IMPROVED SAMPLING OF DIFFUSION MODELS IN FLUID DYNAMICS WITH TWEEDIE'S FORMULA

Youssef Shehata* **Benjamin Holzschuh** **Nils Thuerey**
Technical University of Munich
85748 Garching, Germany

ABSTRACT

Denoising Diffusion Probabilistic Models (DDPMs), while powerful, require extensive sampling due to a high number of function evaluations (NFEs) for accurate predictions, hindering their use in long-term spatio-temporal physics predictions. We address this limitation by introducing two novel sampling strategies: 1) Truncated Sampling Models, which achieve high-fidelity single/few-step sampling by truncating the diffusion process, bridging the gap with deterministic methods; and 2) Iterative Refinement, a reformulation of DDPM sampling as a few-step refinement process. We demonstrate that both methods significantly improve accuracy over DDPMs, DDIMs, and EDMs with NFEs ≤ 10 for compressible transonic flows over a cylinder and provide stable long-term predictions.

Turbulent flow is prevalent in everyday phenomena ranging from natural occurrences (Sullivan & McWilliams, 2024; Pyakurel et al., 2017) to engineering applications (Tulapurkara, 1997; Cheah et al., 2007; Volpe et al., 2014). Computational Fluid Dynamics (CFD) is essential for understanding these flows, with direct numerical simulations being the gold standard. However, it requires high-resolution grids to resolve the full spectrum of turbulent spatial and temporal scales, resulting in computationally intensive simulations (Pope, 2012). This limitation has propelled the recent surge in data-driven approaches. Leveraging the abundance of high- and low-fidelity data, machine learning algorithms offer various opportunities to enhance the accuracy and efficiency of turbulence simulations (Vinuesa & Brunton, 2022), particularly for phenomena like turbulent flows, which are challenging for traditional CFD methods.

Diffusion models (DMs) as surrogates. DMs (Hyvärinen, 2005; Sohl-Dickstein et al., 2015) have demonstrated great potential in various domains (Dhariwal & Nichol, 2021; Wang et al.; Lugmayr et al.; Ho et al.; Li et al., 2023); however, their application to fluid-based problems remains an underexplored area of research. To date, applications involved inverse problems (Holzschuh et al., 2023), high-fidelity reconstruction (Shu et al., 2023), autoregressive sampling in two-dimensional (Yang & Sommer, 2023; Lippe et al., 2023; Kohl et al., 2024) and three-dimensional (Lienen et al., 2024) settings, and an uncertainty-aware surrogate for airfoil simulations (Liu & Thuerey, 2024).

Motivation for using DMs. DMs have been shown to autoregressively generate videos or simulation trajectories, which are unconditionally stable over very long time horizons (Kohl et al., 2024). Temporal stability is difficult to accomplish using supervised loss terms, requiring memory-consuming techniques like multi-step unrolling (Um et al., 2020). Additionally, the probabilistic nature of DMs can deal very well with measurement noise or missing data (Huang et al., 2024), making them highly robust and versatile. Since a prediction from the DM samples from the posterior, this allows small variations due to uncertainty in the input to naturally evolve and amplify over time, creating diverse predictions over many steps, while individual trajectories always remain physically accurate.

Need for faster and more accurate sampling. The main drawback of DMs, especially evident in fluid problems, is the long inference time due to the large number of function evaluations (NFEs) required, and the limited accuracy compared to deterministic baselines (Cachay et al., 2023). Therefore, our objective in this study is to reduce the inference time disparity between DMs and single-step deterministic baselines while concurrently enhancing the accuracy of their predictions. This is achieved through our proposed straightforward training and sampling procedures.

*Correspondence to: y.shehata@tum.de

The main contributions of this work can be summarized as follows:

1. We introduce *Truncated Sampling Models (TSMs)* to enable single-step and few-step sampling while preserving or even augmenting sampling fidelity. We describe how truncation of the diffusion process, typically employed to reduce NFEs, can improve inference accuracy. Additionally, we distinguish our proposed TSMs from related approaches and highlight their efficiency and ease of implementation.
2. We introduce an inherently stochastic *Iterative Refinement (IR)* approach to enable flexible sampling of conditional diffusion models, allowing reduced NFEs with improved accuracy compared to ancestral sampling. We explain the intuition behind the approach and provide a comparative analysis against existing methods in our experiment.
3. We empirically demonstrate the efficacy of the proposed methods in reducing inference steps and improving the accuracy of diffusion models for compressible transonic flows over a cylinder.

1 BACKGROUND

1.1 DENSOISING DIFFUSION PROBABILISTIC MODELS

Denoising Diffusion Probabilistic Models (Ho et al., 2020, DDPMs), a class of generative DMs, convert a data distribution $q(x_0)$ to a prior distribution $q(x_T) \sim \mathcal{N}(0, \boldsymbol{I})$ over T steps through a Markovian forward diffusion process $q(x_t \mid x_{t-1})$ by gradually adding Gaussian noise with noise schedule β_t. We can sample the state x_t directly from x_0 through the parameterized closed form:

$$q(x_t \mid x_0) = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon\,, \tag{1}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\alpha_t = 1 - \beta_t$. The forward process posterior $q(x_{t-1} \mid x_t)$ is approximated using a neural network through a parameterized Gaussian distribution $p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right)$, where μ_θ and Σ_θ are the network predicted mean and variance, respectively. However, in this study, the variance is kept constant according to the noise schedule β_t and is chosen to be $\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$, with $\sigma_t^2 = \beta_t$.

During inference, the reverse process (i.e., ancestral sampling) begins from $x_T \sim \mathcal{N}(0, \boldsymbol{I})$ and iteratively samples $x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$ for T steps until reaching a fully denoised state x_0. The network is trained to estimate the forward process posterior by minimizing the Kullback-Leibler (KL) divergence $\mathrm{KL}(q(x_{t-1} \mid x_t) || p_\theta\left(x_{t-1} \mid x_t\right))$, which reduces to the loss function (Ho et al., 2020):

$$L_{t-1} := \mathbb{E}_{x_0, \epsilon}\left|\left|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\right|\right|^2, \tag{2}$$

where the network only learns to predict the noise at each noise step to perform partial denoising. The denoising step is related to Tweedie's formula (Efron, 2011), which can be used to estimate the posterior mean $\mathbb{E}[\hat{x}_0 | x_t; \theta]$ from a noisy sample x_t via

$$\mathbb{E}[\hat{x}_0 | x_t; \theta] = (x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x_t, t))/\sqrt{\bar{\alpha}_t}. \tag{3}$$

1.2 PROBLEM DEFINITION

The Navier-Stokes (NS) Partial Differential Equations (PDEs) represent a class of problems that epitomize the complex physics encountered in engineering and scientific disciplines. For an arbitrary domain Ω, fluid motion is governed in space and time τ by the NS PDE, defined as:

$$\frac{\partial u}{\partial \tau} + (u \cdot \nabla)u = -\nabla p + \frac{1}{Re}\Delta u + f, \quad \frac{\partial \rho}{\partial \tau} + \rho(\nabla \cdot u) = 0, \tag{4}$$

where f is external forcing, and u, p, and ρ are the velocity, pressure, and density, respectively. Re is the non-dimensional Reynolds number, controlling the severity of diffusive to convective transport.

