

SynBoost: A Synergistic Framework for Fast Sampling of Diffusion Models

Hu Yu, Hao Luo, Xueyang Fu, Jie Huang, Fan Wang, Feng Zhao, *IEEE, Member*

Abstract—Diffusion probabilistic models (DPMs) have demonstrated remarkable success in visual generation. However, their iterative sampling mechanism results in slow inference speeds. While reducing sampling steps offers an intuitive acceleration strategy, it introduces significant discretization error. Existing fast samplers have made substantial progress in mitigating this error through high-order solvers, yet further optimization appears constrained. This limitation prompts a critical question: can sampling efficiency be advanced beyond current paradigms? In this work, we re-examine the composition of sampling errors and identify two distinct components: the well-studied discretization error and the under-explored approximation error. By implementing a dual-error disentanglement strategy, we elucidate the dynamic interplay between these error types across sampling steps. Building on this empirical analysis, we propose SynBoost, a unified and training-free acceleration framework that simultaneously addresses both error sources to minimize total sampling error. Concretely, it mitigates approximation error by blending the current noise estimation with a more accurate prediction from a larger timestep. SynBoost seamlessly integrates with existing samplers, substantially enhancing their speed and output quality, particularly in regimes with extremely few steps. We validate our framework through extensive experiments across unconditional and conditional generation tasks, encompassing both pixel-space and latent-space DPMs.

Index Terms—Diffusion model, inference acceleration, error decomposition.

I. INTRODUCTION

Diffusion probabilistic models (DPMs) [1], [2], [3] have demonstrated impressive success across a broad spectrum of tasks, including image synthesis [4], [5], [6], [7], image editing [8], video generation [9], [10], voice synthesis [11], etc. Compared with other generative models such as GANs [12] and VAEs [13], DPMs not only exhibit superior sample quality but also benefit from a more robust training methodology and an advanced guided sampling technique. However, the inference of DPMs usually requires multiple model evaluations (NFEs), which hinders their practical deployment.

Recently, there has been a surge in endeavors to expedite the sampling processes of DPMs [14], [15], [16], [17],

[18], [19], [20], [21], [22], which are broadly categorized into training-based model distillation and training-free fast sampling approaches. Distillation-based techniques, notable for facilitating generation in a minimal number of steps, are somewhat curtailed by a complex distillation procedure and the necessity for model-specific distillation. For example, Comp-Diff [16] proposes the pruning and distillation framework for compressing diffusion models. Conversely, fast samplers [19], [23], [24], [20], [21], [25], [26], [27], [28], [29], [30] enjoy wider popularity for their inherent training-free quality, allowing seamless integration with readily available pre-trained DPMs. These methods predominantly leverage probability flow ordinary differential equations (ODEs) and can be formulated in a general continuous exponential integrator form. Their main strategy centers on minimizing discretization errors that emerge from approximating such continuous integration with large step size discretization (small sampling steps), and potentially reaching a plateau in terms of optimization. However, an intriguing question arises: *besides minimizing the discretization error, is it possible to further elevate the sampling speed and quality with fewer-step inference?*

In this paper, we start with a detailed examination of the constituents of the total sampling error, discerning it into discretization and approximation errors. The former part originates from the discretization of the continuous exponential integrator, whereas the latter part emerges due to the neural network’s imprecise estimation of the ground truth vector field (score function). Prevailing fast solvers predominantly address the discretization error, inadvertently ignoring the approximation error, thus leaving space for enhanced inference acceleration. Further, we empirically introduce a dual-error disentanglement strategy aimed at isolating these two sub-errors from the aggregate sampling error, subsequently elucidating their interrelations and their distinct associations with the step. Our findings indicate that both errors are of comparable magnitude, underscoring the criticality and significance of reducing the approximation error as part of our contribution. Moreover, we experimentally discover that the approximation error monotonically decreases as step t increases, a relationship that is pivotal for guiding the reduction of approximation error in later designs.

With these insights, we propose an unified and training-free acceleration framework, SynBoost¹, for the fast sampling of DPMs by taking both discretization and approximation errors into consideration to further reduce the total sampling error. To mitigate the discretization error, SynBoost readily incorporates

This work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences. (Corresponding author: Feng Zhao.)

H Yu, X Fu, J Huang, and F Zhao are with MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China. H Luo and F Wang are with Alibaba DAMO Academy for Discovery, Adventure, Momentum and Outlook.

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes more explanations, experiments and analyses. This material is 3.4 MB in size.

¹The name “SynBoost” in our work stands for “Synergistic Boosting”.

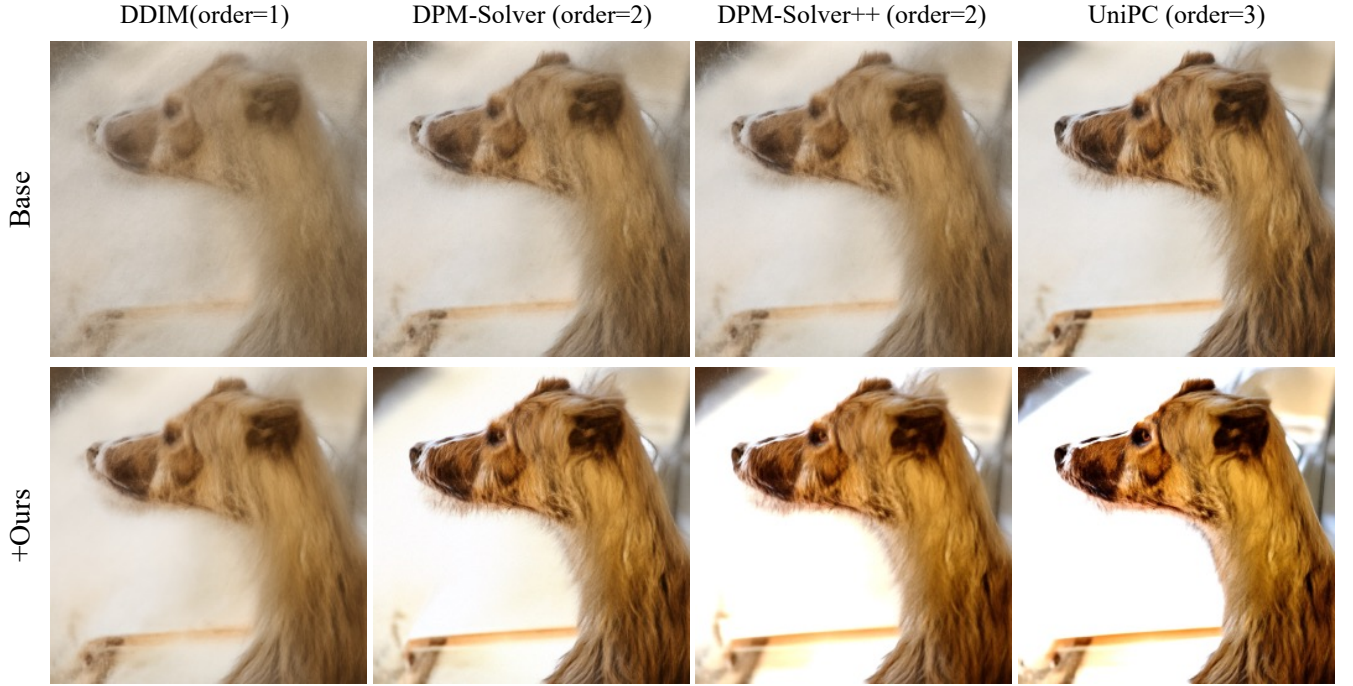


Fig. 1: Qualitative comparisons between existing solvers and our method with class condition. All images are generated by sampling from ADM [4] trained on ImageNet 256×256 with only 7 number of function evaluations (NFEs). We show that our proposed method can significantly elevate the sample quality with better details and contrast than previous base samplers.

existing ODE solver techniques. To address the approximation error, we design a plug-and-play reduction strategy that integrates fluidly with current ODE solvers, thereby facilitating further acceleration. Specifically, we propose a training-free mixing strategy that partially substitutes the noise estimation at the current timestep with a highly accurate prediction from a preceding, larger timestep. Furthermore, we elucidate the process of integrating this approximation error reduction strategy into several representative fast solvers, including DDIM [19], DPM-Solver [20], and DPM-Solver++ [25], spanning various orders and prediction modes. Note that this strategy is analyzed comprehensively and empirically, rather than pursuing strict theoretical rigor.

The efficacy of our SynBoost framework is rigorously substantiated through comprehensive experiments that span a diverse range of models, datasets, solvers, and sampling steps. Specifically, these experiments are structured to encompass two sampling types (unconditional and conditional generation), two condition types (class and text conditions), two sampling spaces (pixel-space and latent-space DPMs), samplers of different orders (1-order DDIM, 2-order DPM-Solver and DPM-Solver++), different prediction modes (noise and data predictions), and various sampling steps. SynBoost significantly improves the sampling quality and efficiency over previous solvers on all the conducted experiments, especially with extremely limited sampling steps. Qualitative comparisons, as depicted in Figure 1, reveal a consistent advantage of our method in producing images of superior structure, details, and color contrast over the base solvers.

II. RELATED WORK

A. Diffusion Probabilistic Models

Diffusion probabilistic models (DPMs) [1], [2], [3] transform complex data distributions into simple noise distributions and learn to recover data from noise. The forward diffusion process starts from a clean data sample $\mathbf{x}_0$ and repeatedly injects Gaussian noise to a simple normal distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ at time $T > 0$ for some $\sigma > 0$. The corresponding transition kernel $q_{t|0}(\mathbf{x}_t | \mathbf{x}_0)$ is as follows:

$$q_{t|0}(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}), \quad (1)$$

where the *signal-to-noise-ratio* (SNR) equals α_t^2 / σ_t^2 .

Training process. Diffusion models are trained by optimizing a variational lower bound (VLB). For each step t , the denoising score matching loss is the distance between two Gaussian distributions, written as:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}_0, \epsilon, t} \left[\omega(t) \|\epsilon_{\theta}(\mathbf{x}_t, t) - \epsilon\|_2^2 \right], \quad (2)$$

where $\omega(t)$ is weighting function, $\epsilon \sim q(\epsilon) = \mathcal{N}(\epsilon | \mathbf{0}, \mathbf{I})$, and $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$.

ODE-based sampling process. Sampling from DPMs can be achieved by solving the following diffusion ODE [3]:

$$\frac{d\mathbf{x}_t}{dt} = f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t} \epsilon_{\theta}(\mathbf{x}_t, t), \quad \mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \quad (3)$$

where the coefficients $f(t) = \frac{d \log \alpha_t}{dt}$, $g^2(t) = \frac{d \sigma_t^2}{dt} - 2 \frac{d \log \alpha_t}{dt} \sigma_t^2$.

