Training-Free Refinement of Flow Matching with Divergence-based Sampling

Yeonwoo Cha Jaehoon Yoo Semin Kim
Yunseo Park Jinhyeon Kwon Seunghoon Hong†

KAIST, South Korea

https://yeonwoo378.github.io/official_fds

Abstract. Flow-based models learn a target distribution by modeling a marginal velocity field, defined as the average of sample-wise velocities connecting each sample from a simple prior to the target data. However, when sample-wise velocities conflict at the same intermediate state, this averaged velocity can misguide samples toward low-density regions, degrading generation quality. To address this issue, we propose the Flow Divergence Sampler (FDS), a training-free framework that refines intermediate states before each solver step. Our key finding reveals that the severity of this misguidance is quantified by the divergence of the marginal velocity field that is readily computable during inference with a well-optimized model. FDS exploits this signal to steer states toward less ambiguous regions. As a plug-and-play framework compatible with standard solvers and off-the-shelf flow backbones, FDS consistently improves fidelity across various generation tasks including text-to-image synthesis, and inverse problems.

Keywords: Flow Matching · Training-Free · Plug-and-Play

1 Introduction

Flow Matching (FM) [6, 23, 31, 38] have emerged as a powerful paradigm for modeling complex target distributions, enabling high-fidelity synthesis across image [32, 33, 41], audio [15, 24], and video [13, 37]. Given a simple prior and a complex target data distribution, FM learns a mapping between them through a time-dependent velocity field. Although each paired source-target sample induces its own sample-wise conditional velocities, FM learns a single marginal velocity at each intermediate state, which is an averaged velocity over all sample-wise velocities passing through that state.

While the marginal velocity field induces a valid mapping from source to target distribution, it introduces a fundamental limitation arising from directional conflicts among the underlying sample-wise velocities. Specifically, these velocities are locally multi-modal and can conflict, pointing in different directions at the same location, as illustrated in Fig. 1. At such states, a single marginal velocity cannot align with any valid mode, but instead follows their average direction. As a consequence, generative trajectories that pass through those states can drift toward low-density regions, producing blurry or degraded outputs.

† Corresponding author.

Prior works [9, 45] have addressed this problem by redesigning the formulation to model the multi-modal distribution of these conflicting velocities. While effective, these approaches require costly training, preventing the direct application of the strong off-the-shelf flow models that are already available.

In this paper, we ask a different question: can this problem be mitigated at inference time *without* retraining the model? Our core idea is to correct the trajectory at inference time rather than modifying the velocity field itself. Before taking the next solver step, we update the current intermediate state to a nearby point where the same pre-trained vector field is expected to provide less ambiguous velocity, and then continue integration from there. Since this refinement operates on the state rather than the model itself, it can be applied in a plug-and-play fashion to any off-the-shelf model and combined with standard solvers such as Euler and Heun.

The central challenge is that this local ambiguity is not directly observable at inference time, as evaluating the underlying dispersion of sample-wise velocities requires access to the training dataset. Our key theoretical finding establishes that this dispersion admits a closed-form expression in terms of the divergence of the marginal velocity field. In practice, this motivates a data-free surrogate based on the divergence of the pre-trained model. Building on this observation, we propose the Flow Divergence Sampler (FDS), which steers trajectories away from locally ambiguous regions during sampling. Our extensive experiments show that, under matched compute budgets, FDS outperforms simply taking more solver steps in various settings.

Our contributions are summarized as follows: **(1)** We show that the conditional mean-squared discrepancy between the marginal velocity and sample-wise velocities can be characterized through the divergence of the marginal velocity field, which in turn motivates a practical data-free proxy at inference time. **(2)** We introduce FDS, a plug-and-play sampling method that refines intermediate states toward locally less ambiguous regions without retraining of the underlying model. **(3)** We demonstrate consistent improvements across diverse flow backbones, solvers, and various settings, including class-conditional generation, text-to-image synthesis, and inverse problems.

2 Preliminaries

2.1 Flow Matching

Flow Matching (FM) [23] constructs a continuous probability path p_t that transports a simple prior distribution $p_0 = \mathcal{N}(0, I)$ to a target data distribution p_1. In general, this path can be defined through an interpolant between the noise sample $x_0 \sim p_0$ and a data sample $x_1 \sim p_1$ by

$$x_t = \alpha_t x_1 + \beta_t x_0, \quad x_t \sim p_t, \tag{1}$$

where α_t and β_t are differentiable schedules that monotonically increase and decrease, respectively, satisfying $(\alpha_0, \beta_0) = (0, 1)$ and $(\alpha_1, \beta_1) = (1, 0)$ such that x_t transitions from noise to data.