Reformulated autoregressive sampling. For time-dependent problems, to reach the target state $\mathbf{x}(\tau_f)$, where $\mathbf{x} = \{u, p, \rho\}$, and with the initial condition $\mathbf{x}_0$, our reformulation of autoregressive sampling is defined as (notice that the notations $\mathbf{x}(\ldots)$ and $\mathbf{x}_{\ldots}$ are used interchangeably):

$$\mathbf{x}(\tau_f) = p_\theta^T(x_T, p_\theta^T(\ldots p_\theta^T(x_T, \mathbf{x}_0, j)\ldots), j), \tag{5}$$

where the prediction stride j denotes how far we sample in the future based on the physical timestep $\delta\tau$. The shortened notation p_θ^T denotes the DDPM iterative sampling, $p_\theta^T(x_T, \mathbf{x}_0, j) = \mathbf{x}(j \cdot \delta\tau) = p_\theta(p_\theta(\dots p_\theta(x_T, \mathbf{x}_0, j)\dots), \mathbf{x}_0, j)$. In essence, p_θ^T maps any fluid state $\mathbf{x}(\tau)$ to $\mathbf{x}(\tau + j \cdot \delta\tau)$, without the need to estimate the intermediate states $\mathbf{x}(\tau + i \cdot \delta\tau)\ \forall i < j$, as typically required by classical numerical solvers to satisfy the Courant–Friedrichs–Lewy (CFL) convergence conditions (de Moura & Kubrusly, 2013). Although conditioning on j has been previously explored in the context of multi-parameter conditioning (Gupta & Brandstetter, 2023), our contribution emphasizes the use of j to facilitate flexible, parallelizable, and potentially more accurate diffusion sampling instead of next-step predictions.

Flexibility in predicting future states. By conditioning a surrogate model on j for $j \in \{0, \dots, \mathcal{T}\}$, we achieve two major benefits. First, the model can predict all possible intermediate states between $\tau = 0$ and $\tau = \mathcal{T} \cdot \delta\tau$. In comparison to methods such as ***DYffusion*** (Cachay et al., 2023) that require independent forecaster and temporal interpolator networks to achieve this task, our formulation enables a single network to predict the next timestep in addition to intermediate ones. Second, we are able to balance between the accuracy of the first-step prediction and error accumulation (Lienen et al., 2024). For instance, smaller j values would lead to first-step predictions with high accuracy; however, they would require longer rollout steps, leading to more error accumulation and vice versa. Thus, we will demonstrate in our experiments how an optimal value for j can lead to better accuracy than next-step sampling.

1.3 Related Work

DMs as flow surrogates. Regarding time-dependent autoregressive predictions,Yang & Sommer (2023) attempted to predict nonlinear fluid fields at specific points in time based on initial conditions. Kohl et al. (2024) introduced an autoregressive conditional diffusion model (ACDM) capable of predicting fluid states over extended time horizons while maintaining sample quality and temporal stability, and provided benchmark results on various datasets and against multiple baselines.

Expedited sampling. The slow sampling time of DMs is a major drawback, prompting extensive research to reduce the computational cost without compromising quality (Nichol & Dhariwal, 2021; Luhman & Luhman, 2021; Salimans & Ho, 2022). Song et al. (2021) presented Denoising Diffusion Implicit Models (DDIMs) that generalize DDPMs via a class of non-Markovian diffusion processes instead of the Markovian diffusion process of DDPMs, leading to shorter deterministic generative processes. By expressing DMs in a common framework known as elucidated DMs (EDMs), Karras et al. (2022) introduce a design space featuring separable design choices that can be optimized to attain expedited sampling with state-of-the-art accuracy. Truncated Diffusion Probabilistic Models (TDPMs) (Zheng et al., 2023) truncate the last steps of the forward diffusion process, leading to a shorter generative process starting from a hidden noisy distribution by leveraging an additional generative adversarial network (GAN)-based implicit generator to match the prior to the aggregated posterior.

2 Novel training and sampling approaches

2.1 TSM: Truncated Sampling Model

Motivation. An interesting phenomenon of DDPMs, particularly when trained with a linear β_t, is the ability to skip a small percentage (i.e., $\leq 20\%$) of the reverse diffusion process while maintaining the sampling quality (Nichol & Dhariwal, 2021). Furthermore, approaches have been devised to target a relevant range of noise levels during training by prioritizing intermediate noise levels (Karras et al., 2022; Choi et al., 2022), thereby enhancing the loss per noise level.

We re-visit these approaches with a new perspective: We truncate a significant part from the last steps of the reverse Markov chain with a high skip percentage s to reduce NFEs and focus the training on noise steps preceding the truncation. We refer to a model trained for a limited part of the diffusion process and sampled with truncated ancestral sampling as *Truncated Sampling Model (TSM)*. Focused training by restricting the sampling window for t parallels approaches by Karras et al. (2022) and Choi et al. (2022) to improve the loss per noise level (see Eq. 2), with our main objective to achieve enhanced sampling accuracy. Hence, we expect $s \gg 0$ to lead to better accuracy and reduced NFEs.

Sampling. Algorithm 1 summarizes the sampling procedure for conditional TSMs, with differences from ancestral sampling highlighted in blue. TSM sampling follows ancestral sampling until x_{t_s}, i.e., $p_\theta(x_{t_s:T}) := p(x_T) \prod_{t=t_s+1}^{T} p_\theta(x_{t-1}|x_t)$, where $t_s = \lfloor s \cdot T \rfloor$ and $s \in (0, 1]$. At $t = t_s$, instead of sampling $x_{t_s-1} \sim p_\theta(x_{t_s-1}|x_{t_s})$, we estimate $\hat{x}_0$ using the posterior mean $\mathbb{E}[\hat{x}_0|x_t; \theta]$, see Eq. 3.

Training. The TSM training procedure involves the choice of the hyperparameter s (skip percentage) and its use as a lower bound for sampling diffusion steps. Hence, the sole adjustment to the DDPM training algorithm from Ho et al. (2020) is defined as $t \sim \text{Uniform}(\{t_s, \dots, T\})$. Based on the skip percentage s, typically > 0.2 for more pronounced outcomes, a balance between NFEs, sampling accuracy, and stochasticity can be achieved. Extreme skip percentages (e.g., $s \approx 1$) are feasible, resulting in unprecedented high-accuracy single-step diffusion sampling, albeit at the cost of reduced stochasticity. Hence, relatively lower s values are optimal for problems of stochastic nature to enable learning all modes of the data distribution.

TSMs vs TDPMs. We highlight the main differences between TDPMs (Zheng et al., 2023) and our proposed TSMs. First, TDPMs apply the truncation to the last steps of the forward diffusion process, whereas TSM focuses on these last steps and conversely truncates the first steps. Second, TDPMs rely on an additional implicit generator network (GAN) to match the prior with the aggregated posterior. This implicit generator requires joint training with the DDPM, thus adding extra complexity to the training process. On the other hand, our training procedure closely resembles the straightforward DDPM training procedure and doesn't add complexity, thus rendering TSMs an appealing approach for enhancing sampling efficiency. Finally, while the sample quality of TDPMs is adversely impacted by large truncations, TSMs enable one-step inference with the same or improved sampling quality, relying on characteristics specific to fluid dynamics datasets as we explain in Section 3.

