PHYSICS-ALIGNED FINITE MAPS: RECASTING PHYSICS-CONSTRAINED GENERATION AS NEXT-STATE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-constrained generative modeling aims to generate realistic physical fields that are both distributionally accurate and physically consistent, with applications to inverse problems, conditional generation, and scientific simulation. Yet these two requirements can impose competing objectives: existing methods typically balance generative and physical losses during training or correct samples at inference, leading to trade-offs among fidelity, consistency, and computational cost. We introduce *Physics-Aligned Finite Maps* (PAFM), which instead factorizes target construction into distributional progression in free coordinates and physics-based completion. At every step, the sample is advanced toward the data distribution in free physical coordinates, while known physics completes this state into a feasible full field. The two objectives are therefore compatible at each teacher next-state, which serves as both the endpoint of the current transition and the starting point of the next. Physics-constrained generation is thus recast as a sequence of finite next-state predictions with sample-wise compatible teacher targets. We instantiate the prior-to-data sequence with a coarse-to-fine heat bridge, and further distill offline physical completion into a lightweight transition recovery network to suppress rollout drift without physical solves between learned transitions. Across three benchmarks, PAFM achieves low physical residuals, competitive distributional fidelity, and millisecond-scale inference, providing an effective framework for physics-constrained scientific generation.

1 INTRODUCTION

Many scientific problems require a distribution of plausible physical fields rather than a single deterministic solution (Kerrigan et al., 2024). Such uncertainty can arise from unknown inputs, uncertain coefficients, or partial observations, making generative modeling a natural tool for inverse inference, conditional generation, and uncertainty quantification. Earlier scientific machine-learning methods such as physics-informed neural networks and neural operators mainly learn individual solutions or input-solution mappings (Raissi et al., 2019; Li et al., 2021; Lu et al., 2021; Li et al., 2024), whereas recent diffusion and flow-based models extend this setting to distributions over coupled physical fields (Huang et al., 2024; Lienen et al., 2024; Long et al., 2025). Thus, a successful physical generator must achieve both distributional fidelity and sample-wise physical consistency.

Existing methods typically introduce physics either during training or during sampling. Training-time approaches add physical objectives to the generative loss, as in physics-informed diffusion and PBFM (Bastek et al., 2025; Baldan et al., 2026), while sampling-time approaches guide, correct, or optimize generated states, as in DiffusionPDE, PCFM, and D-Flow (Huang et al., 2024; Utkarsh et al., 2025; Ben-Hamu et al., 2024). Both strategies face a common trade-off among distributional fidelity, physical consistency, and computational cost: physical objectives can interfere with generative learning, whereas iterative correction or optimization increases inference overhead.

We take a different view of this trade-off. Training-time physical penalties implicitly ask the generative model to discover a trajectory that both follows the desired probability transport and leads to physically valid outputs. Sampling-time correction instead postpones this reconciliation to

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.

inference. In both cases, distributional fidelity and physical consistency remain entangled in the same generative update.

PAFM separates these two roles. We let a subset of physical variables describe how the sample should progress toward the data distribution, while the remaining coupled variables are determined from the selected physical relations. For each step, the former specifies where the next state should lie distributionally, and physics completes it into a full field that satisfies the prescribed coupling. The two objectives therefore share the same sample-wise next-state target, removing the target-level incompatibility between transition matching and physical supervision. This turns the coupled learning problem into a sequence of compatible finite next-state prediction problems.

The key consequence is that every next state can be treated as the endpoint of the current prediction and, at the same time, as the starting point of the next one. Generation can therefore be represented as a chain of finite state transitions rather than a continuous trajectory that must be integrated before its endpoint is assessed. Once the number of transitions is fixed, generation reduces to repeatedly predicting the next physically aligned state until the data distribution is reached. Because each learned map predicts the finite state change that is actually executed at deployment, sampling follows the same transition sequence as training.

We instantiate the prior-to-data sequence with a heat-dissipative bridge, which connects the random prior to the data distribution and organizes the intermediate states from large-scale structure to progressively finer details. We first learn this finite transition process with a BASE generator. Because prediction errors at one endpoint can shift the starting state of the next transition, we further train a lightweight transition recovery distillation (TRD) module to make a small correction after each update. TRD distills the effect of the training-time physical completion into the sampler, so that each predicted next state is recovered toward the physically completed state before being reused. This limits error accumulation while preserving the efficiency of the learned rollout.

Our contributions are threefold. (I) We introduce a physics-aligned finite-transport formulation in which distributional progression is constructed in free physical variables and physically completed into full-field transition targets satisfying the selected relations. Each paired target therefore specifies both the intended distributional move and its corresponding physical coupling, making transition matching and physical supervision sample-wise target-compatible. **(II)** We recast both training and generation as a prescribed sequence of finite next-state predictions: for a given deployment budget, each learned map predicts the next state that is actually executed by the sampler. We learn these physically completed transitions with deployment-aligned finite maps and distill corrections obtained by physically completing BASE rollout states into a lightweight recovery map. Our analysis further characterizes deterministic transition averaging and the resulting constraint averaging gap (CAG). **(III)** We validate PAFM on Kolmogorov Flow, Dynamic Stall, and Darcy Flow, covering linear, local nonlinear, and globally coupled constraints, with controlled ablations of the teacher path, coarse-to-fine bridge, recovery, initialization, and computational cost.

2 Related Work

Joint priors and physical representations. DiffusionPDE models coefficients and solutions jointly, then guides reconstruction with observations and PDE residuals when either field is only partly known (Huang et al., 2024). CoCoGen similarly combines a score-based field prior with residual-guided sampling for forward and inverse problems (Jacobsen et al., 2025). These joint priors capture uncertainty over coupled variables and can be adapted to new conditioning queries, with physical information evaluated during each guided sample.

FunDiff takes a representational route: its function autoencoder uses a streamfunction to make decoded velocities divergence-free, while general PDE relations are encouraged by soft residual losses before latent diffusion is trained (Wang et al., 2026). PAFM assigns physical completion a different role: a prescribed map constructs each teacher endpoint as the full-field target of a finite update, while a task-specific closure supplies offline correction labels for BASE proposals.

Where physical constraints are evaluated. Training-time methods differ in the state on which they measure physics (Tauberschmidt et al., 2026). PIDM evaluates a PDE residual on either the predicted clean mean or an accelerated DDIM sample estimate; the former can suffer a Jensen gap under nonlinear constraints, while the latter adds training computation (Bastek et al., 2025). PBFM

unrolls its learned flow during training to evaluate a terminal-state residual and resolves conflicts between that gradient and velocity matching (Baldan et al., 2026). PIDDM applies a residual directly to the output of a distilled one-step student (Zhang et al., 2026). These methods improve where and how a residual is imposed while their targets remain tied to a diffusion or flow construction. PAFM changes the paired target itself: every adjacent finite update targets a physically completed field, and the residual is evaluated on the field BASE actually proposes.

Inference-time control preserves an existing generator but evaluates the control objective for each new sample (Cheng et al., 2025; Christopher et al., 2024). PCFM corrects the trajectory of a pretrained flow and can enforce selected nonlinear equality constraints at its final output to numerical tolerance (Utkarsh et al., 2025). D-Flow optimizes the source noise of a frozen flow by differentiating through generation to satisfy a user-specified objective (Ben-Hamu et al., 2024); it does not carry PCFM's hard-constraint guarantee. PAFM computes physical completion during training and distills corrections to BASE proposals. Its learned rollout calls neither a physical solver nor a residual-gradient correction, though its generated fields are only approximately consistent.

Finite transitions and structured paths. Directly learned finite transitions are well established (Boffi et al., 2025b; Geng et al., 2025). Consistency Models and Flow Map Matching learn endpoint or two-time maps, Shortcut Models condition their jumps on step size, and Inductive Moment Matching targets the marginals of few-step samplers (Song et al., 2023; Boffi et al., 2025a; Frans et al., 2025; Zhou et al., 2025). A physics-informed consistency model has also applied PDE residuals to single-step and refined two-step generated fields (Chang et al., 2026).

Inverse heat dissipation and blurring diffusion use smoothing as a structured corruption to be reversed during generation (Rissanen et al., 2023; Hoogeboom & Salimans, 2023). Here, heat instead orders the scales of the data contribution along the bridge before physical completion; it is not the governing PDE. The physical completion map supplies a full field satisfying the selected relations up to its numerical tolerance at each node. Thus the bridge specifies which distributional transition to learn, while completion specifies a physically consistent destination for that transition.

3 METHOD

3.1 OVERVIEW

PAFM follows the decomposition introduced in §1: distributional progression is specified through a set of free physical variables, while the selected physical relations determine how these variables are completed into a coupled full field. Rather than asking a generative model to discover both roles simultaneously, we use this decomposition to construct the states that the model is trained to predict.

For a prescribed deployment budget of N transitions, we first construct a sequence of $N+1$ physically completed teacher states, $x_0^\star, x_1^\star, \ldots, x_N^\star$. The free variables of these states follow a prescribed prior-to-data path, while their remaining physical structure is supplied by a task-specific completion map. Consequently, every adjacent pair $(x_k^\star, x_{k+1}^\star)$ defines a finite transition whose destination already combines the intended distributional progression with the selected physical relations.

PAFM learns these transitions directly. The **BASE** model takes a complete state and predicts the finite update to the next completed teacher state. At deployment, however, BASE is applied to its own previous predictions rather than exact teacher states, so small transition errors can move the rollout away from the states seen during training. We therefore use the same physical completion mechanism a second time, now on BASE-generated proposals, to determine how an imperfect predicted state should be physically corrected. These corrections are constructed offline and distilled into a lightweight **transition recovery distillation** (TRD) network. The deployed sampler then alternates a learned finite transition and a learned recovery, without invoking physical completion between transitions.

This construction connects training and deployment at the level of the actual finite updates being executed: the teacher sequence specifies the desired transitions, BASE learns those transitions, and TRD suppresses the physical inconsistency introduced when the learned rollout departs from that sequence.

3.2 PHYSICS-ALIGNED TEACHER SEQUENCE

Let c denote the observed condition together with known physical information, and let $p_{\mathrm{data}}(x \mid c)$ denote the distribution of complete physical fields $x \in \mathcal{X}$. The selected physical relations are represented by $R_c : \mathcal{X} \to \mathbb{R}^{m_c}$. We use three representations throughout this section: $z \in \mathcal{Z}$ denotes the free physical variables used to describe the stochastic degrees of freedom, $x \in \mathcal{X}$ denotes a complete physical field, and $y = \mathcal{A}(x) \in \mathcal{Y}$ denotes the fixed normalized representation used by the neural networks, where $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is invertible.

A task-specific physical completion map $D_c : \mathcal{Z} \to \mathcal{X}$ maps the free variables to a complete field satisfying the selected physical relations up to numerical tolerance. This may include, for instance, a closed-form reconstruction or a numerical solve. Conversely, $E_c : \mathcal{X} \to \mathcal{Z}$ extracts the free coordinates used to construct the generative path. The parameterization of $\mathcal{Z}$ need not be minimal or injective.

For the states used to construct training targets, we assume

$$\|R_c(D_c(z))\|_2 \le \varepsilon_{\mathrm{dec},c}, \tag{1}$$

where $\varepsilon_{\mathrm{dec},c}$ accounts for discretization or solver error. The important role of D_c is that it provides a complete coupled state, rather than only a scalar measure of physical violation. Physics therefore enters the construction of the transition target itself.

We next define how the free variables progress from the random source to the data endpoint. Sample $\xi \sim \mathcal{N}(0, I)$ and $x \sim p_{\mathrm{data}}(\cdot \mid c)$ according to the training coupling, and set

$$z_0 = G_c(\xi), \qquad z_1 = E_c(x), \tag{2}$$

where G_c is the identity in the chosen coordinates.

We connect these endpoints with a heat-dissipative bridge. Let $L_c \succeq 0$ denote a task-specific discrete diffusion operator and $\kappa_c > 0$ its strength. We define

$$z(t) = (1-t)z_0 + t\exp[-\kappa_c(1-t)L_c]z_1, \qquad t \in [0,1]. \tag{3}$$

The bridge interpolates from the random source to the encoded data while allowing large-scale components of the data endpoint to appear before progressively finer structure. The heat operator serves only to organize this distributional progression; it is not the governing physical PDE.

For a deployment grid $0 = t_0 < \cdots < t_N = 1$, we then complete every bridge state in physical space:

$$x_k^\star = D_c(z(t_k)), \qquad k = 0, \dots, N. \tag{4}$$

Thus, the bridge determines how the free coordinates advance, while D_c determines the corresponding complete physical state. Every teacher node satisfies the selected physical relations up to Eq. 1, and the same completed state $x_{k+1}^\star$ serves as the destination of transition k and the starting state of the following teacher transition.

At the final node, $x_N^\star = D_c(E_c(x))$. If $D_c(E_c(x)) = x$ on the data, the terminal teacher distribution is exactly $p_{\mathrm{data}}(\cdot \mid c)$; otherwise, the teacher sequence terminates at the corresponding physically re-completed data distribution. This distinction is analyzed explicitly in Appendix A.

3.3 LEARNING DEPLOYMENT-ALIGNED FINITE TRANSITIONS

Using the fixed normalization, let $y_k^\star = \mathcal{A}(x_k^\star)$. For each interval of the prescribed deployment grid, BASE directly predicts the finite update from one complete teacher state to the next:

$$T_{\theta,N}(y, t_k, c) = y + f_{\theta,N}(y, t_k, c), \qquad \bar{y}_{k+1} = T_{\theta,N}(y_k^\star, t_k, c). \tag{5}$$

Although the teacher path was constructed through $z(t)$, BASE operates only on complete normalized fields. It therefore learns the full state transition that will later be executed by the sampler, rather than separately predicting the free and dependent variables.

The supervised destination of the transition is the next completed teacher state $y_{k+1}^\star$. We additionally evaluate the physical residual on the state that BASE actually predicts, $\mathcal{A}^{-1}(\bar{y}_{k+1})$. With $w_k = (t_{k+1} - t_k)^{-2}$, we train BASE using

$$\mathcal{L}_{\mathrm{BASE}}^{(N)} = \mathbb{E}_{c,x,\xi,k}\Big[w_k\|\bar{y}_{k+1} - y_{k+1}^\star\|_2^2 + \lambda_{\mathrm{phys}} w_k \|R_c(\mathcal{A}^{-1}(\bar{y}_{k+1}))\|_2^2\Big], \tag{6}$$

where k is sampled uniformly from $\{0, \ldots, N-1\}$. The interval-dependent weight compensates for the different magnitudes of finite increments: longer intervals naturally induce larger state changes, and the factor w_k prevents them from dominating the regression objective solely for that reason.

Crucially, the physical term does not introduce a separate transition destination. The supervised endpoint $y^\star_{k+1}$ has already been constructed through physical completion. If BASE predicts this endpoint exactly, the transition error vanishes and its physical residual is bounded by $\varepsilon_{\mathrm{dec},c}$. With exact completion, both terms attain zero at the same prediction. Thus, for each paired teacher transition, distributional transition matching and physical supervision are compatible at the sample level up to the completion tolerance.

This compatibility does not imply that deterministic regression exactly reproduces the teacher process at the population level. A current state can in general correspond to multiple teacher next states, in which case squared transition regression favors their conditional mean. For nonlinear constraints, this mean need not remain feasible even when each individual teacher endpoint is. We refer to the resulting physical error as the *constraint averaging gap* (CAG). Appendix B characterizes its dependence on transition ambiguity, constraint curvature, and transition length, and Appendix A analyzes the corresponding distributional effect of deterministic finite-transition regression.

The deployment grid $\{t_k\}_{k=0}^{N}$ is fixed before training and is used unchanged when constructing teacher states and executing the learned sampler. Changing N changes the finite transitions being learned, so each deployment budget uses its corresponding BASE checkpoint. For multi-step models, the grid may place more transitions earlier in the sequence, where an inaccurate prediction can influence a larger fraction of the subsequent rollout; the schedules used in our experiments are detailed in Appendix D.

3.4 Transition recovery distillation

The teacher objective above trains BASE on exact states $y^\star_k$. During generation, however, the input to transition $k+1$ is the prediction produced at transition k, rather than the exact teacher state. Small prediction errors can therefore shift subsequent inputs away from the teacher-state distribution and accumulate over the rollout. This discrepancy is particularly important for physical generation because a small error in one transition can produce a physically inconsistent starting state for all subsequent updates.

One direct solution would be to apply D_c after every learned transition. Doing so would recover the physical coupling explicitly, but it would also reintroduce the task-specific physical computation into sampling. Instead, we use D_c only offline to teach a lightweight recovery network what correction should follow a BASE proposal.

Specifically, freeze BASE and let the current rollout state y_k produce

$$\bar{y}_{k+1} = T_{\theta,N}(y_k, t_k, c). \tag{7}$$

We map this provisional state back to physical coordinates, extract the corresponding free variables, and apply the same physical completion map used to construct the teacher sequence:

$$y^{\mathrm{phys}}_{k+1} = \mathcal{A}\left(D_c\big(E_c\big(\mathcal{A}^{-1}(\bar{y}_{k+1})\big)\big)\right). \tag{8}$$

For problems with explicitly dependent variables, this operation retains the free quantities proposed by BASE and recomputes only their coupled counterparts. For projection-based constraints, the completion map instead applies the corresponding feasible projection. In either case, the operation answers the same question as in the teacher construction: given the state proposed in the free coordinates, what complete field is implied by the selected physical relations?

The recovery target is the discrepancy between the provisional prediction and this completed state,

$$d^\star_{N,k} = y^{\mathrm{phys}}_{k+1} - \bar{y}_{k+1}. \tag{9}$$

Importantly, y^{phys}_{k+1} is not the paired teacher endpoint $y^\star_{k+1}$. Recovery therefore does not force a rollout sample back onto its original training trajectory. It follows the state actually proposed by BASE and learns only the correction implied by physical completion of that proposal.