B. Fast Sampling of DPMs

A large step size in stochastic differential equations (SDEs) violates the randomness of the Wiener process [31] and often causes non-convergence. Certain methods [32], [33] propose to accelerate SDE solvers but still lag behind ODE solvers in speed. AdaSDE [34] achieves strong results in the low-NFE regime by unifying ODE efficiency and SDE error resilience through a learnable error-correction coefficient. Restart sampling [29] combines the SDE and ODE to boost sampling quality, and also mentions the concept of the approximation error, but it neglects to present the reason, disentanglement, and impact of this error. EPD-Solver [35] leverages parallel gradient evaluations and lightweight distillation to effectively mitigate truncation errors. Chen et al. [36] explored to decompose the score function of the linear subspace data, but under the low-dimensional linear subspace assumption. Hunter et al. [37] reduced the cost of calling the U-net via reusing the attention, while keeping the total steps unchanged. For faster sampling, the probability flow ODE [3] is usually considered as a better choice. Recent works [20], [21], [25] find that ODE solvers built on exponential integrators [38] appear to have faster convergence than directly solving the diffusion ODE Equation 3. The solution $\mathbf{x}_{t_i}$ of the diffusion ODE given the initial value $\mathbf{x}_{t_{i-1}}$ can be computed as [20]:

$$\mathbf{x}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \alpha_{t_i} \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda) d\lambda, \quad (4)$$

where the ODE is changed from the time (t) domain to the log-SNR (λ) domain by the change-of-variables formula, and the notation $\hat{\epsilon}_{\theta}$ and $\hat{\mathbf{x}}_{\lambda}$ denote change-of-variables. Based on the exponential integrator, existing ODE solvers approximate $\hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda)$ via Taylor expansion at time $\lambda_{t_{i-1}}$.

$$\mathbf{x}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \alpha_{t_i} \sum_{n=0}^{k-1} \hat{\epsilon}_{\theta}^{(n)}(\hat{\mathbf{x}}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}}) \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \frac{(\lambda - \lambda_{t_{i-1}})^n}{n!} d\lambda + \mathcal{O}(h_i^{k+1}), \quad (5)$$

where k denotes the order of the Taylor expansion, also known as the order of the solver, and $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$ is the step size. Obviously, high order solver reduces the discretization error to $\mathcal{O}(h_i^{k+1})$. Further, it can be simplified into a general form as follows:

$$\mathbf{x}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \sigma_{t_i} (e^{h_i} - 1) \mathbf{D}_{t_i}. \quad (6)$$

Existing ODE solvers only differ in $\mathbf{D}_{t_i}$. For the first-order DDIM solver, $\mathbf{D}_{t_i} = \epsilon_{\theta}(\hat{\mathbf{x}}_{t_{i-1}}, t_{i-1})$. EDM [39] finds that the discretization error is smaller at larger noise level. DPM-Solver++ [25] considers rewriting Equation 4 using $\hat{\mathbf{x}}_{\theta}$ instead of $\hat{\epsilon}_{\theta}$. UniPC [26] proposes a predictor-corrector method to refine the prediction. A common thread in these approaches is that they attempt to reduce discretization error part via high order Taylor approximation in Equation 5, while ignoring the approximation error induced in Equation 4 when replacing score function with network approximation $\hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda)$. It is worth noting that our framework is also theoretically orthogonal and compatible with linear multi-step methods like iPNDM

[24], though exponential integrators remain our primary focus due to their dominance in the extreme low-step regime.

III. METHOD

A. Motivation and Insights

We delve into the evolution from the exact solution of diffusion ordinary differential equations (ODEs) to its practical implementation form as illustrated in Equation 6:

$$\begin{aligned} \mathbf{x}_{t_i} &= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \alpha_{t_i} \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} [-\sigma_t \nabla_{\mathbf{x}} \log q_t(\mathbf{x}_t)] d\lambda, \\ &= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \alpha_{t_i} \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda) d\lambda + \mathcal{E}_{app}, \\ &= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \sigma_{t_i} (e^{h_i} - 1) \mathbf{D}_{t_i} + \mathcal{E}_{app} + \mathcal{E}_{dis}. \end{aligned} \quad (7)$$

Here, $\mathbf{D}_{t_i}$ signifies the polynomial of $\hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda)$, mirroring varying degrees of discrete approximations to the continuous integral. The above formulation reveals that the induced error consists of two parts: discretization error ($\mathcal{E}_{dis}$) and approximation error ($\mathcal{E}_{app}$). The approximation error arises from the substitution of the precise score function $-\sigma_t \nabla_{\mathbf{x}} \log q_t(\mathbf{x}_t)$ with the network's approximation $\hat{\epsilon}_{\theta}(\hat{\mathbf{x}}_{\lambda}, \lambda)$ due to the denoising network's imprecise score function estimation. Subsequently, discretization error emerges when simulating the continuous integral through discrete implementation. While previous fast samplers predominantly aim at diminishing discretization error via higher-order Taylor expansions, introducing variously ordered samplers such as 1-order DDIM and 2-order DPM-Solver, they overlook the criticality of approximation error. It is noticeable that while some prior methods [29], [37] may recognize the presence of approximation error, they conclude the exploration at this point. In stark contrast, this derivation is the start of our study. We conduct comprehensive analyses and studies to thoroughly unlock the approximation error, and proceed to integrating this error into existing solvers for further acceleration with an unified and general ODE-based acceleration framework.

B. Dual-Error Disentanglement

With the above insights on the total sampling error, we then delve into dissecting the properties of these two errors. To achieve this, we introduce a dual-error disentanglement strategy, effectively isolating these sub-errors as depicted in Figure 2. Specifically, within the time interval $[t_{i-1}, t_i]$, we construct three distinct transition processes, each subjected to varying levels of sampling error. Initially, we define the exact data distributions $\mathcal{P}(\mathbf{x}_{t_{i-1}})$ and $\mathcal{P}(\mathbf{x}_{t_i})$, which are free of both approximation and discretization errors. These distributions are derived from the pristine image distribution $\mathcal{P}(\mathbf{x}_0)$, employing the transition kernel outlined in Equation 1. This procedure establishes an optimal, error-free transition from distribution $\mathcal{P}(\mathbf{x}_{t_{i-1}})$ to $\mathcal{P}(\mathbf{x}_{t_i})$. We then formulate a second distribution transition from $\mathcal{P}(\mathbf{x}_{t_{i-1}})$ to $\mathcal{P}(\mathbf{x}_{t_i}^s)$, which is largely afflicted by approximation error, alleviating the effect of discretization error. To this end, we adopt an extremely

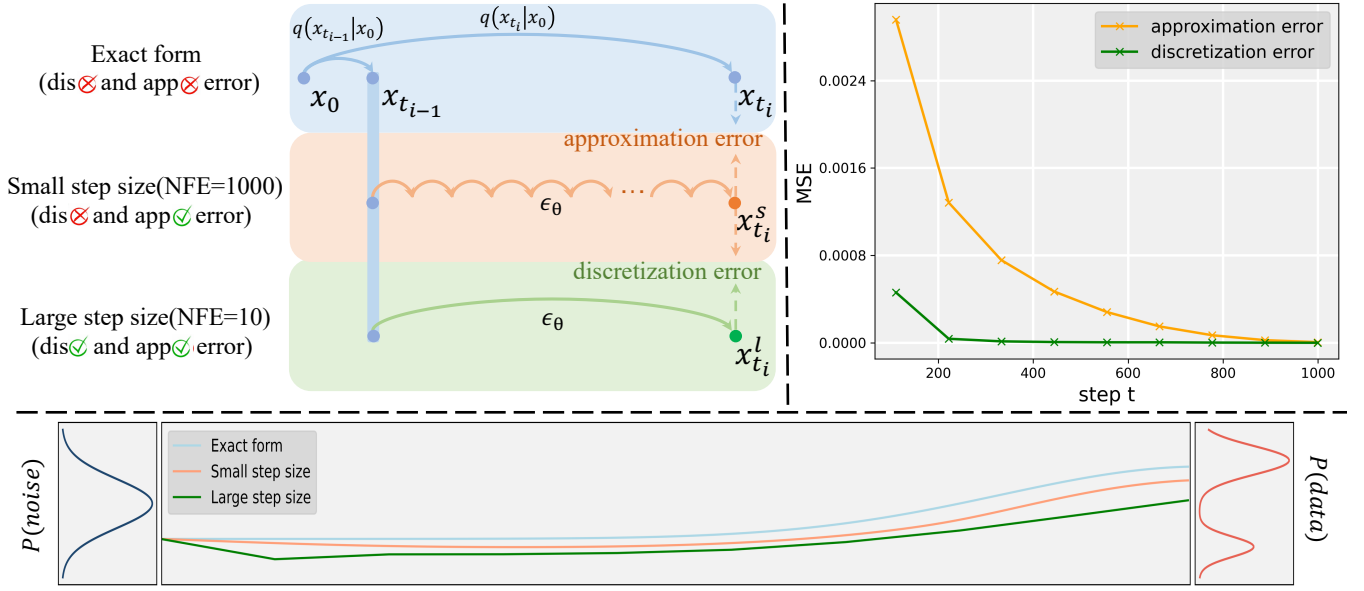


Fig. 2: (Left): Dual-Error Disentanglement. Empirically, for the identical time period, we map out three distinctive distribution transition processes, each subjected to a unique combination of errors. The top one represents exact data distribution and is free of errors. The middle one is liberated from discretization error owing to an extremely fine-grained step size (NFE=1000). The bottom one, with a coarse step size (NFE=10), succumbs to both errors. (Right): The MSE curve of approximation and discretization errors along the step t , evaluated on the unconditional generation task using the ImageNet dataset. Note that this monotonically decreasing trend remains highly consistent across different generation settings (e.g., classifier-guided conditional generation, as detailed in Figure 12 of the experimental part). (Bottom): Overview of ODE-based generation process mapping noise to data distribution, where smaller errors correspond to higher probability density region.

small step size (NFE=1000) to minimize the discretization error when approximating the continuous integral. Subsequently, we chart the third distribution transition from $\mathcal{P}(\mathbf{x}_{t_{i-1}})$ to $\mathcal{P}(\mathbf{x}_{t_i}^l)$, suffering from both discretization and approximation errors, facilitated by a large step size (i.e., NFE=10).

Originating from the identical data distribution $\mathcal{P}(\mathbf{x}_{t_{i-1}})$, we now derive three distributions $\mathcal{P}(\mathbf{x}_{t_i})$, $\mathcal{P}(\mathbf{x}_{t_i}^s)$, and $\mathcal{P}(\mathbf{x}_{t_i}^l)$, thereby disentangling approximation and discretization errors within the time range $[t_{i-1}, t_i]$. Notably, the discrepancy between $\mathcal{P}(\mathbf{x}_{t_i})$ and $\mathcal{P}(\mathbf{x}_{t_i}^s)$ illuminates the approximation error, whereas the divergence between $\mathcal{P}(\mathbf{x}_{t_i}^s)$ and $\mathcal{P}(\mathbf{x}_{t_i}^l)$ manifests the discretization error. We adopt the following formulation to approximately represent these two errors. The rationality of adopting MSE to measure distribution distance is explained in the Analyses part of the Experiment section.

$$\begin{aligned} E_{app} &= \text{MSE}(\mathbf{x}_{t_i}^s, \mathbf{x}_{t_i}), \\ E_{dis} &= \text{MSE}(\mathbf{x}_{t_i}^s, \mathbf{x}_{t_i}^l). \end{aligned} \quad (8)$$

An illustrative MSE curve of these errors across the step t is depicted in Figure 2. Specifically, we divide the total generation period to 9 segments, with each covering the period size of 111. Therefore, NFE=111 and NFE=1 respectively represent the small step manner and large step manner. The right part of Figure 2 is then plotted with these 9 segments. Analysis of the MSE curve yields three pivotal insights: (1) Both errors are of comparable magnitude and the approximation error surpasses the discretization error at most timesteps, highlighting its significant influence in the sampling process and underscoring the criticality of reducing the approximation error for accelerating sampling. (2) The discretization error

declines as the step t increases, which is also consistent with the conclusion in EDM [39]. (3) The approximation error exhibits a strict decline as the step t increases, a principle that subsequently instructs the design of our approximation error reduction strategy.