Algorithm 1 Truncated Ancestral Sampling

Require: ϵ_θ (TSM), s (skip percentage), c (condition)

$x_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, ..., t_s + 1$ **do**
  $z \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $z = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t, c) \right) + \sigma_t z$
end for
if $s > 0$ **then**
  $\hat{x}_0 = (x_{t_s} - \sqrt{1-\bar{\alpha}_{t_s}} \epsilon_\theta(x_{t_s}, t_s, c)) / \sqrt{\bar{\alpha}_{t_s}}$
return $\hat{x}_0$

Algorithm 2 Iterative Refinement

Require: ϵ_θ (DDPM model), x_{init} (initial state), $\gamma = \{t_r, \dots, t_e\}$ (refinement schedule)

$\hat{x}_0 \leftarrow x_{\text{init}}$
for each t in γ **do**
  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
  $x_t = \sqrt{\bar{\alpha}_t} \hat{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon$
  $\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\theta(x_t, t, c) \right)$
end for
return $\hat{x}_0$

2.2 IR: Iterative Refinement

Motivation. A fundamental characteristic of DMs is their iterative sampling process, which systematically reduces noise from a pure Gaussian noise x_T to a noise-free state x_0 over successive iterations. This ancestral sampling procedure is performed by gradually estimating all intermediate states $x_{T-1:1}$ until reaching a fully denoised state x_0, matching the training data distribution $q(x_0)$. Nevertheless, during training, a DDPM network learns to only predict the noise field at each noise level of a predetermined β_t independent of any adjacent states (see Eq. 2). Therefore, for a pre-trained DDPM model, various sampling methods can be employed without re-training, as long as they pertain to β_t originally used for training.

We leverage this property of DMs to introduce an intuitive refinement algorithm as a novel sampling method for DDPMs, which we refer to as *Iterative Refinement (IR)*. In IR sampling, we consider a much shorter noise schedule $\gamma = \{t_r, \dots, t_e\} \subset \{T, \dots, 1\}$ and interpret the different noise levels to essentially correspond to different levels of detail for a given state. We observe for any provided initial state x_{init} with a distribution similar to the training data, there exists a sequence γ that defines the minimal number of noise levels (or levels of detail) sufficient to augment the accuracy of x_{init}. Hence, we optimize γ to ensure that the accuracy of the final prediction closely matches all levels of detail present in the ground truth state x_0.

Sampling. Algorithm 2 summarizes the generative process for IR. Given an initial state x_{init} (assumed to be a noise-free, low-order approximation of x_0) and a refinement schedule γ, we iteratively apply forward diffusion followed by an estimation of the posterior mean to predict a series of gradually enhanced approximations of the noise-free state $\hat{x}_0^i, \forall i = 1, \ldots, N$, where $N = |\gamma|$, at distinct noise levels. Conversely, ancestral sampling $p_\theta(x_{t-1}|x_t)$ gradually removes noise for the same schedule (assuming $\gamma = \{T, \ldots, 1\}$) leading to intermediate states that are partially noisy. IR allows for a flexible choice of the length and distribution of γ to fit the problem under consideration and ensures higher accuracy of predictions using fewer NFEs compared to ancestral sampling. We also argue that x_{init} can be a (partially) noisy state obtained through truncation of ancestral sampling, for example, or even sampled from the prior distribution $q(x_T)$, further relaxing the computational overhead before IR sampling.

Relation to DDIMs. The general recursive sampling formula is defined by Song et al. (2021):

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{predicting } x_0} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{direction pointing to } x_t} + \sigma_t \epsilon_t, \tag{6}$$

with $\sigma_t = 0$ resulting in a deterministic forward process. One can see that Eq. 6 consists of two main components, a prediction of x_0 followed by a direction pointing to x_t. Using $\sigma_t = \sqrt{1-\bar{\alpha}_{t-1}}$ in the previous equation yields IR sampling as described in Algorithm 2, but but with the order of lines 4 and 6 reversed. For deterministic DDIMs, the predicted noise ϵ_θ in Eq. 6 is used to define the forward process instead of a randomly sampled Gaussian noise. This directly results from their choice of the mean function for the inference distribution $q(x_{t-1} \mid x_t, x_0)$ (refer to Song et al. (2021) for more details). Despite the similarity, our results demonstrate that IR consistently outperforms DDIM. We hypothesize that the stochastic nature of IR aids in rectifying errors incurred in earlier sampling steps. However, the dynamics of stochastic sampling are complicated in practice and might introduce additional errors (Karras et al., 2022). Consequently, this observation may not generalize to other datasets and domains.

3 Experiment: Transonic Flow

We consider a two-dimensional (2D) fluid flow scenario, a compressible transonic flow over a cylinder on a 128×64 grid including u_x, u_y, p, and ρ flow fields (`Tra`). This dataset is also a benchmark dataset by Kohl et al. (2024). The case becomes particularly challenging at high Mach Numbers *Ma* due to the presence of shock waves. Interpolation `int` and extrapolation `ext` datasets for evaluation involve $R = 60$ timesteps with *Ma* $\in \{0.66, 0.67, 0.68\}$ and *Ma* $\in \{0.50, 0.51, 0.52\}$, respectively. Temporal stability is tested on `long` with $R = 240$. The prediction stride is defined as $j \in \{0, 1, \ldots, 10\}$, i.e., $\mathcal{T} = 10$, and is provided as an additional input channel to the network. Details regarding training and diffusion-related hyperparameters for this test case can be found in Appendix C.

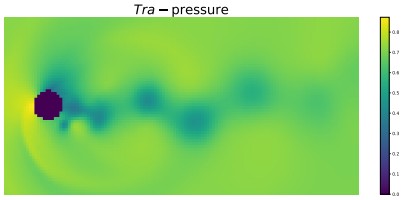


Figure 1: Pressure field for transonic flow.

Evaluation metrics. We evaluate the accuracy of DMs against the ground truth data through various metrics. We consider the temporal-average MSE (TA-MSE), turbulent kinetic energy spectrum (TKE) and temporal correlation $\boldsymbol{\rho}$ for the absolute velocity $|u|$, and temporal stability for significantly long rollout horizons. For results obtained with $j > 1$, we restrict the analysis in this section to sampling with the stride j without sampling the intermediate states (i.e., we only make $\lceil R/j \rceil$ predictions to reach the target timestep). Details regarding the sampling of the intermediate states are presented in Appendix D.

We focus our evaluation on the speedup obtained via a reduction in NFEs, relying on the fact that we utilize the same architectures with almost the same number of parameters for all models. Training and sampling were carried out using NVIDIA GeForce RTX 2080 Ti GPU. This section is solely

dedicated to quantitative analysis of the top-performing models. We provide our comprehensive set of results as well as other baselines (such as ResNet, Fourier neural operators (FNO), PDE-Refiner, latent-space transformers (TF)) in Appendix E.1. Qualitative samples are provided in Appendix E.2. Moreover, we compare our methods against DDPMs, DDIMs, and EDMs and exclude other expedited sampling methods outlined in Section 1, as distillation techniques require extensive retraining without yielding significant accuracy improvements over the teacher model.

Figure 2: Comparison of DDIM and IR (left) and scaled turbulent kinetic energy spectrum (TKE, right) for different methods.

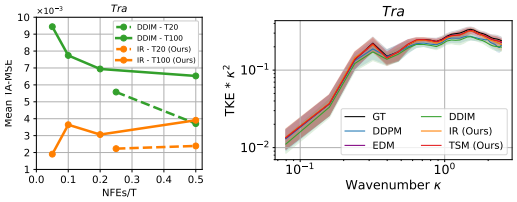


Table 1: TA-MSE values for the top performing models. Standard deviation values are estimated over all timesteps and multiple samples.