We construct these targets offline from BASE-generated states and train a network $g_{\phi,N}$ by

$$\mathcal{L}^{(N)}_{\mathrm{TRD}} = \mathbb{E}\left[\left\| g_{\phi,N}(y_k, \bar{y}_{k+1} - y_k, t_k, c) - d^\star_{N,k} \right\|_2^2\right]. \tag{10}$$

The physical completion used in Eq. 8 is required only to construct these training targets. Once TRD is trained, its forward pass approximates the corresponding correction directly. In this way, the same physical completion mechanism first defines the feasible teacher transitions learned by BASE and then supervises how BASE-generated rollout states should be recovered when those learned transitions are imperfect.

3.5 Sampling and step budget

For the formal PAFM sampler, the initial state is the physically completed prior

$$y_0 = \mathcal{A}\left(D_c(G_c(\xi))\right). \tag{11}$$

Generation then repeatedly applies one learned finite transition followed by one learned recovery:

$$\begin{aligned} \bar{y}_{k+1} &= T_{\theta,N}(y_k, t_k, c), \\ y_{k+1} &= \bar{y}_{k+1} + g_{\phi,N}(y_k, \bar{y}_{k+1} - y_k, t_k, c), \qquad k = 0, \dots, N-1, \\ \widehat{x} &= \mathcal{A}^{-1}(y_N). \end{aligned} \tag{12}$$

The recovered state y_{k+1} is immediately passed to the next finite transition. The sampler therefore follows the same prescribed transition grid used to construct its teacher sequence, while replacing both the finite transition and the physical correction by learned forward maps.

A rollout with budget N requires N BASE evaluations and N lightweight TRD evaluations. The physically completed initialization above is separate from the rollout itself. Appendix E further evaluates the same frozen BASE and TRD models from physics-free Gaussian initialization, while Appendix J reports the cost of completed-prior construction separately.

4 Experiments and Results

4.1 Experimental setup

Benchmarks. We evaluate PAFM on three benchmarks spanning linear, local nonlinear, and globally coupled constraints.

Kolmogorov Flow is conditioned on a scalar Reynolds number and requires a two-channel velocity field to satisfy the linear incompressibility constraint. Dynamic Stall is conditioned on four operating parameters, the free-stream Mach number, mean angle of attack, pitching amplitude, and reduced frequency, and jointly generates six fields linked by the ideal-gas and wall-shear relations. Darcy Flow is unconditional and couples permeability and pressure through a global elliptic PDE. Their prescribed physical completion maps use a Fourier-Leray projection, pointwise evaluation of known relations, and a numerical elliptic solve, respectively. Task definitions and preprocessing are given in Appendix C.

Evaluation protocol. Kolmogorov Flow is evaluated on 16 unseen Reynolds numbers, with 128 generated samples per condition. Dynamic Stall uses 16 unseen operating conditions and the same number of samples per condition. Darcy Flow contains $10{,}000$ training pairs and $1{,}000$ reference pairs on a 64×64 grid; we generate $1{,}024$ samples for evaluation.

Physical consistency is measured by the task-specific residual error (RE). Distributional metrics include one-dimensional Wasserstein distance (WD), histogram Jensen–Shannon divergence (JS), and, for conditional tasks, the MSE of the mean and standard-deviation fields (MMSE and SMSE). Dynamic Stall reports Ideal, Wall, and their sum, Total. In Darcy Flow, we report distributional metrics for pressure p, permeability K, and $\log K$; permeability must remain positive and is modeled in log-space. Valid denotes the fraction of finite generated samples satisfying $K > 0$. Selection uses only training-split conditions or validation entries; the formal test data are reserved for evaluation.

Compared methods and sampling budgets. PBFM is the primary baseline because it uses the same backbone family and is available at all tested sampling budgets. FM-OT provides an unconstrained flow baseline (Lipman et al., 2023; Tong et al., 2024). The task-specific comparisons additionally include DiffusionPDE, D-Flow, PCFM, ECI, PIDM-ME, and CoCoGen, as listed in Tables 1–3, where each method appears only on benchmarks for which its native constraint mechanism is directly applicable. Kolmogorov Flow is evaluated at 20 BASE evaluations; Dynamic Stall and Darcy Flow are

Table 1: Kolmogorov Flow results on 16 unseen Reynolds numbers with 128 generated samples per condition. PAFM w/o TRD is the BASE finite transition map. Metrics are reported with the scaling factors shown in the row labels. Lower is better; best and second-best are bold and underlined.

	Baselines					Ours	
Metric	FM-OT	DiffusionPDE	D-Flow	PCFM	PBFM	w/o TRD	PAFM
NFE	20	20	20	20	20	20	20
RE $\cdot 10^1$	3.4132	0.3843	93.499	$\mathbf{6.13\times10^{-10}}$	1.2393	1.1292	<u>0.1741</u>
WD $\cdot 10^1$	1.1093	1.0867	1.5063	6.1016	1.3786	**1.0629**	<u>1.0642</u>
JS $\cdot 10^2$	6.7225	6.7947	7.0641	12.617	6.9945	**6.6869**	<u>6.6876</u>
MMSE $\cdot 10^4$	2.541	3.110	**2.105**	9.333	4.867	2.418	<u>2.407</u>
SMSE $\cdot 10^4$	3.041	**1.431**	10.385	234.012	4.250	<u>2.139</u>	2.170
IT [ms/sample]	**21.97**	2623.68	5973.78	233.04	22.24	<u>22.22</u>	22.32

evaluated at $N \in \{1, 5, 10, 20\}$. For PAFM, base NFE counts BASE evaluations; each configuration also makes N lightweight TRD calls. For optimization-based or multistage samplers, the table notes distinguish the base budget from actual network, gradient, and correction calls. Fixed high-budget references are listed separately.

Timing and implementation. Inference time (IT) measures the deployed learned generator itself: it starts from an initialized network state and includes all rollout operations, including TRD when used. Importantly, physical initialization is an evaluation convention, not a deployment requirement. As shown in Appendix E, the same frozen BASE+TRD checkpoints can generate directly from physics-free Gaussian initialization, without invoking the physical completion map. We treat completed-prior construction as an optional initialization cost rather than part of the sampler, and report it separately in Appendix J for completeness. Training costs are reported in Appendix L. Wall-clock measurements reported in tables are obtained on the same server, using the median over five repetitions after warm-up. Architectures, optimization, and recovery settings are detailed in Appendix D.

4.2 Results on Physical-Field Benchmarks

4.2.1 Kolmogorov Flow

For this Reynolds-number-conditioned velocity benchmark, Fourier projection constructs divergence-free teacher fields. The constraint is linear, so averaging feasible transitions preserves feasibility; the endpoint residual mainly reflects learned transition and rollout error. Table 1 reports physical, distributional, conditional-statistics, and timing metrics at $N = 20$.

Without TRD, PAFM already improves PBFM's residual (0.1129 versus 0.1239) while retaining FM-OT-level distributional quality. TRD reduces the endpoint residual to 0.0174 and suppresses stepwise divergence error (Figure 1) with negligible latency. Only PCFM attains a lower residual, using online projection at substantially higher inference cost.

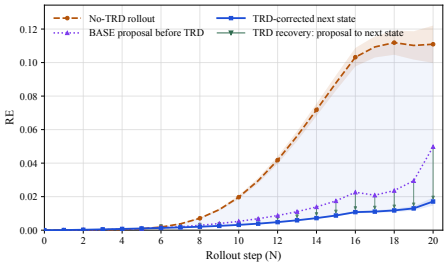


Figure 1: Kolmogorov stepwise divergence residual at $N = 20$. TRD reduces the BASE proposal residual before each subsequent step; shading shows 95% confidence intervals.

4.2.2 Dynamic Stall

Dynamic Stall is our main conditional benchmark. Given four operating parameters, the model jointly generates six high-dimensional, strongly coupled physical fields subject to two nonlinear constraints. The underlying flow also contains highly nonlinear phenomena such as moving shock waves, making this a demanding test of whether PAFM can reproduce the conditional field distribution while maintaining physical consistency across all predicted quantities.

At every shared budget, PAFM improves on PBFM in both Total residual and WD (Table 2). At $N = 20$, those gains are 97.8% and 20.8%, respectively; PBFM retains the better SMSE, indicating less accurate conditional spread for PAFM. Figure 2 shows that PCFM attains a lower residual with

Table 2: Dynamic Stall: physical residual (Total), distributional error, and runtime across budgets. Lower is better. Residuals, WD/JS, and SMSE use the scaling factors 10^6, 10^4, and 10^5, respectively. Full results, including Ideal, Wall, and MMSE, are in Appendix K.

NFE	Method	Total	WD	JS	SMSE	IT (ms)
1	PBFM	46.1802	22.4446	33.7547	6.6768	0.4332
	PAFM	0.6780	9.4064	28.0477	7.2840	0.4644
5	PBFM	0.3985	7.4753	5.0228	1.3030	1.5144
	PAFM	0.0100	6.2387	8.9274	7.2737	1.7051
10	PBFM	0.3035	7.0796	4.2106	0.9682	2.8436
	PAFM	0.0138	5.9905	5.3922	2.7233	3.3636
20	FM-OT	5.5982	6.8774	5.1800	1.3318	4.7030
	D-Flow	1.9796	8.0409	5.0270	1.4256	13288.3153
	PCFM	0.0002	9.0410	9.0251	7.6688	61.6236
	PBFM	0.3376	7.0618	3.9945	0.8781	5.7570
	PAFM	0.0075	5.5943	3.8970	1.4513	6.2478

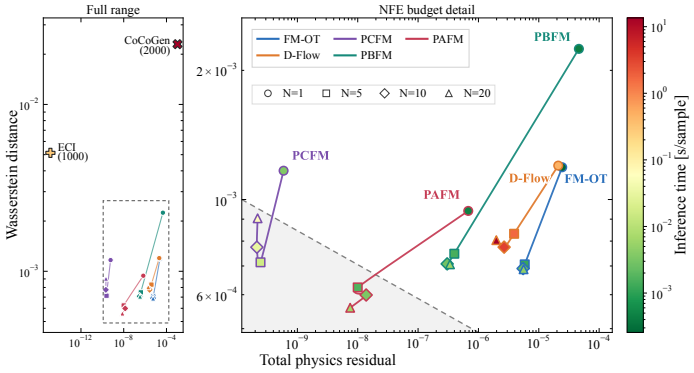


Figure 2: Dynamic Stall trade-offs across budgets. Left: full range; right: budget detail. Shapes indicate base budgets and fill colors indicate inference time.

online corrections, whereas PAFM has lower WD at every shared budget and needs about one tenth of PCFM's inference time at $N = 20$. A one-step PAFM sample has good WD but weaker SMSE; more transitions improve ensemble metrics, while residuals need not decrease monotonically. The full metrics and correction-call accounting appear in Appendix K.

4.2.3 Darcy Flow

Darcy Flow is our most demanding test of global physical coupling. Permeability K and pressure p are linked by a coefficient-dependent elliptic PDE, so a local change in K can influence p across the entire domain. Each teacher next-state must therefore preserve a nonlocal sample-wise coefficient–solution relation, making both transition prediction and rollout recovery substantially more challenging than under local algebraic constraints.

Under fixed forcing, boundary conditions, and pressure gauge, all stochastic variation is carried by $a = \log K$, while p is determined by the generated permeability. Darcy therefore directly tests whether PAFM can learn this global completion well enough to avoid online elliptic solves.

Table 3 and Figure 3 summarize the Darcy results across deployment budgets; complete metrics are reported in Appendix K, with teacher-completion ablations in Appendix I.4. At $N = 20$, PAFM reaches RE 0.0399 and $\mathrm{WD}_K = 0.1311$ in 5.60 ms/sample, improving both over PBFM (RE 0.8427, $\mathrm{WD}_K = 0.5298$). FM-OT attains lower $\mathrm{WD}_K = 0.0764$ but much larger RE 3.7771, while D-Flow is weaker in both and requires 4752 ms/sample. Across budgets, all PAFM points remain valid with RE $< 10^{-1}$, while larger budgets improve the pressure–permeability distribution (Figure 3).

Table 3: Darcy Flow: residual, marginal Wasserstein distances, validity, and runtime. All errors are lower-is-better. The full multi-budget comparison and JS metrics are in Appendix K.

NFE	Method	RE	WD_p	WD_K	Valid	IT (ms)
1	FM-OT	8.7008	0.002349	0.836530	1.0000	0.1715
	PBFM	–	–	–	0.0000	0.1749
	PAFM	0.0301	0.006710	0.797580	1.0000	0.2847
5	PAFM	0.0306	0.005009	0.410762	1.0000	1.4020
10	PAFM	0.0320	0.003218	0.240570	1.0000	2.7983
20	FM-OT	3.7771	0.000671	0.076421	0.9990	3.3712
	D-Flow	1.6144	0.002400	0.185848	0.9990	4752.133
	DiffusionPDE	5.3063	0.006656	0.034896	0.9102	165.5825
	PBFM	0.8427	0.014997	0.529766	0.9971	3.3783
	PAFM	0.0399	0.002474	0.131098	1.0000	5.6011

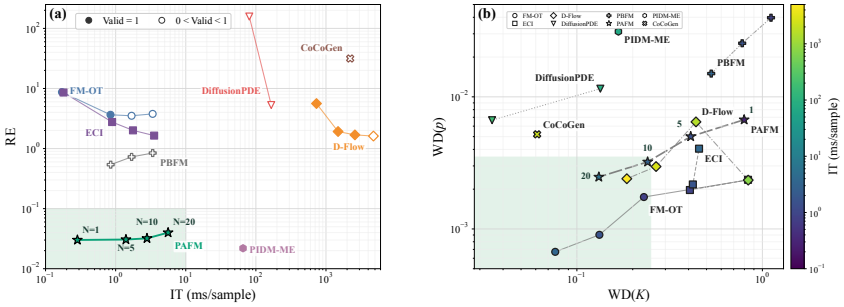


Figure 3: Darcy trade-offs across budgets: (a) residual versus inference time, with marker fill indicating validity; (b) pressure versus permeability WD, with color indicating inference time.

4.2.4 Solver-Free Initialization and Training Cost

The learned rollout does not require a physically completed initial state. Using the same frozen BASE+TRD checkpoints, Gaussian initialization in the free coordinates with zeroed coupled channels leaves the reported Kolmogorov and Dynamic Stall metrics close to the completed-prior setting; for Dynamic Stall, the physical residual is even marginally lower, while WD, JS, MMSE, and SMSE remain in the same range. For Darcy Flow, validity, physical residual, and pressure-side metrics are likewise preserved, with degradation appearing only in the permeability-distribution metrics. Full results are reported in Appendix E.

Training also remains lightweight despite the use of physical completion for supervision. At $N = 20$, on-the-fly teacher completion is already included in the measured BASE update time, which is 0.0558–0.474 s/update across the three benchmarks, with less than 3 GiB peak GPU memory; constructing the complete offline TRD target cache takes at most about 1.5 min (Appendix L). This efficiency follows from finite-transition training: each update uses a single teacher pair and does not backpropagate through a complete rollout or the physical completion map.

5 Limitations and Conclusion

Limitations. PAFM shifts physical computation to training through physically completed teacher states and recovery targets, while deployment uses only learned BASE+TRD transitions. It therefore relies on a sufficiently accurate and affordable completion map, does not guarantee exact physical feasibility of learned rollout states, and requires retraining for each prescribed deployment budget because the finite transition sequence changes with the budget.

Conclusion. Across linear, local nonlinear, and globally coupled constraints, physically completed next-state targets yield strong physical consistency and competitive distributional fidelity at millisecond-scale inference. PAFM thus organizes physics-constrained generation as a sequence of physically aligned finite next-state predictions.

AI USE STATEMENT

Generative AI tools were used to find and summarize related literature and to draft, edit, and format portions of the manuscript and its LaTeX source. We did not use generative AI to implement the method, design or tune experiments, generate or process data, formulate mathematical claims or proofs, or interpret numerical results. The authors reviewed and revised all AI-assisted text, checked references against the cited works, and checked scientific statements against the method and experimental records. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies physical-field generation using numerical simulations and previously released PDE and CFD benchmark datasets; it does not involve human participants or personal data. Generated fields are approximations and can retain nonzero physical residuals. They should therefore be checked with an appropriate physical solver before use in safety-critical engineering decisions. We report physical consistency together with distributional accuracy and inference cost to make these limitations visible.

REPRODUCIBILITY STATEMENT

§3 defines the teacher construction, BASE and TRD objectives, and sampling procedure. §4.1 specifies the evaluation protocol; Appendix C documents dataset provenance, preprocessing, splits, constraints, and metrics for all three benchmarks. Appendix D gives transition grids, architectures, optimization settings, and recovery configurations. Assumptions and proofs are provided in Appendices A and B, and complete multi-budget results are reported in Appendix K.

REFERENCES

ANSYS, Inc. Ansys Fluent 2024 R2 User's Guide, 2024. URL https://ansyshelp.ansys.com/public/Views/Secured/corp/v242/en/flu_ug/flu_ug.html.

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*, 2026. URL https://openreview.net/forum?id=tAf1KI3d4X.

Jan-Hendrik Bastek. Physics-informed diffusion models: Datasets and model checkpoints. ETH Zurich Research Collection, 2024. URL https://doi.org/10.3929/ethz-b-000674074. Data collection, CC BY 4.0.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tpYeermigp.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through flows for controlled generation. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3462–3483. PMLR, 2024. URL https://proceedings.mlr.press/v235/ben-hamu24a.html.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856. URL https://openreview.net/forum?id=cqDH0e6ak2.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2ff3605c9e442d55156d99fe4f5f228d-Paper-Conference.pdf.