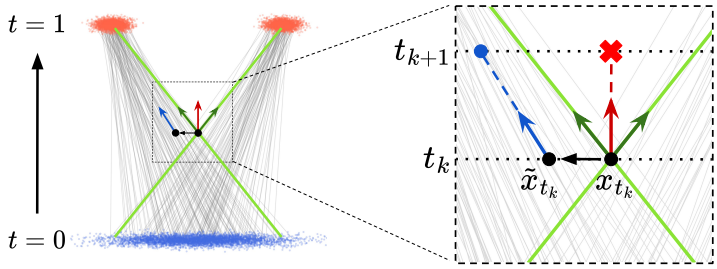


Fig. 1: Overview of FDS. In flow matching, multiple sample-wise velocities (green arrows) may pass through the same intermediate state x_{t_k}. In such cases, the marginal velocity field becomes their average (red arrow), causing subsequent ODE integration to drift toward a low-density region. To counteract this, FDS first updates the current state to a nearby state $\tilde{x}_{t_k}$ where the local directional conflict is weaker. ODE integration from $\tilde{x}_{t_k}$ is then follows more reliable direction towards a mode (blue circle).

A velocity field that transports p_t along this path can be characterized through the *sample-wise velocity* of each interpolant:

$$v_t(x_0, x_1) := \frac{dx_t}{dt} = \dot{\alpha}_t x_1 + \dot{\beta}_t x_0. \tag{2}$$

Importantly, v_t is a variable that varies across sample pairs (x_0, x_1), not a vector field defined over x_t.

The Conditional Flow Matching (CFM) framework trains a neural network $u_\theta(x_t, t)$ to predict this velocity by minimizing

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\left\|u_\theta(x_t, t) - v_t\right\|^2\right], \tag{3}$$

with $t \sim \mathcal{U}[0, 1]$, $x_0 \sim p_0$, and $x_1 \sim p_1$. Because Eq. (3) is a least-squares objective, its pointwise minimizer at each (x_t, t) is the conditional expectation

$$u_t(x_t) := \mathbb{E}\left[v_t \mid x_t\right], \tag{4}$$

termed the *marginal velocity field*. Therefore, a well-trained flow model satisfies $u_\theta(x_t, t) \approx u_t(x_t)$. At inference time, samples are generated by integrating the learned field

$$\frac{dx_t}{dt} = u_\theta(x_t, t), \quad x_0 \sim p_0, \tag{5}$$

from $t = 0$ to $t = 1$ over a finite number of discrete timesteps using standard numerical solvers, such as first-order Euler method or second-order Heun's method.

2.2 Discrepancy Between Marginal and Sample-wise Velocities

In flow matching, interpolants from different sample pairings can pass through the same intermediate location x_t. At such crossings, the sample-wise velocities v_t associated with each sample pair may point toward distinct target modes, while the marginal field $u_t(x_t)$ can only represent their average (Eq. (4)). In

that case, the marginal velocity may point toward a low-density region between modes rather than toward any valid mode (red cross in Fig. 1), causing the ODE trajectory during inference to produce blurry or degraded samples [9, 45].

We quantify the severity of this degradation-inducing term through the residual of the optimal CFM predictor:

$$\mathcal{L}^*_{\mathrm{CFM}}(x_t, t) = \mathbb{E}\left[\left\|u_t(x_t) - v_t\right\|^2 \,\middle|\, x_t\right], \tag{6}$$

which measures the *discrepancy* between v_t around its conditional mean $u_t(x_t)$. The discrepancy is small when the sample-wise velocities are locally consistent, and large when multiple incompatible velocities coexist at the same state. Importantly, this residual is irreducible: it persists even for a perfectly optimized flow model using Eq. (3), since it reflects the intrinsic multi-modality of v_t at x_t rather than optimization error.

To address this issue, recent works [9, 30, 45] redesign training to explicitly resolve crossings, by modeling the full multi-modal velocity distribution with an auxiliary network. However, these approaches require costly training, often together with substantial architectural or objective modifications, and thus cannot directly leverage off-the-shelf flow models pre-trained on large-scale data.

3 Method

To address the issues discussed in Sec. 2.2, we seek an inference-time method that can be attached plug-and-play to off-the-shelf flow models. Rather than retraining the network or attempting to alter the learned marginal velocity field itself, we keep the pre-trained flow model fixed. Instead, our approach intervenes exclusively during the sampling phase, performing targeted spatial refinement at the exact same timestep t.

