PHYSICS-CONSTRAINED STOCHASTIC ROMS FOR UNSTEADY AIRFOIL FLOWS

Giacomo Baldan
Technical University of Munich
Garching, Germany
giacomo.baldan@tum.de

Qiang Liu
Technical University of Munich
Garching, Germany
qiang7.liu@tum.de

Alberto Guardone
Politecnico di Milano
Milan, Italy
alberto.guardone@polimi.it

Nils Thuerey
Technical University of Munich
Garching, Germany
nils.thuerey@tum.de

ABSTRACT

We introduce Physics-Based Flow Matching (PBFM), a novel framework that integrates physical residuals into generative modeling to enforce physical consistency. PBFM leverages conflict-free gradient composition and trajectory unrolling to address the challenges of balancing generative fidelity and physical accuracy, eliminating the need for manual loss weighting. By mitigating Jensen's gap during nonlinear residual evaluation, PBFM achieves improved physical adherence without compromising sample quality. We validate our approach on a challenging unsteady compressible flow problem relevant to real-world engineering applications. PBFM demonstrates state-of-the-art performance, achieving high distributional fidelity and low residual errors at inference costs comparable to standard flow matching. Additionally, we provide a detailed analysis of the impact of noise-level selection and unrolling depth, offering practical insights for advancing physics-constrained generative modeling.

Code and datasets available at https://github.com/tum-pbs/PBFM.

1 INTRODUCTION

Partial differential equations (PDEs) serve as the fundamental framework for describing dynamic physical systems in space and time (Evans, 2010). While they are rigorous, the numerical discretization of PDEs frequently leads to high-dimensional systems that are expensive to solve, particularly when multiscale or nonlinear dynamics are involved (Haber et al., 2018; Valencia et al., 2025). Consequently, data-driven approaches have gained traction as powerful surrogates, approximating PDE solutions with dramatically lower computational overhead (Chen et al., 2021; Fresca et al., 2021; Baldan et al., 2021; Brunton & Kutz, 2024). Among these, Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019; Baldan et al., 2023) integrate constraints directly into the loss function. However, their deterministic nature restricts them from capturing stochastic behaviors essential for uncertainty quantification contexts (Roy & Oberkampf, 2011; Abdar et al., 2021; Liu & Thuerey, 2024). Conversely, generative frameworks such as Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020; Nichol & Dhariwal, 2021), their deterministic counterparts (DDIMs) (Song et al., 2022), and Flow Matching (Lipman et al., 2023) have demonstrated strong capabilities in modeling complex distributions in fields ranging from media generation to graph learning (Rombach et al., 2022; Ho et al., 2022; Kong et al., 2021; Chamberlain et al., 2021). Flow Matching, specifically, stands out for its straightforward formulation and efficiency in generating high-quality samples (Esser et al., 2024). Nevertheless, embedding physical laws into generative frameworks presents a non-trivial optimization problem. Simultaneously training for data distribu-

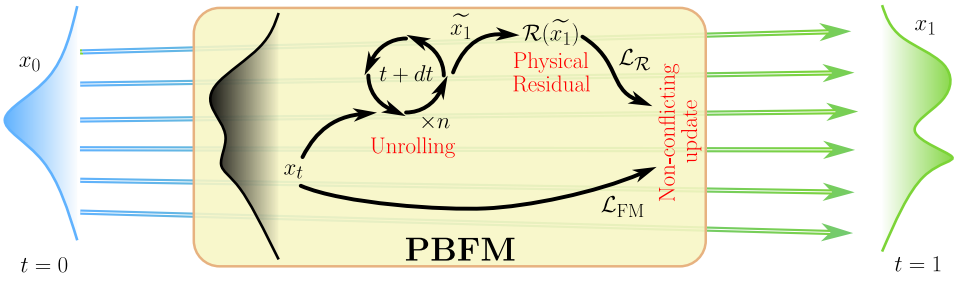

Figure 1: Overview of the proposed approach. During training, the sample x_t at time t is evolved to $t = 1$ over n time steps to compute the residual $\mathcal{R}(\widetilde{x}_1)$. The flow matching loss $\mathcal{L}_{\text{FM}}$ and residual loss $\mathcal{L}_{\mathcal{R}}$ are combined in a conflict-free manner.

tion fidelity and physical consistency often induces conflicting gradient updates, where improving one metric degrades the other (Krishnapriyan et al., 2021). Balancing these competing objectives is challenging. A critical technical aspect is *Jensen's gap*, which creates a discrepancy when constraints are applied to the estimated posterior mean at noisy intermediate steps instead of the clean final output (Zhang & Zou, 2025). While some methods enforce constraints strictly during inference, they typically rely on costly iterative adjustments in contrast with the speed advantages of the generative model (Utkarsh et al., 2025). Furthermore, backpropagating through nonlinear hard constraints for high-dimensional systems is often computationally intractable (Cheng et al., 2025). To overcome these obstacles, we enhance the Flow Matching paradigm by synthesizing techniques such as trajectory unrolling, gradient conflict resolution, and stochastic sampling (Liu et al., 2025), tailored specifically for physics-aware modeling. Fig. 1 summarizes the proposed training pipeline.

Our contributions can be summarized as follows: (**I**) We propose a constrained Flow Matching framework that minimizes PDE and algebraic residuals simultaneously without requiring manual hyperparameter tuning. (**II**) We demonstrate that training with unrolled trajectories effectively bridges Jensen's gap, resulting in lower residual errors and sharper predictions at zero additional inference cost. (**III**) We investigate the impact of Gaussian noise levels on physical constraint satisfaction, and (**IV**) we provide a detailed performance analysis on a real aerodynamic case relevant for helicopter blade and wind turbine design. A key benefit of our method is its ease of integration into existing pipelines, delivering consistent gains in both physical validity and sample quality.

2 Related Work

Deep learning has been extensively applied to tackle intricate challenges within physics and engineering domains (Morton et al., 2018; Wang et al., 2020; Sanchez-Gonzalez et al., 2020; Thuerey et al., 2020). Recently, the focus has shifted towards adapting foundation models for scientific machine learning. Prominent examples include PDEformer (Ye et al., 2024), Poseidon (Herde et al., 2024), Aurora (Bodnar et al., 2024), and Unisolver (Zhou et al., 2025a), all of which strive to create versatile models capable of generalizing across varied physical systems. Concurrently, specialized transformer architectures like OFormer (Li et al., 2023), Transolver (Wu et al., 2024), and Fengbo (Pepe et al., 2025) have been developed specifically for PDE-related tasks.

Focusing on diffusion models for engineering and physical sciences, notable applications range from molecular and drug discovery (Guo et al., 2024; Schneuing et al., 2024; Bose et al., 2024; Guastoni & Vinuesa, 2025) to simulating particle trajectories in high-energy physics (Mikuni et al., 2023), resolving inverse problems (Holzschuh et al., 2023), and modeling turbulent particle dispersion (Li et al., 2024). Parallel to these applications, numerous studies focus on embedding physical priors or constraints directly into the diffusion framework. For example, Huang et al. (2024) leveraged generative priors to infer missing data in partial observations of PDE solutions. Zhou et al. (2025b) introduced a method to embed conservation laws (energy, momentum) and PDE constraints within the diffusion process. Additionally, Rixner & Koutsourelakis (2021) developed a probabilistic model that enforces physics via virtual observables.

Table 1: Comparison between our method and other physics-constrained generative models.