Che-Chia Chang, Chen-Yang Dai, Te-Sheng Lin, Ming-Chih Lai, and Chieh-Hsin Lai. Stabilizing physics-informed consistency models via structure-preserving training. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 10555–10566, 2026. doi: 10.1145/3770855.3819059. URL https://doi.org/10.1145/3770855.3819059.

Chaoran Cheng, Boran Han, Danielle C. 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*, 2025. URL https://openreview.net/forum?id=teE4pl9ftK.

Jacob K. Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a28af221f2f70be183afc16797a56b91-Abstract-Conference.html.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html.

Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OjDkC57x5sz.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb3878c1dcbfff9ee95d5d033e5f5942-Abstract-Conference.html.

Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. CoCoGen: Physically consistent and conditioned score-based generative models for forward and inverse problems. *SIAM Journal on Scientific Computing*, 47(2):C399–C425, 2025. doi: 10.1137/24M1636071. URL https://epubs.siam.org/doi/10.1137/24M1636071.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3934–3942. PMLR, 2024. URL https://proceedings.mlr.press/v238/kerrigan24a.html.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

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. doi: 10.1145/3648506. URL https://doi.org/10.1145/3648506.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3D flow simulation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/15972a9575e0f03bf82f00aebeb40774-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Qiang Liu, Felix Koehler, and Nils Thuerey. TorchFSM: Fourier spectral method with PyTorch. Zenodo, May 2025. URL https://doi.org/10.5281/zenodo.15350210.

Da Long, Zhitong Xu, Guang Yang, Akil Narayan, and Shandian Zhe. Arbitrarily-conditioned multi-functional diffusion for multi-physics emulation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40270–40289. PMLR, 2025. URL https://proceedings.mlr.press/v267/long25b.html.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL https://www.nature.com/articles/s42256-021-00302-5.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Jan Tauberschmidt, Sophie Fellenz, Sebastian Josef Vollmer, and Andrew B. Duncan. Physics-constrained fine-tuning of flow-matching models for generation and inverse problems. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=khBHJz2wcV.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gómez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=cf4etwjY7n.

Sifan Wang, Zehao Dou, Siming Shan, Tong-Rui Liu, and Lu Lu. FunDiff: Diffusion models over function spaces for physics-informed generative modeling. *Nature Communications*, 17 (1):5749, 2026. doi: 10.1038/s41467-026-72292-0. URL https://doi.org/10.1038/s41467-026-72292-0.

Yi Zhang, Peng Wang, and Difan Zou. Physics-informed distillation of diffusion models for PDE-constrained generation. In *ICML 2026 Workshop on Foundations of Deep Generative Models*, 2026. URL https://arxiv.org/abs/2505.22391.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78651–78686. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25c.html.

A Probabilistic Formulation and Transport Error Bounds

This section formulates PAFM as a discrete probabilistic transport model and characterizes its distributional and physical errors. Appendix B then analyzes the constraint averaging gap arising from its deterministic transition objective.

A.1 Physically completed probability paths

Fix a condition c. We work in normalized state coordinates and write

$$D = \mathcal{A} \circ D_c, \qquad E = E_c \circ \mathcal{A}^{-1}, \qquad R = R_c \circ \mathcal{A}^{-1}.$$

where A is the fixed affine normalization. Thus, D applies the fixed physical completion map in normalized coordinates, E extracts task-defined generating coordinates, and R evaluates the physical residual.

Let p_{data} denote the normalized data distribution and define

$$q_0 = \mathcal{N}(0, I), \qquad q_1 = E_{\#} p_{\mathrm{data}}.$$

Here $\#$ denotes pushforward of a probability measure. The source–target coupling π has marginals q_0 and q_1. For $(Z_0, Z_1) \sim \pi$, PAFM constructs

$$Z_t = (1-t)Z_0 + te^{-\kappa(1-t)L} Z_1, \qquad X_t = D(Z_t). \tag{13}$$

On the deployment grid $0 = t_0 < \cdots < t_N = 1$, write $X_k = X_{t_k}$, $h_k = t_{k+1} - t_k$, and $\mu_k = \mathrm{Law}(X_k)$. The bridge endpoints and the physical completion bound give

$$\mu_0 = D_{\#} q_0, \qquad \mu_N = D_{\#} q_1, \qquad \|R(X_k)\|_2 \le \epsilon_{\mathrm{dec}} \quad \text{almost surely.} \tag{14}$$

These statements do not require an ambient-space density or a global smooth-manifold parameterization.

The terminal teacher distribution contains physically re-completed data. Consequently,

$$W_2(\mu_N, p_{\mathrm{data}}) \le \eta_{\mathrm{data}}, \qquad \eta_{\mathrm{data}}^2 = \mathbb{E}_{Y \sim p_{\mathrm{data}}} \|D(E(Y)) - Y\|_2^2. \tag{15}$$

The bound follows by coupling each data state with its re-completed counterpart. It separates re-completion bias from learning error.

A.2 Conditional transitions and finite-increment learning

The teacher path induces a conditional transition kernel

$$K_k(x, B) = \mathbb{P}(X_{k+1} \in B \mid X_k = x).$$

By the law of total probability,

$$\mu_{k+1}(B) = \int K_k(x, B)\, \mu_k(dx). \tag{16}$$

Thus, successively sampling these exact kernels from μ_0 recovers all teacher marginals. This statement does not require the original paired path to be Markov; it concerns its one-time distributions.

PAFM represents each teacher transition by a deterministic finite map. For a given current state x, however, the teacher process may admit multiple possible next states. We therefore define their conditional mean and covariance as

$$m_k(x) = \mathbb{E}[X_{k+1} \mid X_k = x], \qquad \Sigma_k(x) = \mathrm{Cov}(X_{k+1} \mid X_k = x).$$

Under squared transition loss, any deterministic next-state predictor B_k satisfies

$$\begin{gathered} \mathbb{E}[\|B_k(X_k) - X_{k+1}\|_2^2] = \mathbb{E}[\|B_k(X_k) - m_k(X_k)\|_2^2] + V_k, \\ V_k := \mathbb{E}[\mathrm{tr}\, \Sigma_k(X_k)]. \end{gathered} \tag{17}$$

Here V_k is the expected conditional variance of the teacher transition. Equivalently,

$$V_k = \mathbb{E}\big\|X_{k+1} - m_k(X_k)\big\|_2^2, \tag{18}$$

so it measures how much the possible next states remain dispersed even after the current state X_k is known. It is therefore an intrinsic ambiguity of the deterministic transition problem rather than an approximation error of the network.

Because V_k is independent of B_k, the unrestricted Bayes-optimal deterministic prediction is

$$B_k^\star(x) = m_k(x).$$

PAFM parameterizes the same transition through a finite increment,

$$B_k(x) = x + f_{\theta,N}(x, t_k, c),$$

and hence the corresponding optimal increment is

$$f_{\theta,N}^\star(x, t_k, c) = m_k(x) - x. \tag{19}$$

Thus, squared finite-transition regression drives the model toward the conditional mean transition, while any nonzero V_k represents transition variability that a deterministic map cannot reproduce. Positive step-dependent weighting only rescales this transition-only objective and does not change its conditional optimum.

Even when this conditional mean is learned exactly, replacing the stochastic teacher transition by a deterministic mean map generally does not preserve the next-state distribution. By the law of total covariance,

$$\mathrm{Cov}(X_{k+1}) = \mathrm{Cov}(m_k(X_k)) + \mathbb{E}[\Sigma_k(X_k)]. \tag{20}$$

The deterministic Bayes predictor $B_k^\star(X_k) = m_k(X_k)$ retains only the first term. Hence, whenever $V_k > 0$, it discards nonzero conditional variation and cannot exactly reproduce the teacher next-state marginal in general. This is a distributional consequence of finite-transition averaging; whether the averaged prediction also violates the physical constraint depends on the geometry of that constraint and is analyzed in Appendix B.

A.3 Distributional and physical error bounds

Let T_k be the complete deployed transition, including the recovery head when enabled:

$$\widehat{X}_{k+1} = T_k(\widehat{X}_k), \qquad \widehat{\mu}_k = \mathrm{Law}(\widehat{X}_k).$$

Set

$$e_k = W_2(\widehat{\mu}_k, \mu_k), \qquad b_k = \|T_k - m_k\|_{L^2(\mu_k)}.$$

The term b_k includes approximation error and any departure from the conditional mean caused by physical supervision or recovery.

Proposition (finite-budget transport error). Assume finite second moments and that T_k is ℓ_k-Lipschitz on the relevant states. Then

$$e_{k+1}^2 \le (\ell_k e_k + b_k)^2 + V_k. \tag{21}$$

Consequently, with

$$w_k = \prod_{j=k+1}^{N-1} \ell_j, \qquad w_{\mathrm{init}} = \prod_{j=0}^{N-1} \ell_j,$$

and empty products equal to one,

$$W_2(\widehat{\mu}_N, p_{\mathrm{data}}) \le \sum_{k=0}^{N-1} w_k b_k + \left(w_{\mathrm{init}}^2 e_0^2 + \sum_{k=0}^{N-1} w_k^2 V_k \right)^{1/2} + \eta_{\mathrm{data}}. \tag{22}$$

Proof. Couple $\widehat{X} \sim \widehat{\mu}_k$ and $X \sim \mu_k$ optimally. Given X, draw $Y \sim K_k(X, \cdot)$ independently of $\widehat{X}$. Then $Y \sim \mu_{k+1}$, and conditional centering yields

$$\begin{aligned} \mathbb{E}\|T_k(\widehat{X}) - Y\|_2^2 &= \mathbb{E}\|T_k(\widehat{X}) - m_k(X)\|_2^2 + V_k \\ &\le (\ell_k e_k + b_k)^2 + V_k. \end{aligned}$$

This proves Eq. 21. Unrolling the recursion using

$$\sqrt{(u+v)^2+w^2} \le \sqrt{u^2+w^2}+v, \qquad u,v,w \ge 0,$$

and then applying Eq. 15 gives Eq. 22. □

The bound distinguishes prior mismatch, conditional ambiguity, map error and its rollout amplification, and endpoint re-completion error. In particular, if uniformly over a sequence of trained grids

$$\ell_k \le 1 + Lh_k, \qquad V_k \le M^2 h_k^2,$$

then

$$W_2(\widehat{\mu}_N, p_{\mathrm{data}}) \le e^L \left(e_0 + \sum_k b_k + M\sqrt{h_{\max}} \right) + \eta_{\mathrm{data}}, \qquad h_{\max} = \max_k h_k. \tag{23}$$

Thus, stable deployment-aligned maps are consistent with the re-completed target when $h_{\max} \to 0$, $e_0 \to 0$, and $\sum_k b_k \to 0$. These are additional assumptions, not consequences of increasing the sampling budget of a fixed checkpoint.

Physical-closure recovery. Define $P = D \circ E$. For a provisional sampling state $\bar{X}$, write the recovered output as

$$\widehat{X} = P(\bar{X}) + \varepsilon, \qquad \rho = \left(\mathbb{E}\|\varepsilon\|_2^2\right)^{1/2}.$$

Assume the extracted variables are in the D_c's valid domain. If R is L_R-Lipschitz, then

$$\left(\mathbb{E}\|R(\widehat{X})\|_2^2\right)^{1/2} \le \epsilon_{\mathrm{dec}} + L_R\rho. \tag{24}$$

If D is also L_D-Lipschitz and $\bar{q} = \mathrm{Law}(E(\bar{X}))$, then

$$W_2(\mathrm{Law}(\widehat{X}), p_{\mathrm{data}}) \le \rho + L_D W_2(\bar{q}, q_1) + \eta_{\mathrm{data}}. \tag{25}$$

Both follow by comparing $\widehat{X}$ with $P(\bar{X})$. They separate recovery accuracy from underlying distribution accuracy: completion-aligned recovery can control physical error without itself recovering missing conditional variability. Here ρ is evaluated on actual sampling inputs; a small empirical training loss alone does not establish this quantity.

B Constraint Averaging Gap in Finite Transitions

Appendix A establishes that squared finite-increment regression selects a conditional mean. We now characterize when this mean violates the physical constraints, even though every teacher next state is feasible. We retain its notation and normalized coordinates.

B.1 From feasible samples to an averaged prediction

Fix a step k and current state $X_k = x$. Write

$$Y = X_{k+1}, \qquad m = m_k(x), \qquad \delta = Y - m, \qquad \Sigma = \Sigma_k(x).$$

Each realized teacher increment $Y - x$ reaches a state satisfying

$$\|R(Y)\|_2 \le \epsilon_{\mathrm{dec}}.$$

Thus, matching an individual teacher increment is compatible with teacher feasibility. However, the network does not observe the hidden endpoint pairing and generally learns an average over its possible transitions.

We define

$$\mathrm{CAG}_k(x) = \|R(m_k(x))\|_2. \tag{26}$$

CAG is the physical residual of the conditional-mean prediction, not the conditional variance itself. With a numerical completion map, it is interpreted relative to the completion residual floor.

B.2 Curvature and conditional ambiguity

Proposition (curvature–ambiguity bound). Suppose R is twice continuously differentiable on a convex region containing the conditional next states and their mean, with

$$\|D^2R(y)[u,u]\|_2 \le M_R\|u\|_2^2.$$

Then

$$\mathrm{CAG}_k(x) \le \epsilon_{\mathrm{dec}} + \frac{M_R}{2}\operatorname{tr}\Sigma_k(x). \tag{27}$$

Proof. Taylor expansion with integral remainder gives

$$R(m+\delta) = R(m) + J_R(m)\delta + \int_0^1 (1-s)D^2R(m+s\delta)[\delta,\delta]\,ds.$$

Taking conditional expectations eliminates the linear term because $\mathbb{E}[\delta \mid x] = 0$. Hence

$$\begin{aligned} R(m) &= \mathbb{E}[R(Y) \mid x] \\ &\quad - \mathbb{E}\left[\int_0^1 (1-s)D^2R(m+s\delta)[\delta,\delta]\,ds \,\middle|\, x\right]. \end{aligned} \tag{28}$$

The physical completion bound and $\mathbb{E}[\|\delta\|_2^2 \mid x] = \operatorname{tr}\Sigma_k(x)$ prove the claim. □

With exact physical completion, CAG vanishes if the conditional transition is unique or if the constraint is affine. The latter explains why averaging incompressible teacher states can preserve a linear divergence constraint while still changing the generated covariance, as shown in Eq. 20. For nonlinear relations, ambiguity contributes to physical error through the curvature of the constraint.

Dependence on transition length. Suppose locally, for an interval of length h,

$$X_{t+h} - X_t = hU_t + Q_h, \qquad \mathbb{E}[\|Q_h\|_2^2 \mid X_t = x] = O(h^4),$$

with bounded conditional second moment of U_t. Then

$$\operatorname{Cov}(X_{t+h} \mid X_t = x) = h^2 \operatorname{Cov}(U_t \mid X_t = x) + O(h^3),$$

and therefore

$$\mathrm{CAG}_{t,h}(x) \le \epsilon_{\mathrm{dec}} + O(h^2). \tag{29}$$

Shorter transitions reduce the displacement over which ambiguity interacts with constraint curvature; they need not remove the underlying endpoint ambiguity. Heat and linear paths induce different conditional covariances, but the bound does not imply that heat always produces a smaller CAG.

Distributional effect of transition averaging. The conditional ambiguity underlying CAG also affects distributional fidelity. Eq. 20 shows that deterministic transition regression suppresses the conditional variability represented by $\Sigma_k(x)$. Under the local short-step assumption above, $\operatorname{tr}\Sigma_{t,h}(x) = O(h^2)$, so finer transition grids reduce the variability averaged out by each finite map. Constraint curvature determines whether the same averaging also induces physical error: for affine constraints, CAG vanishes even though the distributional averaging effect may remain. The improved SMSE at larger deployment budgets in Dynamic Stall is consistent with this reduced local averaging, while the complete multi-step behavior also depends on approximation and rollout errors.

B.3 Implications for the PAFM objective

By Eq. 17, the unrestricted conditional base objective is, up to an additive constant and a positive interval weight,

$$\min_y \|y - m_k(x)\|_2^2 + \lambda_{\mathrm{phys}}\|R(y)\|_2^2. \tag{30}$$

For any global minimizer y^λ, comparison with $y = m_k(x)$ gives

$$\|y^\lambda - m_k(x)\|_2 \le \sqrt{\lambda_{\mathrm{phys}}}\,\mathrm{CAG}_k(x). \tag{31}$$

Thus, physical supervision can move the prediction away from an infeasible conditional mean, while its idealized displacement is controlled by CAG. It does not reveal the hidden pairing or guarantee

Table 4: Benchmark instantiations. "Condition" denotes information supplied to the neural generator. Dynamic Stall geometry is fixed side information and is not counted among the six generated fields.

Kolmogorov Flow	
Grid	128^2, periodic
Generated state x	(u_x, u_y)
Free variables z	unconstrained two-channel velocity field
Condition c	scalar Re
Field(s) set by physical completion	none in the stored state
Sample-wise constraint	$\nabla \cdot u = 0$
Dynamic Stall	
Grid	128^2
Generated state x	$(p, g_x, g_y, T, \rho, \tau_w)$
Free variables z	$(\log T, \log \rho, q, \eta)$
Condition c	four operating parameters
Field(s) set by physical completion	(p, τ_w)
Sample-wise constraint	ideal-gas and wall-shear relations
Darcy Flow	
Grid	64^2
Generated state x	(p, a), where $a = \log K$
Free variables z	a
Condition c	none
Field(s) set by physical completion	p
Sample-wise constraint	$-\nabla \cdot (e^a \nabla p) = f$

recovery of the full conditional distribution. Physical-closure recovery similarly controls physical error through Eq. 24, while its distributional effect remains covered by Equations 22 and 25.

