2026), computed between the sorted empirical distributions at each spatial location $\omega \in \Omega$ and averaged over Ω. Lower values indicate closer agreement between the generated and reference marginals at every location.

Constraint error. Following Utkarsh et al. (2026), we measure the constraint residual as the mean ℓ^2-norm across the N generated samples, reported separately for the local and global components:

$$\text{CE}_L \;=\; \frac{1}{N}\sum_{n=1}^{N}\big\|\mathcal{R}_{\text{loc}}(\mathcal{V}_n)\big\|_2, \qquad \text{CE}_G \;=\; \frac{1}{N}\sum_{n=1}^{N}\big\|\mathcal{R}_{\text{glob}}(\mathcal{V}_n)\big\|_2 \tag{37}$$

Lower values indicate stricter satisfaction of the constraints. Because the two are different physical quantities and can sit orders of magnitude apart on the same benchmark, we report CE by normalizing each by its value on the unconstrained FFM baseline before averaging:

$$\mathrm{CE} \;=\; \frac{1}{2}\left(\frac{\mathrm{CE}_L}{\mathrm{CE}_L^{\mathrm{FFM}}} \;+\; \frac{\mathrm{CE}_G}{\mathrm{CE}_G^{\mathrm{FFM}}}\right) \tag{38}$$

CE is therefore dimensionless, equals 1 for the unconstrained baseline by construction, and reads as the fraction of baseline violation remaining.

C.4 Performance evaluation

Table 5 reports the inference cost of each sampler on a single GH200. Two competitors set the extremes. PBFM matches the unconstrained FFM wall-clock ($\approx 1.00\times$FFM) since its constraint enters only the training loss and sampling is unchanged. D-Flow is two orders of magnitude slower (97-140×FFM) because guided generation requires 2100 NFE and backpropagation through the ODE trajectory at every optimization step. The projection-based samplers (PCFM, ECI, CAFM, SAPC) sit between these bounds, and their relative cost depends on where the projector is invoked along the trajectory and how its cost scales with the state dimension.

Table 5: Inference cost of each sampler, measured on one NVIDIA GH200 96GB GPU. NFE is the number of network forward evaluations per sample; IT is the wall-clock time per generated sample, ×FFM that time relative to the unconstrained FFM baseline, and $\mathrm{T_{tot}}$ the wall-clock of the complete evaluation run ($n = 1225$ samples, $n = 100$ on Navier-Stokes). Each cell is the median over 3 seeds × 4 held-out test sets.

	Metric	SAPC	PCFM	ECI	CAFM	PBFM	D-Flow	FFM
	NFE	100	100	500	100	100	2100	100
Heat	IT [ms]	100	<u>43.2</u>	497	44.0	**20.8**	2739	20.7
	×FFM	4.85	2.09	24.02	2.13	1.00	133	
	$\mathrm{T_{tot}}$ [s]	123	<u>52.9</u>	608	53.9	**25.4**	3356	25.3
RD	IT [ms]	127	<u>78.6</u>	862	80.9	**25.5**	2755	26.0
	×FFM	4.91	3.03	33.21	3.12	0.98	106	
	$\mathrm{T_{tot}}$ [s]	156	<u>96.3</u>	1056	99.1	**31.2**	3375	31.8
StokesIC	IT [ms]	675	<u>96.4</u>	3295	96.4	**21.6**	2906	20.8
	×FFM	32.43	4.63	158	4.63	1.04	140	
	$\mathrm{T_{tot}}$ [s]	827	<u>118</u>	4036	118	**26.4**	3560	25.5
StokesBC	IT [ms]	669	93.0	3259	<u>91.8</u>	**21.7**	2796	20.9
	×FFM	32.04	4.45	156	4.40	1.04	134	
	$\mathrm{T_{tot}}$ [s]	819	114	3993	<u>113</u>	**26.6**	3425	25.6
Burgers	IT [ms]	102	39.3	503	<u>37.2</u>	**21.9**	2735	21.9
	×FFM	4.65	1.80	22.99	1.70	1.00	125	
	$\mathrm{T_{tot}}$ [s]	125	48.2	616	<u>45.5</u>	**26.8**	3351	26.8
NS	IT [ms]	3913	20401	<u>3664</u>	20414	**249**	25131	258
	×FFM	15.18	79.16	14.22	79.21	0.97	97.51	
	$\mathrm{T_{tot}}$ [s]	391	2040	<u>366</u>	2041	**24.9**	2513	25.8

D Ablation Study

SAPC combines three elements: source projection $P(\mathcal{V}_1)$; a constraint-aware training loss that incorporates the projected source; and endpoint projection $P(\mathcal{V}_{0,\theta})$ at every reverse step. To isolate which of these drives the gain, we compare the full method against three ablations (named S1, S2, and S3) in which one or two components are removed, keeping the FFM backbone and the projector P fixed. Table 6 reports the results over the six datasets.