Method	Physics at Training	Hard Constraints	Gradient Free Inference	Complex Constraints	Balanced Hyperparams
FFM [30]	✗	✓	✓	✗	✗
ProbConserv [21]	✗	✓	✓	✗	✗
CoCoGen [28]	✗	✗	✗	✓	✗
DiffusionPDE [27]	✗	✗	✗	✓	✗
PIDM [6]	✓	✗	✓	✓	✗
D-Flow [7]	✗	✗	✗	✓	✗
ECI [13]	✗	✓	✓	✗	✗
PCFM [55]	✗	✓	✗	✓	✗
PIDDM [61]	✓	✗	✗	✓	✗
PBFM (ours) [3]	✓	✗	✓	✓	✓

Closely related to our approach, Shu et al. (2023) proposed conditioning the diffusion process on residual gradients during both training and inference, primarily for super-resolution in turbulent flows, though they do not enforce the constraint. Similarly, CoCoGen (Jacobsen et al., 2025) incorporates governing equations purely at inference time, enhancing physical consistency at the cost of increased computational time, a significant drawback for practical deployment. Several other inference-time constraint methods exist, such as FFM (Kerrigan et al., 2023), DiffusionPDE (Huang et al., 2024), D-Flow (Ben-Hamu et al., 2024), ECI (Cheng et al., 2025), and PCFM (Utkarsh et al., 2025). These typically rely on iterative sampling to satisfy physics, heavily increasing the computational burden, up to an order of magnitude compared to standard diffusion (Utkarsh et al., 2025). Among these, ECI is restricted to simple constraints, while PCFM handles arbitrary constraints but demands expensive residual differentiation. Towards training-time integration, Bastek et al. (2025) proposed Physics-Informed Diffusion Models (PIDM), adding a residual minimization term to the loss. Zhang & Zou (2025) developed Physics-Informed Distillation of Diffusion Models (PIDDM) to address the Jensen's gap by fine-tuning a student model to minimize residuals. However, this necessitates a dual-model training setup. Both approaches, like other recent fine-tuning methods such as PIRF (Yuan et al., 2025) and PCFT (Tauberschmidt et al., 2025), struggle to fundamentally resolve the gradient conflict between generative and physical objectives. A detailed comparison of our proposed method against these existing approaches is provided in Table 1.

3 Methodology

3.1 Preliminaries

A general time-dependent partial differential equation (PDE) in n spatial dimensions can be expressed as $\mathbf{u}_t(\mathbf{x}, t) = \mathcal{L}[\mathbf{u}(\mathbf{x}, t)] + \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^n$, where $\mathbf{u}(\mathbf{x}, t)$ denotes the state variable, $\mathcal{L}$ is a spatial differential operator, and $\mathbf{f}(\mathbf{x}, t)$ represents external forcing. For numerical solutions, the computational domain Ω and its boundary $\partial\Omega$ are discretized, and continuous operators are replaced by suitable discrete counterparts (Karniadakis & Sherwin, 2005).

Physical consistency can be enforced by minimizing a residual function derived from the governing equations. Depending on the modeling setting, residual stochasticity in generative frameworks can be broadly classified into three categories. (i) For *steady-state* PDEs, uncertainty in physical parameters induces a distribution over solutions, and the residual is computed directly from the governing equations, e.g., $\mathcal{R}(\mathcal{L}(\mathbf{u}) + \mathbf{f}) = 0$. (ii) For *time-dependent* PDEs, the system evolves dynamically, and residuals are formulated to enforce conservation laws, such as mass, momentum, or energy, across temporal states. (iii) *Algebraic constraints* may further be imposed to enforce consistency between coupled physical quantities, thereby strengthening adherence to the underlying physics.

3.2 Physics-based flow matching

Incorporating physical constraints into generative models presents a fundamental trade-off: enforcing physical fidelity often degrades distributional accuracy, while prioritizing generative perfor-

mance can weaken physical consistency. Existing diffusion-based approaches (Shu et al., 2023; Bastek et al., 2025) typically address this trade-off using a weighted objective,

$$\arg\max_{\theta} \mathbb{E}_{x_1\sim q(x_1)}[\log p_\theta(x_1)] + \mathbb{E}_{x_1\sim p_\theta(x_1)}\left[\log q_{\mathcal{R}}(\hat{r}=0 \mid x_1)\right] \tag{1}$$

where $\hat{r}$ denotes virtual observables (Rixner & Koutsourelakis, 2021) derived from the residual $\mathcal{R}(x_1)$. Within the flow-matching framework, this objective reduces to (Bastek et al., 2025)

$$\mathcal{L} = w_{\text{FM}}\mathcal{L}_{\text{FM}} + w_{\mathcal{R}}\mathcal{L}_{\mathcal{R}} = w_{\text{FM}}\|u_t^\theta(x_t,t) - u_t(x_t)\|_2 + w_{\mathcal{R}}\|\mathcal{R}(x_1(x_t,t))\|_2 \tag{2}$$

where w_{FM} and $w_{\mathcal{R}}$ are manually tuned weights balancing the generative and physical objectives. Selecting these weights is nontrivial: increasing the residual weight often deteriorates sample quality, whereas emphasizing the generative loss compromises physical validity.

To overcome this limitation, we adopt a multi-task optimization strategy that explicitly resolves conflicting gradient directions. In particular, we employ the *ConFIG* method (Liu et al., 2025) to construct conflict-free parameter updates for physics-based flow matching:

$$\mathbf{g}_{\text{update}} = (\mathbf{g}_{\text{FM}}^\top \mathbf{g}_v + \mathbf{g}_{\mathcal{R}}^\top \mathbf{g}_v)\,\mathbf{g}_v, \tag{3}$$

$$\mathbf{g}_v = \mathcal{U}[\mathcal{U}(\mathcal{O}(\mathbf{g}_{\text{FM}}, \mathbf{g}_{\mathcal{R}})) + \mathcal{U}(\mathcal{O}(\mathbf{g}_{\mathcal{R}}, \mathbf{g}_{\text{FM}}))]\,, \tag{4}$$

where $\mathbf{g}_{\text{FM}}$ and $\mathbf{g}_{\mathcal{R}}$ are the gradients of $\mathcal{L}_{\text{FM}}$ and $\mathcal{L}_{\mathcal{R}}$, respectively. The operator

$$\mathcal{O}(\mathbf{g}_1, \mathbf{g}_2) = \mathbf{g}_2 - \frac{\mathbf{g}_1^\top \mathbf{g}_2}{\|\mathbf{g}_1\|^2}\mathbf{g}_1$$

extracts the component of $\mathbf{g}_2$ orthogonal to $\mathbf{g}_1$, and $\mathcal{U}(\mathbf{g}) = \mathbf{g}/\|\mathbf{g}\|$ normalizes vectors to unit length. The resulting update direction satisfies $\mathbf{g}_{\text{update}}^\top \mathbf{g}_{\text{FM}} > 0$ and $\mathbf{g}_{\text{update}}^\top \mathbf{g}_{\mathcal{R}} > 0$, ensuring simultaneous descent for both objectives. Figure 2 illustrates this gradient composition.

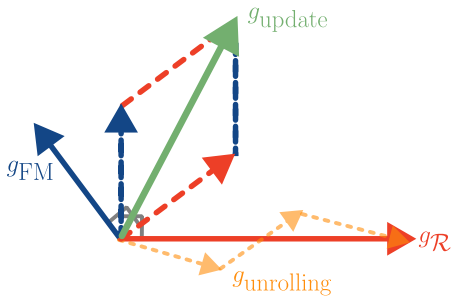

Figure 2: PBFM method leverages ConFIG to combine the unit vectors of the orthogonal components and scales the result using the projection length of the FM and the physical residual $\mathcal{R}$ gradients.

By adaptively aligning gradients, this approach eliminates the need for manual loss weighting and prevents any single objective from dominating the optimization. Consequently, it achieves high distributional fidelity while maintaining strong physical consistency, outperforming fixed-weight formulations across a wide range of weight choices (see Appendix E).

3.3 Improving Physical and Distributional Accuracy

For scientific applications, generative models must produce samples that not only match the target distribution but also achieve physical accuracy comparable to traditional numerical solvers. A central challenge in this setting is *Jensen's gap*, which arises whenever a nonlinear function f is applied to a random variable Z, since generally $\mathbb{E}[f(Z)] \neq f(\mathbb{E}[Z])$. When physical constraints are enforced on the posterior mean $\mathbb{E}[x_1 \mid x_t]$ at intermediate noise levels, rather than on the final clean sample x_1, this discrepancy can significantly degrade physical fidelity (Zhang & Zou, 2025). To alleviate this issue, we emphasize accurate reconstruction of the final noise-free sample during training when evaluating the physical residual. Concretely, we employ *unrolling*, in which the intermediate state at time t is integrated forward to $t = 1$ using multiple ODE steps. This procedure evaluates the residual on a more accurate approximation of x_1, rather than relying on a single-step estimate. Unrolling over n steps of size $(1-t)/n$ yields a closer approximation to the true trajectory and effectively reduces Jensen's gap. The number of unrolling steps is increased gradually via a curriculum during training. To further suppress noisy residual estimates near $t = 0$, we weight the residual loss by a factor t^p, with an optimal choice $p_{\text{opt}} = 1$. This linear scaling is consistent with the linear noise schedule in flow matching and aligns the residual weighting with the underlying dynamics. A comparison of alternative power laws is provided in Appendix D. Accurate reconstruction of x_1 substantially improves the evaluation of the physical residual in Eq. 2, leading to more

informative optimization directions and reduced residual errors, without increasing inference-time cost. The main drawback of unrolling is increased memory usage during training, as intermediate states must be stored for backpropagation. Table 4 reports the associated computational overhead for different unrolling depths.