Finally, fixed training pairs should not be confused with a deterministic population coupling. A finite bank can assign a unique endpoint to each stored source without establishing unique transitions for unseen sources. For example, under independent source–target pairing and a single step,

$$m_0(x) = \mathbb{E}[X_1],$$

so transition-only population regression selects a constant output. The guarantees therefore retain conditional ambiguity and learning error explicitly, rather than identifying exact teacher feasibility with exact finite-budget generation.

C Benchmark Definitions, Data, and Evaluation Protocols

This section specifies the generated random field, conditioning information, physical constraints, data split, and evaluation protocol for each benchmark. The three tasks cover a linear incompressibility constraint, local nonlinear algebraic relations, and a globally coupled elliptic PDE. Throughout this paper, *BASE* denotes the finite-transition backbone before transition recovery distillation (TRD). BASE predicts increments in the complete normalized state, whereas the free-variable representation is used to construct physically consistent teacher paths and to identify the channels that TRD may correct. Table 4 summarizes the three instantiations.

C.1 Kolmogorov Flow

Dataset provenance and numerical generation. We use the Kolmogorov Flow HDF5 datasets released with PBFM (Baldan et al., 2026). The data were generated at spatial resolution 128×128

using the vorticity–streamfunction formulation and the GPU-accelerated Fourier spectral solver TorchFSM (Liu et al., 2025). The corresponding TorchFSM Kolmogorov configuration uses the periodic domain $[0, 2\pi)^2$ and evolves vorticity according to

$$\partial_t \omega + \boldsymbol{u} \cdot \nabla \omega = \frac{1}{\mathrm{Re}} \Delta \omega - 0.1\,\omega - 4\cos(4y),$$

with the velocity recovered from vorticity through the periodic streamfunction solve. TorchFSM uses a fourth-order stable exponential time-differencing Runge–Kutta integrator (SETDRK4) by default for nonlinear operators, together with a default $2/3$ de-aliasing rule. The PBFM simulations use the time step $\Delta t = 1/\mathrm{Re}$. Initial vorticity fields follow the diffused-random-noise construction provided by TorchFSM.

For each Reynolds-number condition, the PBFM generation protocol runs 256 simulations from slightly perturbed initial states. Each simulation is first advanced for 10,000 time steps to reach a statistically steady state, after which snapshots are sampled every 4,000 time steps. This procedure yields 1,024 stored velocity snapshots per condition (Baldan et al., 2026). We use the released HDF5 fields directly and do not regenerate the simulation trajectories.

Field, conditioning, and residual. Each realization is a two-component velocity field $u = (u_x, u_y) \in \mathbb{R}^{2\times 128\times 128}$ on the periodic domain $[0, 2\pi)^2$. The scalar Reynolds number indexes the conditional data distribution and is normalized before entering the model as

$$\widetilde{\mathrm{Re}} = \frac{\mathrm{Re} - 300}{200} = \frac{\mathrm{Re} - 100}{200} - 1. \tag{32}$$

Thus, the reference interval $\mathrm{Re} \in [100, 500]$ is mapped to $[-1, 1]$. Independently, both velocity components are divided by the fixed field scale 5.0. Physical consistency is measured only through incompressibility,

$$R_{\mathrm{K}}(u) = \partial_x u_x + \partial_y u_y, \qquad \mathrm{RE}_{\mathrm{K}} = \mathbb{E}\left[\|R_{\mathrm{K}}(u)\|_2^2\right]. \tag{33}$$

Both derivatives follow the same periodic spectral convention as Leray projection and evaluator. The reported RE is therefore the divergence residual, not the full Navier–Stokes momentum residual.

Physical completion and physical prior. Let P_{L} denote the discrete Fourier–Leray projector. For Kolmogorov Flow, the free-variable state z is an unconstrained two-channel velocity field, while the complete teacher state is obtained by Leray projection,

$$D(z) = P_L\, z, \tag{34}$$

We use the identity prior in the free-variable space,

$$z_0 = \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{35}$$

and project it to the physical initial state

$$x_0 = P_L\, z_0. \tag{36}$$

The teacher sequence uses a periodic Fourier heat semigroup with $\kappa = 0.01$ and the same Leray convention at every physically completed node.

Data split and preprocessing. The training HDF5 contains 32 Reynolds conditions and 1,024 fields per condition. The disjoint test HDF5 contains 16 unseen Reynolds conditions and 1,024 reference fields per condition. Formal evaluation generates 128 samples for each unseen condition, for 2,048 generated fields in total. BASE and TRD selection use fixed conditions from the training file only; the formal test file is accessed only after selection. Velocity channels are divided by five during training and converted back to physical units before evaluation.

C.2 Dynamic Stall

Dataset provenance and CFD generation. We use the Dynamic Stall datasets released with PBFM directly, without regenerating the underlying CFD solutions (Baldan et al., 2026). The datasets

were generated by solving the unsteady, compressible, two-dimensional RANS equations around a sinusoidally pitching NACA0012 airfoil using ANSYS Fluent 2024 R2 (ANSYS, Inc., 2024). The computational mesh is an O-grid with 512 nodes along the airfoil surface and 128 nodes in the wall-normal direction. Spatial discretization uses a second-order upwind scheme, time integration uses a second-order implicit scheme, gradients are reconstructed with the least-squares cell-based method, and pressure–velocity coupling is handled by the coupled solver with Rhie–Chow momentum interpolation. Turbulence closure is provided by the SST model augmented with an intermittency transport equation.

The sinusoidal pitching motion is imposed as rigid-body motion of the mesh,

$$\dot{\alpha}(t) = \omega \alpha_s \cos(\omega t), \tag{37}$$

with the initial mesh rotated to the prescribed mean incidence α_0. Each pitching cycle is resolved with 2,048 time steps, and the simulations are continued until periodic convergence (Baldan et al., 2026). The wall quantities over one complete pitching cycle are subsequently represented on a 128×128 surface-location–cycle-phase grid. Hence, the two axes of each stored field correspond to position along the airfoil surface and position within the pitching cycle, respectively, rather than to the two Cartesian coordinates of the full CFD domain.

State and local physical relations. One sample is a six-channel field

$$x = (p, g_x, g_y, T, \rho, \tau_w) \in \mathbb{R}^{6\times 128\times 128}. \tag{38}$$

Two additional geometry arrays encode the known local unit tangent. For clarity, we denote this tangent by $\boldsymbol{t} = (t_x, t_y)$ and its unit normal by $\boldsymbol{n} = (-t_y, t_x)$. These arrays are fixed side information used by the physical completion map, residual, and recovery head; they are neither generated nor taken from a future target. At each pixel, the state is required to satisfy

$$p = \rho R_{\text{air}} T, \qquad R_{\text{air}} = \frac{8.31446261815324}{28.966} \times 10^3, \tag{39a}$$

$$\tau_w = \operatorname{sgn}(\mu(T)\boldsymbol{g}\cdot\boldsymbol{t})\, \mu(T) \|\boldsymbol{g}\|_2, \qquad \boldsymbol{g} = (g_x, g_y), \tag{39b}$$

$$\mu(T) = 1.716 \times 10^{-5} \frac{273.11 + 110.56}{T + 110.56} \left(\frac{T}{273.11}\right)^{3/2}. \tag{39c}$$

The Ideal and Wall metrics in the main paper are the squared normalized residuals of Equations 39a and 39b; Total is their sum.

Free variables and analytical completion. We parameterize the gradient field by a signed magnitude q and an angular coordinate δ,

$$\boldsymbol{g} = q\big(\cos\delta\, \boldsymbol{t} + \sin\delta\, \boldsymbol{n}\big), \qquad \delta = \left(\frac{\pi}{2} - 10^{-4}\right) \tanh\eta. \tag{40}$$

Together with $\log T$ and $\log\rho$, the standardized tuple $z = (\log T, \log\rho, q, \eta)$ provides four independent coordinates. The fixed completion map exponentiates the first two coordinates, reconstructs (g_x, g_y) using Eq. 40, and sets $p = \rho R_{\text{air}} T$ and $\tau_w = \mu(T) q$. Only training-set statistics are used for standardization. Before exponentiation, $\log T$ and $\log\rho$ are clipped to the corresponding training range enlarged by a 10% margin. A four-channel Gaussian draw can therefore be lifted pointwise to a complete six-channel physical prior.

The six stored channels are normalized with means $(10^5, 10^7, 4{\times}10^5, 290, 1.17, -2.55)$ and standard deviations $(8\times10^4, 1.8\times10^7, 2\times10^7, 56, 0.8, 400)$. These affine transformations are coordinate encodings and do not alter the physical definitions above.

Conditions and split. Each operating condition is one four-dimensional vector, rather than one entry of a Cartesian product. Its components are the free-stream Mach number, mean angle of attack, pitching amplitude, and reduced frequency. The physical operating ranges are $M_\infty \in [0.3, 0.5]$, $\alpha_0, \alpha_s \in [5^\circ, 10^\circ]$, and $k \in [0.05, 0.1]$ (Baldan et al., 2026). To reproduce the released preprocessing, we use the condition channels exactly as stored in the PBFM HDF5 files and apply the released channelwise normalization unchanged.

The training file contains 128 distinct nominal condition vectors and 32 realizations per nominal condition. The test file contains 16 unseen nominal condition vectors with 32 reference realizations each, and formal evaluation generates 128 samples per test condition. Training-file conditions 0–111 are used for model fitting, while conditions 112–127 are reserved for BASE and TRD selection. None of the formal test conditions are used for training or model selection.

C.3 DARCY FLOW

Target-data provenance and permeability distribution. We use the Darcy Flow coefficient–solution pairs released with Physics-Informed Diffusion Models (Bastek et al., 2025; Bastek, 2024). The permeability is log-normal,

$$K(\xi) = \exp G(\xi), \qquad G(\cdot) \sim \mathcal{N}(0, \kappa), \tag{41}$$

with exponential covariance

$$\kappa(\xi, \xi') = \exp\left(-\frac{\|\xi - \xi'\|_2}{0.1}\right). \tag{42}$$

Thus, G has zero mean, unit marginal variance, and correlation length $l = 0.1$. In our notation, the free variable is the log-permeability $a(\xi) = \log K(\xi) = G(\xi)$. The random field is represented by the leading 64 terms of its Karhunen–Loève expansion,

$$G(\xi) = \sum_{i=1}^{64} \sqrt{\lambda_i}\, z_i \phi_i(\xi), \qquad z_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1), \tag{43}$$

where the eigenpairs (λ_i, ϕ_i) are ordered by decreasing eigenvalue.

The corresponding pressure fields are computed on a 64×64 grid using second-order central finite differences. The discrete Darcy equations, homogeneous no-flux boundary conditions, and zero-mean pressure constraint are assembled into an overdetermined linear system, which is solved using `scipy.linalg.lstsq`. The released dataset contains 10,000 training pairs and 1,000 validation pairs.

Coefficient–solution distribution. The generated state is $x = (p, a)$ on $\Omega = [0, 1]^2$, discretized on a 64×64 grid, with permeability $K = e^a > 0$. Pressure satisfies

$$-\nabla \cdot (K \nabla p) = f \qquad \text{in } \Omega, \tag{44a}$$

$$\nabla p \cdot n = 0 \qquad \text{on } \partial\Omega, \tag{44b}$$

$$\int_\Omega p \, \mathrm{d}x = 0. \tag{44c}$$

The forcing is fixed for every sample:

$$f(x, y) = \begin{cases} 10, & (x, y) \in [0, 0.125]^2, \\ -10, & (x, y) \in [0.875, 1]^2, \\ 0, & \text{otherwise.} \end{cases} \tag{45}$$

Thus, the forcing, homogeneous Neumann boundary condition, and zero-mean gauge are fixed properties of the benchmark rather than learned or sampled quantities. All stochastic variation is carried by $a = \log K$. The completion map exponentiates a and solves Equations 44a– 44c for p. The boundary condition imposes zero normal pressure gradient, while the gauge removes the additive null space of the all-Neumann elliptic system.

Data and initial prior. We use 10,000 training coefficient–solution pairs and 1,000 disjoint reference pairs. Formal metrics are computed from 1,024 generated samples. The physical prior is obtained by drawing a Gaussian field a_0, setting $K_0 = e^{a_0}$, and solving the same discretized Darcy system once for p_0. The 1,024 formal-evaluation priors are stored only to keep checkpoint and method comparisons deterministic. Sampling begins from a prepared prior; the subsequent learned updates invoke no PDE solver.

Table 5: Candidate $N = 20$ schedule family. The final four step sizes are fixed to 0.05 in every configuration.

Schedule	(a, b), first 16 steps	$h_{16:19}$	Total
Uniform 5–5	$(0.05, 0.05)$	0.05	1.00
Mild 4–6	$(0.04, 0.06)$	0.05	1.00
Strong 3–7	$(0.03, 0.07)$	0.05	1.00
Aggressive 2–8	$(0.02, 0.08)$	0.05	1.00

Training path sampling. Training examples are traversed through seeded shuffled permutations of all $10{,}000$ targets, without replacement within each data epoch. Each visited target is paired with a newly drawn independent Gaussian a_0, from which the two finite Heat states required by the sampled transition are physically completed on demand. BASE therefore sees the full target set repeatedly together with fresh source realizations.

Validity and metrics. Darcy RE is computed with the released strong-form residual under the same interior, Neumann-boundary, and gauge conventions. Valid is the fraction of finite samples whose permeability is strictly positive throughout the domain. Marginal WD and JS are reported separately for p, K, and $\log K$; the log-space metrics are computed only from valid samples. These marginal scores do not measure whether each pressure field is paired with the correct permeability realization, which is why they are reported together with RE.

D Implementation and Reproducibility Details

D.1 Finite-transition grids

For a deployment budget N, PAFM is trained directly on the finite transitions executed by the sampler. Given a grid

$$0 = t_0 < t_1 < \cdots < t_N = 1, \qquad h_k = t_{k+1} - t_k, \tag{46}$$

the supervision target at step k is

$$\Delta_k = x_{t_{k+1}} - x_{t_k}. \tag{47}$$

Some configurations normalize the transition error by the interval length h_k during training. In increment coordinates this is equivalent to an h_k^{-2} loss weight; on a uniform grid it reduces to the constant N^2. No corresponding scaling is applied to the deployed state update. For a given budget, exactly the same node set $\{t_k\}_{k=0}^{N}$ is used for teacher construction, BASE training, TRD-cache construction, and inference.

Candidate $N = 20$ schedule family. We define a common family of candidate non-uniform schedules at the largest evaluated deployment budget, (N=20). For the first sixteen transitions,

$$h_k = a + (b - a)\frac{k}{15}, \qquad k = 0, \ldots, 15, \tag{48}$$

while the final four intervals are fixed to 0.05. Since $(a + b)/2 = 0.05$, the first sixteen intervals sum to 0.8, and every schedule therefore terminates exactly at $t_{20} = 1$. Table 5 lists the candidate family. The notation "3–7", for example, denotes a linear increase from 0.03 to 0.07 over the first sixteen transitions. Keeping the last four intervals at 0.05 avoids excessively coarse discretization close to the data endpoint, where the heat bridge restores finer spatial structure.

For reference, the (N=20) Strong 3–7 grid is

$$\begin{aligned}\mathcal{T}_{20} = \{&0,\ 0.030000,\ 0.062667,\ 0.098000,\ 0.136000,\\ &0.176667,\ 0.220000,\ 0.266000,\ 0.314667,\ 0.366000,\\ &0.420000,\ 0.476667,\ 0.536000,\ 0.598000,\ 0.662667,\\ &0.730000,\ 0.800000,\ 0.850000,\ 0.900000,\ 0.950000,\ 1\}.\end{aligned} \tag{49}$$

Table 6: Finite-transition grid used for each task and deployment budget.

Task	$N = 1$	$N = 5$	$N = 10$	$N = 20$
Kolmogorov Flow	Endpoints	–	–	Strong 3–7
Dynamic Stall	Endpoints	Uniform	Uniform	Uniform
Darcy Flow	Endpoints	Strong 3–7	Strong 3–7	Strong 3–7

Nested grids for smaller budgets. The lower-budget grids are obtained by nested subsampling of $\mathcal{T}_{20}$, rather than by independently redefining a new schedule. Specifically, $N = 10$ retains every second node of the $N = 20$ grid, $N = 5$ retains every fourth node, and $N = 1$ reduces to the two endpoints. This gives

$$\mathcal{T}_1 = \{0, 1\}, \tag{50}$$

$$\mathcal{T}_5 = \{0,\ 0.136000,\ 0.314667,\ 0.536000,\ 0.800000,\ 1\}, \tag{51}$$

$$\mathcal{T}_{10} = \{0,\ 0.062667,\ 0.136000,\ 0.220000,\ 0.314667,\ 0.420000,\ 0.536000,\ 0.662667,\ 0.800000,\ 0.900000,\ 1\}. \tag{52}$$

The same construction is applied to the other candidate (N=20) schedules when forming their lower-budget candidates. This preserves exact endpoints and nested node locations across deployment budgets while retaining the relative early/late density pattern of the parent schedule.

Task- and budget-specific selection. The schedules above define a common candidate family; the final grid is not selected on one benchmark and then transferred to the others. Instead, for each task and deployment budget, we perform the same training-side pilot selection over the corresponding candidate grids and retain the selected schedule for that configuration. Appendix G.2 reports the Kolmogorov Flow pilot in detail as a representative example of this selection procedure; analogous selection is performed independently for Dynamic Stall and Darcy Flow. No formal test conditions are used for grid selection. Table 6 summarizes the grids used in the reported experiments.

Because the grid determines both the teacher states and the finite increments being regressed, it is part of the training specification rather than an inference-time sampler choice. Separate deployment-aligned models are therefore trained for $N \in \{1, 5, 10, 20\}$.

D.2 BASE ARCHITECTURES AND OPTIMIZATION

All BASE models use a DiT backbone with 4×4 patchification, fixed two-dimensional sine–cosine positional embeddings, adaLN-Zero conditioning, GELU MLPs with expansion ratio four, and a zero-initialized output layer. The architectures differ across tasks because the spatial resolution, conditioning dimension, and required coupling range are different. Table 7 reports the selected $N = 20$ configurations. "Stored" includes the fixed positional tensor, whereas "optimized" counts parameters with `requires_grad=True`.