C. Acceleration Framework-SynBoost

In this section, we introduce our unified and training-free SynBoost framework, crafted to synergize with prevailing fast samplers while concurrently mitigating the approximation error for augmented acceleration. For the discretization error, we seamlessly incorporate the Taylor approximation mechanism employed by preceding fast ODE solvers. For the approximation error, we formulate a reduction strategy by leveraging the property of the approximation error that it monotonically decreases with advancing step t . This characteristic, where the network's estimation of the actual score function gains accuracy at larger step, guides us to partially substitute the noise prediction at the current step t_i with that of a preceding, larger step τ , given its diminished approximation error.

$$\epsilon_{\theta}^{\text{new}}(\mathbf{x}_{t_i}, t_i) = (1 + c)\epsilon_{\theta}(\mathbf{x}_{t_i}, t_i) - c\epsilon_{\theta}(\mathbf{x}_{\tau}, \tau), \quad (9)$$

where c is the mixing coefficient, and step τ is larger than current step t_i . We provide comprehensive theoretical and empirical justifications for this formulation in the supplementary material. Specifically, we detail: 1) its strict mathematical derivation from the exact DDIM sampling step; 2) the theoretical expansion of the direct substitution baseline; and 3) empirical evidence demonstrating the severe numerical instability (e.g., generation failure) inherently caused by direct

substitution. The mixing coefficient c should monotonically increase as current step t_i decreases, echoing the tendency that smaller steps correspond to larger approximation error and thus necessitates more substantial rectification. Besides, we adopt the initial noise $\mathbf{x}_T$ as $\epsilon_\theta(\mathbf{x}_T, \tau)$. Note that this does not introduce extra NFEs, since the prediction $\epsilon_\theta(\mathbf{x}_T, T)$ is computed once at the first step and cached in memory. A detailed examination and ablation study pertaining to the configurations of c and τ are presented in Sec. IV-D. Besides, while Equation 9 is expressed within the context of noise prediction, it is equally applicable to data prediction due to the equivalent transformation between these modes. Note that, SynBoost only needs one NFE per step.

The rationality of Equation 9 can also be verified by its similarity to the multi-step method [24], which averages the history noise prediction to reduce error.

SynBoost can be seamlessly integrated into existing fast ODE solvers to achieve further speedup. We provide a thorough mathematical integration process of this approximation error reduction strategy into existing solvers. For instance, we select three common and representative fast ODE samplers: DDIM, DPM-Solver, and DPM-Solver++, spanning various orders and prediction modes. Furthermore, our framework is fundamentally compatible with other state-of-the-art exponential integrators such as UniPC [26]. Since SynBoost operates by refining approximation error, it is completely orthogonal to UniPC’s multi-step discretization rules, allowing for seamless plug-and-play integration without structural conflicts. We also present the integration of our method with UniPC in the experimental part. Here, we again emphasize that Equation 6 is the general formulation of the ODE-based inference, with the only difference between different samplers in $\mathbf{D}_{t_i}$.

Ours-DDIM. DDIM is the classical ODE sampler that proposes a non-Markovian diffusion process following the same training objective as DDPMs. This non-Markovian process leads to deterministic generation and faster speed. The sampling formulation of DDIM is as follows:

$$\mathbf{x}_{t_i}^{base} = \alpha_{t_i} \mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + \sigma_{t_i} \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}). \quad (10)$$

Further, we can rewrite the above equation in the unified form of Equation 6 with following $\mathbf{D}_{t_i}$:

$$\mathbf{D}_{t_i}^{base} = \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}). \quad (11)$$

To reduce the approximation error, we replace the noise estimation part in Equation 11 with Equation 9.

$$\mathbf{D}_{t_i}^{ours} = (1 + c) \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) - c \epsilon_\theta(\mathbf{x}_T, \tau). \quad (12)$$

Ours-DPM-Solver. We apply multi-step, thresholding strategy [25] and second-order to DPM-Solver, and get the base version termed as DPM-Solver(2M). Note that the only difference between DPM-Solver(2M) and DPM-Solver++(2M) is the prediction mode.

DPM-Solver reveals that diffusion ODEs have a semi-linear structure and derives the formulation of the solutions by analytically computing the linear part of the solutions, avoiding the corresponding discretization error. Concretely,

DPM-Solver(2M) can be directly written in the formation of Equation 6 with the corresponding $\mathbf{D}_{t_i}$:

$$\mathbf{D}_{t_i}^{base} = \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + a_1 [\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) - \epsilon_\theta(\mathbf{x}_{t_{i-2}}, t_{i-2})], \quad (13)$$

where a_1 is the coefficient for the second part of DPM-Solver. Then, we adopt a similar way as in previous DDIM part to reduce the approximation error.

$$\mathbf{D}_{t_i}^{ours} = [(1 + c) \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) - c \epsilon_\theta(\mathbf{x}_T, \tau)] + a_1 [\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) - \epsilon_\theta(\mathbf{x}_{t_{i-2}}, t_{i-2})]. \quad (14)$$

Ours-DPM-Solver++. DPM-Solver++ is the SOTA and default sampler in the stable diffusion model. It finds that previous high-order fast samplers suffer from instability issue, and solves the diffusion ODE with the data prediction model. Due to employing different prediction modes, DPM-Solver++ reformulates Equation 6 as follows:

$$\mathbf{x}_{t_i} = \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} \mathbf{x}_{t_{i-1}} - \alpha_{t_i} (e^{-h_i} - 1) \mathbf{D}_{t_i}, \quad (15)$$

where $\mathbf{D}_{t_i}$ is expressed in the data-prediction mode $\mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$ as follows:

$$\mathbf{D}_{t_i}^{base} = \mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + a_2 [\mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) - \mathbf{x}_\theta(\mathbf{x}_{t_{i-2}}, t_{i-2})], \quad (16)$$

where a_2 is the coefficient for the second part of DPM-Solver++. The data prediction $\mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$ and noise prediction $\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$ can be mutually derived from each other with Equation 1.

$$\mathbf{x}_{t_{i-1}} = \alpha_{t_{i-1}} \mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}) + \sigma_{t_{i-1}} \epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1}). \quad (17)$$

Therefore, we first convert the first order part $\mathbf{x}_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$ in Equation 16 to noise prediction $\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$, and apply Equation 9 to the converted $\epsilon_\theta(\mathbf{x}_{t_{i-1}}, t_{i-1})$, and finally convert it back to data prediction.

IV. EXPERIMENTS

In this section, we show that our method can significantly boost the sampling quality and speed of existing solvers through extensive experiments. We employ FID and the human preference model HPD v2 [40] for comprehensive evaluation. Following the setting of recent training-free samplers [25], [26], all values are reported using 10k images unless specifically mentioned. This ensures computational efficiency across our extensive ablation sweeps while maintaining strictly fair across all baseline methods. We first present main results in Section IV-A and provide more analyses in Section IV-D.

A. Main Results

We illustrate the effectiveness of our method on few-step sampling with comprehensive experiments and analyses. Our experiments cover the main fast ODE samplers with different orders (1-order DDIM, 2-order DPM-Solver and DPM-Solver++), two model prediction modes (noise and data prediction), three general generation types (unconditional, class-conditional, and text-conditional), two existing guiding strategies (classifier-guided [4] and classifier-free guided [41])

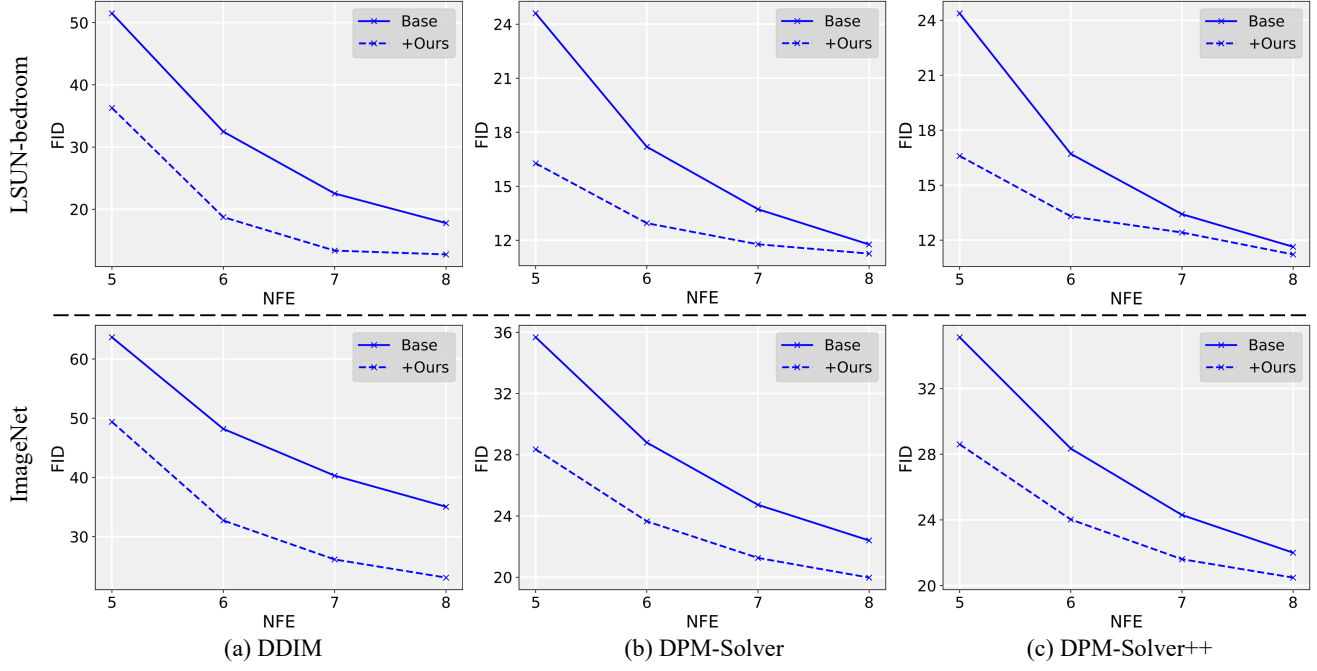


Fig. 3: Unconditional sampling results. We compare SynBoost with baseline samplers on LSUN Bedroom and ImageNet datasets. We report the FID $\downarrow$ of the methods with different NFEs. Experimental results show that SynBoost is consistently better than baselines on pixel-space DPMs.

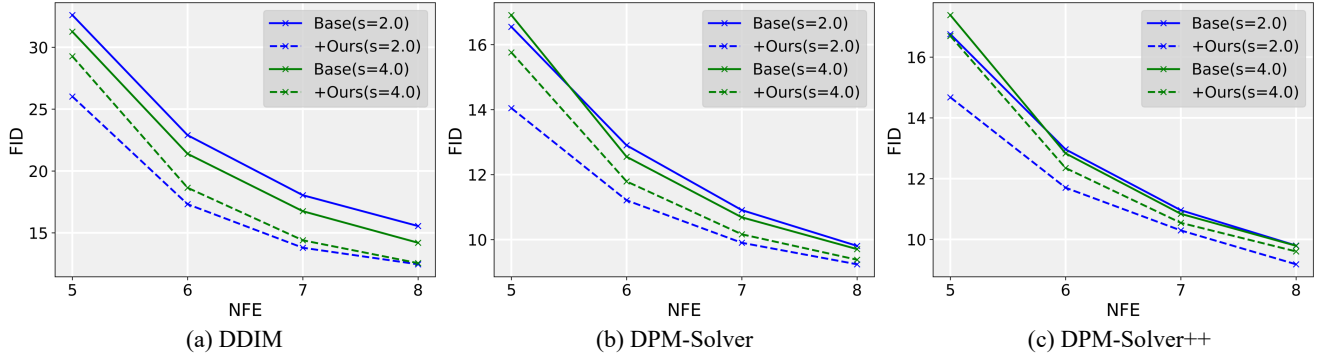


Fig. 4: Class-conditional sampling results. Quantitative comparisons between existing samplers and our method with classifier-guided class condition, employing various classifier scales ($s=2.0$ and 4.0) and NFEs. SynBoost achieves consistent performance improvement over baseline samplers.