Another critical design choice in physics-based flow matching is the minimum noise level σ_{min} added to training data, as dictated by the mixture-of-Gaussians formulation. While values such as $\sigma_{\text{min}} = 10^{-3}$ are common in computer vision (Lipman et al., 2023; Esser et al., 2024), excessive noise can corrupt physical residuals and degrade performance. The choice of σ_{min} therefore determines the minimum achievable residual error. As shown in Table 3, injecting Gaussian noise of scale σ_{min} induces an irreducible residual MSE of approximately σ_{min}^2, motivating the guideline $\sigma_{\text{min}} \lesssim \mathcal{R}_{\text{min}}$. Following practices in image generation (Esser et al., 2024), we also sample the time variable t from a logit-normal distribution with zero mean and unit variance during training, rather than uniformly. This choice emphasizes regions where flow matching errors are typically largest, particularly around $t \approx 0.5$.

Algorithm 1 Training procedure for PBFM

$n \leftarrow$ number of unrolling steps
$dt \leftarrow (1-t)/n$ ▷ Compute dt to reach the final state at $t = 1$
$\widetilde{t} \leftarrow t$ ▷ Keep the starting time t to weight the residual loss
$u_t^\theta \leftarrow \text{model}(x_t,\ t)$ ▷ Flow matching velocity u_t^θ is not part of unrolling
$\widetilde{x}_1 \leftarrow x_t + dt \cdot u_t^\theta$
for $i = 1,\ i < n$ **do** ▷ Improved state prediction via unrolling
  $\widetilde{t} = \widetilde{t} + dt$
  $\widetilde{u}_t^\theta \leftarrow \text{model}(\widetilde{x}_1,\ \widetilde{t})$
  $\widetilde{x}_1 \leftarrow \widetilde{x}_1 + dt \cdot \widetilde{u}_t^\theta$ ▷ Current fields $\widetilde{x}_1$ are updated until $t = 1$
end for
$\mathcal{R} \leftarrow \text{compute residual}(\widetilde{x}_1)$
$\mathcal{L}_{\mathcal{R}} \leftarrow \|t^p \cdot \mathcal{R}\|_2$ ▷ Residuals are weighted with $t^p,\ p_{\text{opt}} = 1$
$\mathcal{L}_{\text{FM}} \leftarrow \|u_t^\theta - u_t\|_2$
$\nabla_\theta \leftarrow$ compute $\mathbf{g}_{\text{update}}$ via Eq. 3 ▷ Conflict-free update
AdamW optimizer step with ∇_θ

Algorithm 1 summarizes the resulting training procedure. The flow-matching loss is computed at time t, while unrolling is used to obtain an accurate estimate of the final state for residual evaluation. The initial time is retained to properly weight the residual term.

At inference time, samples are usually generated by deterministically integrating an initial Gaussian noise sample $\mathcal{N}(0, I)$ forward in time using an ODE solver (Gao et al., 2025). In contrast, DDPM-based samplers rely on stochastic transitions (Ho et al., 2020). Since both diffusion models and flow matching can be viewed as instances of generative modeling under arbitrary Markov processes (Holderrieth et al., 2025), we additionally investigate a stochastic sampling strategy inspired by the ECI method (Cheng et al., 2025). The key idea is to evolve the system forward to $t = 1$ and then step backward to $t + dt$ using a newly sampled noise realization. This backward step injects additional stochasticity, improving distributional coverage while remaining compatible with physics-based flow matching. The complete sampling procedure is described in Algorithm 2.

4 Experimental Setup

We evaluate the generative performance of our method, denoted with *Physics-Based Flow Matching* (PBFM) in the following, on a real-world engineering scenario: the unsteady aerodynamic phenomenon of dynamic stall. Across the experiments, we employ a diffusion transformer (DiT) backbone architecture (Peebles & Xie, 2023), with minor modifications detailed in Appendix F.

4.1 Residual formulation

This setup involves spatio-temporal fields over a pitching NACA0012 airfoil, capturing the phenomenon of *dynamic stall*. Dynamic stall is a complex, unsteady aerodynamic effect that leads to

Table 2: Generative performance metrics using 20 FM steps. RE: physical Residual MSE, WD: Wassserstein Distance, JS: Jensen-Shannon divergence, MMSE: MSE of the mean fields, SMSE: MSE of the standard deviation fields, NFE: Number of function evaluations, IT: Inference wall-clock time. **Best** and second best results.

Metric	PBFM	OT-FM	DiffusionPDE	D-Flow	PCFM
RE $\cdot 10^6$	0.339	11.02	12.20	11.32	**0.143**
WD $\cdot 10^4$	**1.814**	2.707	2.509	3.484	4.013
JS $\cdot 10^2$	**0.680**	0.983	1.029	1.014	1.206
MMSE $\cdot 10^5$	**1.490**	2.791	2.626	2.507	5.669
SMSE $\cdot 10^5$	**0.874**	1.458	1.236	1.372	7.674
IT $[ms]$	60.47	**59.75**	171.7	138.9	3906

significant load fluctuations, making it highly relevant for practical applications such as helicopter and wind turbine blades. The solutions are derived by solving the compressible Navier-Stokes equations, with the resulting flow fields exhibiting shock waves and intricate flow features that are challenging to model. Each sample consists of 128×128 grids representing six quantities: absolute pressure, temperature, density, skin friction, and tangential velocity gradients (in both x and y directions). The operating conditions and airfoil motion are defined by four conditioning parameters. The training dataset includes 128 base configurations, each perturbed 32 times to account for uncertainty, while the validation dataset contains 16 unseen configurations. Physical consistency across the fields is enforced using two analytical, point-wise residual constraints: $\mathcal{R}_{\text{ig}}$, which enforces the ideal gas law, and $\mathcal{R}_\tau$, which minimizes skin friction based on Sutherland's law.

$$\mathcal{R}_{\text{ig}} = P - \rho R T \,, \qquad \mathcal{R}_\tau = \tau_w - \mu_0 \frac{T_0 + S}{T + S} \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \sqrt{\left(\frac{\partial u}{\partial x}\right)^2_{n=0} + \left(\frac{\partial u}{\partial y}\right)^2_{n=0}}$$

5 RESULTS

The dynamic stall scenario is particularly challenging, involving two intricate physical constraints and six predicted fields. The underlying dynamics are highly non-linear, featuring shock waves. Fig. 3 illustrates an example of the predicted fields. Notably, regions such as the center display strongly varying, small-scale shock waves that stochastically oscillate under perturbed operating conditions: a critical phenomenon for industrial applications.

Table 3: Metric comparison for different values of $\sigma_{\min}$ on the dynamic stall problem, 20 FM steps. **Best** and second best results.

Metric	$\sigma_{\min}$ 0.0	10^{-4}	10^{-3}	10^{-2}
RE $\cdot 10^6$	**0.339**	0.468	0.473	0.466
WD $\cdot 10^4$	**1.814**	3.547	3.246	3.566
JS $\cdot 10^2$	**0.680**	0.716	0.718	0.728
MMSE $\cdot 10^5$	1.490	1.360	1.298	**1.200**
SMSE $\cdot 10^5$	0.874	0.831	**0.789**	0.916

Table 2 presents the quantitative results. PBFM demonstrates superior performance across nearly all metrics, achieving the lowest distributional errors and MSE for both the mean and standard-deviation fields. This indicates that PBFM effectively captures both first- and second-order statistics of the complex flow fields. While PCFM achieves the lowest residual error, it does so at the expense of poorer distributional metrics and higher mean and standard-deviation errors. Additionally, PCFM suffers from significantly higher inference times (approximately $65\times$ that of PBFM) due to its iterative correction process.