Full attention is used for Darcy deliberately. In Kolmogorov Flow and Dynamic Stall, the teacher constraints are diagonal in the Fourier basis or pointwise once geometry is supplied, so linear attention provides an efficient global token mixer. In Darcy Flow, a local change in $K = e^a$ changes the elliptic operator and may alter pressure throughout the domain. Full attention in BASE and a domain-scale bottleneck in TRD provide a stronger receptive field for this coefficient-dependent nonlocal response. The task-specific BASE optimization settings and heat-path conventions are summarized in Table 8.

D.3 TRD HEADS, TARGETS, AND OPTIMIZATION

Every TRD head predicts a direct correction in normalized state coordinates. If $b_k = f_{\theta,N}(y_k, t_k, c) = \bar{y}_{k+1} - y_k$ denotes the BASE increment, the deployed update is

$$y_{k+1} = y_k + b_k + g_{\phi,N}(y_k, b_k, t_k, c). \tag{53}$$

Any interval normalization used internally by BASE is undone before b_k is formed, so both terms added in Eq. 53 are state increments in the same coordinates. The task-specific TRD architectures, inputs, corrected variables, and parameter counts are summarized in Table 9.

Table 7: BASE architectures for the selected $N = 20$ configurations.

Kolmogorov Flow	
Input / output	$2/2$ channels at 128^2 resolution
Patch size / tokens	$4/1024$
Transformer width	256
Depth / heads	$8/8$
Attention	linear
Condition injection	learned MLP embedding of Re together with time
Parameters	10,076,192 stored; 9,814,048 optimized
Dynamic Stall	
Input / output	$6/6$ channels at 128^2 resolution
Patch size / tokens	$4/1024$
Transformer width	128
Depth / heads	$4/4$
Attention	linear
Condition injection	repeated four-dimensional operating condition together with time
Parameters	1,425,632 stored; 1,294,560 optimized
Darcy Flow	
Input / output	$2/2$ channels at 64^2 resolution
Patch size / tokens	$4/256$
Transformer width	256
Depth / heads	$8/8$
Attention	full
Condition injection	time only
Parameters	9,813,280 stored; 9,747,744 optimized

Table 8: BASE optimization and heat path for the $N = 20$ models.

Task	Batch	Optimizer	LR	Weight decay	EMA	Seed	Heat convention
Kolmogorov Flow	64	AdamW	10^{-4}	0	0.999	15	periodic spectral, $\kappa = 0.01$
Dynamic Stall	64	AdamW	10^{-4}	0	0.999	15	normalized periodic–reflect, $\kappa = 1$
Darcy Flow	64	AdamW	3×10^{-5}	0	0.999	15	normalized reflect–reflect, $\kappa = 1$

Note. Kolmogorov Flow uses the periodic spectral heat operator on both spatial directions. For Dynamic Stall, the normalized spectral heat operator uses a periodic boundary condition along the first spatial axis (H, axis -2) and a reflective boundary condition along the second (W, axis -1). Darcy Flow uses reflective boundary conditions along both spatial axes. The normalization affects only the heat-path construction and does not alter the physical variables used for evaluation.

All TRD output layers are zero initialized, so BASE+TRD is initially identical to the frozen BASE. The Kolmogorov head is global and periodic but linear, matching the structure of Leray closure. The Dynamic Stall head is pointwise because Equations 39a–39b are local once geometry and the four generating fields are known. Darcy instead requires a larger spatial receptive field, since the pressure correction depends nonlocally on the permeability over the entire domain. To limit both training and inference overhead, we nevertheless use a relatively compact TRD head that preserves this domain-scale coupling while providing sufficient recovery accuracy.

TRD state distributions. The recovery-state distribution is task-specific. A teacher-current entry starts from an exactly constructed state at node t_k. This source is used only for Darcy Flow, where the pressure corresponding to the current permeability field is obtained from the exact offline solver. A rollout-current entry instead starts from the deployment prior and executes a frozen sampler prefix to obtain the normalized current state y_k. Kolmogorov Flow and Dynamic Stall use rollout-current entries only. Their R0 prefixes use BASE alone, whereas R1 prefixes use BASE together with the

Table 9: Task-specific TRD implementations. TRD/BASE percentages compare the TRD parameter count with the full BASE parameter counts in Table 7.

Kolmogorov Flow	
Architecture	complex 2×2 Fourier multiplier applied independently to each of the 128×65 nonredundant Fourier modes
Head input	FFT of the normalized provisional endpoint $\bar{y}_{k+1} = y_k + b_k$; mode-wise learned complex 2×2 multiplier
Corrected field(s)	(u_x, u_y) increments
Parameters	66,560
TRD / BASE	0.661%
Dynamic Stall	
Architecture	deterministic pointwise feature construction followed by two independent 1×1 branches, each $4 \to 32 \to 32 \to 1$, with SiLU activations and zero-initialized output layers
Head input	External inputs: normalized state y_k (6), BASE increment b_k (6), and geometry tangent (t_x, t_y) (2). From $r := y_k + b_k = \bar{y}_{k+1}$, the pressure branch uses $(r_p, r_T, r_\rho, r_T r_\rho)$, and the wall-shear branch uses $(r_{\tau_w}, r_T, q, r_T q)$.
Corrected field(s)	p and τ_w increments
Parameters	2,498
TRD / BASE	0.175%
Darcy Flow	
Architecture	compact coordinate-aware U-Net with a $7 \to 16$ five-by-five stem, channel widths $16/24/32/48/56$, four downsampling stages to a 4×4 global bottleneck, skip connections, and a symmetric bilinear upsampling path
Head input	current normalized state y_k (2), BASE increment b_k (2), time t_k (1), and fixed spatial coordinates (2)
Corrected field(s)	p increment
Parameters	496,633
TRD / BASE	5.061%

selected frozen R0 head. Darcy Flow uses a balanced mixture of teacher-current and frozen-BASE rollout-current entries and does not employ the two-round R0/R1 procedure.

For every cache entry, regardless of its current-state source, the frozen BASE first produces a provisional next state. A task-specific offline physical closure is then applied to this provisional state to obtain its physically closed counterpart, and their difference defines the recovery target. For Dynamic Stall and Darcy Flow, the free components proposed by BASE are held fixed and only the coupled components are reconstructed. Prefix rollout, BASE evaluation, and target construction are all performed without gradient tracking. No future teacher node, test realization, physical residual, or closure target is provided as input to the deployed recovery head. The corresponding TRD cache construction and optimization settings are summarized in Table 10.

Which channels TRD corrects. Dynamic Stall TRD changes p and τ_w, but not (g_x, g_y, T, ρ); Darcy TRD changes p, but not $a = \log K$. Kolmogorov Flow is the linear-subspace case: both velocity channels are corrected jointly because they encode the same divergence-free constraint. After the initial prior has been prepared, each online update consists of one BASE call and one TRD call, with no call to the exact physical closure or any numerical optimization.

E Solver-free generation from physics-free Gaussian initialization

The main experiments use a physically completed prior as the starting state of the learned transport. The main results show that this physical consistency is not required at initialization. Here we keep

Table 10: TRD cache construction, optimization, and selection for the selected $N = 20$ configurations. The BASE backbone is frozen throughout TRD training.

Kolmogorov Flow	
Prefix distribution	R0: BASE rollout; R1: BASE + R0 rollout
Cache	10,240 transitions
Batch size	32
Training schedule	200 epochs (R0) + 100 epochs (R1)
Learning rate	2×10^{-4} (R0), 1×10^{-4} (R1)
EMA	0.995
Dynamic Stall	
Prefix distribution	R0: BASE rollout; R1: BASE + R0 rollout
Cache	4,096 transitions per round, with $3{,}584/512$ fit/holdout split
Batch size	8 with four-step gradient accumulation (effective batch size 32)
Training schedule	20 epochs (R0) + 40 epochs (R1)
Learning rate	3×10^{-3} (R0), 1×10^{-4} (R1)
EMA	0.990
Darcy Flow	
Prefix distribution	balanced mixture of exact-teacher and deployed-rollout states
Cache	$4{,}096/512$ train/validation transitions; $8{,}192/1{,}024$ exact solves
Batch size	64
Training schedule	selection up to 10,000 epochs
Learning rate	2×10^{-4}
EMA	0.995

the trained BASE and TRD checkpoints fixed, change only the initial state, and report the resulting metrics.

This setting is practically important because it separates the physics used to teach the transport from the information required to execute it. At deployment, a new sample can be generated from independent Gaussian noise without first constructing a PDE-consistent state or supplying a sample-specific physical solution. The resulting pipeline therefore behaves like a conventional black-box generator: once training is complete, generation requires only random sampling and forward evaluations of the learned BASE and TRD maps. The governing relations remain encoded in the learned transport through the physically aligned teacher states used during training, rather than being re-evaluated for each new sample.

The Gaussian variables need not coincide with the final physical channels. The initial state uses each task's generating-coordinate convention; where needed, a fixed coordinate lift places those variables in the normalized network state before the learned transitions begin. The experiment therefore tests whether the initially coupled fields must already satisfy the selected relations, rather than whether generating coordinates can be dispensed with.

The results below therefore isolate a deployment property rather than a new training configuration. They show how much of the physically aligned source construction remains necessary after the transport has already been learned.

Kolmogorov Flow. Kolmogorov Flow is a special case because its generating coordinates are an unconstrained two-channel velocity field in normalized coordinates and incompressibility is enforced by the linear Leray projector. The formal network prior is $y_0 = P_{\mathrm{L}}\xi$, with $\xi \sim \mathcal{N}(0, I)$. Table 11 compares this prior with the unprojected iid source $y_0 = \xi$. Removing the projection causes only a small degradation in the reported metrics; all generated fields remain finite. This shows that the frozen learned maps remain robust even when the source itself is not made divergence-free.

Table 11: Kolmogorov Flow results under Gaussian deployment initialization using the same trained BASE and TRD checkpoint.

Initialization	RE $\cdot 10^1$	WD $\cdot 10^1$	JS $\cdot 10^2$	MMSE $\cdot 10^4$	SMSE $\cdot 10^4$	Valid
Unprojected iid, $y_0 = \xi$	0.2440	1.0689	6.6953	2.461	2.317	1.000
Projected iid, $y_0 = P_{\mathrm{L}}\xi$ (formal)	**0.1741**	**1.0642**	**6.6876**	**2.407**	**2.170**	1.000

Note. The Kolmogorov free variables contain both unconstrained velocity channels. The formal network prior is exactly $y_0 = P_{\mathrm{L}}\xi$, where ξ is normalized iid Gaussian noise. Valid here records finite generated fields; it does not assert exact incompressibility. The alternative is the unprojected source $y_0 = \xi$.

Dynamic Stall. Dynamic Stall shows that the initial coupled channels need not already satisfy the two constitutive relations, while the placement of randomness matters. As shown in Table 12, replacing the formal completed prior by iid Gaussian samples in its four free coordinates, with the normalized p and τ_w channels initialized to zero, leaves the final physical residual essentially unchanged. Its Ideal, Wall, and Total residuals are even marginally lower than those of the formal prior, while WD, JS, MMSE, and SMSE remain close to the formal setting.

By contrast, lifting iid Gaussian noise into all six channels still allows the same frozen BASE+TRD model to recover nearly identical physical residuals, but substantially worsens WD, JS, and MMSE. Thus, physical feasibility of the initial complete state is not itself required at deployment. What matters is that stochasticity is introduced through the free variables from which the physically completed training states are constructed.

Table 12: Dynamic Stall results under alternative deployment initializations using the same trained BASE and TRD checkpoint.

Initialization	RE $\cdot 10^6$	WD $\cdot 10^4$	JS $\cdot 10^4$	MMSE $\cdot 10^5$	SMSE $\cdot 10^5$
Lifted six-channel iid	0.007617	16.1753	9.9993	5.7793	**1.3167**
Lifted four-channel iid, $y_{0,p} = y_{0,\tau_w} = 0$	**0.007402**	5.7990	4.3636	1.5894	1.6094
Physically completed prior	0.007462	**5.5943**	**3.8970**	**1.4428**	1.4513

Darcy Flow. Darcy Flow provides the strongest practical consequence because initialization without physical completion removes the only numerical PDE solve otherwise required before the learned rollout. Sampling $a = \log K$ directly from its Gaussian reference and setting the normalized pressure channel to zero preserves full validity and a low final residual, while increasing the permeability-distribution error relative to the completed physical prior. Randomizing both p and $\log K$ further improves the pressure-side metrics and slightly lowers RE, but leads to a larger degradation of the K marginal. These results again show that the frozen learned maps are highly tolerant to inaccurate dependent-variable initialization, whereas the placement of stochasticity has a stronger effect on the final generative distribution.

Table 13: Darcy Flow results under alternative deployment initializations using the same trained BASE and TRD checkpoint.

Initialization	RE	WD_p	WD_K	JS_p	JS_K	Valid
$p, \log K$ iid Gaussian	**0.0339**	**0.001698**	0.300047	**0.000459**	0.018474	1.0000
$\log K$ iid Gaussian, $p = 0$	0.0354	0.002504	0.201088	0.000605	0.011372	1.0000
Physically completed prior	0.0399	0.002474	**0.131098**	0.000555	**0.004213**	1.0000

Across the three benchmarks, the same deployment behavior emerges. Physical completion is important for constructing the training transport, but the initial state need not be physically completed once this structure has been learned by the finite maps. PAFM can therefore start from simple Gaussian randomness and generate complete physical fields without solving the governing relations during deployment. However, the Dynamic Stall and Darcy results also show that the Gaussian source should still be introduced through the free variables used during training; arbitrary full-field initialization can degrade distributional fidelity. PAFM therefore separates the physical information needed to learn the transport from the physical computation needed to execute it.

Table 14: Comparison between PAFM and Terminal Completion (TC) at deployment. PAFM generates the complete field using learned maps, whereas TC generates only the free variables and performs one physical completion after sampling.

Task	Method	RE	WD	JS	IT
Kolmogorov	PAFM	1.74×10^{-2}	1.06×10^{-1}	6.69×10^{-2}	22.32 ms
	TC	1.95×10^{-10}	5.12×10^{-2}	2.27×10^{-2}	23.80 ms
Darcy	PAFM	0.0399	0.1311	0.0042	5.60 ms
	TC	0.0012	0.0805	0.0086	19.60 ms

For Darcy Flow, WD and JS are computed on the free permeability variable K only.

F Generation in Free Physical Variables with Terminal Completion

PAFM generates the complete physical field during the learned rollout. An alternative is to generate only the free variables and recover the remaining coupled quantities once, after generation has finished. We refer to this strategy as Terminal Completion (TC). In the notation of §3,

$$\hat{z} \sim q_\theta(z \mid c), \qquad \hat{x} = D_c(\hat{z}), \tag{54}$$

where the generative model produces only the free variables and the final physical field is obtained by one explicit physical completion.

TC enforces the selected physical relations to the accuracy of the final physical completion. It also reduces the learning problem to the variables that carry the stochastic degrees of freedom. However, the physical model remains part of deployment. Generating a sample requires not only the learned generator, but also the physical information and computation needed to recover the coupled variables.

This distinction is minor when physical completion is inexpensive. For Kolmogorov Flow, for example, the final velocity field can be recovered through a spectral divergence-free construction at very small additional cost. TC can therefore enforce incompressibility almost exactly while retaining a practical sampling speed. However, this advantage depends on the completion operator remaining available during inference.

The difference becomes more important for globally coupled systems. In Darcy Flow, generating permeability alone is insufficient to produce the complete state: each new permeability realization requires solving

$$-\nabla \cdot (\hat{K} \nabla \hat{p}) = f \tag{55}$$

to obtain the corresponding pressure field. TC therefore introduces one new coefficient-dependent PDE solve for every generated sample. In contrast, PAFM uses these solves only to construct training states and offline recovery targets, and generates both permeability and pressure directly with the learned BASE and TRD maps. Table 14 summarizes the two deployment modes.

The two approaches therefore represent different deployment regimes rather than a simple accuracy hierarchy. TC can provide very strong physical consistency by enforcing the governing relations explicitly at the final sample: it reduces RE on both benchmarks and also improves several distributional metrics. When completion is inexpensive, as in Kolmogorov Flow, this comes with only a small increase in inference time. For Darcy Flow, however, the terminal PDE solve increases the sampling cost from 5.60 to 19.60 ms per sample, while still requiring the physical solver and its associated information at deployment. PAFM instead uses physical completion only to construct training targets and recovery supervision, and executes generation entirely through the learned BASE and TRD networks. It therefore trades explicit terminal enforcement for a solver-free learned rollout, which becomes increasingly attractive when physical completion is expensive or difficult to access during deployment.