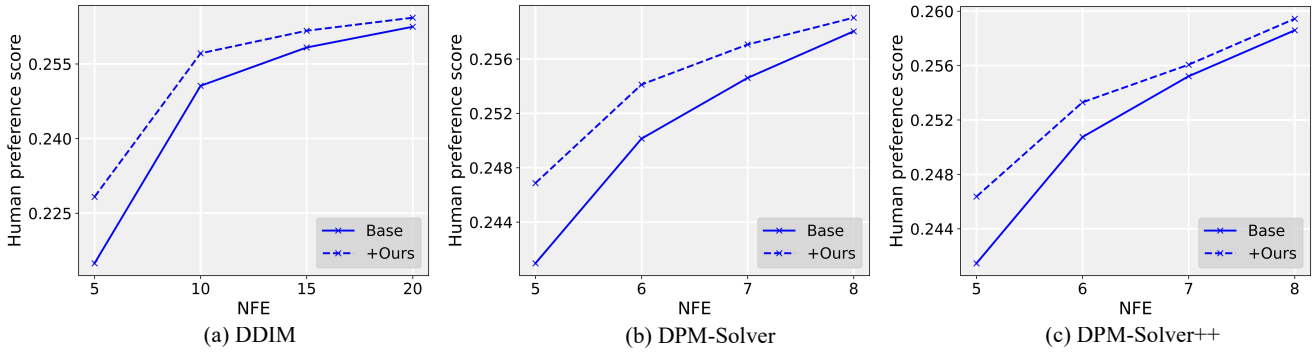


Fig. 5: Text-conditional sampling results. The results are reported on Stable diffusion model with the MS-COCO2014 validation dataset and guidance scale 7.5. We adopt the human preference score $\uparrow$ from HPD v2 [40]. SynBoost performs better than previous baselines on latent-space DPMs.

models), two mainstream sampling space (image space [4] and latent space [5]), main-stream datasets (LSUN-bedroom and ImageNet), as well as various guidance scales.

Unconditional sampling. We first compare the unconditional sampling quality of different methods on LSUN Bedroom [42] and ImageNet [43] datasets in Figure 3. The pre-trained diffusion models are from [4]. Our method substantially boosts existing fast ODE solvers with better sampling quality and faster speed. The performance lift is especially obvious with fewer NFEs, which demonstrates the potential and effectiveness of SynBoost for the practical deployment of generative diffusion models.

Class-conditional sampling. Besides the unconditional sampling, we adopt the class label as condition information. To this end, we employ the classifier-guided sampling strategy and the pre-trained models from [4]. We validate the effectiveness of our method over baseline samplers under different guidance scales ($s = 2.0$ and 4.0). The results are shown in Figure 4. SynBoost achieves consistent performance improvement with various solvers and guidance scales.

Ablations on c and τ . Figure 8 shows the sensitivity analysis and robustness of the framework’s hyper-parameters, which investigates the two primary components of Equation 9: (a) the mixing coefficient c and (b) the target timestep τ . The results in (a) show that a linearly decreasing strategy for c (starting at 0.5 and reaching 0.0 at step T) outperforms constant coefficients, aligning with the empirical observation that approximation error decreases as steps advance. Part (b) demonstrates that choosing a larger τ (closer to initial noise T) generally yields better performance due to the higher accuracy of the network at those timesteps. Overall, the curves reflect high robustness, as a wide range of parameter choices consistently result in a substantial performance lift over the base DPM-Solver.

Text-conditional sampling. To further assess our method’s performance across different condition types, we explore its application in a text-to-image stable diffusion model [5], which works in latent space utilizing classifier-free guidance strategy [41]. The guidance scale is set as 7.5 following the common setting. We concatenate conditional and unconditional inputs channel-wise into a single forward pass, consuming exactly 1 NFE per step. We sample the first 10K captions from the MS-COCO2014 validation dataset [44] for input texts. Acknowledging the limitations of the FID metric in text-to-image scenarios [25], [26], and the inadequacy of MSE for evaluating distribution convergence, we instead employ the HPD v2 [40], a state-of-the-art model that predicts human preferences for images generated by text-to-image diffusion models. Results, illustrated in Figure 5, show our method consistently outperforms baseline samplers across varying orders, as evidenced by higher human preference scores.

B. Visual Results

Class-conditional sampling. We provide a qualitative comparison between our method and previous sampling methods in Figure 6. We adopt NFE=7 with various classifier scales ($s = 0.0, 2.0$, and 4.0) for the presented samples. We find that the

sampling quality generally increases as the order of the sampler increases under the same NFE. Besides, the class labels of the images are increasingly obvious due to larger classifier guidance scale. While, our method consistently improves the image quality with better details, color, and contrast regardless of the sampler and guidance scale. For example, DualFast can even boost DDIM to achieve comparable visual results to DPM-Solver++. The above results strongly demonstrate the effectiveness of our method on pixel-space based sampling. Our method consistently improves the image quality with better details, color, and contrast regardless of the samplers and guidance scales. For example, DualFast can even boost DDIM to achieve comparable visual results to DPM-Solver++. The above results strongly demonstrate the effectiveness of our method.

Text-conditional sampling. Besides the pixel-space sampling results, we additionally provide visual results on stable diffusion in Figure 7. The NFE is set as only 5 to validate the performance bound of the compared methods. The classifier-free guidance scale is 7.5. Our method consistently generate more realistic images with fewer visual flaws and better structures than previous samplers. The above results illustrate that our method generalizes well to both pixel and latent space. Besides the pixel-space sampling results, we additionally provide a qualitative comparison between our method and previous sampling methods on stable diffusion model in Figure 7. The NFE is set as only 5 to validate the performance bound of existing sampling methods and our method. The classifier-free guidance scale is 7.5. Our method consistently generate more realistic images with fewer visual flaws and better structures than previous samplers of various orders. The above results illustrate that our method generalizes well to both pixel and latent space generation.

C. Ours-UniPC

UniPC is the recent state-of-the-art high-order ODE solver. It employs a predictor-corrector framework, consisting of a predictor and a corrector. The unified corrector (UniC) can be applied after any existing DPM sampler to increase the order of accuracy without extra model evaluations, and the unified predictor (UniP) supports arbitrary order. In this part, we take the most widely used third-order UniPC sampler as example. Concretely, the predictor in UniPC first gets an estimation of $\mathbf{x}_t^p$, then with the corresponding $\mathbf{D}_{t-1}$:

$$\begin{aligned} \mathbf{D}_{t-1}^{base} = & \epsilon_\theta(\mathbf{x}_t, t) + \sum_{m=0}^{p-2} a_m^p [\epsilon_\theta(\mathbf{x}_{t+m}, t+m) \\ & - \epsilon_\theta(\mathbf{x}_{t+1+m}, t+1+m)], \end{aligned} \quad (18)$$

where p is the order of the predictor. Then the corrector in UniPC refines the estimation $\mathbf{x}_t^p$ with the corresponding $\mathbf{D}_t$:

$$\begin{aligned} \mathbf{D}_{t-1}^{base} = & \epsilon_\theta(\mathbf{x}_t, t) + \sum_{m=0}^{p-2} a_m^c [\epsilon_\theta(\mathbf{x}_{t+m}, t+m) \\ & - \epsilon_\theta(\mathbf{x}_{t+1+m}, t+1+m)] \\ & + a_0^c [\epsilon_\theta(\mathbf{x}_{t-1}, t-1) - \epsilon_\theta(\mathbf{x}_t, t)]. \end{aligned} \quad (19)$$

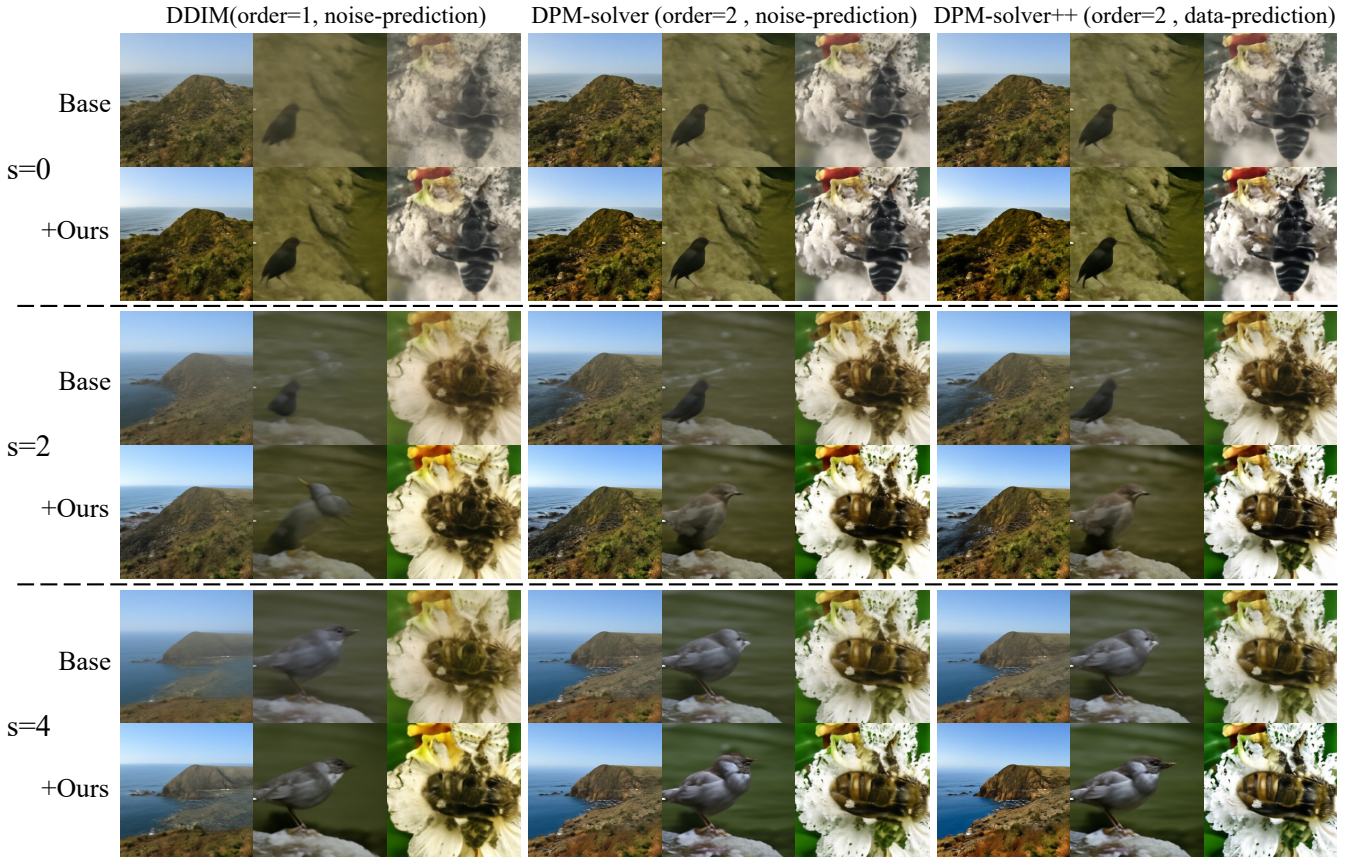


Fig. 6: Qualitative comparisons with class-conditional sampling in pixel-space. All images are generated by sampling from a DPM trained on ImageNet 256×256 [4] with NFE= 7. The classifier scale s is respectively set as 0.0, 2.0, and 4.0. Our method can generate more plausible samples with more visual details and higher contrast compared with previous samplers.



“A room with blue walls and a white sink and door”

“A bathroom sink with toiletries on the counter”

Fig. 7: Qualitative comparisons with text-conditional sampling in latent-space. All images are generated with NFE = 5 and classifier-free guidance scale = 7.5. Our method can consistently generate more realistic images with fewer visual flaws than previous samplers of various orders.