Inference times for PBFM are comparable to FM-OT. DiffusionPDE and D-Flow incur higher computational costs because evaluating their physical residuals is more complex, although the cost difference is limited since the residuals are pointwise.

Fig. 4 illustrates how residual error evolves with the number of unrolling steps used during training and it demonstrates that unrolling effectively mitigates Jensen's gap, leading to lower residual errors.

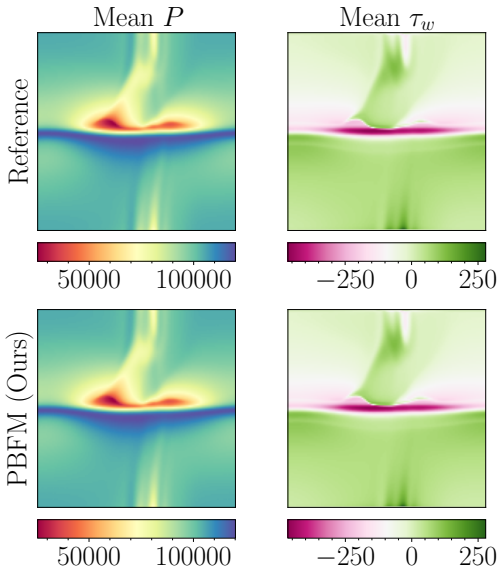

Figure 3: Dynamic stall, comparison of mean for different methods computed over 128 samples using 20 FM steps.

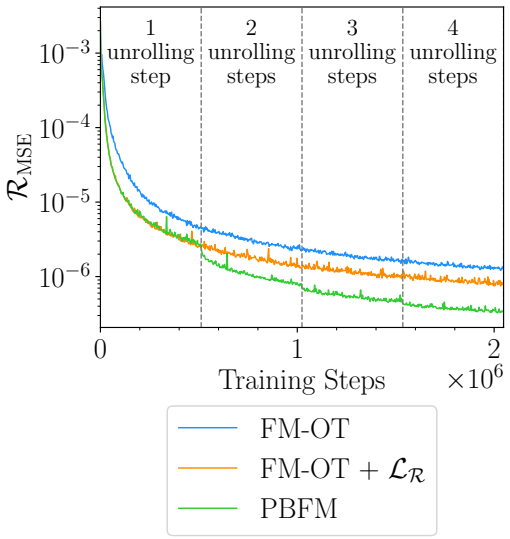

Figure 4: Physical residual MSE as a function of training steps.

Furthermore, it highlights how just adding the physics loss with ConFIG is not sufficient to reduce the residuals significantly without unrolling. This confirms that accurately reconstructing the final state is crucial for effective physics-constrained generative modeling.

We analyze the influence of varying $\sigma_{\min}$, which is particularly critical in this setup due to the very small magnitude of the residuals. Table 3 presents the results. Setting $\sigma_{\min} = 0$ achieves the best distributional metrics and residual error, highlighting the importance of minimizing noise perturbation in this challenging scenario. Increasing $\sigma_{\min}$ to 10^{-4} slightly worsens the metrics, while further increases to 10^{-3} and 10^{-2} result in more significant performance degradation. This trend underscores the sensitivity of physical residuals to noise levels, emphasizing the importance of carefully tuning $\sigma_{\min}$ in physics-constrained generative modeling. However, higher noise levels lead to reduced mean and standard deviation errors. A practical observation is that adding Gaussian noise of scale $\sigma_{\min}$ introduces a residual MSE approximately equal to $\sigma_{\min}^2$ in an ideal reconstruction scenario. For dynamic stall, this implies an upper bound: $\sigma_{\min}^2 \lesssim 10^{-7} \Rightarrow \sigma_{\min} \lesssim 3 \times 10^{-4}$. It is worth noting that this sensitivity is not specific to our method but is inherent to flow matching formulations, as the noise level directly affects the scale of the denoising target and the achievable physical accuracy.

5.1 PBFM VARIANTS

We tested two variants on the PBFM framework to assess whether decomposing the physics objective or modifying gradient flow could yield further gains. The evaluated variants are:

- *PBFM 3 losses*: the physics objective is split into two separate residual losses, $\mathcal{R}_{\text{ig}}$ and $\mathcal{R}_{\tau}$, trained alongside the FM loss.
- *PBFM last step*: same as the base PBFM but the gradient is computed only at the last step of the unrolled predictions when computing the residual loss to reduce backward memory and computational time.

We analyze the training performance by examining the impact of the residual loss combined with ConFIG and the effect of unrolling. Table 4 presents the wall-clock time, in seconds, for a single training iteration on an NVIDIA A100 64 GB GPU for the three variants. For comparison, FM-OT without physics constraints requires $4.29 \cdot 10^{-2}$ seconds and 4.25 GB of memory.

Table 4: Comparison of wall-clock time in seconds for one training iteration and memory usage in GB on an NVIDIA A100 64GB GPU for the proposed approaches. FM-OT without physics constraints requires $4.29 \cdot 10^{-2}$ seconds and 4.25 GB of memory. Batch size is 64 for all cases. Inference time is unchanged.

Method		**1 step**	**2 steps**	**3 steps**	**4 steps**
PBFM	[s]	$8.14 \cdot 10^{-2}$	$1.18 \cdot 10^{-1}$	$1.55 \cdot 10^{-1}$	$1.90 \cdot 10^{-1}$
	[GB]	4.41	7.18	9.90	12.58
PBFM	[s]	$1.08 \cdot 10^{-1}$	$1.68 \cdot 10^{-1}$	$2.29 \cdot 10^{-1}$	$2.84 \cdot 10^{-1}$
3 losses	[GB]	4.41	7.20	9.91	12.59
PBFM	[s]	$8.35 \cdot 10^{-2}$	$1.09 \cdot 10^{-1}$	$1.23 \cdot 10^{-1}$	$1.34 \cdot 10^{-1}$
last step	[GB]	4.41	8.48	8.48	8.48

Table 5: Generative performance metrics for dynamic stall problem using 20 FM steps. RE: physical Residual MSE, WD: Wassserstein Distance, JS: Jensen-Shannon divergence, MMSE: MSE of the mean fields, SMSE: MSE of the standard deviation fields.

Dataset	**Metric**	**PBFM**	**PBFM 3 losses**	**PBFM last step**
	RE $\cdot 10^6$	0.339	0.315	0.392
	WD $\cdot 10^4$	1.814	2.077	3.038
Dynamic	JS $\cdot 10^2$	0.680	0.653	0.722
Stall	MMSE $\cdot 10^5$	1.490	1.696	1.814
	SMSE $\cdot 10^5$	0.874	0.842	0.809

Table 5 summarizes the generative performance metrics using 20 FM steps. Splitting the physics terms into separate losses does not yield clear overall improvements. While the three-loss variant achieves a slightly lower residual MSE (0.315 vs. 0.339), the Wasserstein distance increases from 1.814 to 2.077. Additionally, the three-loss setup increases per-iteration training time by approximately 30-40% due to extra gradient computations, with only a marginal rise in peak memory usage. Combining the physics constraints into a single residual loss and resolving conflicts using ConFIG offers the best balance between accuracy and training efficiency for the dynamic stall case. The *last step* variant reduces backward computation costs, with only minor impacts on final accuracy. Gradient alignment between the two physics terms was also measured, showing a high average cosine similarity (approximately 0.76). This explains why aggregating the physics terms into a single loss is both effective and computationally efficient in practice.

5.2 Extrapolation

To provide a complete overview of the PBFM framework, we evaluate extrapolation settings in which at least one conditioning parameter lies outside the design space, by up to $\pm 10\%$ of its bounds. Figure 5 reports the mean squared error of the physical residuals for both interpolation and extrapolation regimes, comparing standard flow matching (FM-OT) with physics-based flow matching (PBFM). The residuals correspond to the ideal gas law and wall shear stress. PBFM consistently reduces physical residuals by approximately one order of magnitude compared to FM-OT. Notably, the resulting extrapolation error with PBFM is lower than the interpolation error obtained with the standard FM approach, indicating improved generalization beyond the training distribution. This suggests that incorporating physical residuals during training leads to more physically consistent models. For 50 FM steps, a slight increase in residual error is observed. However, this effect is of limited practical relevance, as the typical number of function evaluations ranges between 10 and 20.