The central idea of our method is to refine the trajectory, rather than the velocity field. When interpolant crossings make the learned marginal velocity locally unreliable, we do not attempt to re-estimate the underlying multi-modal velocity distribution. Instead, at a solver state x_t, we seek a nearby state $\tilde{x}_t$ where the same pre-trained velocity field is expected to be more reliable, and then continue integration from $\tilde{x}_t$. In this sense, our method performs a spatial refinement of the intermediate state x_t, rather than a modification of the model itself (Fig. 1).

A natural objective for this refinement is the discrepancy $\mathcal{L}^*_{\mathrm{CFM}}(x_t, t)$ (Eq. (6)) between the marginal and sample-wise velocities, since it directly measures the local ambiguity of the marginal velocity at the current state. However, this quantity is not observable at inference time, as evaluating Eq. (6) requires access to the ground-truth sample-wise velocities v_t that depend on training data.

To address this challenge, we introduce the **Flow Divergence Sampler (FDS)**. In this section, we first theoretically prove that this intractable, data-dependent discrepancy can be explicitly measured using a data-free surrogate at inference time (Sec. 3.1). Building upon this, we outline our sampling strategy designed to seamlessly navigate the intermediate state x_t toward reliable, low-discrepancy regions without excessive computational overhead (Sec. 3.2).

3.1 Flow Divergence as Discrepancy Surrogate

Recall that the marginal velocity is the conditional mean (Eq. (4)). Thus, the discrepancy in Eq. (6) measures the local spread of sample-wise velocities around their average $u_t(x_t)$. When multiple paths passing through the same x_t, this spread becomes large, and the averaged direction may become unreliable. Since the individual velocities are unobservable at inference-time, we instead examine how their average changes under small spatial perturbations of x_t. This leads to the spatial derivative of the marginal field, $J_{x_t} u_t(x_t)$, whose trace is the divergence $\nabla_{x_t} \cdot u_t(x_t)$.

The following theorem shows that the ambiguity is exactly captured by the divergence of the marginal velocity, up to known schedule-dependent coefficients.

Theorem 1. *For any t such that $\alpha_t \neq 0$, the optimal CFM residual satisfies*

$$\mathcal{L}^*_{CFM}(x_t, t) = \mathbb{E}\left[\|u_t(x_t) - v_t\|^2 \;\middle|\; x_t\right] = \frac{\dot{\alpha}_t \beta_t - \alpha_t \dot{\beta}_t}{\alpha_t}\Big(\beta_t \nabla_{x_t} \cdot u_t(x_t) - \dot{\beta}_t d\Big), \quad (7)$$

where d is the dimensionality of the data.

The proof is provided in the App.A. Thm. 1 shows that the inaccessible discrepancy can be expressed entirely in terms of the divergence of the marginal velocity field and known schedule-dependent coefficients α_t and β_t. Since all coefficients are constant with respect to x_t for a fixed t, it implies that minimizing $\mathcal{L}^*_{\mathrm{CFM}}(x_t, t)$ over nearby candidates is equivalent to minimizing the spatial divergence $\nabla_{x_t} \cdot u_t(x_t)$.

In practice, the true marginal velocity field u_t is unavailable, so we replace it with the pre-trained model u_θ, i.e., $u_\theta(x_t, t) \approx u_t(x_t)$, and define the following inference-time surrogate:

$$\hat{\delta}_t(x) = \nabla_x \cdot u_\theta(x, t). \quad (8)$$

We use $\hat{\delta}_t(x)$ as a surrogate for local reliability: a lower divergence indicates a lower discrepancy $\mathcal{L}^*_{\mathrm{CFM}}$ and, consequently, a lower risk of discrepancy-induced degradation. This quantity is computable entirely from the pre-trained model u_θ and serves as a data-free, inference-time surrogate for discrepancy. To compute this divergence efficiently, we adopt Hutchinson's trace estimator [16], following its successful application in the literature [8]. The reliability of this surrogate is further discussed and evaluated in App.D.

3.2 Flow Divergence Sampler

Building upon the theoretical foundations established in Sec. 3.1, we introduce the Flow Divergence Sampler (FDS). At each inference timestep t, the core operational objective of FDS is to optimize the solver state x_t to $\tilde{x}_t$ such that the refined state exhibits lower discrepancy. A direct way to achieve this would be to optimize $\hat{\delta}_t(x)$ with respect to x via gradient descent, yet this is computationally unattractive for large generative models. Since $\hat{\delta}_t(x)$ already contains a first-order spatial derivative of the flow model, differentiating it with respect to x

would require second-order derivatives, which are expensive in high-dimensional settings.