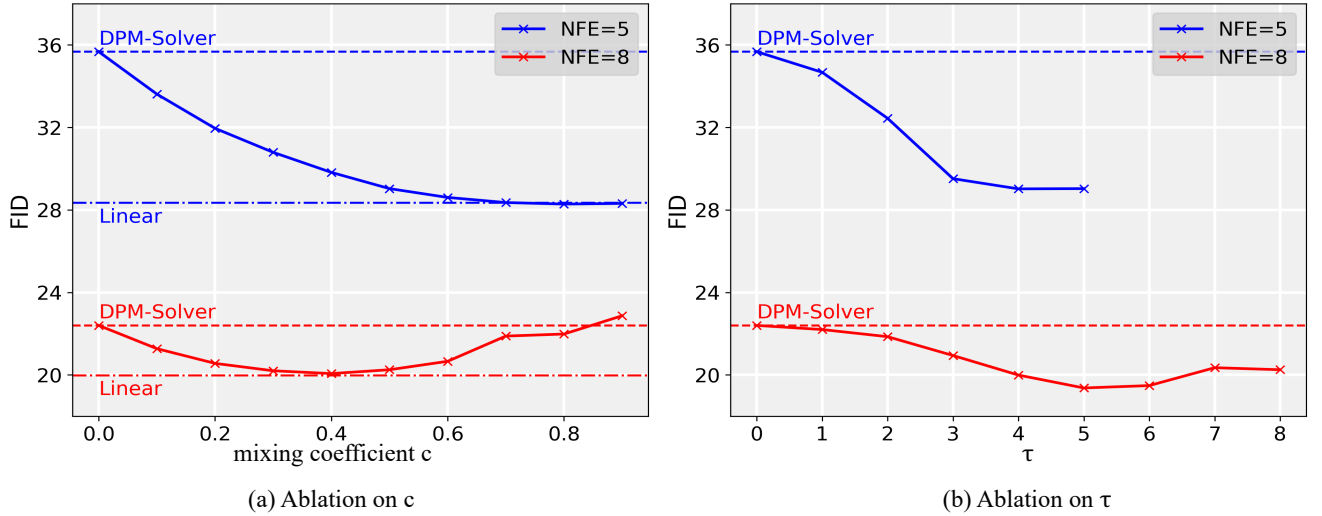


Fig. 8: Sensitivity analysis and robustness of the framework’s hyper-parameters. This figure investigates the two primary components of Equation 9: (a) the mixing coefficient c and (b) the target timestep τ . The results in (a) is achieved with τ fixed to T , showing that linearly decreasing strategy for c outperforms constant coefficients. Part (b) demonstrates that, under the linear mixing strategy, choosing a larger τ generally yields better performance. Overall, the curves reflect high robustness, as a wide range of parameter choices consistently result in a substantial performance lift over the base DPM-Solver.

TABLE I: Reduced error. Detailed MSE reduction analysis between DPM-Solver and SynBoost at various NFEs. By using the results of 1000-step sampling as pseudo-ground truth, the experiment demonstrates that SynBoost consistently achieves lower MSE than the base DPM-Solver across all steps. Since the discretization error remains identical for both methods at a given NFE, the observed reduction stems directly from SynBoost’s ability to mitigate the “approximation error”.

MSE(10^{-3})	NFE								
	5	6	7	8	9	10	12	15	20
Base	10.97	8.19	6.23	4.93	3.96	2.63	1.82	1.11	0.61
Ours	7.81	5.28	3.69	2.79	2.15	2.08	1.44	0.96	0.53

TABLE II: Performance enhancement of unconditional sampling on the high-order UniPC solver. This table quantifies the improvement in FID scores when the SynBoost framework is integrated into UniPC, a high-order predictor-corrector ODE solver. While UniPC is designed to minimize discretization error through its multi-step architecture, the further reduction in FID achieved by “UniPC (Ours)” proves that approximation error remains a bottleneck even for advanced third-order solvers.

Sampling Method \ NFE	4	5	6	7
UniPC (Base)	44.617	29.472	23.324	20.676
UniPC (Ours)	42.066	28.367	22.882	20.451

Then the modified version with our approximation error reduction strategy are presented as follows:

$$\begin{aligned}
D_{t-1}^{\text{base}} &= \epsilon_{\theta}(\mathbf{x}_t, t) + \sum_{m=0}^{p-2} a_m^c [\epsilon_{\theta}(\mathbf{x}_{t+m}, t+m) \\
&\quad - \epsilon_{\theta}(\mathbf{x}_{t+1+m}, t+1+m)] + a_0^c [(1+c)\epsilon_{\theta}(\mathbf{x}_{t-1}, t-1) \\
&\quad - c\epsilon_{\theta}(\mathbf{x}_{\tau}, \tau) - \epsilon_{\theta}(\mathbf{x}_t, t)].
\end{aligned} \tag{20}$$

We also show the results on UniPC sampler with quantitative comparison in Table II and visual results in Figure 9. Our method can also boost the performance of this 3-order solver.

D. Analyses

In this section, we will provide more detailed analyses and ablation studies (including: the choices of hyper-parameters,

MSE analysis, higher guidance scale, larger NFEs, and compatibility with DiT [45] architecture) to further evaluate the effectiveness of SynBoost. Due to page limit, **we leave more experiments and analyses in the supplementary material, including** the comparison with SDE-based samplers, performance upper bound of SynBoost, sampling diversity metric, as well as more visual results.

Ablation on the choices of c and τ . SynBoost has two main hyper-parameters within equation 9. For the mixing coefficient c , based on the prior that approximation error linearly decreases with steps, we adopt a linearly decreasing strategy (from 0.5 to 0.0), which starts from 0.5 at step 0 and reaches 0.0 at step T . We also compare with constant mixing coefficient in Figure 8, where our linearly decreasing weight achieves higher performance than constant one. For the choice of step τ , we investigate the performance of different τ in

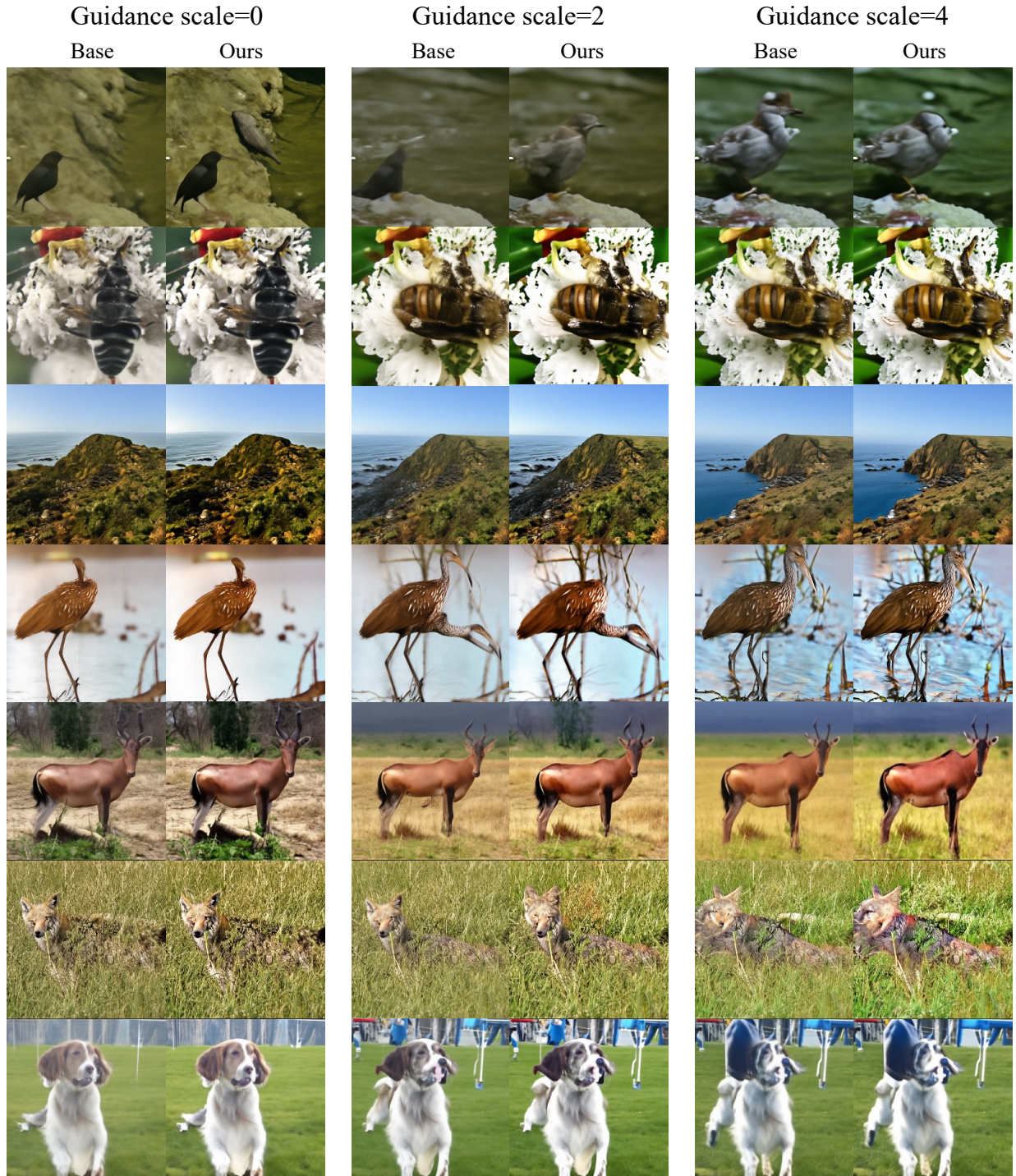


Fig. 9: Qualitative comparison of SynBoost integration with UniPC sampler on ImageNet dataset with various guidance scales. These images provide a visual validation of the quantitative data in Table II. Our method exhibits noticeably better structural details, higher color contrast, and fewer artifacts compared to the base UniPC sampler. This demonstrates that by correcting the approximation bias at each step, SynBoost enables high-order solvers to produce more realistic and plausible samples in the pixel-space, even when the underlying solver is already highly optimized for discretization errors.

Figure 8. Higher τ usually corresponds to better performance. For simplicity and ease-of-use, we directly employ $\tau = T$. This means that we employ initial noise x_T as $\epsilon_\theta(x_T, \tau)$. Besides the above analyses, we additionally emphasise that due to our efficient paradigm, the choices of c and τ are quite robust. Different choices all lead to substantial performance lift compared to the baseline solver.

We also expanded the ablation study on the mixing coefficient c to include the 3rd-order UniPC solver in Figure 10. The experiment is conducted on the ImageNet dataset with unconditional generation. We adopt NFE=5 for lower computational burden. The results confirm that our linear mixing strategy for c remains highly effective and robust even when integrated with higher-order solvers like UniPC, further demonstrating the generality of SynBoost.

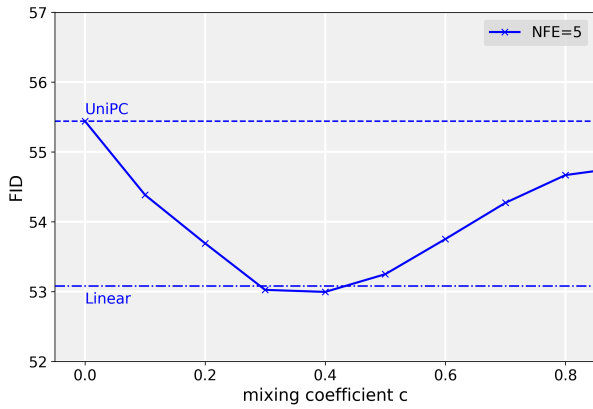


Fig. 10: Generality of the mixing strategy across different solver: ablation study of the mixing coefficients c on the UniPC sampler using $NFE = 5$. The U-shaped FID curve confirms that an optimal mixing coefficient c (around 0.3 to 0.4) significantly improves generation quality compared to the base solver ($c = 0.0$). The consistency of this trend with the results for 2nd-order solvers in Figure 8 reinforces the generality of the SynBoost framework. It proves that the linear mixing strategy remains a robust and effective method for reducing total sampling error regardless of the mathematical order or complexity of the base ODE solver.

Reduced error. We verify the effectiveness of SynBoost with the common FID metric as well as numerical visual results. Besides, we also depict the MSE analysis in Table I. Specifically, we adopt the results of 1000-step sampling as pseudo GT. Then we generate samples under various NFEs with both baseline solver and our SynBoost. Since the NFEs and sampler orders are kept identical between the baseline solver and our SynBoost, the discretization errors are also same between these two pairs. Thus the additional MSE error reduction stems from the decreased approximation error brought by our SynBoost. More concretely, at each step t , the network prediction is modified with higher accuracy, thus contributing to smaller final error compared to the pseudo GT. **Higher Guidance Scale.** Guided sampling can significantly boost the sample quality compared to unconditional sampling. But high guidance scale would also cause the instability of the

sampler [25] and poor sample quality. In this part, we show the results with guidance scale of 6.0 in Table III, and avoid higher guidance scale. Compared with the results of smaller guidance scales, the sample quality is worse with guidance scale 6.0. But, SynBoost can still consistently achieve better performance than the base sampler.