Tra (10^{-3})			
Model	NFEs	ext	int
ACDM T20 (Kohl et al., 2024)	20	2.3 ± 1.4	2.7 ± 2.1
UNet$_{ut}$ (Kohl et al., 2024)	1	1.6 ± 0.7	1.5 ± 1.5
DDPM T100	100	3.0 ± 2.7	4.1 ± 3.7
EDM - Deterministic Euler	4	$\mathbf{1.3} \pm \mathbf{1.3}$	$\mathbf{1.1} \pm \mathbf{1.0}$
DDIM T20	10	3.2 ± 2.7	4.2 ± 3.9
IR T100 - $\mathcal{N}$ γ_1 (Ours)	5	1.6 ± 1.3	2.0 ± 1.7
TSM T100 s =1 (Ours)	1	$\mathbf{1.2} \pm \mathbf{1.1}$	1.5 ± 1.5

We present TA-MSE results in Table 1, where UNet$_{ut}$ refers to UNet with unrolled training. In IR, T100/T80 is the base model, $\mathcal{N}$ is Gaussian noise used as x_{init}, and the absence of γ implies linear sampling steps. Using $j = 1$, our approaches significantly surpass DDPMs and DDIMs accuracy while requiring only a fraction of the NFEs, as low as single-step inference. Our approaches transcend the benchmark ACDM (Kohl et al., 2024) with $\times 4$ and $\times 20$ (i.e., single-step inference) speedup for IR and TSM, respectively, while preserving all turbulent scales present in the flow (see Fig.2, right). Additionally, the single-step TSM outperforms the best baseline from Kohl et al. (2024), while IR yields marginally lower accuracy. The most accurate result from EDM is reported here, with comprehensive tests for various samplers detailed in Fig. 5 (Appendix E.1), demonstrating marginal increase in accuracy compared to TSM, albeit requiring $4\times$ the NFEs.

Comparison of IR and DDIM. In Fig. 2 (left), we compare IR against DDIMs with varying NFEs for identical sampling schedules. Besides IR's consistency in transcending DDIMs across various base DDPMs, we notice a discernible correlation between NFEs and accuracy for DDIMs, whereas IR does not manifest a noticeable trend. The stochastic nature of IR makes it difficult to find clear relations across different datasets and NFEs as it is case-by-case tuned (Karras et al., 2022).

Limitations. In TSMs, since the training was focused on a limited part of the diffusion process, they exhibit inflexible sampling; neither DDIMs nor IR could be applied unless the sampling steps $t = \{t_{\text{start}}, \ldots, t_{\text{end}}\} \subset \{T, \ldots, t_s\}$, which is not the case for high s values. Also, while IR and TSMs enable single- and few-step sampling with increased accuracy compared to ancestral sampling, they are not guaranteed to supersede the accuracy of deterministic baselines. However, we have empirically shown that our methods improve over DDPMs, DDIMs, and EDMs in terms of speed and/or accuracy for the transonic flow experiment.

4 Conclusion

We have introduced two novel training and sampling approaches to enable single- and few-step sampling of DDPMs without compromising inference quality. Our first contribution is a Truncated Sampling Model (TSM), capable of achieving single-step inference while maintaining or even enhancing accuracy through early truncation of the reverse process. Additionally, our second contribution, Iterative Refinement (IR), targets pre-trained DDPMs by formulating the sampling process as a refinement endeavor to facilitate high-fidelity inference with reduced NFEs compared to existing sampling methods, such as DDIMs. We have showcased the efficacy of TSMs and IR in minimizing the disparity between DDPMs and deterministic baselines for a challenging transonic flow experiment. We believe our proposed methods significantly enhance sampling speed and quality in fluid dynamics simulations, and we posit their potential applicability in other domains for which diffusion models are considered state-of-the-art.

REFERENCES

Salva Rühling Cachay, Bo Zhao, Hailey James, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

K. W. Cheah, T. S. Lee, S. H. Winoto, and Z. M. Zhao. Numerical flow simulation in a centrifugal pump at design and off-design conditions. *International Journal of Rotating Machinery*, 2007:1–8, 2007. ISSN 1023-621X. doi: 10.1155/2007/83641.

J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Carlos A. de Moura and Carlos S. Kubrusly. *The Courant–Friedrichs–Lewy (CFL) Condition*. Birkhäuser Boston, Boston, 2013. ISBN 978-0-8176-8393-1. doi: 10.1007/978-0-8176-8394-8.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. ISSN 01621459. URL http://www.jstor.org/stable/23239562.

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

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Dietrich Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Benjamin Holzschuh, Simona Vegetti, and Nils Thuerey. Solving inverse physics problems with score matching. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation, 2024.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. 2024.

Yifan Li, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Diffusion models for non-autoregressive text generation: A survey. In Peter Stone and Edith Elkind (eds.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6692–6701, California, 2023. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/750.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3d flow simulation. In *The Twelfth International Conference on Learning Representations*, 2024.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Qiang Liu and Nils Thuerey. Uncertainty-aware surrogate models for airfoil flow simulations with denoising diffusion probabilistic models. *AIAA Journal*, pp. 1–22, 2024. ISSN 0001-1452. doi: 10.2514/1.J063440.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. URL http://arxiv.org/pdf/2201.09865v4.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed, 2021.

Haomiao Ni, Changhao Shi, Kai Li, Sharon X. Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18444–18455. IEEE, 2023. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.01769.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 2012. ISBN 9780521591256. doi: 10.1017/CBO9780511840531.

Parakram Pyakurel, James H. VanZwieten, Manhar Dhanak, and Nikolaos I. Xiros. Numerical modeling of turbulence and its effect on ocean current turbines. *International Journal of Marine Energy*, 17:84–97, 2017. ISSN 2214-1669. doi: 10.1016/j.ijome.2017.01.001. URL https://www.sciencedirect.com/science/article/pii/S2214166917300012.

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

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.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Peter P. Sullivan and James C. McWilliams. Oceanic frontal turbulence. *Journal of Physical Oceanography*, 54(2):333–358, 2024. ISSN 0022-3670. doi: 10.1175/JPO-D-23-0033.1.

E. G. Tulapurkara. Turbulence models for the computation of flow past airplanes. *Progress in Aerospace Sciences*, 33(1-2):71–165, 1997. ISSN 03760421. doi: 10.1016/S0376-0421(96)00002-4.

Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.

Ricardo Vinuesa and Steven L. Brunton. Enhancing computational fluid dynamics with machine learning. *Nature computational science*, 2(6):358–366, 2022. doi: 10.1038/s43588-022-00264-7.

Raffaele Volpe, Valérie Ferrand, Arthur Da Silva, and Luis Le Moyne. Forces and flow structures evolution on a car body in a sudden crosswind. *Journal of Wind Engineering and Industrial Aerodynamics*, 128:114–125, 2014. ISSN 01676105. doi: 10.1016/j.jweia.2014.03.006.

Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7589–7599. IEEE, 2023. ISBN 979-8-3503-0718-4. doi: 10.1109/ICCV51070.2023.00701.

Gefan Yang and Stefan Sommer. A denoising diffusion model for fluid field prediction. 2023.

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. In *The Eleventh International Conference on Learning Representations*, 2023.

A EXPERIMENTAL SETUP

We summarize the details for our dataset in Table 2. For detailed information regarding the generation of the datasets, please refer to the corresponding paper (Kohl et al., 2024). For the generation, only a single parameter is varied, namely *Re*, and is provided as an input channel to all models. We consider velocity (u_x and u_y), pressure (p), and density (ρ) fields as input and output fields. Additionally, the models are conditioned on the prediction stride j, provided as an additional input channel to the network.