S3 applies source projection only during sampling, using a backbone trained on unconstrained noise. As shown in Table 6, constraint errors CE, CE_L, and CE_G are of the same order as those of the unconstrained FFM across all datasets. Projecting the source onto $\mathcal{H}$ alone therefore does not ensure that the final sample satisfies the constraints, as the learned velocity field can transport samples away from the manifold, degrading constraints satisfaction. The distributional accuracy of S3 varies instead across datasets. These results highlight the importance of aligning the source distributions used during training and sampling. SAPC achieves this alignment through a constraint-aware loss that learns transport from the anchored source. Source anchoring with matched training improves distributional accuracy, while endpoint projection enforces the constraints.

The two variants that project only the endpoint estimate (S2 and S1, with and without training) bring the constraint errors CE, CE_L, and CE_G down to values comparable with the full SAPC method. The distributional metrics MMSE, SMSE, FPD, and W_2 nonetheless remain close to those of the unconstrained backbone: on Heat, RD, and Burgers, MMSE and FPD are two to three orders of magnitude worse than SAPC, while on NS the gap narrows to a factor of three to four. Enforcing the constraint pointwise on the endpoint estimate is therefore sufficient to make each generated field satisfy the PDE, but does not push the ensemble towards the target solution distribution. The network produces admissible but distributionally biased samples.

Comparing S2 and S1 columns of Table 6 isolates the effect of folding the endpoint projection into the training loss, at fixed sampling procedure. On all distributional metrics, the two variants are mostly indistinguishable across every benchmark; on the constraint errors they agree to within a factor of two on Heat, RD, and the two Stokes tasks, while on Burgers and NS the zero-shot variant is one to two orders of magnitude better on CE_L. Endpoint projection therefore does not benefit from training-time embedding. This identifies the pairing of source anchoring with a matched training loss as the mechanism driving SAPC's improvement.

For completeness, the last column of Table 6 reports condFFM, a soft-conditioning FFM that receives the IC/BC as an additional input channel concatenated with the Gaussian source. This variant assesses the distributional accuracy achievable through conditioning alone, without projection. On MMSE, SMSE, FPD, and W_2, condFFM is competitive with SAPC and occasionally better. This confirms that soft conditioning on the IC/BC captures much of the shape of the constrained data distribution. The trade-off appears on the constraint errors: CE_L of condFFM is six to thirteen orders of magnitude worse than SAPC's on every benchmark, and CE_G spans 10^{-2} to 10^{0} against SAPC's 10^{-8} to 10^{-2} across datasets. Soft conditioning is therefore a valid approach to distributional quality but leaves the physical constraint essentially unenforced, whereas SAPC addresses both.

E Additional Results

Table 7 complements the main results by reporting the Wasserstein-2 distance W_2 and the constraint errors CE_L and CE_G. Three observations reinforce the previous results. First, W_2 confirms the ranking obtained from MMSE, SMSE, and FPD in Table 2. Indeed, SAPC attains the best distributional match across all datasets, one to two orders of magnitude below the closest competitor on Heat, RD, and Burgers, and by a factor of 1.6 to 3 on the two Stokes tasks and NS. Second, the decomposition into CE_L and CE_G distinguishes local constraint satisfaction from global conservation accuracy. PCFM and CAFM reduce local errors to approximately 10^{-12}–10^{-14} on Heat, RD, and Burgers. SAPC also achieves very small local errors, although not uniformly as low, while matching or improving global constraint accuracy: it attains the lowest CE_G on Heat and RD and remains within a factor of two of the best result on NS and Burgers. Stokes BC is the exception: CAFM achieves $\mathrm{CE}_G = 8.4 \times 10^{-7}$, compared with 2.8×10^{-2} for SAPC. These values should be interpreted relative to the numerical NLEC residual on exact held-out solutions, which ranges from 8.5×10^{-2} to 1.1×10^{-1}. SAPC falls below this reference range by a factor of approximately three to four, whereas CAFM falls about five orders of magnitude below it. Nevertheless, CAFM yields higher MMSE, FPD, and W_2 than SAPC, demonstrating that a smaller numerical constraint residual does not necessarily imply closer agreement with the target distribution. Finally, the soft-constraint competitors (PBFM, D-Flow) fail to reduce either constraint errors below the unconstrained FFM level across most datasets. This confirms that penalty-based enforcement does not translate into satisfaction of the physical law at inference time.