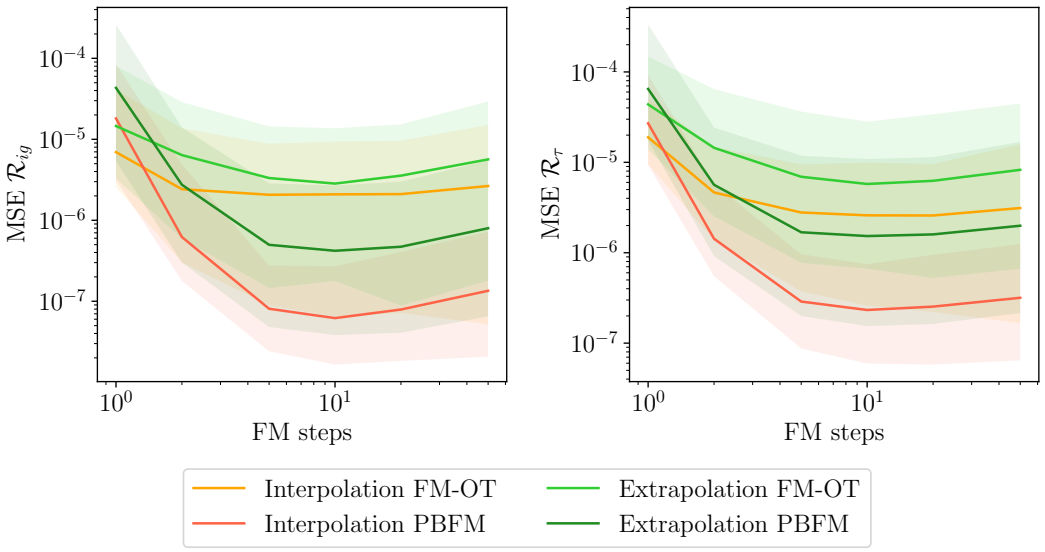

Figure 5: Mean squared error in interpolation and extrapolation conditions of the residuals for standard FM and PBFM. The residual corresponds to ideal gas law $\mathcal{R}_{ig}$ and wall shear stress $\mathcal{R}_{\tau}$,

6 Discussion and conclusions

This work introduces Physics-Based Flow Matching (PBFM), a practical training-time framework that brings physical fidelity into flow-matching generative models via conflict-free gradient composition and trajectory unrolling. Across a challenging unsteady aerodynamic benchmark, PBFM consistently reduces residuals and improves distributional statistics while preserving the fast inference characteristics of flow matching. The unrolling mechanism effectively narrows Jensen's gap by yielding more accurate final-state estimates for residual evaluation, and ConFIG stabilizes multi-objective training by removing manual loss balancing. Together these components enable reliable, physically informed sample generation suitable for large-scale uncertainty quantification and surrogate modeling.

From a practitioner perspective, PBFM (Baldan et al., 2026) offers clear benefits: stronger physical consistency without inference-time slowdown, straightforward integration into existing flow-matching pipelines, and robustness across a range of noise schedules and unrolling depths. Recommended settings for many engineering problems are modest unrolling (2-4 steps) and minimal $\sigma_{\min}$ consistent with numerical stability; these choices balance residual accuracy and memory footprint (see Table 4).

A notable limitation is applicability to very large 3D datasets: residual computations and their gradients can become expensive in high-resolution 3D domains, increasing both runtime and memory requirements during training. Practical mitigations include using coarser residual evaluations or surrogate residual models, localized or adaptive residual sampling, checkpointing and gradient checkpoint strategies, and distributed evaluation of residuals. Exploring such efficiency techniques is a priority for future work.

In summary, PBFM narrows a crucial gap between generative modeling and physics-based accuracy by combining principled gradient handling with a pragmatic unrolling strategy. The method delivers strong empirical gains on challenging flows, and with targeted efficiency improvements it can be extended to broader, higher-dimensional physical systems.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: 10.1016/j.inffus.2021.05.008. URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

ANSYS. Fluent 2024R2, 2024. URL https://www.ansys.com/products/fluids/ansys-fluent.

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Physics vs distributions: Pareto optimal flow matching with physics constraints. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tAf1KI3d4X.

Marco Baldan, Giacomo Baldan, and Bernard Nacke. Solving 1D non-linear magneto quasi-static Maxwell's equations using neural networks. *IET Science, Measurement & Technology*, 15(2): 204–217, 2021. doi: 10.1049/smt2.12022.

Marco Baldan, Paolo Di Barba, and David A. Lowther. Physics-informed neural networks for inverse electromagnetic problems. *IEEE Transactions on Magnetics*, 59(5):1–5, 2023. doi: 10.1109/TMAG.2023.3247023.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-Informed Diffusion Models. In *The Thirteenth 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 *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 2024.

Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian FATRAS, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M. Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kJFIH23hXb.

Steven L. Brunton and J. Nathan Kutz. Promising directions of machine learning for partial differential equations. *Nature Computational Science*, 4(7):483–494, Jul 2024. ISSN 2662-8457. doi: 10.1038/s43588-024-00643-2.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph Neural Diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 18–24 Jul 2021.

Wenqian Chen, Qian Wang, Jan S. Hesthaven, and Chuhua Zhang. Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of Computational Physics*, 446: 110666, 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110666.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=teE4pl9ftK.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010. ISBN 9780821849743 0821849743.

Stefania Fresca, Luca Dede', and Andrea Manzoni. A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs. *Journal of Scientific Computing*, 87(2):61, Apr 2021. ISSN 1573-7691. doi: 10.1007/s10915-021-01462-7.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025. URL https://openreview.net/forum?id=C8Yyg9wy0s.

Luca Guastoni and Ricardo Vinuesa. A new perspective on the simulation of stochastic problems in fluid mechanics with diffusion models: Fluid dynamics. *Nature Machine Intelligence*, pp. 1–2, 2025. doi: 10.1038/s42256-025-01060-4.

Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen, Duolin Wang, Dong Xu, and Jianlin Cheng. Diffusion models in bioinformatics and computational biology. *Nature Reviews Bioengineering*, 2(2):136–154, Feb 2024. ISSN 2731-6092. doi: 10.1038/s44222-023-00114-9.

Eldad Haber, Lars Ruthotto, Elliot Holtham, and Seong-Hwan Jun. Learning across scales—multiscale methods for convolution neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Derek Hansen, Danielle C. Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W. Mahoney. Learning physical models that can respect conservation laws. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient Foundation Models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=JC1VKK3UXk.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646. Curran Associates, Inc., 2022.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator Matching: Generative modeling with arbitrary Markov processes. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RuP17cJtZo.

Benjamin Holzschuh, Simona Vegetti, and Nils Thuerey. Solving inverse physics problems with score matching. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=z0I2SbjN0R.

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.

George Karniadakis and Spencer Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 06 2005. ISBN 9780198528692. doi: 10.1093/acprof:oso/9780198528692.001.0001.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional Flow Matching, 2023. URL https://arxiv.org/abs/2305.17209.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks, 2021. URL https://arxiv.org/abs/2109.01050.

T. Li, L. Biferale, F. Bonaccorso, M. A. Scarpolini, and M. Buzzicotti. Synthetic lagrangian turbulence by generative diffusion models. *Nature Machine Intelligence*, 6(4):393–403, Apr 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00810-0.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Qiang Liu and Nils Thuerey. Uncertainty-Aware Surrogate Models for Airfoil Flow Simulations with Denoising Diffusion Probabilistic Models. *AIAA Journal*, 62(8):2912–2933, 2024. doi: 10.2514/1.J063440.

Qiang Liu, Mengyu Chu, and Nils Thuerey. ConFIG: Towards Conflict-free Training of Physics Informed Neural Networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.11104.

Vinicius Mikuni, Benjamin Nachman, and Mariel Pettee. Fast point cloud generation with diffusion models in high energy physics. *Phys. Rev. D*, 108:036025, Aug 2023. doi: 10.1103/PhysRevD.108.036025. URL https://link.aps.org/doi/10.1103/PhysRevD.108.036025.