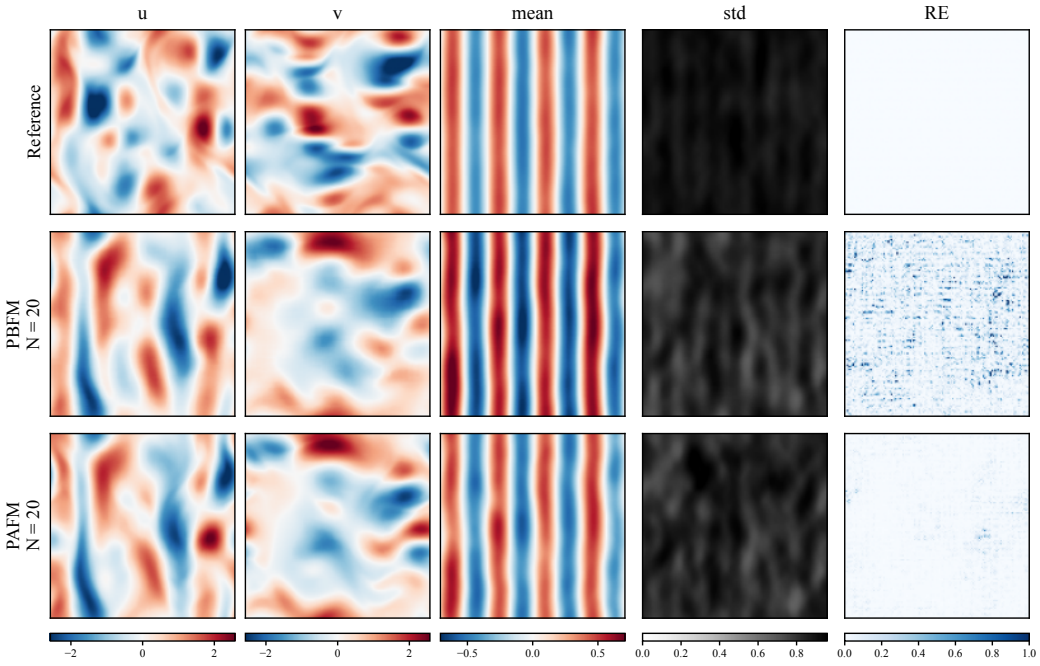


Figure 4: Kolmogorov qualitative comparison. Reference, PBFM, and PAFM fields at the same condition are shown together with conditional mean, standard-deviation, and divergence-error views. Color limits are shared across methods for each displayed quantity.

G Supplementary Kolmogorov Flow Results

G.1 Qualitative samples and recovery behavior

Figure 4 compares generated velocity fields, conditional means and standard deviations, and divergence errors. PAFM preserves coherent vortical structure and reference ensemble variability; the effect of recovery is most visible in the divergence map rather than as artificial smoothing of the velocity channels. This agrees with the formal $N = 20$ result: TRD reduces RE from 0.11292 to 0.01741 (84.58%), while WD, JS, MMSE, and SMSE change by at most 1.5% relative to the frozen BASE.

G.2 Non-uniform-grid study

We compared the four candidate grids for $2{,}000$ BASE updates from a common initialization, using training conditions only. To avoid selecting a schedule that improves one metric at the expense of a severe regression in another, we used the geometric mean of metric ratios

$$Q = \left[\prod_{m \in \{\mathrm{RE},\mathrm{WD},\mathrm{JS},\mathrm{MMSE},\mathrm{SMSE}\}} \frac{m_{\mathrm{schedule}}}{m_{\mathrm{uniform}}} \right]^{1/5}, \tag{56}$$

where lower values are better. Table 15 shows that Strong 3–7 attains the lowest joint ratio, narrowly ahead of Mild 4–6. Aggressive 2–8 gives the lowest RE and JS but regresses WD, MMSE, and SMSE enough to lose under the joint criterion. We therefore selected Strong 3–7 for the subsequent non-uniform experiment.

Table 15: Kolmogorov common-initialization schedule pilot after 2k updates. All metrics and the joint ratio are lower-is-better.

Grid	RE	WD	JS	MMSE·10^4	SMSE·10^4	Joint ratio Q
Uniform 5–5	150.7368	0.174907	0.127703	9.5955	4.1322	1.00000
Mild 4–6	157.3750	**0.161062**	0.126165	**7.3206**	3.5536	0.90975
Strong 3–7	151.6692	0.161591	0.123363	7.5316	**3.4242**	**0.89804**
Aggressive 2–8	**145.7484**	0.176373	**0.123128**	9.0593	5.0938	1.01814

G.3 Multiscale energy statistics

The preceding results evaluate physical consistency and distributional agreement in state space, but do not directly measure how generated kinetic energy is distributed across spatial scales. This is particularly relevant for Kolmogorov Flow, where two models with similar WD or low-order moments may still differ substantially in spectral organization or small-scale vortical activity. We therefore supplement the standard metrics with scale-aware diagnostics computed on the de-normalized physical velocity fields.

For each Reynolds number, we compute the mean kinetic energy

$$K = \frac{1}{2}\left\langle u_x^2 + u_y^2 \right\rangle, \tag{57}$$

the shell-integrated radial Fourier energy spectrum

$$E(k) = \frac{1}{2}\sum_{\boldsymbol{\kappa}\in\mathcal{B}_k}\left(|\widehat{u}_x(\boldsymbol{\kappa})|^2 + |\widehat{u}_y(\boldsymbol{\kappa})|^2\right), \tag{58}$$

where $\boldsymbol{\kappa} = (\kappa_x, \kappa_y)$ is a discrete Fourier wavevector and $\mathcal{B}_k$ denotes the set of wavevectors whose radial wavenumbers fall in the frequency bin centered at k, i.e.,

$$\mathcal{B}_k = \left\{\boldsymbol{\kappa} : k - \frac{\Delta k}{2} \le \|\boldsymbol{\kappa}\|_2 < k + \frac{\Delta k}{2}\right\},$$

with Δk equal to one fundamental wavenumber. Under the adopted orthonormal FFT normalization, summing $E(k)$ over all radial bins recovers the total kinetic energy. We further compute the enstrophy

$$\Omega = \frac{1}{2}\left\langle (\partial_x u_y - \partial_y u_x)^2 \right\rangle. \tag{59}$$

Spec-L1 measures the total discrepancy between generated and reference energy spectra, while Spec-W1 compares their normalized spectral distributions along the wavenumber axis. We additionally report relative kinetic-energy and enstrophy errors, the error in the reference-defined high-frequency energy fraction, and the relative error of the spectral centroid. All quantities are computed independently for each of the 16 unseen Reynolds numbers and then averaged across conditions.

To isolate the contribution of the teacher path itself, we evaluate the *BASE-only* models without TRD. PAFM (Heat) uses the formal dissipative bridge, while PAFM (Linear) is a strictly controlled variant obtained by replacing H_τ with the identity. The prior, Leray projection, architecture, training budget, physical supervision, EMA, and Euler20 rollout are otherwise unchanged. Thus, any difference between Heat and Linear reflects the path used to train the discrete transport. PBFM and FM-OT are evaluated under the same test conditions, with no post-processing.

Figure 5 provides a distribution-level view complementary to the scalar errors below. All generated models slightly shift the energy statistics relative to the reference ensemble, but PAFM-Heat consistently reduces this discrepancy and more closely preserves the distribution of small-scale vortical activity.

The scale-aware metrics reveal a clearer advantage of the heat path than the standard distributional metrics alone. Heat and Linear have very similar WD (0.1063 versus 0.1088) and nearly identical JS (0.066869 versus 0.066884), while MMSE differs by less than one percent. In contrast, Table 16

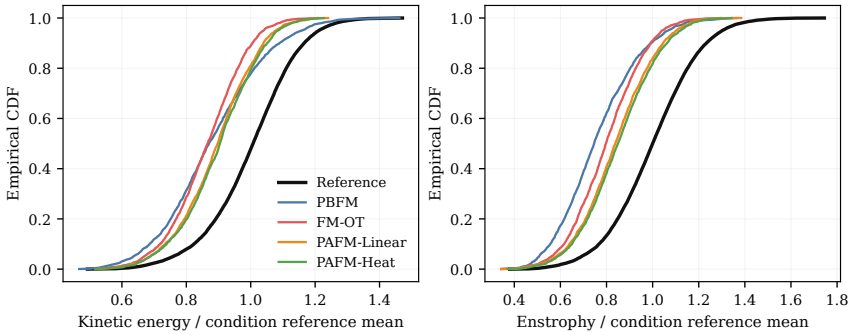


Figure 5: Energy and vortical statistics for Kolmogorov Flow. Empirical CDFs of per-sample kinetic energy (left) and enstrophy (right), each normalized by the corresponding condition-wise reference mean.

Table 16: Scale-aware statistics on 16 unseen Reynolds numbers. All metrics are computed on de-normalized velocity fields and averaged across conditions. Lower is better; best and second-best values are shown in bold and underline.

Method	Spec-L1	Spec-W1	KE Err.	Enstrophy Err.	HF Err.	Centroid Err.
FM-OT	0.138452	1.205×10^{-3}	0.136868	0.199840	0.017406	0.036683
PBFM	0.162886	1.981×10^{-3}	0.118231	0.237427	**0.008067**	0.061332
PAFM BASE (Linear)	<u>0.109585</u>	<u>8.218×10^{-4}</u>	<u>0.107879</u>	<u>0.159896</u>	0.009928	<u>0.026014</u>
PAFM BASE (Heat)	**0.102715**	**7.305×10^{-4}**	**0.101052**	**0.148603**	<u>0.008241</u>	**0.022982**

consistently favors the heat bridge: relative to Linear, it reduces Spec-L1 by 6.27%, Spec-W1 by 11.11%, kinetic-energy error by 6.33%, enstrophy error by 7.06%, high-frequency error by 16.99%, and spectral-centroid error by 11.66%. The benefit of heat is therefore expressed primarily in the organization of energy across spatial scales rather than through a simple change in field amplitude. Importantly, both variants are evaluated at the BASE level, so this separation arises before recovery is introduced and directly reflects the learned teacher-path transport. We additionally evaluated the corresponding BASE+TRD models; all reported spectral and energy metrics changed by less than 5%, while the relative advantage of the heat path was preserved. This stability is consistent with the role of TRD in PAFM: recovery primarily removes the constraint-violating component of the provisional state rather than relearning the distribution encoded by the physics-aligned teacher sequence. In Kolmogorov Flow, this amounts to correcting the non-solenoidal component toward the Leray-consistent subspace while largely preserving the solenoidal content that carries the learned multiscale energy statistics. The spectral advantage induced by the heat path is therefore retained naturally after recovery.

The comparison with the baselines leads to the same conclusion. PAFM BASE (Heat) reduces Spec-W1 by 63.1% relative to PBFM and by 39.4% relative to FM-OT, while lowering enstrophy error by 37.4% and 25.6%, respectively. PBFM attains a slightly smaller high-frequency fraction error than PAFM, but its overall spectrum, spectral centroid, and enstrophy remain substantially less accurate. This distinction shows that matching a single high-frequency energy ratio, or achieving a low incompressibility residual, does not by itself imply a correct multiscale flow distribution.

Together with the standard metrics, these BASE-only results show that the advantage of the PAFM transport already extends beyond divergence reduction before recovery is applied. The heat bridge yields velocity fields with more faithful energy allocation and small-scale vortical statistics than the controlled linear path, demonstrating that the physically structured teacher path itself contributes to the quality of the learned generative transport.

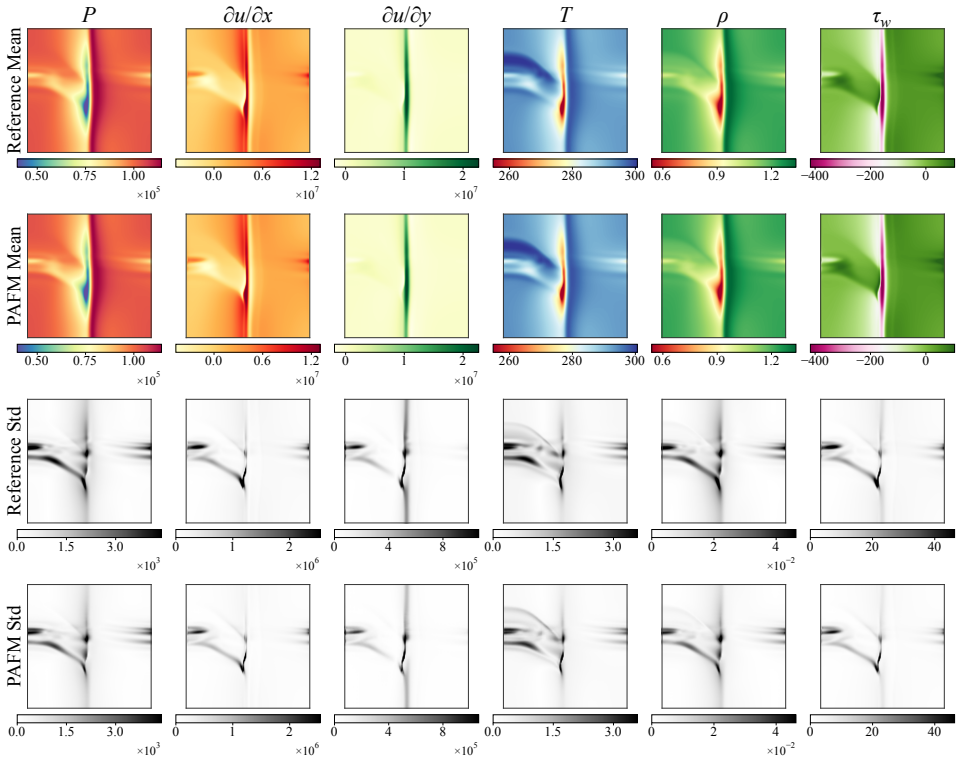


Figure 6: Dynamic Stall field statistics. Rows show reference mean, PAFM mean, reference standard deviation, and PAFM standard deviation across the six channels. Corresponding reference and PAFM panels share color scales.

H Supplementary Dynamic Stall Results

H.1 Qualitative field statistics

Figure 6 compares the reference and generated mean and standard-deviation fields. Across the six channels, PAFM reproduces the dominant transition structures in the mean fields. The reference standard deviations are concentrated around narrow bands and curved transition regions. PAFM captures these main localization patterns, while differences in their local intensity and extent remain. This distinction helps interpret Table 2: accurate mean fields can coexist with less accurate standard-deviation fields. Physical consistency of individual generated states is evaluated separately by the Ideal and Wall residuals.

H.2 Heat versus linear teacher paths

The Heat and Linear models use the same initialization, architecture, training stream, seed, optimizer, Strong 3-7 grid, physical weight, and 337k-update budget. The only intervention is replacing H_s by the identity. Evaluation uses the training-file holdout conditions 112–127 and does not access the formal test set. Table 17 reports the selected aggregate metrics. Heat improves all five quality metrics, with the largest gains in WD (14.58%) and MMSE (18.50%). The measured inference times differ by 4.62%; both samplers omit the heat operator at inference, so this difference does not reflect an additional online heat computation.

Figure 7 examines the scale ordering produced by Heat. For pressure frequencies $|k| \geq 61$, the Heat/Linear power ratios at $t = 0.1, 0.3, 0.5, 0.7, 0.9$ are $0.219, 0.310, 0.437, 0.612, 0.851$. Aggregating the high-frequency energy over all four free-variable channels and normalizing by the shared endpoint gives $0.299, 0.389, 0.507, 0.663, 0.871$ at the same times and one at $t = 1$; Linear remains at one because it scales all frequencies equally. Heat therefore does not shorten the Euclidean endpoint

Table 17: Strict Dynamic Stall bridge-only ablation after 337k updates on the training-condition holdout. Residuals are multiplied by 10^6, WD/JS by 10^4, and MMSE/SMSE by 10^5. All metrics are lower-is-better.

Path	RE	WD	JS	MMSE	SMSE	ms/sample
Heat	**8.2744**	**8.6729**	**6.7451**	**1.7287**	**1.3272**	5.646
Linear	8.3645	10.1537	7.2331	2.1210	1.3762	**5.397**

displacement. It reorders teacher information from coarse to fine and delays derivative-sensitive high-frequency content until later transitions.

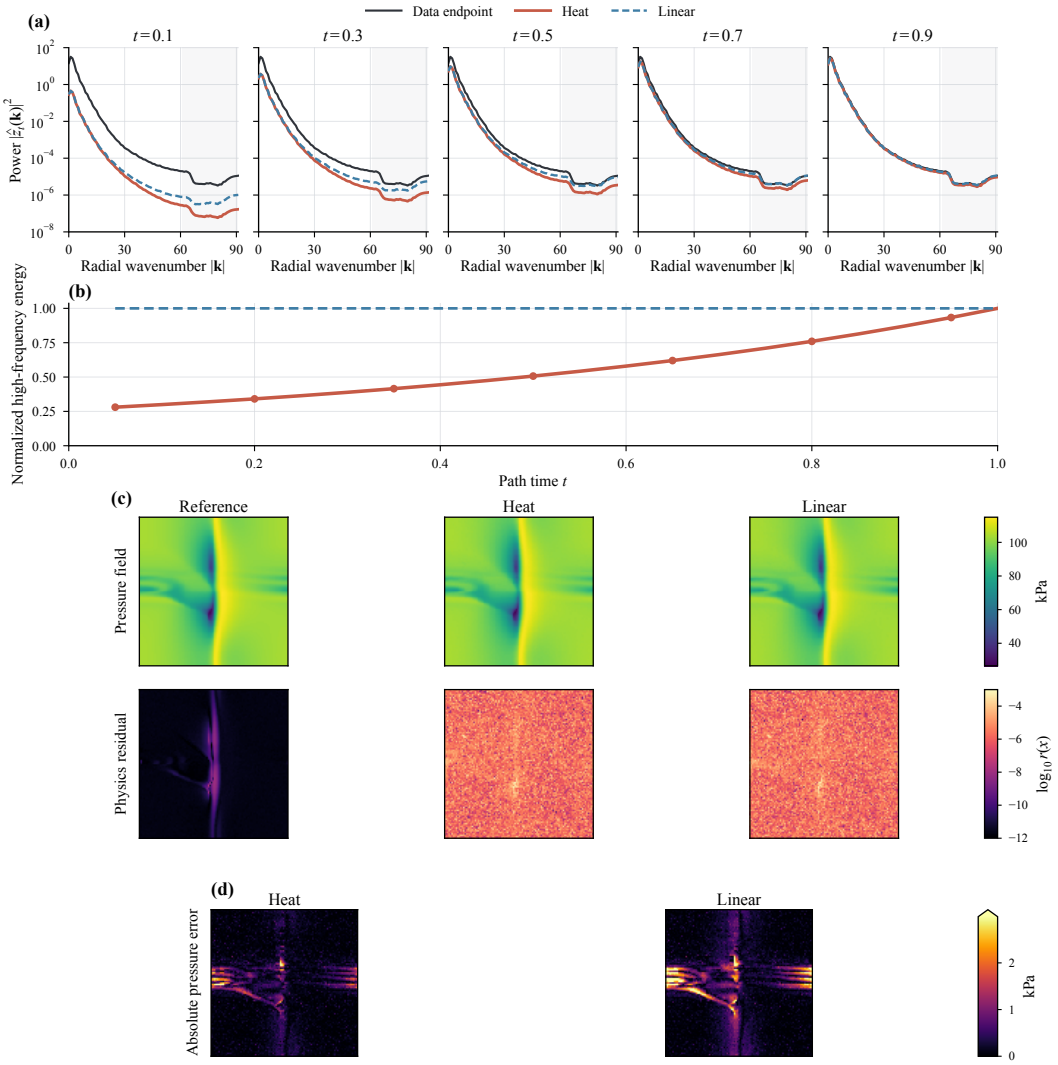


Figure 7: Dynamic Stall Heat-versus-Linear mechanism. (a) Radial pressure spectra along the two teacher paths. (b) Normalized high-frequency energy over the four free-variable channels. (c) Reference, Heat, and Linear pressure and physical-residual maps. (d) Absolute pressure error. The displayed field is the preselected largest-gain holdout case (condition 113, generated sample 11, reference 3); aggregate evidence is reported in Table 17.