Figures 11–15 provide the same comparison of pointwise means and standard deviations as Figure 3 for the remaining datasets. In each panel, the test trajectories share the prescribed IC or BC and differ only in the remaining free parameter. The ground-truth (GT) standard deviation therefore captures the variability under that fixed condition. SAPC and the projection-based samplers (PCFM, ECI, and CAFM) enforce the prescribed IC or BC exactly, while D-Flow approximates it through guidance. PBFM and FFM impose no constraints during sampling, allowing the IC or BC to vary according to the training distribution and consequently overestimating the pointwise standard deviation.

Figure 16 extends the trajectory analysis on Heat, Stokes (IC), Stokes (BC) and NS. The left column shows that SAPC produces endpoint estimates $\mathcal{V}_{0,\theta}$ with the lowest constraint residual throughout sampling on Heat, Stokes BC, and NS. On Stokes IC, SAPC reduces the residual by a smaller factor of two to three over most of the trajectory. Near $t = 0$, however, ECI and CAFM reach lower residuals of 0.84 and 0.60, respectively, compared with 1.05 for SAPC. These values describe endpoint estimates before projection, indicating that ECI and CAFM predict states that satisfy the constraints more closely at this stage. Nevertheless, projecting these estimates enforces the prescribed constraints without guaranteeing agreement with the target distribution. The middle column is consistent with Equation 4. On Heat and NS, where $\mathcal{R}$ is affine, SAPC maintains a low, nearly constant residual throughout sampling. On Stokes, the constraints combine a linear IC or BC condition with a non-linear energy balance, and the residual grows from initially small values. Unlike RD, Stokes does not exhibit a symmetric residual profile. Although the Gauss–Newton projector converges to a small residual, subsequent sampling updates can move intermediate states away from the curved constraint manifold. That is the reason why, following Utkarsh et al. (2026), we apply an additional projection at $t = 0$ to remove residual constraint violations arising from non-linear projection approximations and numerical integration. The right column shows that SAPC achieves the smallest deviation Δ and excess KE among the competing methods on Heat, Stokes BC, and NS. On NS, however, SAPC and its three ablated variants (S1–S3) have similar excess KE ($\approx 2 \times 10^{-3}$), suggesting that source anchoring improves constraint satisfaction without substantially changing the trajectory geometry. Stokes IC is the exception: SAPC achieves lower excess KE (0.20) than CAFM (0.61) and ECI (17), but higher than PCFM (0.09). S3 attains the lowest value (3.2×10^{-3}), despite its larger constraint violations. This is consistent with the larger iterate residual in the middle column, suggesting that maintaining constraint satisfaction on Stokes IC requires greater deviation from the linear interpolant.

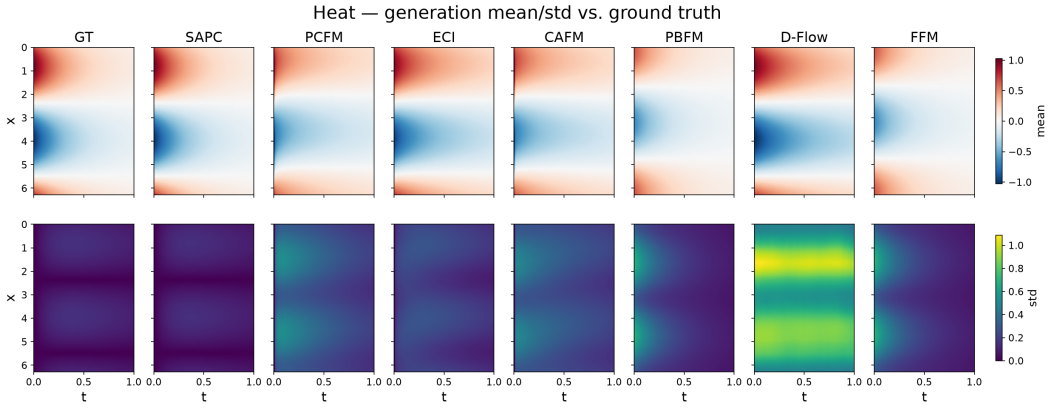


Figure 11: Generated trajectories on Heat for one test split: pointwise mean (top) and standard deviation (bottom) over 1225 trajectories sharing the split's IC at $\phi = \pi/4$.