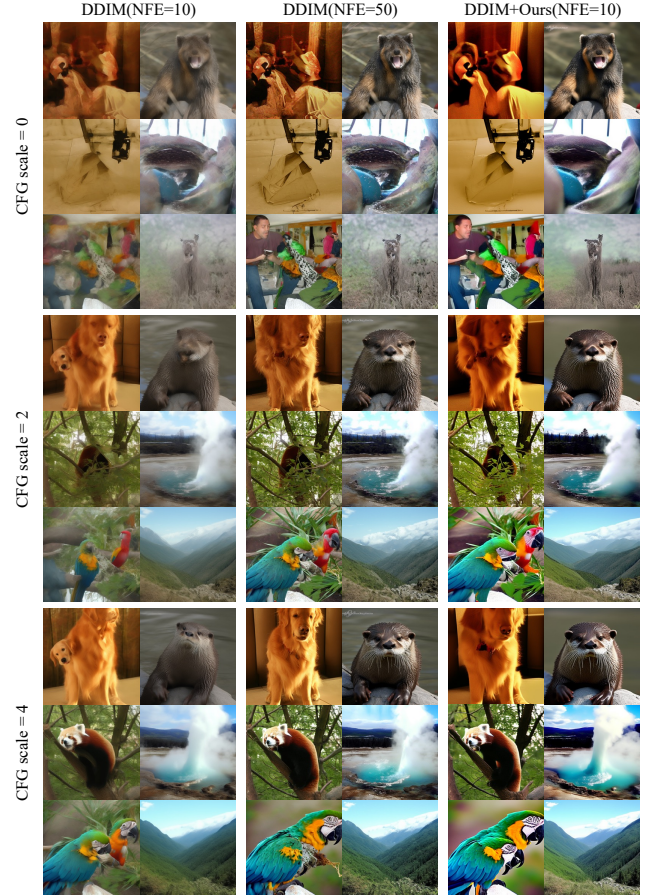


Fig. 11: Visual results on DiT with various guidance scales. All images are generated by sampling from DiT-XL/2 [45] with class condition. Our method significantly boosts the quality and speed of DDIM sampler, even achieving comparable visual results to DDIM of 50 NFEs with only 10 NFEs.

Performance on DiT. We present the efficacy of our method on DiT [45], which adopts transformer [46] architecture. Specifically, we adopt DiT-XL/2 with various guidance scales in Figure 11. Our method significantly boosts the quality and speed of DDIM sampler, even achieving comparable visual results to DDIM of 50 NFEs with only 10 NFEs.

Larger NFEs. In this part, we also validate the effectiveness of our method on larger NFEs in Table IV. SynBoost improves the FID of DDIM from 28.906 to 20.559 when NFE=10, achieving two times acceleration (comparable to 20-step DDIM sampling).

Extension of the observed trend. The statistical results in Figure 2 are evaluated on the unconditional generation task using the ImageNet dataset. To verify the consistency of this observation on different generation settings, we present the error decomposition curve for classifier-guided class-

TABLE III: Robustness and stability under high classifier guidance ($s = 6.0$) of class-conditional sampling. High guidance scales often lead to sampler instability and degraded sample quality in standard diffusion models. However, the data shows that SynBoost consistently outperforms the base DDIM (e.g., reducing FID from 23.408 to 19.845 at NFE=6). This demonstrates that SynBoost effectively stabilizes the sampling trajectory even when aggressive guidance is applied.

Guidance Scale	Sampling Method \ NFE	5	6	7	8
6.0	DDIM (Base)	36.129	23.408	17.407	14.705
	DDIM (Ours)	34.976	19.845	14.341	12.740

TABLE IV: Sample quality of unconditional sampling on ImageNet dataset with larger NFes. This table explores the effectiveness of SynBoost as the number of sampling steps increases from 10 to 20. A pivotal finding is that DDIM (Ours) at only 10 NFes (FID 20.559) achieves comparable or superior results to the base DDIM at 20 NFes (FID 20.344). This indicates a roughly 2x acceleration in inference speed while maintaining image quality.

Sampler	DDIM (Base)			DPM-Solver (Base)			DDIM (Ours)	DPM-Solver (Ours)
NFE	10	15	20	10	15	20	10	10
FID	28.906	22.781	20.344	21.250	20.004	19.533	20.559	19.727

conditional generation on ImageNet in Figure 12. The results exhibit the exact same monotonic trend that approximation error decreasing monotonically as timestep t increases.

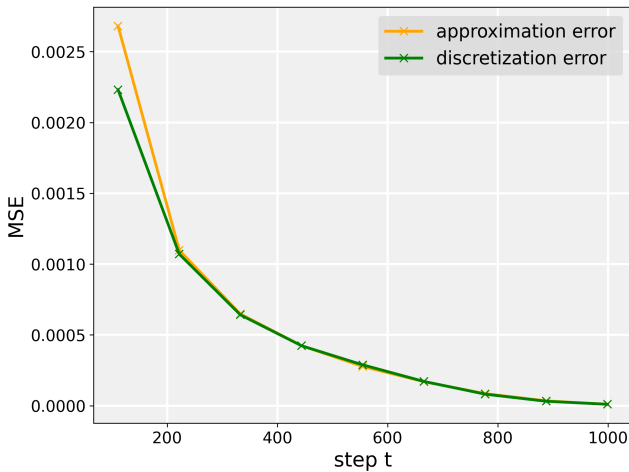


Fig. 12: Empirical verification of error trends in guided sampling tasks. To ensure the observations from Figure 2 are not limited to unconditional task, this figure illustrates the error disentanglement for classifier-guided generation on ImageNet. The curve confirms the universal monotonic trend that approximation error consistently decreases as the timestep t increases toward T . This consistency across different generation settings provides the fundamental theoretical grounding for the SynBoost design.

MSE for distributions measurement in diffusion models.

Adopting MSE to measure the distributions divergence in diffusion models is grounded with both theoretical guarantee and sufficient empirical support from classical and representative papers. (1) Employing MSE to measure the distribution divergence in this special case is theoretically guaranteed. [47] discusses MSE as a special case of maximum likelihood estimation when the error follows a Gaussian distribution. In the context of deep learning, [48] discusses the application of MSE, particularly in error measurement in generative models.

Since in the context of diffusion models, gaussian distribution is the essential and default choice. MSE is thus a rational metric to measure the distribution divergence. (2) It is also a common practice of previous sampler pappers, that employing MSE to measure distribution distance. For example, the mainstream samplers (our baselines), including DPM-Solver++ [25] and UniPC [26], also employs MSE (l2 distance) to compare the convergence error between the results of different methods and 1000-step DDIM. Besides, EDM [39] focuses on the discretization error and also proposes to leverage root mean square error (RMSE) to measure the distribution distance between one Euler iteration and a sequence of multiple smaller Euler iterations, representing the ground truth.

Discussions and limitations. Besides, despite the effectiveness of SynBoost, it still lags behind training-based methods [14], [18] with one-step generation. How to further close the gap between training-free methods and training-based methods requires future efforts.

V. CONCLUSION

In this work, we re-examine the fundamental sampling process of diffusion probabilistic models and reveal that the total sampling error is composed of two distinct parts: the well-studied discretization error and the under-explored approximation error. While existing fast samplers primarily focus on mitigating discretization error through high-order Taylor expansions, our dual-error disentanglement strategy elucidates that approximation error, arising from the neural network’s imprecise estimation of the ground-truth score function, is of a comparable magnitude and significantly impacts sampling quality. Building on the empirical discovery that approximation error monotonically decreases as the timestep t increases, we propose SynBoost, a unified and training-free acceleration framework that simultaneously addresses both error sources. By blending current noise estimations with more accurate predictions from larger timesteps, SynBoost seamlessly integrates as a plug-and-play strategy with existing solvers like DDIM, DPM-Solver, and UniPC. We verify the effectiveness of our method through extensive experiments across unconditional and conditional generation tasks in both

pixel-space and latent-space, demonstrating that SynBoost significantly elevates sample quality and efficiency, particularly in regimes with extremely limited sampling steps.

REFERENCES

- [1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep Unsupervised Learning Using Nonequilibrium Thermodynamics,” in *International Conference on Machine Learning*, 2015, pp. 2256–2265.
- [2] J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-Based Generative Modeling through Stochastic Differential Equations,” in *International Conference on Learning Representations*, 2020, pp. 1–36.
- [4] P. Dhariwal and A. Nichol, “Diffusion Models Beat GANs on Image Synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.
- [5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [6] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical Text-Conditional Image Generation with CLIP Latents,” *arXiv preprint arXiv:2204.06125*, 2022.
- [7] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.
- [8] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations,” in *International Conference on Learning Representations*, 2021, pp. 1–33.
- [9] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video Diffusion Models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.
- [10] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, “Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets,” *arXiv preprint arXiv:2311.15127*, 2023.
- [11] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, “WaveGrad: Estimating Gradients for Waveform Generation,” in *International Conference on Learning Representations*, 2020, pp. 1–15.
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 1–9, 2014.
- [13] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” in *International Conference on Learning Representations*, 2014.
- [14] T. Salimans and J. Ho, “Progressive Distillation for Fast Sampling of Diffusion Models,” in *International Conference on Learning Representations*, 2022, pp. 1–21.
- [15] C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans, “On Distillation of Guided Diffusion Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 297–14 306.
- [16] L. Yu, W. Xiang, K. Han, G. Liu, and R. Kompella, “Comp-Diff: A Unified Pruning and Distillation Framework for Compressing Diffusion Models,” *IEEE Transactions on Multimedia*, vol. 27, pp. 8486–8497, 2025.
- [17] B. Qin, W. Ye, C. Zhang, Q. Yu, W. Zhang, S. Tang, and Y. Zhuang, “Truncate Diffusion: Efficient Video Editing With Low-Rank Truncate,” *IEEE Transactions on Multimedia*, vol. 27, pp. 7093–7108, 2025.
- [18] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency Models,” in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 32 211–32 252.
- [19] J. Song, C. Meng, and S. Ermon, “Denoising Diffusion Implicit Models,” in *International Conference on Learning Representations*, 2020, pp. 1–20.
- [20] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in around 10 Steps,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 5775–5787, 2022.
- [21] Q. Zhang and Y. Chen, “Fast Sampling of Diffusion Models with Exponential Integrator,” in *International Conference on Learning Representations*, 2022, pp. 1–33.
- [22] H. Yu, L. Shen, J. Huang, M. Zhou, H. Li, and F. Zhao, “Debias the Training of Diffusion Models,” *arXiv preprint arXiv:2310.08442*, 2023.
- [23] Q. Zhang, M. Tao, and Y. Chen, “gDDIM: Generalized Denoising Diffusion Implicit Models,” in *International Conference on Learning Representations*, 2022, pp. 1–31.
- [24] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, “Pseudo Numerical Methods for Diffusion Models on Manifolds,” in *International Conference on Learning Representations*, 2022, pp. 1–23.
- [25] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models,” *Machine Intelligence Research*, vol. 22, no. 4, pp. 730–751, 2025.
- [26] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu, “UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 49 842–49 869, 2023.
- [27] X. Wang, A.-D. Dinh, D. Liu, and C. Xu, “Boosting Diffusion Models with an Adaptive Momentum Sampler,” in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 1416–1424.
- [28] G. Zhang, K. Niwa, and W. B. Kleijn, “Lookahead Diffusion Probabilistic Models for Refining Mean Estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1421–1429.
- [29] Y. Xu, M. Deng, X. Cheng, Y. Tian, Z. Liu, and T. Jaakkola, “Restart Sampling for Improving Generative Processes,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 76 806–76 838, 2023.
- [30] K. Zheng, C. Lu, J. Chen, and J. Zhu, “DPM-Solver-v3: Improved Diffusion ODE Solver with Empirical Model Statistics,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 55 502–55 542, 2023.
- [31] P. E. Kloeden and R. Pearson, “The Numerical Solution of Stochastic Differential Equations,” *The ANZIAM Journal*, vol. 20, no. 1, pp. 8–12, 1977.
- [32] H. Guo, C. Lu, F. Bao, T. Pang, S. Yan, C. Du, and C. Li, “Gaussian Mixture Solvers for Diffusion Models,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 25 598–25 626, 2023.
- [33] M. Gonzalez, N. Fernandez Pinto, T. Tran, H. Hajri, N. Masmoudi *et al.*, “SEEDS: Exponential SDE Solvers for Fast High-Quality Sampling from Diffusion Models,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 061–68 120, 2024.
- [34] R. Wang, B. Zhu, J. Li, L. Yuan, and C. Zhang, “Adaptive Stochastic Coefficients for Accelerating Diffusion Sampling,” in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, vol. 36, 2025, pp. 1–24.
- [35] B. Zhu, R. Wang, T. Zhao, H. Zhang, and C. Zhang, “Distilling Parallel Gradients for Fast ODE Solvers of Diffusion Models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 19 557–19 566.
- [36] M. Chen, K. Huang, T. Zhao, and M. Wang, “Score Approximation, Estimation and Distribution Recovery of Diffusion Models on Low-Dimensional Data,” in *International Conference on Machine Learning*, PMLR, 2023, pp. 4672–4712.
- [37] R. Hunter, L. Dudziak, M. S. Abdelfattah, A. Mehrotra, S. Bhattacharya, and H. Wen, “Fast Inference Through The Reuse Of Attention Maps In Diffusion Models,” *arXiv preprint arXiv:2401.01008*, 2023.
- [38] M. Hochbruck and A. Ostermann, “Exponential Integrators,” *Acta Numerica*, vol. 19, pp. 209–286, 2010.
- [39] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the Design Space of Diffusion-Based Generative Models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 565–26 577, 2022.
- [40] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li, “Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis,” *arXiv preprint arXiv:2306.09341*, 2023.
- [41] J. Ho and T. Salimans, “Classifier-Free Diffusion Guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2022, pp. 1–14.
- [42] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop,” *arXiv preprint arXiv:1506.03365*, 2015.
- [43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