Jeremy Morton, Antony Jameson, Mykel J Kochenderfer, and Freddie Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems*, 2018.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Alberto Pepe, Mattia Montanari, and Joan Lasenby. Fengbo: a Clifford Neural Operator pipeline for 3D PDEs in Computational Fluid Dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VsxbWTDHjh.

M. Raissi, P. Perdikaris, and G.E. 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. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045.

Maximilian Rixner and Phaedon-Stelios Koutsourelakis. A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables. *Journal of Computational Physics*, 434:110218, 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110218. URL https://www.sciencedirect.com/science/article/pii/S0021999121001133.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Christopher J. Roy and William L. Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2131–2144, 2011. ISSN 0045-7825. doi: 10.1016/j.cma.2011.03.016. URL https://www.sciencedirect.com/science/article/pii/S0045782511001290.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to Simulate Complex Physics with Graph Networks, 2020. URL https://arxiv.org/abs/2002.09405.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L. Blundell, Pietro Lio, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4 (12):899–909, Dec 2024. ISSN 2662-8457. doi: 10.1038/s43588-024-00737-x.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.111972. URL https://www.sciencedirect.com/science/article/pii/S0021999123000670.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models, 2022. URL https://arxiv.org/abs/2010.02502.

Jan Tauberschmidt, Sophie Fellenz, Sebastian J. Vollmer, and Andrew B. Duncan. Physics-constrained fine-tuning of flow-matching models for generation and inverse problems, 2025. URL https://arxiv.org/abs/2508.09156.

Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, 2020.

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 Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints, 2025. URL https://arxiv.org/abs/2506.04171.

Mario Lino Valencia, Tobias Pfaff, and Nils Thuerey. Learning distributions of complex fluid simulations with diffusion graph networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uKZdlihDDn.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-Attention with Linear Complexity, 2020. URL https://arxiv.org/abs/2006.04768.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. PDEformer: Towards a Foundation Model for One-Dimensional Partial Differential Equations. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024. URL https://openreview.net/forum?id=GLDMCwdhTK.

Mingze Yuan, Pengfei Jin, Na Li, and Quanzheng Li. PIRF: Physics-informed reward fine-tuning for diffusion models, 2025. URL https://arxiv.org/abs/2509.20570.

Yi Zhang and Difan Zou. Physics-informed distillation of diffusion models for PDE-constrained generation, 2025. URL https://arxiv.org/abs/2505.22391.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-Conditional Transformers Are Universal Neural PDE Solvers. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025a. URL https://openreview.net/forum?id=6HlgUqkjmW.

Zihan Zhou, Xiaoxue Wang, and Tianshu Yu. Generating physical dynamics under priors. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=eNjXcP6C0H.

USE OF LARGE LANGUAGE MODELS

Large language models were used exclusively for text editing and correction. They were not involved in the generation of ideas, analyses, or substantive content.

A FLOW MATCHING

Flow matching has recently gained attention as a compelling alternative to diffusion models, offering a more direct and computationally efficient framework for generative modeling (Lipman et al., 2023). It enables high-quality sample generation with significantly fewer function evaluations (Esser et al., 2024). Given a known source distribution p and an unknown target distribution q, flow matching learns a vector field u_t^θ, parameterized by a neural network, that generates a probability path p_t interpolating from $p_0 = p$ to $p_1 = q$ (Lipman et al., 2024). The learning objective is defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x\sim p_t} \|u_t^\theta(x) - u_t(x)\|_2$$

where θ denotes the model parameters. While multiple formulations exist for the target velocity field u_t, a particularly simple and effective one leverages optimal transport (OT) (Tong et al., 2024). In this setting, samples from the base distribution $p_0 = \mathcal{N}(0, I)$ are linearly transported to p_1 via the conditional flow:

$$u_t(x \mid x_1) = \frac{x_1 - (1 - \sigma_{\min})x}{1 - (1 - \sigma_{\min})t},$$

with $\sigma_{\min} \sim 10^{-3}$, and the corresponding interpolant:

$$\psi_t(x) = (1 - (1 - \sigma_{\min})t)x + tx_1.$$

This yields a straight-line conditional flow with a time-independent vector field. Sampling from the trained model requires integrating the learned field over time:

$$x_1 = \int_0^1 u_t^\theta(x_t)\, dt,$$

typically using numerical ODE solvers such as Euler's method. Although the true OT vector field is constant, the learned approximation typically is not, and integration quality still depends on the time discretization.

B DATASET GENERATION

The dynamic stall datasets for training and validation are generated by solving the unsteady, compressible, two-dimensional Reynolds-Averaged Navier-Stokes (RANS) equations around a sinusoidally pitching NACA0012 airfoil. An O-grid mesh is utilized, consisting of 512 nodes along the airfoil surface and 128 nodes in the normal direction. The simulations are performed using ANSYS Fluent 2024R2 (ANSYS, 2024).

The governing equations are discretized using a second-order upwind scheme for spatial accuracy and a second-order implicit scheme for time integration. Gradient reconstruction is carried out with a least-squares cell-based method, while fluxes are computed using the Rhie-Chow momentum interpolation. Pressure-velocity coupling is handled via a coupled solver.

The airfoil's pitching motion is modeled by prescribing a rigid-body motion to the entire mesh, defined by the angular velocity function $\dot{\alpha}(t) = \omega\ \alpha_s \cos(\omega t)$. The mean angle of attack, α_0, is applied at the start of each simulation by appropriately rotating the mesh. The SST turbulence model, enhanced with an intermittency transport equation, is employed to close the RANS equations. Each oscillation cycle is resolved with 2,048 time steps, and the simulations are run until periodic convergence is achieved.

The design space is represented as a four-dimensional hypercube, where each axis corresponds to a conditioning input for the neural network. These inputs include the free-stream Mach number, the mean angle of attack α_0, the pitching amplitude α_s, and the reduced frequency $k = \omega c/2V_\infty$. The ranges for these variables are provided in Table 6.

Table 6: Range of conditioning inputs that define the operating conditions of the pitching airfoil.

Variable	Min	Max
Mach	0.3	0.5
α_0	5°	10°
α_s	5°	10°
k	0.05	0.1

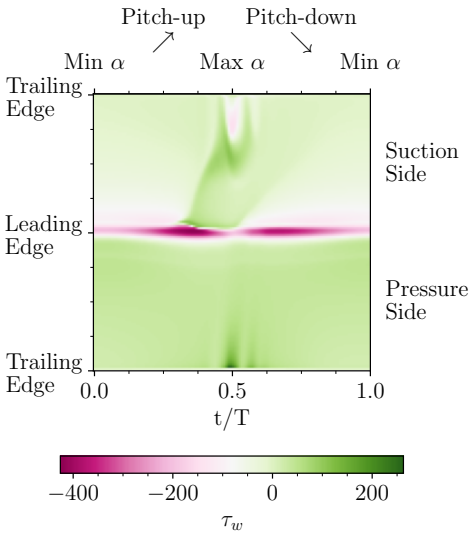

Figure 6: Example of a spatio-temporal contour of the post-processed τ_w distribution over an entire pitching cycle.

The training and testing datasets are generated using Halton sequences. The hypercube is sampled at 128 points for training and 16 points for testing, with each point representing a nominal operating condition. To introduce variability, each nominal condition is perturbed as follows:

$$x_{\text{perturbed}} = (1 + \mathcal{N}(0, 0.02))\ x_{\text{nominal}},$$

where $\mathcal{N}(0, 0.02)$ is a Gaussian noise term with zero mean and a standard deviation of 0.02. This results in 32 perturbed variations per nominal condition, yielding a total of 128×32 simulations for training and 16×32 for testing.

To reduce computational costs, all simulation fields are downsampled to a resolution of 128×128. The stored quantities include fields of absolute pressure, density, temperature, signed skin friction, and tangential velocity gradients along the airfoil surface over a complete pitching cycle. Figure 6 illustrates a spatio-temporal contour of the wall shear stress (τ_w) over one full pitching cycle, highlighting the pitch-up and pitch-down phases and the structured mapping of surface data over time.