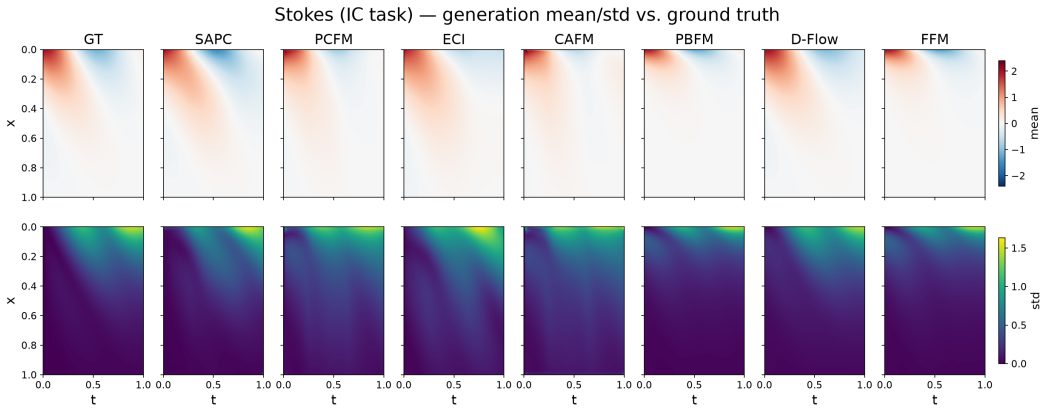


Figure 12: Generated trajectories on Stokes for one test split: pointwise mean (top) and standard deviation (bottom) over 1225 trajectories sharing the split's IC at $k = 4$, differing in forcing frequency.

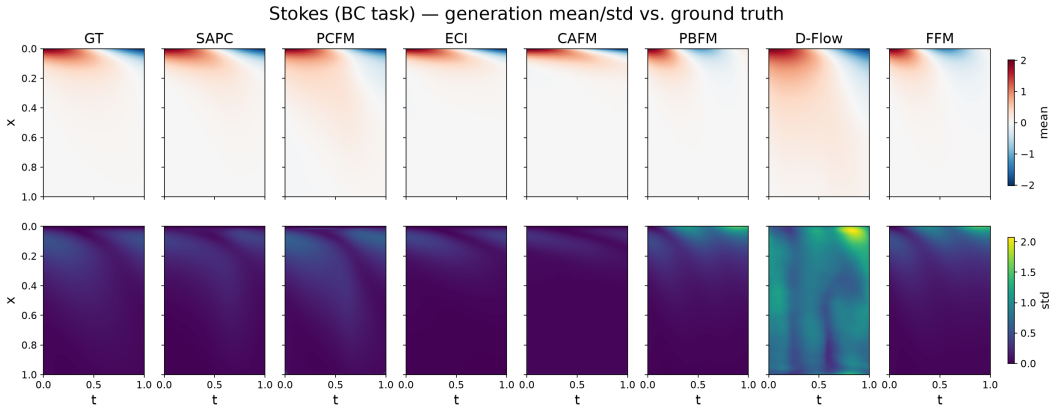


Figure 13: Generated trajectories on Stokes for one test split: pointwise mean (top) and standard deviation (bottom) over 1225 trajectories sharing the split's BC at $\omega = 3$, differing in decay rate.

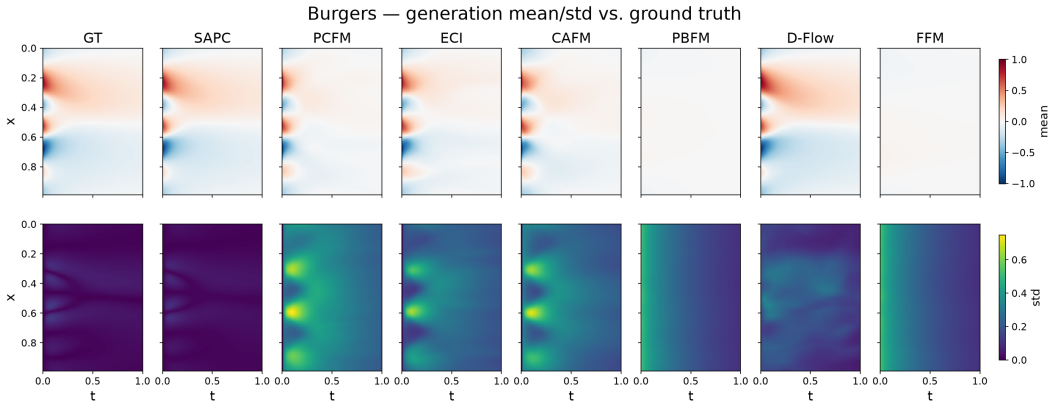


Figure 14: Generated trajectories on Burgers for one test split: pointwise mean (top) and standard deviation (bottom) over 1225 trajectories sharing the split's IC.