Table 2: Parameter values for all datasets.

Dataset	Param	Purpose	Values	Sequences per Param	Total Sequences	R	Total Frames
`Tra` (128×64)	Ma	training	$\{0.53, 0.54, \ldots, 0.63\} \cup \{0.69, 0.70, \ldots, 0.90\}$	1	33	501	16533
		test	`ext`: $\{0.50, 0.51, 0.52\}$	2	6	60	360
			`int`: $\{0.66, 0.67, 0.68\}$	2	6	60	360
			`long`: $\{0.64, 0.65\}$	2	4	240	960

B REFINEMENT SCHEDULES FOR IR

The most critical component of IR sampling is γ, balancing between inference quality and NFEs. Hence, an optimized γ should consider the initial state x_{init}, the model's accuracy L_{t-1} at each noise step (defined as in Eq. 2), and the nature of the problem. All our proposed γ schedules are optimized using a greedy algorithm, which we found to easily lead to highly satisfactory results with minimal effort and computational cost. A direct consequence of using this greedy algorithm is that the final output $\hat{x}_0^N$ provides a better approximation of x_0 than x_{init} and all preceding approximations:

$$\mathbb{E}[\|\hat{x}_0^N - x_0\|_2] < \mathbb{E}[\|\hat{x}_0^{N-1} - x_0\|_2] < \ldots < \mathbb{E}[\|\hat{x}_0^1 - x_0\|_2] < \mathbb{E}[\|x_{\text{init}} - x_0\|_2]. \tag{7}$$

We believe that better results could be achieved through more sophisticated optimization of γ, although this will incur additional training overhead.

We start the optimization by choosing the max possible length of γ (N) and a value for the first refinement step (K). If K is not provided, the default starting value is T (i.e., the number of noise steps of the pre-trained DDPM). For the first step, we consider K possible options for refinement, i.e., $\{x \in \mathbb{Z} \mid 1 \leq x \leq K\}$. As long as the validation loss L is decreasing, we keep looping over all K values. However, if the loss does not improve after tol steps (i.e., a tolerance set by the user to achieve early stopping for non-promising optimizations), we continue to the next refinement step. The starting point for the next timestep is defined to be one step smaller the current optimized step to further reduce the computational cost. If for the current step no value was found to reduce the current best loss L_{best}, the entire optimization is terminated, leading to the loss in Eq. 7, even if $|\gamma| < N$. These restrictions are imposed to reduce the computational cost of optimization, though they might lead to non-optimal results. More expensive gradient-based and gradient-free optimization algorithms offer the potential to yield better results; however, we believe that the accuracy gain would not compensate for the concomitant computational cost.

Using the greedy algorithm, the optimization schedules utilized in all our experiments are summarized as follows (and visualized in Figure 3):

$$\begin{aligned}
\gamma_{1,i} &= 0.805 - 0.2i, \quad \forall i = 0, 1, 2, 3, 4, \\
\gamma_{2,i} &= 0.805 - 0.1i, \quad \forall i = 0, 1, 2, \ldots, 8, \\
\gamma_{3,i} &= 0.655 - 0.05i, \quad \forall i = 0, 1, 2, \ldots, 13, \\
\gamma_{4,i} &= 0.905 - 0.05i, \quad \forall i = 0, 1, 2, \ldots, 18.
\end{aligned} \tag{8}$$

Also, $\gamma_5 = \{1/T\}$. When no γ schedule is provided, we use a linear sampling schedule by default:

$$\gamma_{\text{linear}}(N) = \{x \in \mathbb{N} \mid 0 \leq x < T, \text{for } N \in \mathbb{N}\}. \tag{9}$$

Algorithm 3 Greedy optimization of γ with early stopping

Require: N (Max length of γ), K (starting value)
Require: tol (tolerance value), x_{init} (Initial state, ϵ_θ (pre-trained DDPM)

$\gamma \leftarrow \{\}$
for $i = 1$ to N **do**
  $L_{\text{best}} \leftarrow \infty$
  $t_{\text{opt}} \leftarrow -1$
  counter $\leftarrow 0$
  for $j = K$ to 1 **do** ▷ Go over all possible values.
    $\gamma_{\text{temp}} \leftarrow$ append(γ, j) ▷ Update γ_{temp} by appending step j to γ (unchanged).
    $L =$ recursive_sampling$(x_{\text{init}}, \gamma_{\text{temp}}, \epsilon_\theta)$ ▷ Evaluate γ_{temp} on the validation dataset.
    if $L < L_{\text{best}}$ **then** ▷ Update the current step if accuracy is higher.
      $L_{\text{best}} \leftarrow L$
      $t_{\text{opt}} \leftarrow j$
      counter $\leftarrow 0$
    else
      counter $\leftarrow$ counter $+1$
      if counter $> tol$ **then**
        break ▷ Terminate current step: failed to improve L_{best} after tol trials.
      end if
    end if
  end for
  if $t_{\text{opt}} = -1$ **then**
    break ▷ Terminate optimization: no improvement found.
  end if
  append(γ, t_{opt})
  $K \leftarrow t_{\text{opt}} - 1$ ▷ Consider only smaller values for the next refinement step.
end for
return γ

C SIMULATION PARAMETERS

The training hyperparameters are presented in Table 3. Our training is limited to the L2 loss for DDPMs, TSMs, EDMs and any baselines. All models were trained with max overlap (i.e., 10 frames overlapping) but with early stop once the validation loss stabilizes, typically much earlier than the number of epochs presented in Table 3.

The utilized network architecture and the diffusion-related hyperparameters are summarized in Table 4. We use the exact same network architecture as provided in the benchmark paper for fair comparison.

EDM training and sampling. Our implementation of EDMs is based on the work by Karras et al. (2022). We implement their Algorithm 1 (deterministic sampler) and Algorithm 2 (stochastic sampler) using 1st-order Euler and 2nd-order Heun's methods with design choices from the last column of Table 1 (Karras et al., 2022). Additionally, we consider preconditioning and a weighted loss function using the parameters recommended by the authors. For each case, we ran the model using different combination of deterministic/stochastic samplers and Euler's/Heun's methods. For the

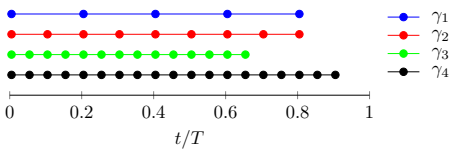


Figure 3: Illustration for the refinement schedules defined in Equation 8. Each node represents a noise step. All schedules start from the node with the largest t/T.

Table 3: Summary of the training hyperparameters

Parameter	`Tra`
Data size (frames)	16533
Resolution	128×64
Batch size	32
Epochs	3000
Learning rate	10^{-4}
Learning rate schedule	None
Optimizer	AdamW
Weight decay	10^{-2}
EMA decay	0.999
Early stop?	Yes

stochastic sampler, the parameters $\{S_{churn}, S_{tmin}, S_{tmax}, S_{noise}\}$ were non-comprehensively tuned to attain the best possible results. We found the values 10, 0, ∞, and 1 for $S_{churn}, S_{tmin}, S_{tmax}$, and S_{noise}, respectively, to generally yield satisfactory results. Noteworthy, EDMs introduce more hyperparameters for tuning compared to DDPMs, DDIMs, and our approaches.