C Additional samples and analysis

For the dynamic stall case, Fig. 7 illustrates the mean and standard deviation of the predicted fields generated using PBFM. The predictions align closely with the reference data, accurately capturing both the overall distribution and the extrema of each variable. A slight discrepancy is observed in the standard deviation of the skin friction, where the predicted values slightly exceed the reference.

Fig. 8 compares the performance of the proposed models as a function of the number of FM integration steps. The first panel highlights the physical residual MSE, showing that the best performance is achieved with 10 FM steps, with PBFM consistently reducing the residual by an order of magnitude compared to the baseline. The second panel presents the Wasserstein distance, demonstrating that PBFM also improves alignment with the reference data distribution, reaching its optimal value at 20

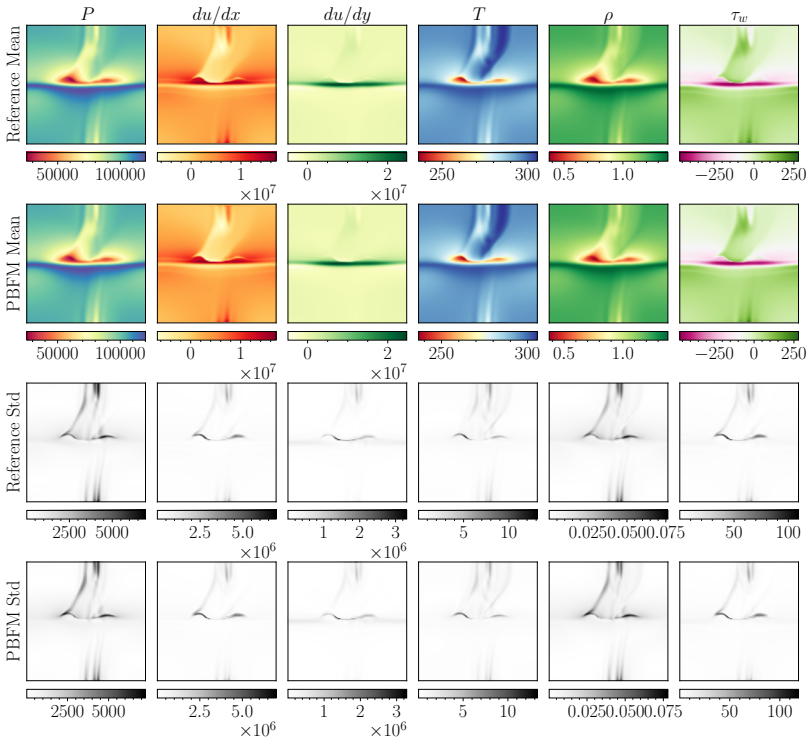

Figure 7: Example of mean and standard deviation for the dynamic stall problem, computed over 128 samples with 20 FM steps.

FM steps. Lastly, the MSE of the mean and standard deviation indicates that the proposed methods maintain the strong performance of the baseline, with only minor deviations.

Additional qualitative samples generated by the trained PBFM model are shown in Fig. 9. These samples showcase the diverse range of physical behaviors modeled in this case, from small oscillation attached flows at the top to deep dynamic stall scenarios with shock wave formation at the bottom.

D Residual loss scaling laws

During training, the residual for each sample is computed starting from a given time t and evolved forward until $t = 1$. Samples initialized closer to $t = 0$ typically exhibit larger errors, especially when only a single integration step is used. To address this, we apply a weighting scheme based on a power law t^p, scaling the residual loss according to the starting time t. We analyze the effect of different power exponents p, focusing on the challenging dynamic stall case.

Fig. 10 compares the MSE for physical residuals, as well as the mean and standard deviation, for both the FM-OT with Config and PBFM frameworks. The results indicate that unrolling regularizes the error, leading to a monotonic increase in error as p increases, while also reducing sensitivity to p within the range $[1, 4]$. Both frameworks achieve their best performance when residuals are scaled linearly with time ($p = 1$). In contrast, using no scaling ($p = 0$) results in significantly higher errors, highlighting the importance of appropriately weighting residuals based on the starting time.

E Conflict-free updates and weighted loss terms

Introducing a second loss term for physical residual minimization transforms the framework into a multi-task learning problem. A straightforward approach is to combine the two losses using a fixed weighting hyperparameter. However, this method requires manual tuning of the relative weights

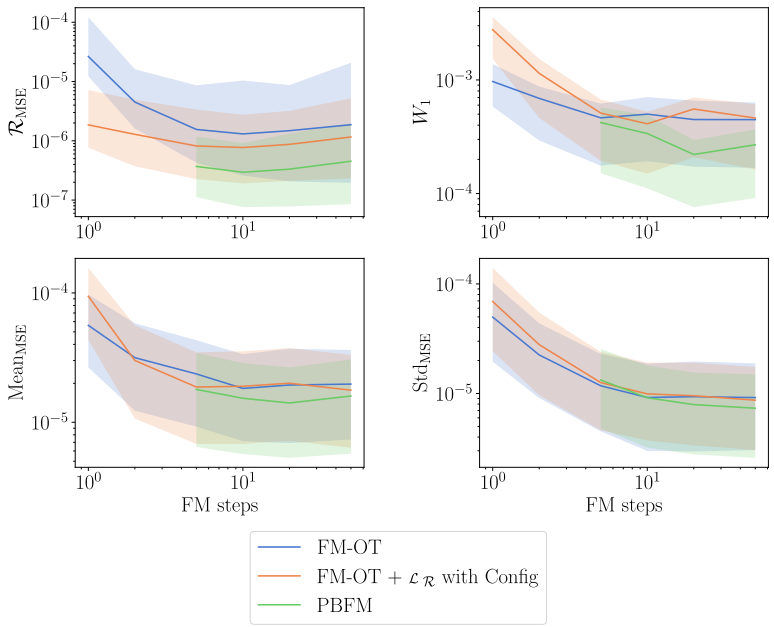

Figure 8: Comparison of the proposed approaches for physical residual and Wasserstein distance, mean and standard deviation MSE as a function of FM steps for dynamic stall case. Error bars refer to the different conditioning values within the validation dataset.

and often results in suboptimal performance. Specifically, optimization can become trapped in a local minimum of one loss due to conflicting gradients between the tasks. To address this issue, we employ conflict-free updates (Liu et al., 2025), which resolve gradient conflicts by computing a non-conflicting optimization direction using the inverse of the gradient covariance matrix. This approach has the potential to achieve improved optimization outcomes.

We evaluate this setup on the dynamic stall case, the most challenging scenario considered, comparing ConFIG to models trained with various fixed loss weights. Fig. 11 shows the MSE for the physical residual, predicted mean, and standard deviation across all configurations. ConFIG consistently outperforms the fixed-weight approaches, achieving the lowest error for each metric and delivering the best overall performance.

To further evaluate ConFIG's effectiveness, we measured gradient conflicts using pairwise cosine similarity between the physical and FM losses. For the best model with 4-step unrolling and ConFIG, the average conflict was 6.93%, compared to 19.98% for a model with fixed residual scaling at 500.

F Architecture and Training Details

The framework is implemented using PyTorch v2.5.1, with Distributed Data Parallel (DDP) employed for scalable training. All experiments utilize the AdamW optimizer, configured with a weight decay of 0, $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a fixed learning rate. To ensure consistent learning rates across different hardware setups, a constant global batch size is maintained, regardless of the number of GPUs. An Exponential Moving Average (EMA) of the model parameters is maintained throughout training with a decay rate of 0.999, and the EMA parameters are used during sampling.

The backbone of our flow matching model is based on the Diffusion Transformer (DiT) architecture introduced by Peebles & Xie (2023), with minor modifications. Conditioning is applied through adaptive layer normalization (adaLN-Zero) blocks, which replace the standard normalization layers. The scale and shift parameters in these blocks are computed from the sum of the embedding vectors for the time step t (used in flow matching) and the conditioning signal c.