Table 6: Ablation of SAPC over the constraint-aware training, the projection $P(\mathcal{V}_1)$ of the source, and the projection $P(\mathcal{V}_{0,\theta})$ of the endpoint estimate at every reverse step. FFM and condFFM are also reported. Each cell reports the mean m and standard error s across seeds and held-out test sets in the compact form $m(s)p$, meaning $(m \pm s) \times 10^p$. Lower values indicate better performance, with the best result in bold and the second best underlined.

		SAPC	SAPC (S3)	SAPC (S2)	SAPC (S1)	FFM	condFFM
	training	✓	×	✓	×	×	×
	conditioning	×	×	×	×	×	✓
	$P(\mathcal{V}_1)$	✓	✓	×	×	×	×
	$P(\mathcal{V}_{0,\theta})$	✓	×	✓	✓	×	×
Heat	MMSE	**2.2(0.8)−05**	1.5(0.5)−02	3.9(0.9)−02	4.0(1.0)−02	5.7(1.3)−02	2.1(0.8)−04
	SMSE	**2.3(0.4)−05**	2.1(0.0)−02	2.3(0.0)−02	2.5(0.0)−02	3.7(0.1)−02	1.4(0.5)−05
	FPD	**4.6(1.3)−03**	1.6(0.1)+00	1.9(0.2)+00	1.9(0.2)+00	2.7(0.3)+00	2.3(0.8)−02
	W_2	**1.4(0.1)−02**	1.4(0.1)−01	1.9(0.1)−01	1.9(0.1)−01	2.3(0.1)−01	1.8(0.2)−02
	CE	2.0(0.0)−06	6.8(0.6)−01	**1.9(0.0)−06**	2.0(0.1)−06	1.0(0.0)+00	3.8(0.6)−01
	CE_L	**6.0(2.3)−11**	3.9(0.2)+00	2.8(0.4)−08	4.0(0.5)−08	6.6(0.5)+00	4.7(0.1)−02
	CE_G	5.9(0.0)−06	1.1(0.2)+00	**5.7(0.1)−06**	6.1(0.2)−06	1.5(0.1)+00	1.1(0.2)+00
RD	MMSE	**5.8(0.9)−05**	7.6(1.1)−02	3.3(1.1)−02	3.3(1.1)−02	4.2(1.2)−02	1.1(0.2)−04
	SMSE	**1.9(0.2)−05**	3.1(0.1)−02	2.6(0.0)−02	2.6(0.0)−02	3.2(0.1)−02	6.5(0.8)−05
	FPD	**5.9(2.0)−01**	2.8(0.5)+02	1.1(0.3)+02	1.1(0.4)+02	1.2(0.3)+02	8.4(3.0)−01
	W_2	**7.0(0.2)−03**	3.0(0.1)−01	2.2(0.2)−01	2.2(0.2)−01	2.5(0.2)−01	1.2(0.1)−02
	CE	**3.3(0.2)−06**	1.7(0.1)+00	4.3(0.1)−06	4.2(0.1)−06	1.0(0.0)+00	8.2(1.2)−01
	CE_L	**1.9(0.2)−14**	5.1(0.0)+00	1.5(0.1)−07	1.5(0.1)−07	4.9(0.1)+00	1.3(0.1)−01
	CE_G	**3.1(0.2)−07**	1.1(0.1)−01	4.1(0.1)−07	3.9(0.1)−07	4.7(0.0)−02	7.5(1.1)−02
Stokes IC	MMSE	**3.0(1.2)−03**	1.4(0.2)−02	9.9(3.6)−03	1.0(0.4)−02	1.5(0.4)−02	8.0(2.9)−04
	SMSE	**1.1(0.4)−03**	1.7(0.1)−02	7.7(1.0)−03	8.0(1.1)−03	1.9(0.2)−02	3.6(1.0)−04
	FPD	**2.8(1.2)−01**	5.6(1.6)+00	2.7(1.2)+00	2.9(1.3)+00	7.9(2.5)+00	1.3(0.5)−01