H.3 SAME-SOURCE RECOVERY ABLATION

The final compact-recovery experiment freezes the $N = 20$ BASE at update 10,050 and constructs every recovery target from that same backbone. A legacy head trained on a different BASE is excluded from initialization and model selection. Table 18 reports formal results on the 16 test conditions with 128 generated samples per condition. The deployed compact head reduces Total from 8.0776 to 0.0075 in the table's 10^6 scale, a 99.91% reduction with only 2,498 parameters. It also improves WD and JS relative to BASE, with only a 0.15% increase in MMSE and a small improvement in SMSE. The larger same-source head further reduces WD and JS, but more than doubles latency and uses $96.0\times$ as many parameters. It is included to expose the capacity–latency trade-off rather than as the deployed design.

Table 18: Dynamic Stall $N = 20$ recovery ablation with the same frozen BASE. Residuals are multiplied by 10^6, WD/JS by 10^4, and MMSE/SMSE by 10^5. Compact is the deployed configuration; the medium and large heads are retained as architecture and capacity ablations.

Recovery	Params.	Ideal	Wall	Total	WD	JS	MMSE	SMSE	ms/sample
None	0	7.1145	0.9631	8.0776	5.6425	4.3254	1.4407	1.4569	**4.8266**
Compact	2,498	**0.0022**	**0.0053**	**0.0075**	5.5943	3.8970	1.4428	1.4513	6.2478
Medium	5,378	0.1394	0.7738	0.9131	5.6697	4.0357	1.4468	**1.4504**	6.9151
Large	239,874	0.0521	0.2313	0.2835	**5.5727**	**3.8492**	**1.4365**	1.4511	13.146

I SUPPLEMENTARY DARCY FLOW RESULTS

I.1 QUALITATIVE PRESSURE–PERMEABILITY CONSISTENCY

Figure 8 compares methods at $N = 10$. A physically meaningful sample requires more than a plausible K texture or a matching pressure marginal: the pressure must be induced by the *same* permeability realization under the fixed forcing, boundary condition, and gauge. The shared residual scale prevents visually smooth samples with a large coefficient–solution mismatch from appearing artificially physical.

I.2 GLOBAL PRESSURE RECOVERY IN DARCY FLOW

Darcy provides a particularly stringent recovery problem because pressure is globally coupled to the generated permeability through an elliptic equation. We first examine why this structure requires nonlocal recovery, and then study how the frozen BASE maps and the learned pressure-only TRD heads interact across deployment budgets.

Why Darcy recovery requires global context. A local perturbation of the permeability can induce a pressure response over the entire domain. Figure 9 illustrates this effect: perturbing an 8×8 region of $\log K$ changes the re-solved pressure globally, with 87.1% of the squared pressure-change magnitude lying outside the perturbed region. Hence the physical pressure correction $\delta p^\star = \mathcal{S}(K) - \bar{p}$, with $\bar{p}$ in physical units, cannot in general be represented by a pointwise relation. Estimating the appropriate correction at one location may require information about the generated permeability elsewhere in the domain.

We therefore use a spatial TRD head with a domain-scale receptive field that updates only pressure while leaving $a = \log K$ unchanged, matching the nonlocal coefficient–solution coupling of Darcy Flow.

Recovery across deployment budgets. We next evaluate each frozen BASE map with and without its associated TRD head. All comparisons use paired initial states from the same locked $1{,}024$-sample physical-prior bank and the selected $N \in \{1, 5, 10, 20\}$ checkpoints. As shown in Table 19, TRD reduces endpoint RE by 98.73%–99.74% while preserving full validity at every budget. At the same time, BASE-only endpoint RE rises markedly with N, from 2.3732 to 15.5038. However, this does not contradict the short-step result in Appendix B, where $\mathrm{CAG}_{t,h}(x) \le \epsilon_{\mathrm{dec}} + O(h^2)$ characterizes

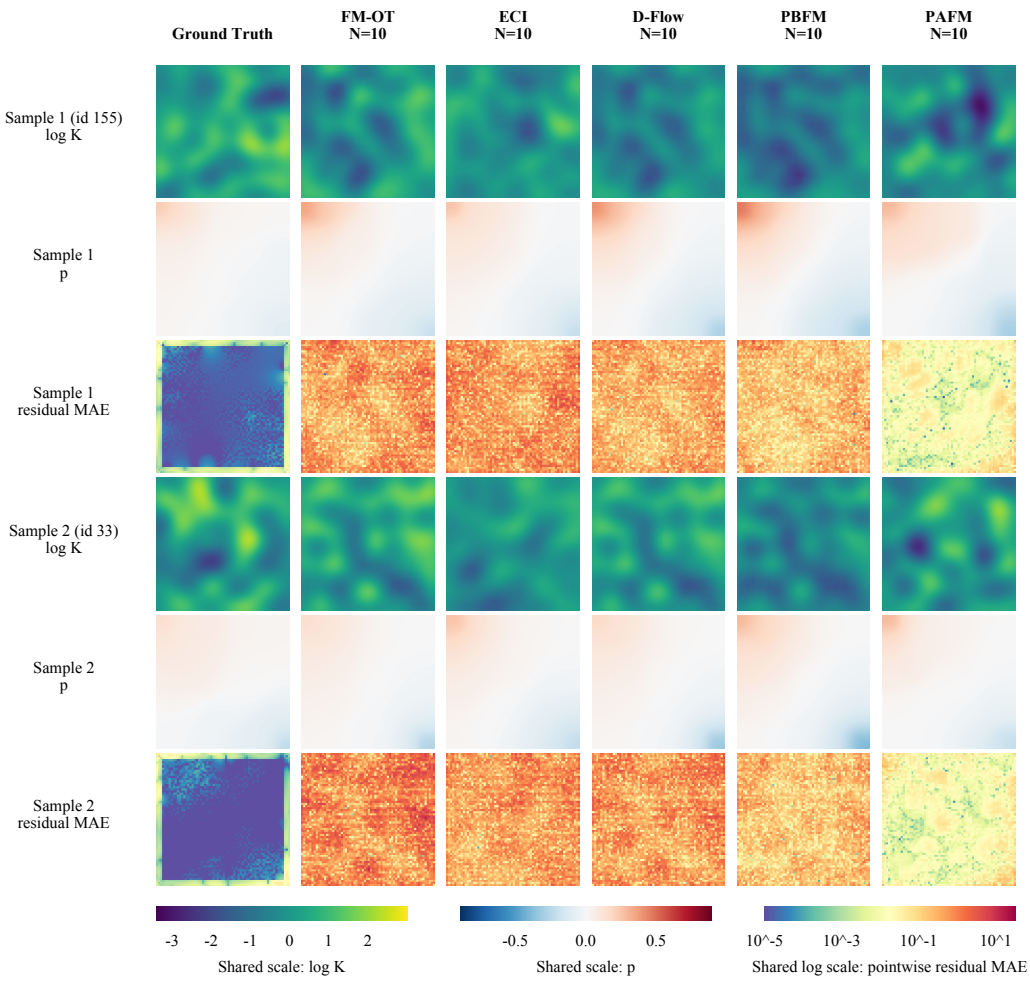


Figure 8: Darcy qualitative comparison at $N = 10$. For two fixed sample IDs, rows show $\log K$, pressure p, and the strong-form residual magnitude for the ground truth and five generators. Color limits are shared across methods within each physical quantity.

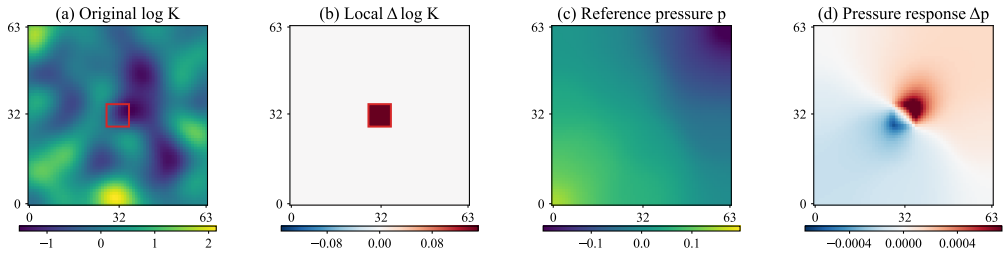


Figure 9: Nonlocal pressure response in Darcy Flow. (a) Original $\log K$ field. (b) A localized 8×8 perturbation with $\Delta \log K = +0.15$. (c) The corresponding pressure solution $p = \mathcal{S}(K)$. (d) The induced pressure change after re-solving the Darcy equation.

the local averaging-induced error of a teacher transition, rather than the error accumulated by a learned rollout. To distinguish this nominal short-step effect from deployment-induced off-path drift, we therefore measure the exact pressure correction required under two state distributions: teacher-current states, which remain on the physically completed teacher sequence and probe the local transition difficulty, and BASE-rollout-current states, which include errors accumulated through preceding learned maps. Comparing the two reveals whether the local benefit of shorter transitions persists once the model is deployed autoregressively.

We quantify this difference by the exact physical-closure pressure correction required after a provisional BASE update. Let $\bar{y}_p$ be the normalized provisional pressure and $\bar{K}$ the proposed physical permeability. With $y_p^\star = \mathcal{A}_p(\mathcal{S}(\bar{K}))$, where $\mathcal{A}_p$ is the fixed pressure-channel normalization, we use

$$\mathrm{RMS}(\delta p^\star) = \left(\frac{1}{G} \sum_{i=1}^{G} (y_{p,i}^\star - \bar{y}_{p,i})^2 \right)^{1/2}, \qquad G = 64^2. \tag{60}$$

For the multi-step models, this correction decreases on held-out teacher-current states from 0.0371 at $N = 5$ to 0.0352 at $N = 10$ and 0.0285 at $N = 20$, consistent with the local short-step behavior characterized in Appendix B. On BASE-rollout-current states, however, the values remain nearly unchanged at 0.0383, 0.0382, and 0.0389. Hence, the reduced local correction associated with finer teacher transitions does not carry over to the states actually visited by the learned rollout, where accumulated transition error shifts the sampler away from the teacher sequence. The $N = 1$ model is a direct endpoint map and is therefore excluded from this multi-step comparison.

Table 19: Paired Darcy recovery diagnostic. Endpoint RE is evaluated on the same $1{,}024$ initial states with and without TRD. Correction RMS measures the physical-closure pressure correction in normalized coordinates on held-out teacher-current and BASE-rollout-current states. All variants have validity 1.0000.

	Endpoint RE		Correction RMS	
N	BASE	BASE+TRD	Teacher state	Rollout state
1	2.3732	0.03014	0.0129	0.0129
5	8.6923	0.03059	0.0371	0.0383
10	11.2319	0.03201	0.0352	0.0382
20	15.5038	0.03987	0.0285	0.0389

Stepwise recovery at $N = 20$. Figure 10 follows 128 fixed physical-prior samples through the $N = 20$ rollout. Each BASE transition reintroduces substantial coefficient–pressure inconsistency, while TRD reduces the residual before the corrected state is passed to the next map. Averaged across all 20 transitions, the mean physical residual decreases from 1.488 for the provisional BASE state to 0.209 after learned recovery, corresponding to an 85.9% reduction. At the final transition, RE decreases from 1.8599 to 0.04128. An exact pressure closure, evaluated only as an offline diagnostic while retaining the same proposed permeability, reaches 0.001215.

These results show that TRD consistently removes most of the physical inconsistency introduced by BASE, while remaining an approximate learned recovery rather than an exact replacement for the elliptic pressure solve.

I.3 Sensitivity to the physical-loss weight

The sample-wise compatibility of the transition and physical objectives does not imply that the physical-loss weight is irrelevant for a finite-capacity learned map. As discussed in Appendix B.3, population regression may favor an averaged prediction under transition ambiguity, while approximation error can further move the learned map away from the completed teacher state. We therefore examine the sensitivity of BASE to λ_{phys}.

We use the Darcy $N = 20$ configuration and vary only the physical-loss weight around its default value $\lambda_0 = 5.8935 \times 10^{-8}$. All other training, teacher-path, initialization, and evaluation settings are

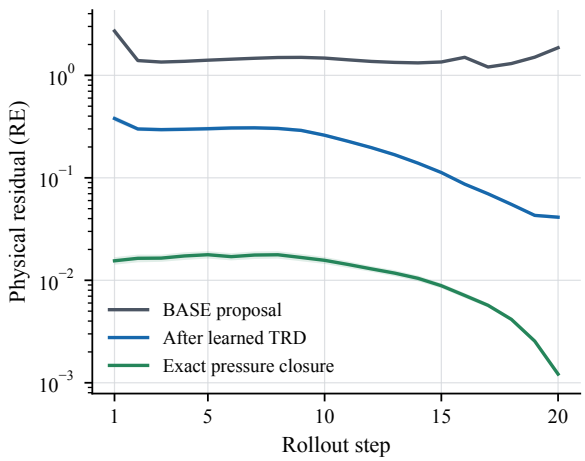


Figure 10: Stepwise physical recovery for the Darcy $N = 20$ sampler on 128 fixed physical-prior samples. Curves show the sample-mean physical residual (RE) of each provisional BASE proposal, the state after learned TRD, and an offline exact pressure closure retaining the same proposed permeability.

kept unchanged. To isolate the BASE objective itself, this sweep is evaluated *before TRD*; attaching a recovery head would otherwise mask part of the dependence on λ_{phys}.

Table 20: Darcy $N = 20$ BASE-only sensitivity to the physical-loss weight. $\lambda_0 = 5.8935 \times 10^{-8}$ is the default setting. All models use the same physical teacher path and differ only in λ_{phys}. Lower is better except Valid.

λ/λ_0	λ_{phys}	RE	WD_p	WD_K	$\text{WD}_{\log K}$	JS_p	JS_K	$\text{JS}_{\log K}$	Valid	IT [ms]
0	0	24.92463	0.002930	0.131263	0.080487	0.000676	0.004261	0.004433	1.0000	4.4194
0.1	5.8935×10^{-9}	22.17045	**0.002918**	**0.130639**	**0.079708**	**0.000654**	**0.004189**	**0.004357**	1.0000	4.4144
1	5.8935×10^{-8}	15.50376	0.003106	0.131538	0.080243	0.000734	0.004220	0.004394	1.0000	4.4252
10	5.8935×10^{-7}	**10.14121**	0.003077	0.132720	0.080987	0.000731	0.004300	0.004470	1.0000	4.4162

Increasing λ_{phys} produces a clear reduction in BASE physical error: RE decreases from 24.92 without the explicit physical term to 15.50 at the default weight and 10.14 at $10\lambda_0$. At the same time, the distributional metrics vary much less over the sweep. The $0.1\lambda_0$ model gives the lowest WD and JS values, whereas stronger physical weighting slightly worsens several permeability metrics. The default λ_0 therefore represents an intermediate operating point rather than an individually optimal setting for either objective.

These results clarify the role of the physical term in PAFM. Physical completion makes transition matching and physical supervision compatible at each paired teacher target, but it does not eliminate finite-capacity or population-level averaging effects. Consequently, explicit physical supervision can still move the learned BASE prediction toward lower residual error, while sufficiently strong weighting can introduce a modest physics–distribution trade-off. Importantly, this sensitivity is already visible before recovery; the subsequent TRD stage addresses rollout-state physical inconsistency rather than defining the trade-off observed here.

I.4 Ablation of the physical teacher path

Darcy Flow provides a stringent ablation of the role of the physical teacher path because pressure is globally coupled to permeability through the elliptic PDE. To test the combined contribution of the physically completed source and teacher sequence, we compare the standard PAFM construction with an ambient-teacher control in which both source and intermediate teacher states are formed directly in the normalized $(p, \log K)$ space. Specifically, the no-completion model starts from iid Gaussian noise in both channels and applies the same heat interpolation without solving the Darcy equation

at intermediate nodes. Its BASE model is retrained from scratch under the same deployment grid, while the next-state physical loss remains active. TRD is then retrained on the corresponding rollout distribution using the same pressure-only recovery target. Thus, the comparison changes the physical structure of the teacher path while retaining the same finite-map learning and recovery framework.

Table 21: Darcy $N = 20$ ablation of the physical teacher path.