- [44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
- [45] W. Peebles and S. Xie, "Scalable Diffusion Models with Transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.
- [46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6000–6010, 2017.
- [47] G. E. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, 2011.
- [48] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [49] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.
- [50] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "Completely Blind" Image Quality Analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2012.



Hu Yu received the B.Eng. degree in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2021.

He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (USTC), Hefei, China. His research interests include generative models, and representation learning.



Hao Luo received the BS and PhD degrees from Zhejiang University, China, in 2015 and 2020, respectively. He is currently with the DAMO Academy, Alibaba Group and Hupan Lab, Zhejiang Province. His research interests include computer vision, vision transformer, generate models, and deep learning.

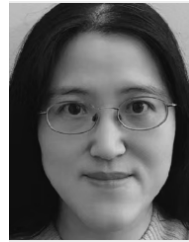


Xueyang Fu received the Ph.D. degree in signal and information processing from Xiamen University, Xiamen, China, in 2018.

He was a Visiting Scholar with Columbia University, New York, NY, USA, sponsored by the China Scholarship Council, from 2016 to 2017. He is currently a Professor and PhD Advisor with the Department of Automation, University of Science and Technology of China, Hefei, China. His research interests include machine learning and image processing.



Jie Huang received the Ph.D. degree in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2024. He is currently an algorithm expert in Joy Future Academy, JD Group. His research interests include vision restoration and generation.



Fan Wang received the B.S. and M.S. degrees from the Department of Automation, Tsinghua University, Beijing, China, and the Ph.D. degree from the Department of Electrical Engineering, Stanford University, CA, USA. She is currently with the Alibaba Group as a Senior Staff Algorithm Engineer. Her research interests include object tracking and recognition, 3D vision, and multi-sensor fusion.



Feng Zhao (Member, IEEE) received the B.Eng. degree in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2000, and the M.Phil. and Ph.D. degrees in computer vision from The Chinese University of Hong Kong, Hong Kong, in 2002 and 2006, respectively. From 2006 to 2018, he was a Post-Doctoral Fellow, Research Fellow, and permanent Senior Lecturer in Hong Kong, Singapore, and the United Kingdom. Since 2019, he has been a Professor and a Ph.D. Advisor with USTC. He has

authored or coauthored more than 170 papers in referred international journals and conferences, including IEEE Transactions (e.g., TPAMI, TIP, TNNLS, TMM), Pattern Recognition, NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, IJCAI, AAAI, ACM MM, EMNLP, and ACL. His research interests include foundation model, AIGC, AI4Science, machine learning, computer vision, and image processing. Prof. Zhao was the recipients of the Best Paper Award in ICIDM 2018 and the Outstanding Paper Award in ACL 2024, the Champion of IJCAI-PRICAI 2020 3D AI Challenge, the Runner-up of CVPR 2021 Streaming Perception Challenge, and the Champions of ICCV 2023 Continual Test-time Adaptation for Semantic Segmentation Challenge and Multitask Learning for Robustness Challenge. He was the Program Committee Chairs of the International Conference on AI-Generated Content (AIGC) 2023-2025, and the Area Chairs of ACM MM 2023-2026, ACL 2025-2026, and AAAI 2026.

Supplementary Material for “SynBoost: A Synergistic Framework for Fast Sampling of Diffusion Models”

This supplementary document is organized as follows:

- Section S-A shows the quantitative comparisons.
- Section S-B shows comparison with SDE-based sampler.
- Section S-C explores the performance upper bound of the baselines and SynBoost.
- Section S-D shows the sampling diversity metric.
- Section S-E presents more explanations of equation 9 in the manuscript.
- Section S-F depicts more visual results.

APPENDIX S-A

QUANTITATIVE COMPARISON

In the main manuscript, we show the quantitative comparison with NFE-FID curve. In this part, we additionally present all comparisons in Table V, VI and VII. In Table V, we depict the results of unconditional sampling with FID metric on LSUN Bedroom and ImageNet datasets. In Table VI, we show the results of class-conditional sampling with FID metric and different guidance scales. In Table VII, we show the results on text-conditional sampling with human preference score $\uparrow$ with the human preference model HPD v2 [40].

APPENDIX S-B

COMPARISON WITH SDE-BASED SAMPLER

Large step size in stochastic differential equations (SDEs) violates the randomness of the Wiener process [31] and often causes non-convergence. Therefore, SDE-based sampler usually adopts hundreds of NFEs for inference. Certain methods [32], [33] propose to accelerate SDE solvers but still require hundreds of steps for inference. Restart [29] proposes to combine SDE and ODE via introducing stochasticity into the ODE process. These methods make promising attempts to accelerate SDE samplers, while still lag behind ODE solvers in speed. Besides, SDE-based sampler leads to stochastic generation, compared to the deterministic generation of ODEs. As shown in Figure 13, we depict the visual comparison between these various samplers, including the SDE-based sampler DDPM [2], the SDE-ODE combined sampler Restart [29], as well as the ODE-based sampler DDIM [19] and its enhanced version with our SynBoost. DDPM suffers from blurry results with small NFEs. Restart effectively elevates the speed of DDPM but still generates low-quality images with small NFEs. Besides, both DDPM and Restart lead to stochastic generation. In contrast, DDIM sampler performs better with finer details and structure. Further, with our SynBoost framework, the sampling quality and speed of DDIM are substantially boosted.

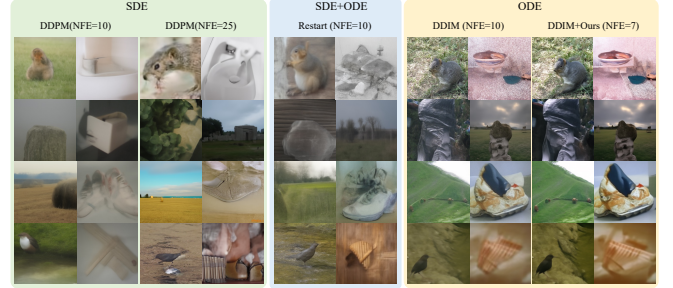


Fig. 13: Visual comparison with SDE-based samplers. All images are generated by sampling from a DPM trained on ImageNet 256x256 [4]. Our method is superior to both SDE and ODE samplers in quality and speed, generating more plausible samples with more visual details and higher contrast.

APPENDIX S-C

EXPLORING THE UPPER BOUND OF SYNBOOST

It is important and of practical value to explore the performance upper bound of existing samplers and our SynBoost for fast sampling. Concretely speaking, we desire to reveal the minimal sampling step required by existing samplers to generate visually clear and pleasing images. We adopt human preference as well as two well-known no-reference image quality assessment indicators: BRISQUE $\downarrow$ [49] and NIQE $\downarrow$ [50] to assess the visual results. As shown in Figure 14, we depict the visual comparison under minimal sampling step of different samplers, and conclude two main conclusions. SynBoost achieves a larger minimal step reduction than increasing sampler order. For example, 2-order DPM-Solver reduces the minimal step from 15 to 8, compared to 1-order DDIM. While, our SynBoost enables DDIM to achieve minimal-step of 7.

Besides, SynBoost can also significantly reduce the minimal step requirement of high order samplers, like DPM-Solver and DPM-Solver++. For example, SynBoost further reduces the minimal sampling step of DPM-Solver from 8 to 6. This validates the generality and robustness of SynBoost.

APPENDIX S-D

SAMPLING DIVERSITY

We also investigate the diversity of the images generated by SynBoost. In Table VIII, we compare the sampling diversity of SynBoost and base samplers with the inception score (IS) metric on ImageNet dataset. SynBoost can consistently improve the sampling diversity on various NFEs.

TABLE V: Sample quality of unconditional sampling measured by FID ↓ on LSUN Bedroom and ImageNet datasets, varying the number of function evaluations (NFE).

Dataset	Sampling Method \ NFE	5	6	7	8
LSUN Bedroom	DDIM (Base)	51.482	32.470	22.533	17.775
	DDIM (+Ours)	36.288	18.738	13.352	12.744
	DPM-Solver (Base)	24.607	17.191	13.722	11.766
	DPM-Solver (+Ours)	16.270	12.947	11.776	11.257
	DPM-Solver++ (Base)	24.378	16.705	13.414	11.635
	DPM-Solver++ (+Ours)	16.594	13.295	12.416	11.218
ImageNet	DDIM (Base)	63.653	48.191	40.295	35.047
	DDIM (+Ours)	49.379	32.729	26.147	23.084
	DPM-Solver (Base)	35.673	28.797	24.729	22.400
	DPM-Solver (+Ours)	28.353	23.653	21.261	19.974
	DPM-Solver++ (Base)	35.118	28.342	24.298	22.003
	DPM-Solver++ (+Ours)	28.602	24.022	21.607	20.486

TABLE VI: Sample quality of class-conditional sampling measured by FID ↓ on ImageNet 256×256 [4], varying the number of function evaluations (NFE) and guidance scale.

Guidance Scale	Sampling Method \ NFE	5	6	7	8
2.0	DDIM (Base)	32.599	22.894	18.043	15.552
	DDIM (+Ours)	26.018	17.316	13.798	12.460
	DPM-Solver (Base)	16.549	12.901	10.907	9.807
	DPM-Solver (+Ours)	14.046	11.210	9.902	9.240
	DPM-Solver++ (Base)	16.753	12.962	10.960	9.799
	DPM-Solver++ (+Ours)	14.675	11.701	10.296	9.582
4.0	DDIM (Base)	31.258	21.389	16.751	14.202
	DDIM (+Ours)	29.265	18.645	14.401	12.554
	DPM-Solver (Base)	16.909	12.548	10.685	9.707
	DPM-Solver (+Ours)	15.759	11.790	10.163	9.380
	DPM-Solver++ (Base)	17.381	12.832	10.842	9.791
	DPM-Solver++ (+Ours)	16.695	12.350	10.546	9.605

TABLE VII: Sample quality of text-conditional sampling measured by human preference score ↑ (human preference model HPD v2 [40]) with captions from MS-COCO2014 validation dataset, varying the number of function evaluations (NFE).