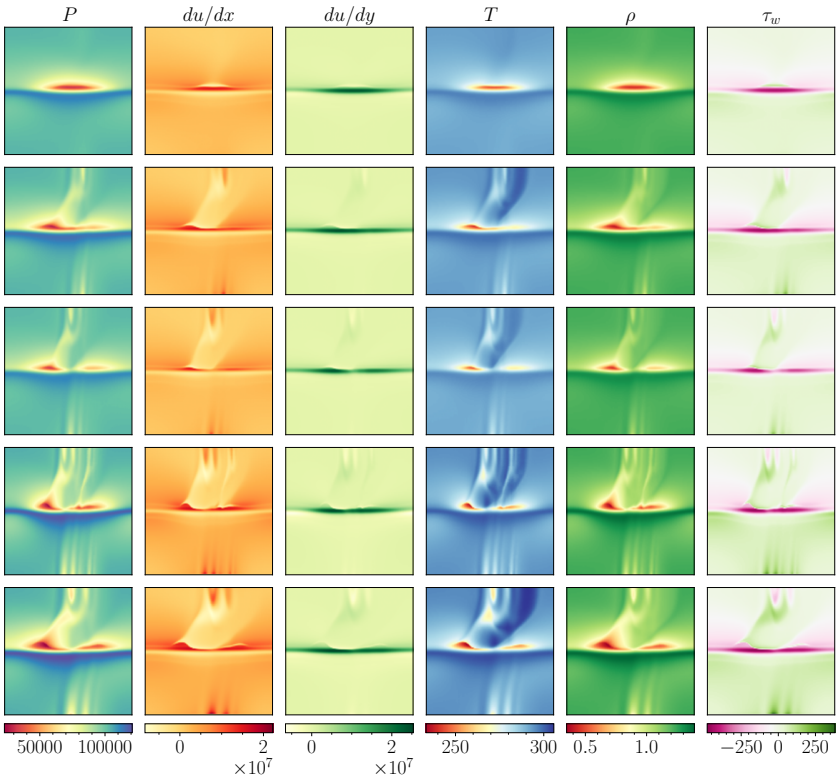

Figure 9: Example of dynamic stall flow field samples generated with PBFM using 20 FM steps. The outputs vary substantially depending on the conditioning inputs, illustrating the model's sensitivity to different flow scenarios.

We introduce two key modifications to the original DiT implementation. First, we integrate linear attention (Wang et al., 2020) alongside the standard quadratic attention mechanism. Second, the original label embedder is replaced with a two-layer module: a linear layer followed by a SiLU activation function, and another linear layer that generates the final conditioning embedding.

The hyperparameters are summarized in Table 7.

Table 7: Architecture and training hyperparameters for dynamic stall.

Hyperparameter	**Value**
Training iterations	2048000
Learning rate	$1 \cdot 10^{-4}$
Batch size	64
Conditioning size	4
Output size	6
Patch size	4
Hidden size	128
DiT depth	4
Attention heads	4
Attention type	Linear
Parameters (M)	1.29
Gflops	1.68

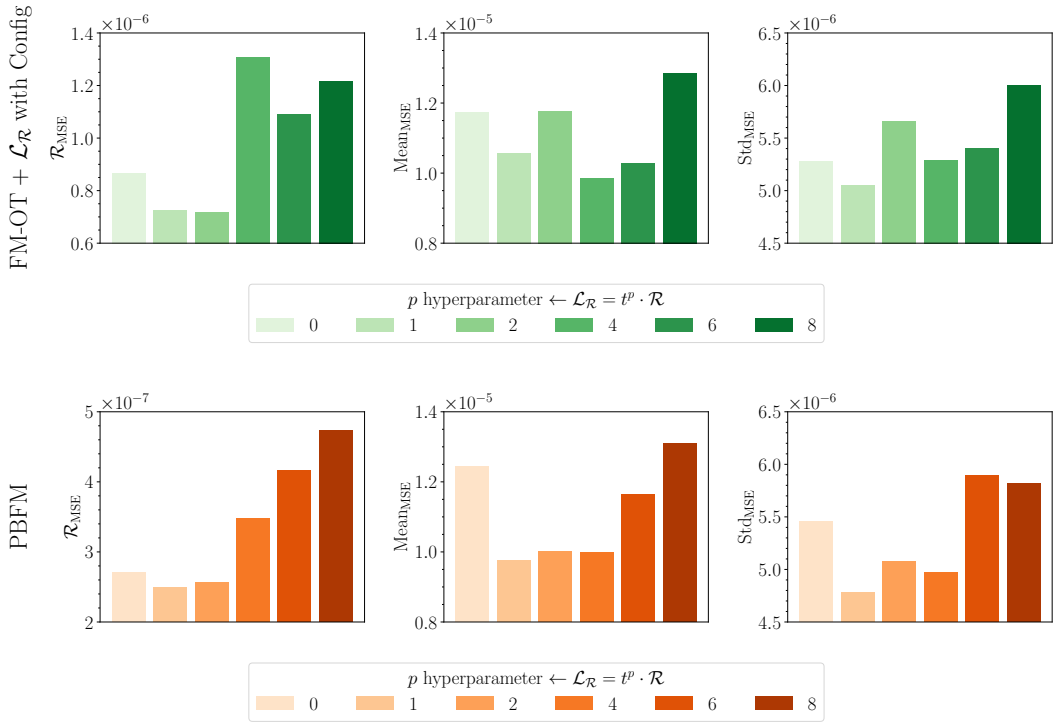

Figure 10: Comparison of physical residual, mean and standard deviation MSE for dynamic stall under various power-law scalings in the residual loss. Unrolling reduces sensitivity to the scaling exponent.

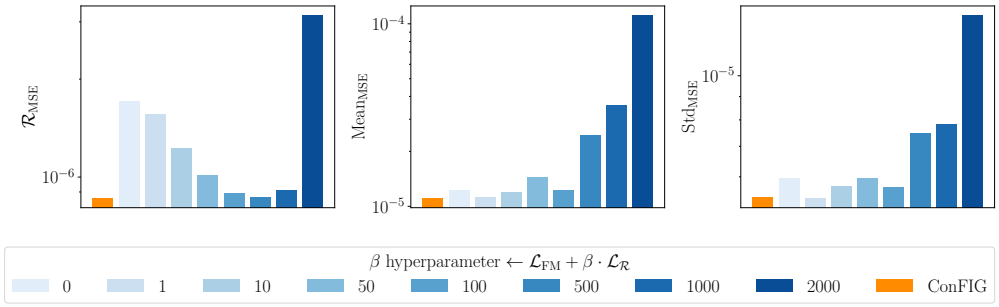

Figure 11: Comparison of MSE for physical residual, predicted mean, and standard deviation in the dynamic stall case, using ConFIG and fixed β hyperparameters for loss weighting. The optimal configuration minimizes the error across all three metrics simultaneously.

G Sampler Implementation and Additional Details

Algorithm 2 outlines the proposed sampling procedure, which is implemented using the explicit Euler integration scheme with n equispaced time steps. The process begins by initializing x_t, which holds the sample values at time $t = 0$, with Gaussian noise. During the integration loop, each time step can be computed using either the standard deterministic FM sampler or the proposed stochastic variant. The choice between the two is governed by a user-defined boolean control parameter and is restricted to the initial segment of the trajectory, up to a threshold time t^*. In our experiments, we set $t^* = 0.2$, introducing additional stochasticity during the early phase of sampling while preserving high sample quality in later stages. In the stochastic sampler, the velocity u_t^θ is used to generate the

final sample in a single forward step, followed by a backward update to time $t + dt$ using a new Gaussian noise sample.

Algorithm 2 Deterministic and Stochastic Samplers

$dt \leftarrow 1/n$ ▷ n is the number of integration steps
$x_t \leftarrow x_0 = \mathcal{N}(0, 1)$
for $i = 0,\ i < n$ **do**
  if $t < t^*$ **and** use stochastic sampler **then**
    $x_t \leftarrow x_t + (1 - t) \cdot u_t^\theta$ ▷ Integrate to $t = 1$
    $t \leftarrow t + dt$
    $x_t \leftarrow (1 - t) \cdot \mathcal{N}(0, 1) + t \cdot x_t$ ▷ Return to $t + dt$ using new generated normal noise
  else
    $x_t \leftarrow x_t + dt \cdot u_t^\theta$ ▷ Standard deterministic sampler, stochasticity is embedded in x_0
    $t \leftarrow t + dt$
  end if
end for