To circumvent this computational overhead, FDS performs a *zero-order local refinement* by applying small random perturbations, a mechanism inspired by [29]. These perturbations are simply scaled by a decaying schedule σ_t, which gradually decreases from σ_{max} to 0. At a solver state x_t, we construct a set of nearby M candidates:

$$x^{(0)} = x_t, \quad x^{(m)} = x_t + \sigma_t \xi^{(m)}, \quad \xi^{(m)} \sim \mathcal{N}(0, I), \quad m = 1, ..., M. \tag{9}$$

Then for each candidate, we evaluate its discrepancy using Eq. (8), and select the best candidate,

$$m^* = \underset{m \in \{0,\dots,M\}}{\arg\min} \ \hat{\delta}_t(x^{(m)}), \qquad \tilde{x}_t \leftarrow x^{(m^*)}, \tag{10}$$

and repeat this procedure for N refinement iterations, using the updated $\tilde{x}_t$ as the base state $x^{(0)}$ if desired. The resulting state is denoted by $\tilde{x}_t$.

As demonstrated, unlike conventional advanced samplers, the key distinction of FDS is that it performs spatial corrections on the state toward a reliable region. Standard high-order ODE solvers primarily improve generation by reducing temporal integration error once the current state x_t is fixed. In contrast, FDS performs a spatial intervention at a fixed time t, relocating the intermediate state itself before the next solver step is taken. Because this refinement is applied externally to the base solver, FDS can be attached to Euler, Heun, and other off-the-shelf solvers.

One sampling step with FDS therefore takes the form,

$$\tilde{x}_{t_k} = \text{Refine}(x_{t_k}, t_k; u_\theta), \quad x_{t_{k+1}} = \text{SolverStep}(\tilde{x}_{t_k}, t_k, t_{k+1}; u_\theta). \tag{11}$$

Thus, FDS is a plug-in inference-time module—readily compatible with various base solvers and guidance methods—rather than their replacement. A psuedocode of our framework is provided in the App.B.

Computational Overhead While FDS introduces extra computations to the base solver proportional to the number of refinement iterations N and the number of search candidate M, empirical evaluations reveal a crucial practical insight: successfully bypassing high-discrepancy regions does not demand an expensive computational budget. In practice, we found that setting $N{=}M{=}1$ is sufficient to significantly elevate generation fidelity compared to baseline models with more NFEs, as demonstrated by the extensive evaluations in Sec. 5.2.

To further maximize efficiency, we explicitly deactivate FDS in the early stages of generation ($t < T_{\text{trunc}}$), empirically setting the truncation threshold $T_{\text{trunc}} = 0.5$. Since severe trajectory conflicts predominantly manifest during these earlier denoising stages [2], this temporal constraint ensures optimal resource allocation without much sacrificing quality. Collectively, these spatial and temporal design choices establish FDS as a low-overhead, highly practical inference-time solution. An in-depth analysis of these components—including the choices for the temporal boundary (T_{trunc}) and N, M—is provided in our ablation study (Sec. 6.2).

4 Related Works

4.1 Resolving Crossings on Train Time

Recent studies increasingly highlight the limitations induced by intersecting trajectories in flow matching. To resolve this, some methods fundamentally change the formulation to allow for crossings; HRF [45] and VRFM [9] reformulate the framework to capture the full multi-modal velocity distribution (Fig. 1), and CAF [30] utilizes an acceleration-based approach to navigate away from empty data regions. Conversely, other works [18,20] attempt to straighten the trajectory by introducing auxiliary encoders. While these strategies successfully mitigate trajectory crossings, their strict reliance on fundamental architectural changes and prohibitively expensive from-scratch training renders them highly inefficient, strongly motivating the need for a training-free alternative.

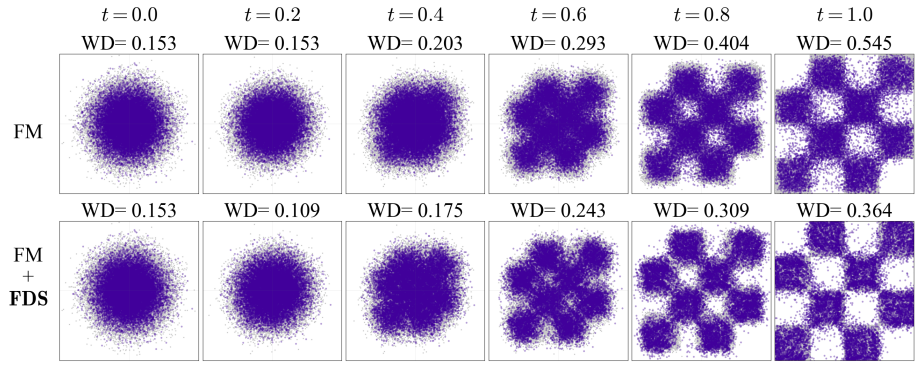
4.2 Enhancing Generation Quality at Inference-Time