D PARALLEL RECURSIVE SAMPLING

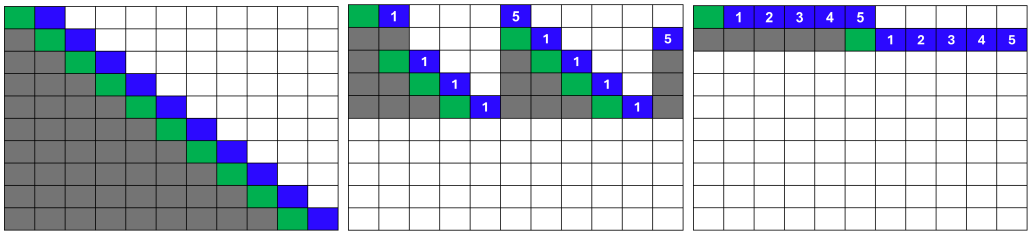
Speedup through parallel sampling. One of the benefits of conditioning a surrogate model on the prediction stride j is to enable parallel sampling of transient test cases. To date, related work concerned with physics-based simulations of transient nature focus on autoregressive sampling only. Nonetheless, we posit that our formulation of the problem definition (see Section 1.2), combined with our proposed improvement approaches (i.e., TSMs and IR), would enable parallel sampling with reduced inference time for the same computational budget.

Inspired by the progress in video diffusion models (Ho et al.; 2022; Ni et al., 2023; Wu et al., 2023; Harvey et al., 2022), we present two examples of sampling schemes that allow for parallel sampling of diffusion models in transient test cases. As demonstrated in Fig. 4 for a sample problem with $R = 10$ states, while autoregressive sampling (see Fig. 4a) requires 10 successive predictions to sample the entire simulation, it only requires 5 steps for parallel sampling with a batch size n = 2 (cf. Fig. 4b). With n = 5, max parallelism can be achieved to sample the entire simulation in 2 steps only (cf. Fig. 4c). This corresponds to a speedup factor $\approx n$, assuming that $\mathcal{T} \geq n$. For the two parallel sampling schemes, we assume that sampling with $j = 5$ will lead to lower TA-MSE compared to sampling with $j = 1$ for the entire simulation.

Higher accuracy when sampling intermediate timesteps. For models that allow sampling using $j > 1$, we not only enable flexible parallel sampling to reduce inference time as discussed, but also we improve the overall prediction accuracy. Typically when intermediate states are sampled with a stride j_{sec} smaller than the primary sampling stride j, we gain (marginally) higher overall accuracy compared to ignoring intermediate states for sampling (using j only). In Table 5, we demonstrate

Table 4: Network architecture and diffusion-related hyperparameters. c_{in} and c_{out} refer to the network's number of input and output channels, respectively. β_{start} and β_{end} of the noise schedule β_t are similar to Ho et al. (2020) but are scaled by a factor depending on the chosen noise steps T, as defined in Nichol & Dhariwal (2021) and Kohl et al. (2024).

Parameter	`Tra`
Architecture	Kohl et al. (2024)
c_{in}	10
c_{out}	4
β_{start}	$10^{-4} \cdot$ (500/T)
β_{end}	$0.02 \cdot$ (500/T)
Schedule	Linear



(a) Autoregressive sampling ($n = 1$). (b) Parallel sampling with $n = 2$. (c) Max parallelism with $n = 5$.

Figure 4: Parallel sampling of a time-dependent simulation with $R = 10$, enabled through models conditioned on the prediction stride j. n is the batch size used for sampling. Each row represents the state of the entire simulation, where the states evolve in time going downwards through the rows. Numbers within blue cells (cells to predict) correspond to the value of j used to sample from the nearest initial condition (green cell). Gray cells are states already sampled and are not needed for current or subsequent predictions. For (b) and (c), we assume that $j = 5$ leads to highest prediction accuracy compared to other j values.

this by sampling the best IR and TSM models. We see marginal improvement in the mean TA-MSE for small j; however, the improvement is more pronounced for higher j values. This is primarily attributed to the argument discussed by Lienen et al. (2024) pertaining to higher accuracy of predicting with small timestep sizes for a short rollout horizon ($R = j$ for sampling the intermediate states) before error accumulation predominates, leading to lower accuracy for longer rollout horizons. This greatly motivates future research in flexible sampling of time-dependent physics-based simulations similar to Harvey et al. (2022).

Table 5: Mean TA-MSE results for predictions with $j > 1$, and with and without sampling the intermediate states. For the former, the model names include "**(all)**". TA-MSE is averaged over `ext` and `int` regions. Standard deviation values are calculated over all timesteps, both regions, and multiple samples. The results show consistent improvement when sampling the intermediate states with $j_{\text{sec}} = 1$ over sampling with the primary prediction stride j only.

	j				
Model	2	4	6	8	10
	`Tra` (10^{-3})				
IR T100 - $\mathcal{N}$ γ_1	2.06 ± 1.91	2.73 ± 2.61	3.15 ± 2.93	4.18 ± 3.39	3.54 ± 2.98
IR T100 - $\mathcal{N}$ γ_1 **(all)**	1.94 ± 1.82	2.33 ± 2.27	2.75 ± 2.83	3.06 ± 2.82	2.72 ± 3.39
TSM T100 s =1	1.46 ± 1.55	1.78 ± 1.82	2.17 ± 2.11	2.35 ± 2.07	2.32 ± 2.12
TSM T100 s =1 **(all)**	1.36 ± 1.47	1.59 ± 1.68	1.93 ± 2.22	1.74 ± 1.74	2.09 ± 3.22

E Full results of experiment

In this section, we present our full results in addition to the top performing models presented in Section 3. The results are split into quantitative and qualitative sections.

E.1 Quantitative results

The results for using $j = 1$ and optimal $j > 1$ are presented in Table 6. For TA-MSE values obtained with $j = 1$, several insights can be observed. First, in comparison to ACDM (Kohl et al., 2024), we are able to provide more accurate results through IR and TSMs with single-step and few-step sampling (NFEs ≤ 5), even though our problem formulation (defined in Section 1) is much harder compared to ACDM for the same utilized architecture. Noteworthy, the ACDM results are obtained by conditioning over two previous timesteps, whereas all our models are conditioned on the previous timestep only. Second, various TSMs are capable of improving over all considered baselines with and without advanced training mechanisms, with the most accurate model being a single-step solver. IR only improves over the base DDPMs and surpasses DDIMs by a significant margin for both similar and optimized sampling schedules. In our experiments, we focused on using random Gaussian noise

$\mathcal{N}(0, \mathbf{I})$ for the initial state x_{init} to reduce the computational overhead before starting the refinement process and also because we found in early experiments that this setup yields slightly more accurate results than starting from a prediction obtained through truncated sampling of the base DDPM. Third, IR and DDIM were able to provide noise-free results even when the base DDPM produces noisy output (e.g. DDPM T20). Fourth, in comparison to EDMs, we observe that although using a deterministic Euler solver yields the best results for this case, with a negligible margin over TSM, it requires 4 NFEs. In contrast, TSM is the only model that enables single-step, high-fidelity inference. Note that we evaluate several solvers for EDM (see Fig. 5), but we only report the optimal settings for each solver in Table 6. Finally, we note that PDE-Refiner performs suboptimally with a significant gap compared to other baselines. The model was found to be highly sensitive to its key hyperparameters (Kohl et al., 2024), rendering the hyperparameter tuning for this model challenging and ultimately resulting in suboptimal performance.