	W_2	**3.7(1.1)−02**	1.4(0.1)−01	1.0(0.1)−01	1.0(0.1)−01	1.5(0.1)−01	1.9(0.4)−02
	CE	**6.1(3.3)−02**	1.0(0.0)+00	5.1(1.9)−01	4.8(1.8)−01	1.0(0.1)+00	5.0(1.3)−01
	CE_L	**3.0(1.7)−08**	1.9(0.1)+00	2.8(1.0)−07	2.4(0.9)−07	2.2(0.2)+00	9.1(0.5)−02
	CE_G	**4.6(2.5)−02**	4.3(0.2)−01	3.8(1.4)−01	3.7(1.4)−01	3.8(0.1)−01	3.6(1.0)−01
Stokes BC	MMSE	**1.4(0.1)−03**	6.6(2.4)−03	1.9(0.2)−02	1.8(0.2)−02	5.3(0.8)−02	3.5(1.0)−04
	SMSE	**3.8(0.4)−03**	4.0(0.4)−03	1.0(0.1)−02	1.1(0.1)−02	3.2(0.0)−02	4.8(2.4)−04
	FPD	1.7(0.2)+00	**6.0(0.9)−01**	5.6(1.5)+00	5.5(1.5)+00	2.1(0.5)+00	1.8(0.9)−01
	W_2	**6.1(0.3)−02**	7.3(0.8)−02	1.2(0.1)−01	1.2(0.1)−01	1.4(0.1)−01	2.7(0.4)−02
	CE	**3.6(0.4)−02**	1.6(0.1)+00	1.5(0.2)−01	1.4(0.2)−01	1.0(0.0)+00	4.6(0.4)−01
	CE_L	**1.3(0.3)−09**	3.0(0.2)+00	2.2(0.1)−07	1.7(0.1)−07	1.2(0.1)+01	1.0(0.0)−01
	CE_G	**2.8(0.3)−02**	1.2(0.0)+00	1.1(0.1)−01	1.0(0.1)−01	3.8(0.1)−01	3.4(0.3)−01
Burgers	MMSE	**6.4(1.8)−05**	3.5(0.5)−02	6.2(1.0)−02	6.3(1.0)−02	9.0(1.2)−02	4.1(1.1)−05
	SMSE	**5.5(1.1)−05**	4.5(0.1)−02	3.3(0.1)−02	3.4(0.1)−02	5.0(0.1)−02	3.8(1.0)−05
	FPD	**1.5(0.4)−02**	2.1(0.3)+00	2.5(0.5)+00	2.5(0.5)+00	2.8(0.4)+00	7.3(1.8)−03
	W_2	**9.9(0.6)−03**	2.5(0.1)−01	2.8(0.1)−01	2.8(0.1)−01	3.2(0.1)−01	9.1(0.5)−03
	CE	1.7(0.2)−06	6.4(0.1)−01	2.7(0.7)−05	**1.4(0.1)−06**	1.0(0.0)+00	3.3(0.1)−01
	CE_L	**4.4(1.3)−10**	5.5(0.1)+00	1.1(0.3)−06	2.9(0.3)−08	7.6(0.1)+00	4.5(0.2)−02
	CE_G	5.0(0.5)−06	8.3(0.3)−01	8.2(2.0)−05	**4.1(0.2)−06**	1.5(0.1)+00	9.7(0.4)−01
Navier-Stokes	MMSE	**5.3(0.6)−02**	1.4(0.1)−01	1.8(0.1)−01	1.9(0.1)−01	1.8(0.1)−01	2.5(0.4)−02
	SMSE	**3.1(0.4)−02**	6.0(0.1)−02	6.9(0.3)−02	7.0(0.3)−02	6.9(0.3)−02	1.4(0.2)−02
	FPD	**1.1(0.1)+00**	2.4(0.1)+00	2.6(0.1)+00	2.7(0.1)+00	2.7(0.1)+00	9.3(1.6)−01
	W_2	**2.5(0.1)−01**	3.8(0.1)−01	4.2(0.1)−01	4.2(0.1)−01	4.2(0.2)−01	1.8(0.1)−01
	CE	7.3(0.1)−07	7.4(0.4)−01	3.6(0.1)−06	**7.1(0.1)−07**	1.0(0.0)+00	4.4(0.4)−01
	CE_L	**6.3(0.9)−09**	1.6(0.0)+01	4.5(0.1)−06	1.9(0.2)−08	2.2(0.1)+01	2.7(0.1)+00
	CE_G	8.8(0.1)−08	4.5(0.4)−02	4.2(0.1)−07	**8.5(0.1)−08**	6.0(0.5)−02	4.5(0.4)−02