Method	RE	WD_p	WD_K	JS_p	JS_K	Valid
PAFM BASE	15.5038	0.003106	0.131538	0.000734	0.004220	1.0000
PAFM (2k)	0.0778	0.002669	0.131133	0.000644	**0.004212**	1.0000
No-completion BASE	5.4987	0.041800	1.119381	0.044545	0.350639	1.0000
No-completion + TRD (2k)	0.1829	0.123248	0.998642	0.135424	0.240277	1.0000
PAFM (10k, formal)	**0.0399**	**0.002474**	**0.131098**	**0.000555**	0.004213	1.0000

Note. PAFM (2k) and No-completion + TRD (2k) use the same 2,000-epoch recovery budget for a strictly matched comparison. The formal PAFM result uses 10,000 TRD epochs and is reported separately. The no-completion variant is not extended to 10,000 epochs because its distributional quality degrades substantially beyond 2,000 recovery epochs.

Table 21 shows that the physical path is crucial for preserving the generated distribution. Before recovery, the no-completion BASE attains a somewhat lower residual than the physically completed BASE (5.4987 versus 15.5038), indicating that an unconstrained ambient path can be easier to regress locally. This apparent advantage, however, comes with a severe loss of generative fidelity: WD_p increases from 0.003106 to 0.041800 and WD_K from 0.131538 to 1.119381, with similarly large degradation in both JS metrics. Hence, easier finite-step regression does not imply a better generative transport. Despite the greater geometric complexity introduced by PDE-based completion, the physical teacher path provides a unified sample-wise coefficient–solution target for every finite transition, leading to substantially better distributional fidelity.

The contrast becomes stronger after TRD. On the physical PAFM path, recovery reduces RE from 15.5038 to 0.0778 while simultaneously improving every reported distribution metric: WD_p decreases to 0.002669, WD_K remains essentially unchanged at 0.131133, and both JS divergences decrease slightly. TRD therefore acts almost purely as a physical correction when applied to states already organized by the physical teacher path. In the no-completion model, recovery also reduces the residual, from 5.4987 to 0.1829, but cannot recover the missing distributional structure. In particular, the pressure distribution deteriorates further, with WD_p increasing to 0.123248 and JS_p to 0.135424. Although the permeability metrics improve modestly, they remain far worse than those of the physically completed model.

This ablation indicates that the physical path and TRD play complementary rather than interchangeable roles. Physical completion organizes finite teacher transitions around the correct sample-wise coefficient–solution relation, thereby preserving the target distribution seen by BASE. TRD then removes the residual inconsistency introduced by imperfect learned transitions without needing to reconstruct the underlying distribution. When the physical path is removed, the recovery head can still reduce the PDE residual, but it must operate on rollout states that were never organized around the correct physical relation, and distributional fidelity is no longer preserved.

The formally selected PAFM model further supports this interpretation. Extending TRD training from $2{,}000$ to $10{,}000$ epochs reduces RE from 0.0778 to 0.0399 and further improves the pressure metrics, while the permeability distribution remains essentially unchanged ($\mathrm{JS}_K = 0.004212$ versus 0.004213). Thus, once the transport itself is physically aligned, additional recovery optimization continues to improve sample-wise PDE consistency without degrading permeability distributional fidelity. This behavior is not observed for the no-completion construction, which begins to suffer substantial distribution degradation beyond $2{,}000$ recovery epochs.

Figure 11 summarizes the same effect in the physics–distribution plane. The physical-path PAFM trajectory moves predominantly toward lower residual while remaining in essentially the same low-error permeability region. The no-completion trajectory starts from a substantially worse distributional state and remains far from the PAFM region even after recovery. The ablation therefore supports the central design choice of PAFM: physically feasible intermediate states are not only useful for reducing

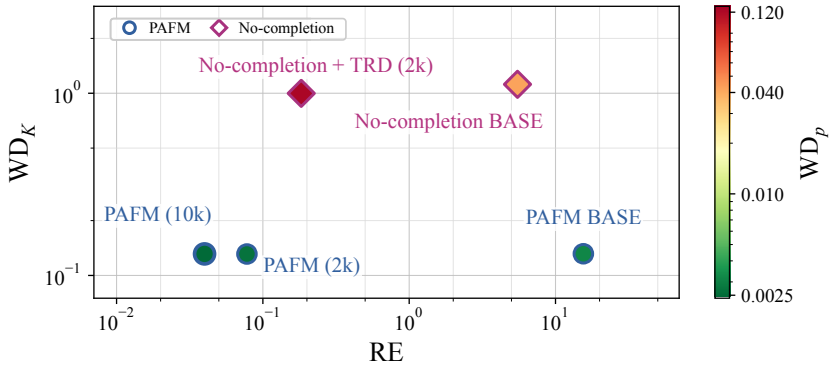


Figure 11: Effect of the physical teacher path in Darcy Flow at $N = 20$. The axes report physical residual (RE) and permeability distribution error (WD_K), while color encodes pressure distribution error (WD_p). PAFM with the physical teacher path (circles) maintains substantially better joint physical–distributional behavior than the no-completion variant (diamonds), both before and after TRD. Lower is better for all three quantities.

residuals, but also provide a better structured transport on which lightweight learned recovery can operate without trading away distributional fidelity.

J Initial-Prior Construction and Online Timing Boundaries

Table 22 separates initial-prior construction from BASE teacher construction and TRD target construction. Kolmogorov and Dynamic Stall measurements begin with iid Gaussian generation and end with a normalized physical prior ready for the learned rollout. They are measured on an RTX 4090 in float32, with CUDA synchronization around every batch; each entry is the median of five repetitions after warm-up. Darcy prior construction is timed on the same 12-core Intel Xeon Platinum 8352V host. The reported value is measured over the 1,024 priors used in formal evaluation. The prior bank is constructed once and reused only to make the evaluation deterministic.

Table 22: Initial physical-prior construction cost. The reported cost is paid once before the learned rollout and is separate from BASE teacher construction and TRD target construction.

Task	Prior construction	Samples	Construction cost
Kolmogorov Flow	Gaussian two-channel field followed by Fourier–Leray projection	2,048	0.0219 s total 0.0107 ms/sample
Dynamic Stall	Four-channel Gaussian field followed by analytical completion to six channels	2,048	0.2269 s total 0.1108 ms/sample
Darcy Flow	Gaussian $\log K$ field followed by a global pressure solve for pressure	1,024	11.779 s total 11.503 ms/sample

For Kolmogorov Flow and Dynamic Stall, constructing the physically completed initial state is inexpensive. Darcy Flow is the only benchmark for which this initialization requires a global PDE solve: given the sampled permeability, we solve the corresponding sparse elliptic system for pressure, with a measured cost of 11.503 ms per sample.

This solve is optional at deployment. As shown in Appendix E, the same trained BASE and TRD models can instead start from Gaussian free variables with a zero-initialized pressure channel, eliminating the initial elliptic solve. This solver-free initialization preserves low physical residuals, with some degradation in distributional fidelity. Accordingly, the 11.503 ms initialization cost is not included in the inference times reported in the main tables.

If a physically completed initial state is desired, it can be computed once before the learned rollout or precomputed and cached. In either case, no physical completion or numerical PDE solve is required between PAFM transitions.

K Full Multi-Budget Benchmark Tables

The main text reports the metrics needed to compare physics, distribution, and inference cost. The complete measured results are reproduced here, including all per-field metrics and high-budget references.

Table 23: Complete Dynamic Stall results under different sampling budgets. Ideal, Wall, and Total are multiplied by 10^6; WD and JS by 10^4; MMSE and SMSE by 10^5. IT denotes inference time per generated sample. All metrics are lower-is-better, and bold denotes the best measured result within each group.

Base NFE	Method	Ideal	Wall	Total	WD	JS	MMSE	SMSE	IT (ms/sample)
1	FM-OT	6.8294	17.5925	24.4218	11.8901	20.2415	2.7458	10.0544	**0.2562**
	D-Flow†	5.3057	15.7050	21.0107	11.9960	**20.0595**	2.7830	9.9521	456.3973
	PCFM‡	**1.5131×10^{-8}**	**0.0006**	**0.0006**	11.6722	21.0537	2.6501	10.1701	3.6924
	PBFM	17.8550	28.3252	46.1802	22.4446	33.7547	6.0551	**6.6768**	0.4332
	PAFM	0.1193	0.5587	0.6780	**9.4064**	28.0477	**2.2364**	7.2840	0.4644
5	FM-OT	1.8551	3.9943	5.8495	7.0606	6.1842	1.7370	2.2669	**1.1824**
	D-Flow†	0.9228	2.9782	3.9011	8.3206	6.7029	2.0032	2.4608	1639.6320
	PCFM‡	**1.5120×10^{-8}**	**0.0002**	**0.0002**	7.1389	5.5604	**1.6578**	**1.0686**	16.4112
	PBFM	0.0939	0.3047	0.3985	7.4753	**5.0228**	1.8909	1.3030	1.5144
	PAFM	0.0035	0.0065	0.0100	**6.2387**	8.9274	2.2538	7.2737	1.7051
10	FM-OT	2.0108	3.4837	5.4945	6.9045	5.2578	**1.6201**	1.6070	**2.3826**
	D-Flow†	0.6314	2.0336	2.6651	7.7449	5.3890	1.7559	1.7075	3296.0870
	PCFM‡	**1.5128×10^{-8}**	**0.0002**	**0.0002**	7.7424	5.8277	2.0574	2.1048	31.4927
	PBFM	0.0641	0.2394	0.3035	7.0796	**4.2106**	1.6534	**0.9682**	2.8436
	PAFM	0.0062	0.0076	0.0138	**5.9905**	5.3922	1.8575	2.7233	3.3636
20	FM-OT	2.2520	3.3463	5.5982	6.8774	5.1800	1.5574	1.3318	**4.7030**
	D-Flow†	0.4897	1.4899	1.9796	8.0409	5.0270	1.7562	1.4256	13288.3153
	PCFM‡	**1.5127×10^{-8}**	**0.0002**	**0.0002**	9.0410	9.0251	3.8658	7.6688	61.6236
	PBFM	0.0796	0.2580	0.3376	7.0618	3.9945	1.5903	**0.8781**	5.7570
	PAFM	0.0022	0.0053	0.0075	**5.5943**	**3.8970**	**1.4428**	1.4513	6.2478
1000	ECI¶	5.6074×10^{-10}	8.2182×10^{-10}	1.3826×10^{-9}	51.2573	82.1576	47.6389	102.9190	278.0336
2000	CoCoGen§	8.3154	1070.3255	1078.6409	229.6978	322.4083	416.6552	258.5966	13541.1196

†D-Flow optimizes the source with Strong-Wolfe L-BFGS through an Euler-integrated flow generator. For Base NFE $1/5/10/20$, it averages $19.2/103.9/226.8/483.9$ DiT calls, $18.2/19.8/21.7/23.2$ backward passes, and $19.2/20.8/22.7/24.2$ physics evaluations per sample. IT includes the full optimization.

‡PCFM reuses the frozen FM-OT model with 20 relaxed physics updates per outer step. For Base NFE $1/5/10/20$, it performs $22/106/211/421$ constraint evaluations and $2/6/11/21$ projections. IT includes all online corrections.

¶ECI follows the official $n_{\text{step}} = 200$, $n_{\text{mix}} = 5$ protocol, giving 1000 DiT calls and 1000 constraint corrections per sample.

§CoCoGen uses its native PF-ODE sampler with 2000 score evaluations and 60 physics-gradient corrections per sample.

Table 24: Complete Darcy Flow results on 1,024 generated samples. Error metrics are lower-is-better; Valid is the fraction of samples with finite generated fields and $K = \exp(a) > 0$ everywhere. Metrics involving $\log K$ are computed only on valid samples. A dash denotes no valid samples, and bold marks the lowest error within each matched-NFE block.

NFE	Method	RE	WD_p	WD_K	$\mathrm{WD}_{\log K}$	JS_p	JS_K	$\mathrm{JS}_{\log K}$	Valid	IT (ms/sample)
1	FM-OT	8.7008	0.002349	0.836530	0.672157	**0.006409**	0.534615	**0.555444**	1.0000	0.1715
	ECI	8.5648	0.002355	0.836661	0.672280	0.006517	0.534642	0.555666	1.0000	0.1804
	D-Flow †	5.6145	**0.002342**	0.838783	0.673721	0.006648	0.550313	0.567443	1.0000	732.029
	DiffusionPDE‡	–	–	–	–	–	–	–	0.0000	4.2762
	PBFM	–	–	–	–	–	–	–	0.0000	0.1749
	PAFM	**0.0301**	0.006710	**0.797580**	**0.637845**	0.010235	**0.515427**	0.639790	1.0000	0.2847
5	FM-OT	3.6494	**0.001744**	**0.229731**	**0.165104**	**0.000539**	**0.021072**	**0.022265**	1.0000	0.8443
	ECI	2.7635	0.001971	0.406690	0.281263	0.001427	0.065074	0.067566	1.0000	0.8900
	D-Flow†	1.9209	0.006464	0.438500	0.269320	0.002442	0.048406	0.050486	1.0000	1487.524
	DiffusionPDE‡	–	–	–	–	–	–	–	0.0000	38.1527
	PBFM	0.5395	0.039701	1.112755	1.463946	0.041953	0.361478	0.367438	0.4082	0.8493
	PAFM	**0.0306**	0.005009	0.410762	0.278946	0.003376	0.064211	0.066086	1.0000	1.4020
10	FM-OT	3.5152	**0.000903**	**0.132235**	0.088866	**0.000239**	0.005717	0.006119	0.9990	1.6867
	ECI	2.0090	0.002172	0.421856	0.268795	0.001377	0.055527	0.057793	1.0000	1.7664
	D-Flow†	1.6893	0.002967	0.267195	0.145927	0.000679	0.013408	0.014057	1.0000	2586.431
	DiffusionPDE‡	157.7207	0.011529	0.133823	**0.074045**	0.009222	**0.001659**	**0.001973**	0.0293	80.5596
	PBFM	0.7267	0.025450	0.778208	0.670397	0.021378	0.118017	0.119017	0.9980	1.6905
	PAFM	**0.0320**	0.003218	0.240570	0.151623	0.001190	0.016323	0.016933	1.0000	2.7983
20	FM-OT	3.7771	**0.000671**	0.076421	0.047981	**0.000154**	0.001637	0.001769	0.9990	3.3712
	ECI	1.6436	0.004052	0.455475	0.279781	0.001835	0.052868	0.054835	1.0000	3.5401
	D-Flow†	1.6144	0.002400	0.185848	0.092919	0.000434	0.004965	0.005127	0.9990	4752.133
	DiffusionPDE‡	5.3063	0.006656	**0.034896**	**0.023051**	0.002585	**0.000311**	**0.000389**	0.9102	165.5825
	PBFM	0.8427	0.014997	0.529766	0.389719	0.008768	0.039020	0.039176	0.9971	3.3783
	PAFM	**0.0399**	0.002474	0.131098	0.080243	0.000555	0.004213	0.004385	1.0000	5.6011
100	PIDM-ME	0.0219	0.031189	0.167445	0.097833	0.100009	0.005917	0.006599	1.0000	65.6041
2000	CoCoGen§	31.5452	0.005203	0.061045	0.063732	0.001819	0.001097	0.001148	0.6514	2213.5475

† D-Flow varies its flow integration steps; its released per-sample LBFGS loop makes the actual network calls $21 \times \mathrm{NFE}$.

‡ DiffusionPDE uses the released deterministic EDM Heun recurrence. The table's 1/5/10/20 reverse steps entail 1/9/19/39 denoiser calls and 1/5/10/20 residual-gradient calls, respectively. IT includes all of them.

§ CoCoGen uses $\tau = 2000$ probability-flow ODE steps, physical-consistency corrections during the final $N = 3$ reverse steps, and $M = 1$ additional post-sampling correction.

L Training and Teacher-Construction Cost

We report the computational cost of the final $N = 20$ configurations on a single NVIDIA GeForce RTX 4090 with an Intel Xeon Platinum 8352V CPU. Across all three benchmarks, PAFM follows the same computational pattern: the physical teacher state required by each sampled transition is constructed on the fly during BASE training, while TRD uses a small target set constructed once before optimization for each recovery round. The former cost is therefore already included in the measured BASE update time, whereas the latter is reported separately in Table 25. Because the same trained checkpoints admit deployment without a completed prior, the inference times reported in the main tables measure only the learned BASE+TRD rollout. The cost of constructing the physically completed prior used in formal evaluation is reported separately in Appendix J.

Table 25: Training and teacher-construction cost at $N = 20$. Stage-I teacher construction is performed on the fly and is included in BASE time per update; Stage II reports the total offline cost of constructing the TRD target cache(s). Peak GPU memory is measured with batch size 64.

Task	Physical completion D_c	Stage I: BASE construction	Stage II: TRD construction	BASE s/update	GPU training	Peak memory
Kolmogorov Flow	Fourier–Leray projection	On the fly included in update	10,240 targets ≈ 1.5 min	0.201	32.5 GPU-h	2.65 GiB
Dynamic Stall	Pointwise known relations	On the fly included in update	8,192 targets 75.10 s	0.0558	9.2 GPU-h	2.97 GiB
Darcy Flow	Numerical PDE solver	On the fly included in update	4,608 transitions 50.26 s	0.474	6.6 GPU-h	2.91 GiB

Table 25 shows that the same training strategy remains practical across substantially different physical constraints. BASE updates take only 0.0558–0.474 s and require less than 3 GiB of peak GPU memory, while construction of the complete TRD target set takes at most about 1.5 minutes. Even for Darcy Flow, where teacher construction requires a global sparse PDE solve, the optimized solver can be executed within the training loop without introducing a separate large-scale preprocessing stage.

The low memory footprint follows directly from finite-transition training. Each optimization step samples only one finite teacher pair; PAFM does not backpropagate through a complete generative rollout and does not retain the completion map or solver in the GPU computation graph. Consequently, increasing the computational complexity of D_c mainly affects the cost of constructing the current teacher transition rather than the memory complexity of optimization. This allows all three final models, including their physically structured supervision, to be trained on a single commodity GPU with modest additional construction overhead.