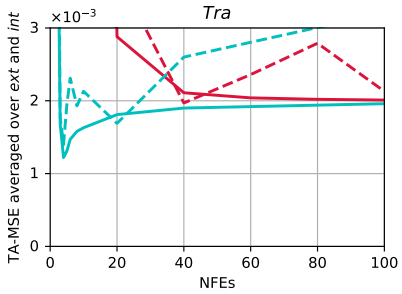


Figure 5: Evaluating different combinations of EDM samplers for the `Tra` test case. Standard deviation regions are omitted for clarity.

For $j > 1$, we observe substantial improvements in accuracy, even for low-fidelity models, except for EDMs. Suboptimal few-steps DDIMs are able to achieve comparable accuracy to the best baselines, which is also true for IR and TSMs. Noteworthy, the numbers reported for $j > 1$ are limited to sampling with j only without considering the intermediate states. We show in Appendix D that sampling intermediate states could further enhance the overall accuracy, particularly for large j.

Temporal analysis. In Fig. 6, we see how the $|u|$ correlation to the ground truth evolves over time for the most accurate models, demonstrating consistency with the TA-MSE metric. In addition, we evaluate the temporal stability of these top-performing models on the $\texttt{Tra}_{\texttt{long}}$ dataset, which has much longer sequence length (i.e., $R = 240$ instead of 60), to ensure that the models' predictions remain physically consistent even after the predicted trajectories have significantly deviated from the ground truth. In Fig. 7, we compare the time evolution of the stability parameter, defined as the rate of change of each flow field $\mathbf{x}$, by evaluating $||(\mathbf{x}_r - \mathbf{x}_{r-1})/\Delta\tau||_1$. Compared to the ground truth results, we see that stability parameter remains bounded within the range $[0.015, 0.02]$ for both the ground truth and all models, except for a low-fidelity PDE-Refiner model that produces non-physical predictions beyond $r \approx 200$. While DDIM, DDPM, and IR models slightly exceed the lower bound, they maintain temporal stability.

E.2 Qualitative results

We provide sample results in this section, see Fig. 8.

Table 6: TA-MSE (10^{-3}) values of the `Tra` test case for $j = 1$ (left) and the best results using the optimum j (right). UNet_{tn} and UNet_{ut} refer to a UNet with training noise and unrolled training, respectively. The reader is referred to Kohl et al. (2024) for a full description of the baselines reported. Lowest TA-MSE values for each group of models are highlighted in **bold**. Standard deviation values are obtained over all timesteps and multiple samples. *Model produces noisy output.

		$j = 1$			optimum j	
Model	NFEs	`ext`	`int`	j	`ext`	`int`
Kohl et al. (2024)						
ACDM	20	2.3 ± 1.4	2.7 ± 2.1			
UNet_{tn}	1	$\mathbf{1.4 \pm 0.8}$	1.8 ± 1.1			
UNet_{ut}	1	1.6 ± 0.7	$\mathbf{1.5 \pm 1.5}$			
$\text{ResNet}_{\text{dil.}}$	1	1.7 ± 1.0	1.7 ± 1.4			
FNO_{16}	1	4.8 ± 1.2	5.5 ± 2.6			
TF_{Enc}	1	3.3 ± 1.2	6.2 ± 4.2			
PDE-Refiner	4	5.4 ± 2.1	7.1 ± 2.1			
DDPM						
T20*	20	3.5 ± 3.1	$\mathbf{2.6 \pm 2.1}$	4	$\mathbf{1.6 \pm 1.3}$	$\mathbf{2.1 \pm 2.0}$
T100	100	$\mathbf{3.0 \pm 2.7}$	4.1 ± 3.7	5	1.9 ± 1.8	3.1 ± 2.7
EDM						
Euler - Deterministic	4	$\mathbf{1.3 \pm 1.3}$	$\mathbf{1.1 \pm 1.0}$	1	$\mathbf{1.3 \pm 1.3}$	$\mathbf{1.1 \pm 1.0}$
Stochastic	4	1.7 ± 1.7	$\mathbf{1.1 \pm 1.0}$	1	1.7 ± 1.7	$\mathbf{1.1 \pm 1.0}$
Heun - Deterministic	40	2.7 ± 2.7	1.6 ± 1.4	2	2.2 ± 2.1	$\mathbf{1.2 \pm 1.1}$
Stochastic	40	2.0 ± 2.2	2.0 ± 1.8	2	1.8 ± 2.0	1.8 ± 1.6
DDIM						
T20	5	4.4 ± 3.3	6.7 ± 5.4	4	1.9 ± 1.7	2.5 ± 2.2
	10	$\mathbf{3.2 \pm 2.7}$	$\mathbf{4.2 \pm 3.9}$	4	$\mathbf{1.4 \pm 1.3}$	$\mathbf{1.8 \pm 1.7}$
T100	5	7.8 ± 5.2	6.1 ± 4.3	10	2.8 ± 1.9	4.1 ± 3.0
	10	8.8 ± 5.5	6.7 ± 4.3	10	3.3 ± 2.2	4.5 ± 3.0
	20	7.8 ± 5.2	6.1 ± 4.3	10	2.8 ± 1.9	4.1 ± 3.0
	50	7.1 ± 4.7	6.0 ± 4.4	9	3.2 ± 2.2	3.3 ± 2.3
IR (Ours)						
T20	5	2.8 ± 2.9	1.7 ± 1.7	4	1.3 ± 1.4	1.5 ± 1.5
	10	3.2 ± 2.9	1.6 ± 1.5	4	$\mathbf{1.2 \pm 1.1}$	1.6 ± 1.6
T100	5	1.7 ± 1.4	2.1 ± 1.8	1	1.7 ± 1.4	2.1 ± 1.8
	10	3.8 ± 3.2	3.5 ± 3.3	2	2.6 ± 2.3	2.7 ± 2.4
	20	3.1 ± 3.1	3.0 ± 2.9	5	1.9 ± 1.7	3.0 ± 2.8
	50	4.4 ± 4.0	3.5 ± 3.7	5	1.7 ± 1.5	4.1 ± 3.8
T20, γ_1	5	2.3 ± 2.3	2.0 ± 1.9	4	1.3 ± 1.4	1.7 ± 1.8
γ_2	9	3.1 ± 2.9	1.6 ± 1.6	4	1.3 ± 1.2	1.7 ± 1.7
γ_3	14	1.7 ± 1.8	1.7 ± 1.8	3	1.7 ± 1.7	1.5 ± 1.6
γ_4	19	2.4 ± 2.6	$\mathbf{1.2 \pm 1.1}$	3	1.3 ± 1.3	$\mathbf{1.4 \pm 1.4}$
T100, γ_1	5	$\mathbf{1.6 \pm 1.3}$	2.0 ± 1.7	1	1.6 ± 1.3	2.0 ± 1.7
γ_2	9	3.7 ± 3.0	2.7 ± 2.4	2	2.7 ± 2.4	2.7 ± 2.4
γ_3	14	3.2 ± 3.0	2.9 ± 2.8	2	2.5 ± 2.8	2.7 ± 2.5
γ_4	19	2.6 ± 2.6	3.1 ± 2.9	2	2.2 ± 2.4	2.6 ± 2.3
TSM (Ours)						
T15, $s = 0.25$	11	1.8 ± 1.8	1.7 ± 1.5	2	$\mathbf{1.2 \pm 1.2}$	$\mathbf{1.5 \pm 1.6}$
$s = 0.5$	7	2.1 ± 2.2	$\mathbf{1.2 \pm 1.1}$	3	$\mathbf{1.2 \pm 1.1}$	1.6 ± 1.4
$s = 0.75$	3	1.4 ± 1.4	2.4 ± 2.3	6	1.4 ± 1.1	1.6 ± 1.4
$s = 1$	1	1.3 ± 1.2	1.6 ± 1.5	2	$\mathbf{1.2 \pm 1.1}$	1.6 ± 1.6
T100, $s = 1$	1	$\mathbf{1.2 \pm 1.1}$	1.5 ± 1.5	1	$\mathbf{1.2 \pm 1.1}$	$\mathbf{1.5 \pm 1.5}$