Table 7: Supplement to Table 2: the Wasserstein-2 distance (W_2) and the two constraint-errors CE_L and CE_G (see Section C.3 for their definition). Each cell reports the mean m and standard error s across seeds and held-out test sets in the compact form $m(s)p$, meaning $(m \pm s) \times 10^p$. Lower values indicate better performance, with the best result in bold and the second best underlined

	Metric	SAPC	PCFM	ECI	CAFM	PBFM	D-Flow	FFM
Heat	W_2	**1.4(0.1)−02**	2.6(0.1)−01	2.6(0.5)−01	3.3(0.5)−01	2.3(0.2)−01	2.3(1.0)+00	2.3(0.1)−01
	CE_L	6.0(2.3)−11	**7.7(4.2)−14**	6.2(0.1)−08	3.9(2.1)−13	6.6(0.5)+00	3.1(0.8)+00	6.6(0.5)+00
	CE_G	**5.9(0.0)−06**	6.0(0.1)−06	7.9(0.4)−06	6.6(0.4)−06	4.3(0.0)−01	5.8(2.7)+00	1.5(0.1)+00
RD	W_2	**7.0(0.2)−03**	3.2(0.2)−01	1.9(0.1)−01	2.7(0.1)−01	2.5(0.2)−01	3.2(1.8)+00	2.5(0.2)−01
	CE_L	**1.9(0.2)−14**	1.1(0.1)−12	1.4(0.0)−07	2.2(0.2)−13	4.9(0.1)+00	5.6(2.0)+00	4.9(0.1)+00
	CE_G	**3.1(0.2)−07**	3.1(0.2)−07	3.4(0.1)−07	3.1(0.1)−07	1.7(0.2)−02	2.8(2.5)+00	4.7(0.0)−02
StokesIC	W_2	**3.7(1.1)−02**	1.7(0.0)−01	1.1(0.2)−01	1.8(0.0)−01	1.5(0.1)−01	4.9(2.4)−01	1.5(0.1)−01
	CE_L	**3.0(1.7)−08**	1.0(0.2)−06	1.3(0.2)−07	7.1(1.7)−08	2.2(0.2)+00	8.2(2.0)−01	2.2(0.2)+00
	CE_G	4.6(2.5)−02	4.7(0.7)−01	1.4(0.4)−01	**3.8(1.2)−02**	1.5(0.1)−01	7.8(6.4)+00	3.8(0.1)−01
StokesBC	W_2	**6.1(0.3)−02**	5.6(2.2)−01	1.0(0.1)−01	1.7(0.0)−01	1.5(0.1)−01	2.3(0.9)+00	1.4(0.1)−01
	CE_L	1.3(0.3)−09	3.4(0.3)−07	5.0(0.4)−08	**1.4(0.3)−10**	1.2(0.0)+01	5.8(1.4)+00	1.2(0.1)+01
	CE_G	2.8(0.3)−02	3.5(0.3)−01	2.1(0.3)−02	**8.4(1.3)−07**	1.2(0.0)−01	1.8(1.6)+02	3.8(0.1)−01
Burgers	W_2	**9.9(0.6)−03**	4.1(0.3)−01	3.7(0.6)−01	2.9(0.1)−01	3.2(0.2)−01	1.2(0.2)−01	3.2(0.1)−01
	CE_L	4.4(1.3)−10	**0.0(0.0)+00**	7.5(0.4)−08	4.4(3.1)−12	7.5(0.2)+00	1.4(0.0)+00	7.6(0.1)+00
	CE_G	5.0(0.5)−06	4.7(0.1)−06	6.2(0.5)−06	**4.2(0.1)−06**	4.2(0.2)−01	7.6(1.1)−01	1.5(0.1)+00
NS	W_2	**2.5(0.1)−01**	5.6(0.1)−01	7.8(0.0)−01	5.5(0.1)−01	4.2(0.2)−01	1.2(0.9)+00	4.2(0.2)−01
	CE_L	6.3(0.9)−09	**3.8(0.7)−12**	6.2(0.0)−07	4.5(1.7)−08	2.1(0.1)+01	1.4(0.7)+01	2.2(0.1)+01
	CE_G	8.8(0.1)−08	**8.4(0.0)−08**	8.5(0.1)−08	8.5(0.1)−08	4.3(0.3)−02	10.0(3.1)−02	6.0(0.5)−02