Sampling Method \ NFE	5	10	15	20
DDIM (Base)	0.21492	0.25061	0.25831	0.26247
DDIM (+Ours)	0.22828	0.25714	0.26168	0.26430
Sampling Method \ NFE	5	6	7	8
DPM-Solver (Base)	0.24097	0.25014	0.25461	0.25803
DPM-Solver (+Ours)	0.24687	0.25412	0.25707	0.25904
Sampling Method \ NFE	5	6	7	8
DPM-Solver++ (Base)	0.24146	0.25075	0.25521	0.25859
DPM-Solver++ (+Ours)	0.24637	0.25330	0.25607	0.25944

TABLE VIII: Comparisons of sampling diversity. We compute the IS score on ImageNet dataset, where SynBoost consistently improves the sampling diversity.

IS	DDIM				DPM-Solver				DPM-Solver++			
	5	6	7	8	5	6	7	8	5	6	7	8
Base	29.9	40.3	46.8	52.6	54.5	61.8	67.5	72.5	54.5	62.1	68.7	71.3
Ours	38.7	54.4	64.2	69.9	61.6	70.0	74.3	75.4	58.6	67.1	71.4	72.7

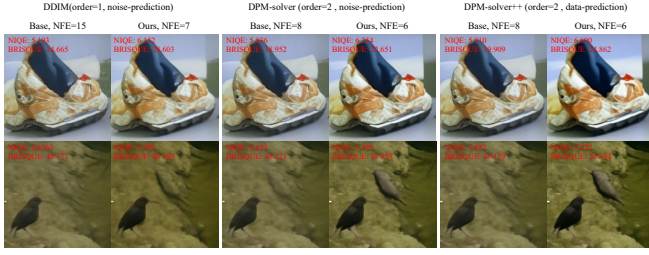


Fig. 14: The minimal steps required to generate visually clear images. All images are generated unconditionally. Our method can consistently lower the minimal steps for clear image generation. Best viewed in color.

APPENDIX S-E

MORE EXPLANATIONS OF EQUATION 9 IN THE MANUSCRIPT

Specifically, we detail: 1) its strict mathematical derivation from the exact DDIM sampling step; 2) the theoretical expansion of the direct substitution baseline; and 3) empirical evidence demonstrating the severe numerical instability (e.g., generation failure) inherently caused by direct substitution.

- **Why mixing outperforms direct substitution.** While $\epsilon_\theta(x_\tau, \tau)$ provides a more accurate estimation of the score function for timestep τ , completely replacing the current update direction D_{t_i} with it would introduce a severe temporal mismatch. The network would ignore the local gradient at t_i . Our mixing formula (Eq. 9) elegantly avoids this by preserving the current step’s data estimation while replacing only the noise estimation. Specifically, the exact sampling formulation of DDIM is:

$$x_{t_i}^{base} = \alpha_{t_i} x_\theta(x_{t_{i-1}}, t_{i-1}) + \sigma_{t_i} \epsilon_\theta(x_{t_{i-1}}, t_{i-1}). \quad (21)$$

When expressed in the unified ODE form, the base update direction is:

$$D_{t_i}^{base} = \epsilon_\theta(x_{t_{i-1}}, t_{i-1}). \quad (22)$$

If we substitute the noise estimation part with the more accurate prior $\epsilon_\theta(x_\tau, \tau)$, but **strictly retain the current local data prediction** $x_\theta(x_{t_{i-1}}, t_{i-1})$, the modified sampling step becomes:

$$x_{t_i}^{ours} = \alpha_{t_i} x_\theta(x_{t_{i-1}}, t_{i-1}) + \sigma_{t_i} \epsilon_\theta(x_\tau, \tau). \quad (23)$$

By algebraically rewriting this modified step back into the unified ODE solver format (Eq. 6), the corresponding update direction $D_{t_i}^{ours}$ naturally emerges as:

$$D_{t_i}^{ours} = \left(1 + \frac{1}{e^{h_i} - 1}\right) \epsilon_\theta(x_{t_{i-1}}, t_{i-1}) - \frac{1}{e^{h_i} - 1} \epsilon_\theta(x_\tau, \tau). \quad (24)$$

This derivation proves that our mixing strategy is mathematically equivalent to leveraging the accurate noise prior while strictly grounding the transition in the current step’s data prediction. The mixing coefficient $c = 1/(e^{h_i} - 1)$ is not arbitrary, but analytically derived from this process.

- **Mathematical flaw of direct substitution.** To provide deeper mathematical insight into why direct substitution fails, we can eliminate the step size variable h_i and fully

expand the update rule. In ODE-based methods, h_i is defined as $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$, where $\lambda_t = \log\left(\frac{\alpha_t}{\sigma_t}\right)$. Consequently, e^{h_i} can be expanded as:

$$e^{h_i} = \exp\left(\log \frac{\alpha_{t_i}}{\sigma_{t_i}} - \log \frac{\alpha_{t_{i-1}}}{\sigma_{t_{i-1}}}\right) = \frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\sigma_{t_i} \alpha_{t_{i-1}}}. \quad (25)$$

Substituting this into the step size coefficient $\sigma_{t_i}(e^{h_i} - 1)$ yields:

$$\sigma_{t_i}(e^{h_i} - 1) = \sigma_{t_i} \left(\frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\sigma_{t_i} \alpha_{t_{i-1}}} - 1 \right) = \frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\alpha_{t_{i-1}}} - \sigma_{t_i}. \quad (26)$$

By applying this coefficient, the direct substitution update rule becomes:

$$x_{t_i}^{direct} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} x_{t_{i-1}} - \left(\frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\alpha_{t_{i-1}}} - \sigma_{t_i} \right) \epsilon_\theta(x_\tau, \tau). \quad (27)$$

To compare this with the standard DDIM formulation ($x_{t_i} = \alpha_{t_i} x_\theta + \sigma_{t_i} \epsilon_\theta$), we fully expand $x_{t_{i-1}}$ as $x_{t_{i-1}} = \alpha_{t_{i-1}} x_\theta(x_{t_{i-1}}, t_{i-1}) + \sigma_{t_{i-1}} \epsilon_\theta(x_{t_{i-1}}, t_{i-1})$. Plugging this into the equation yields:

$$\begin{aligned} x_{t_i}^{direct} = & \alpha_{t_i} x_\theta(x_{t_{i-1}}, t_{i-1}) + \frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\alpha_{t_{i-1}}} \epsilon_\theta(x_{t_{i-1}}, t_{i-1}) \\ & - \left(\frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\alpha_{t_{i-1}}} - \sigma_{t_i} \right) \epsilon_\theta(x_\tau, \tau). \end{aligned} \quad (28)$$

This fully expanded equation mathematically exposes the critical flaw of direct substitution. Instead of preserving the elegant variance schedule of diffusion models (as shown in our clean $x_{t_i}^{ours}$ formulation), direct substitution results in a disjointed and mathematically unprincipled mixture of local noise $\epsilon_\theta(t_{i-1})$ and prior noise $\epsilon_\theta(\tau)$.

- **Empirical evidence: numerical explosion.** Furthermore, when we implement Equation 9 via direct substitution, the generated images appear completely grey. This empirical failure perfectly corroborates our mathematical derivation and serves as strong evidence against direct substitution.

The root cause of this failure is severe numerical instability. Based on the expanded formula above, we define the coefficient for the noise term as $C = \frac{\alpha_{t_i} \sigma_{t_{i-1}}}{\alpha_{t_{i-1}}}$. In the early sampling stages (i.e., when t is large, approaching $T = 1000$), $\alpha_{t_{i-1}}$ is an extremely small value (e.g., 4×10^{-5} or less). With $\alpha_{t_{i-1}}$ in the denominator, the coefficient C explodes, approaching infinity. This causes the values within the tensor to overflow, producing NaNs or astronomically large numbers. During image saving, these anomalous values are forcibly clipped or normalized, inevitably resulting in the entirely grey images. This proves that direct substitution fundamentally destroys the numerical stability of the ODE solver.

APPENDIX S-F

MORE VISUAL RESULTS

In this part, we show more visual results with various samplers (DDIM, DPM-Solver, and DPM-Solver++), sampling types (unconditional, class-conditional, and text-conditional sampling), sampling spaces (pixel and latent space), NFEs (5, 6, 7, and 8), and guidance scales (0.0 and 2.0).

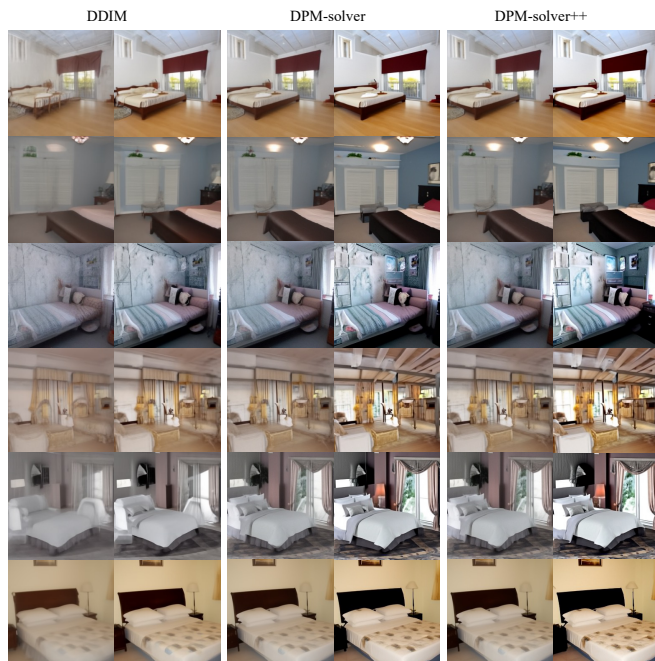


Fig. 15: Visual results of unconditional sampling on LSUN Bedroom dataset.



Fig. 16: Visual results of unconditional sampling on ImageNet dataset.

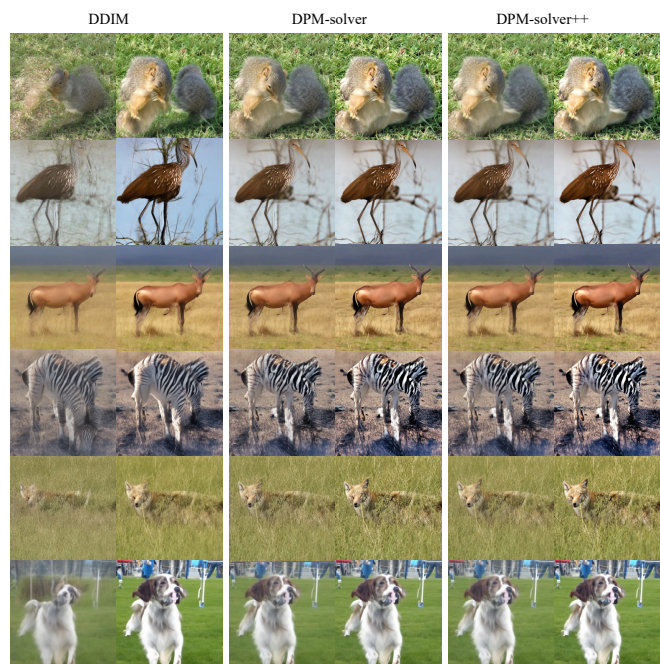


Fig. 17: Visual results of class-conditional sampling on ImageNet dataset with guidance scale 2.0.