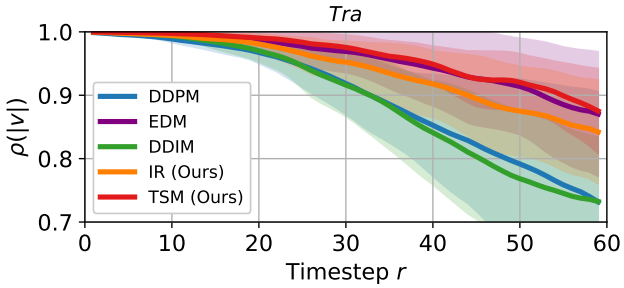


Figure 6: `Tra` absolute velocity $|u|$ correlation to ground truth using $j = 1$ for the models reported in Table 1. Values are averaged over `ext` and `int` regions and the shaded area represents the standard deviation from both regions and multiple samples.

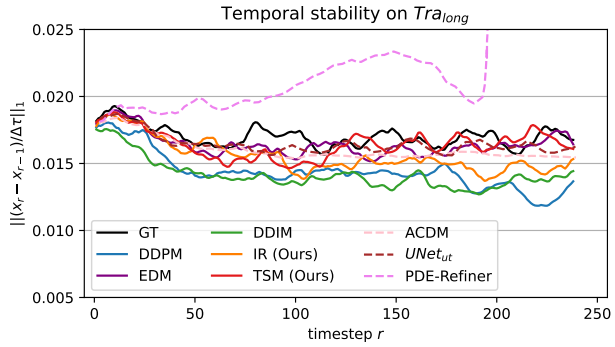
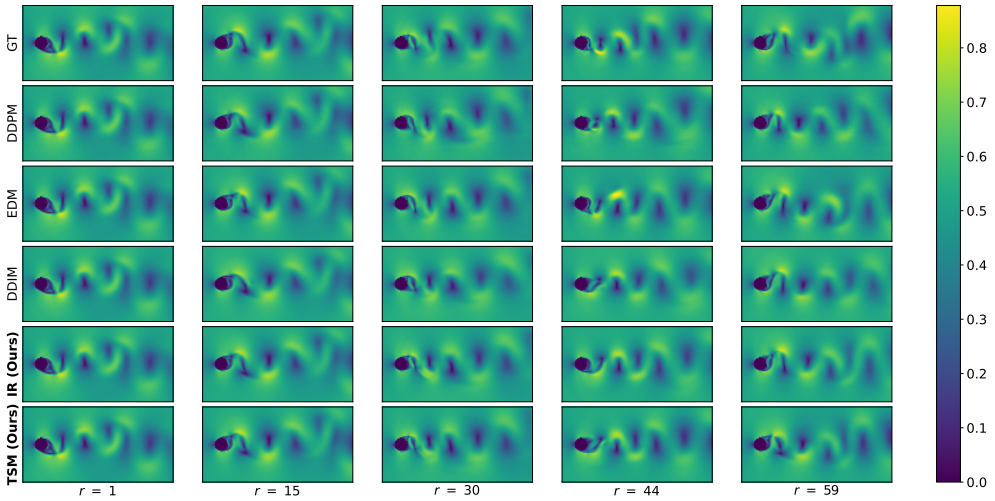
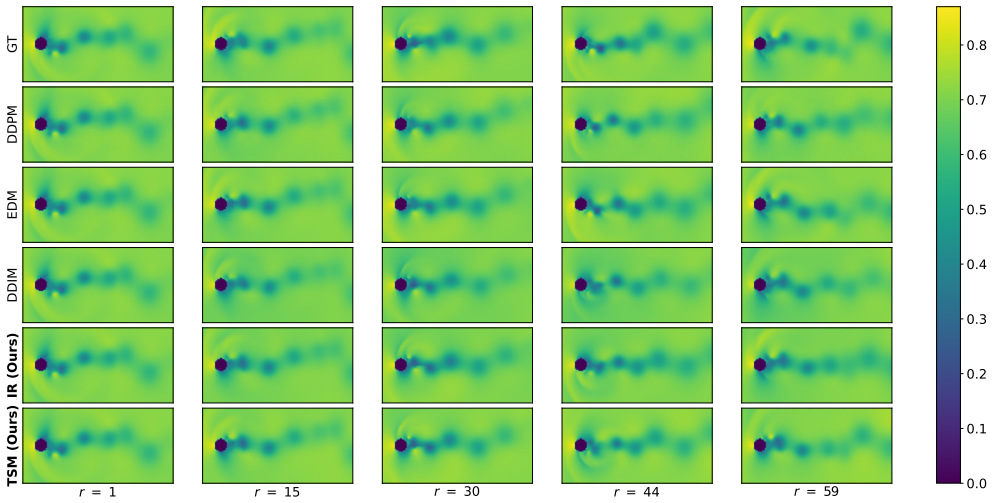


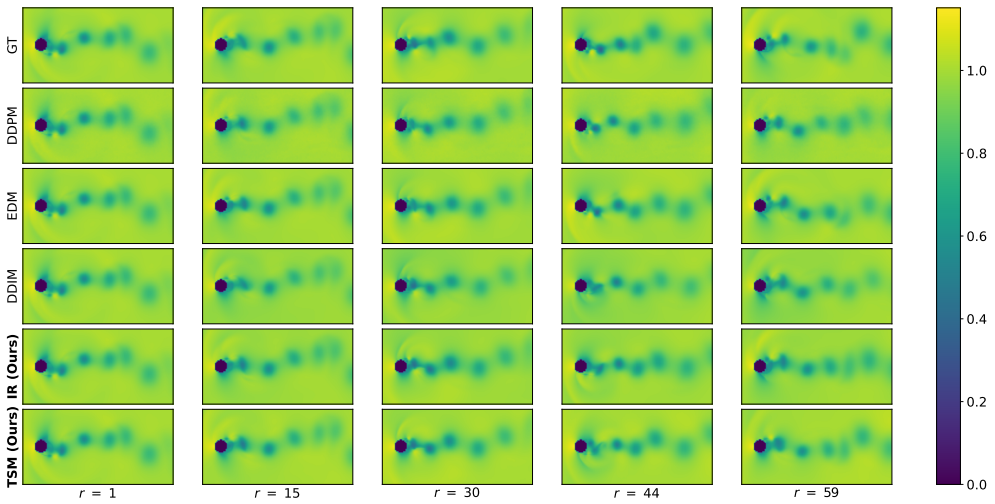
Figure 7: Time evolution for the temporal stability parameter between consecutive timesteps for top-performing models (see Table 1) compared to the ground truth simulation (GT). Dashed lines are obtained from Kohl et al. (2024). Standard deviation regions are omitted for clarity.



(a) $|u|$



(b) p



(c) ρ

Figure 8: Sample predictions of the different flow fields from the $\texttt{Tra}_{\texttt{ext}}$ test case with $Ma = 0.52$. Results are based on the top performing models presented in Table 1.