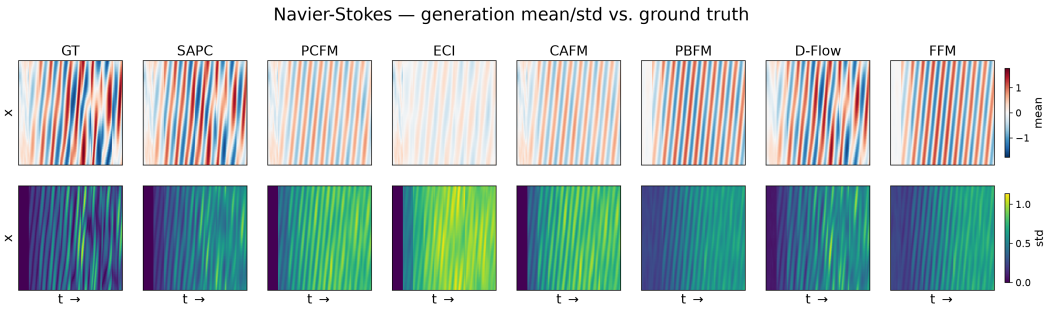


Figure 15: Generated trajectories on Navier-Stokes for one test split. Each column tiles 10 vorticity frames left to right in t; trajectories share the initial vorticity and differ in the forcing phase.

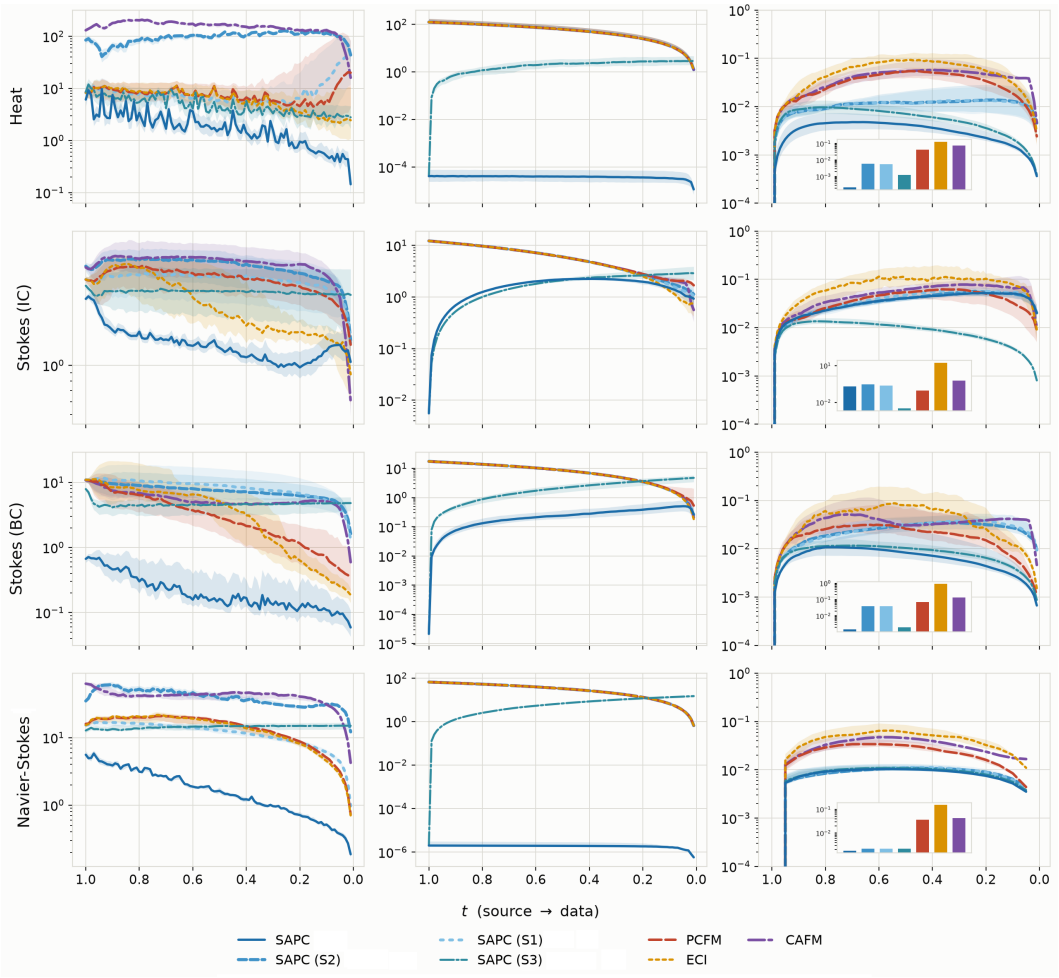


Figure 16: The same as Figure 5, but for the Heat, Stokes (IC and BC) and Navier-Stokes datasets.