To boost generative fidelity and image quality without requiring additional training, recent literature has explored advanced inference solvers to reduce temporal error. Foundational methods [25, 26] employ higher-order numerical approximations to follow trajectory curvature, and techniques like UniPC [46] utilize history buffers to optimize this process for few-step generation. Additionally, recent advanced samplers [27] have been proposed to dynamically allocate computational resources by evaluating trajectory stiffness. However, all these approaches primarily focus on mitigating numerical discretization errors, restricting them to temporal adjustments along a fixed integration path. Our work fundamentally differs by providing spatial enhancements, actively steering the generation away from severe trajectory crossings.

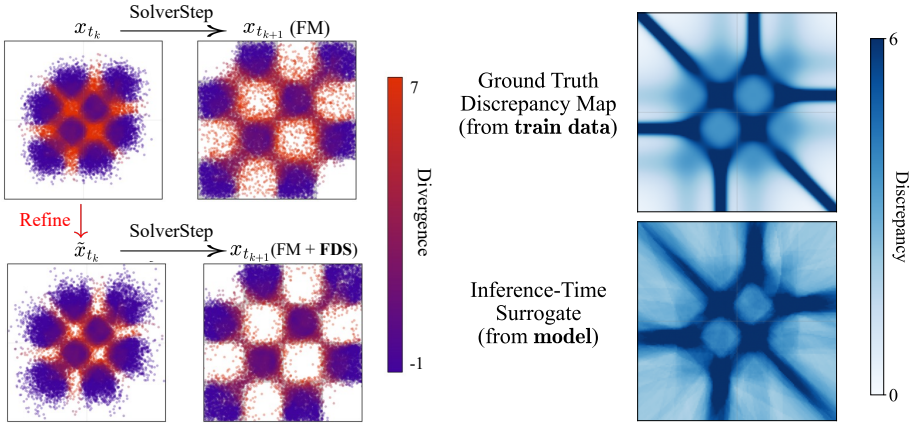
Another line of research focuses on advanced guidance mechanisms that spatially update the state x_t at inference time. For instance, recent studies [1, 39] formulate objectives composed with conditional and unconditional velocities, but are primarily restricted to classifier-free guidance [14] settings. Similarly, other works [10, 43] refine x_t using external reward models or predefined objectives, inherently requiring external signals. In contrast, our approach operates as a versatile, plug-and-play module that requires no external signals and seamlessly integrates with existing inference-time enhancements.

5 Experiments

We validate the effectiveness of FDS across a variety of settings. We first illustrate the intuition and underlying mechanism of FDS on 2D synthetic data in Sec. 5.1. Furthermore, we mainly validate FDS on ImageNet 256×256 [4] and CIFAR-10 [19] in Sec. 5.2, demonstrating its consistent robustness across diverse backbones and solvers. Finally, we highlight the plug-and-play versatility of FDS in diverse tasks such as text-to-image synthesis and inverse problems in Sec. 5.3. Detailed setups for each experiment are placed in the App. C.



(a) FDS yields improved generation quality, reflected by a lower Wasserstein Distance (WD).



(b) FDS *refines* x_t into $\tilde{x}_t$ by steering it toward lower-divergence regions.

(c) The discrepancy map measured by full train data (GT) and the model at $t = 0.6$.

Fig. 2: 2D Synthetic Experiment shows our divergence-based criterion correlates with sample quality. **(a)** FDS achieves more accurate modeling of the target distribution than standard FM, yielding a lower Wasserstein Distance (WD). **(b)** Standard FM passes x_{t_k} directly to the ODE solver, whereas FDS refines x_{t_k} into $\tilde{x}_{t_k}$, moving it to a low-divergence region. **(c)** Discrepancy maps computed from the ground-truth sample-wise velocities (Top) and from our inference-time surrogate using the pre-trained model (Bottom) are highly consistent.

5.1 Synthetic 2D Experiments

To intuitively illustrate the core mechanism of FDS, we first conduct a synthetic experiment mapping a standard Gaussian prior to a 2D checkerboard distribution. Fig. 2a visualizes the generation processes of standard Flow Matching (FM) with an Euler solver alongside our FDS. We quantitatively assess the generation quality using the Wasserstein Distance (WD) to the exact path, reported above each snapshot. As t progresses, FDS consistently achieves a lower WD compared to standard FM. Notably, at $t = 1.0$, FDS significantly reduces the number of stray samples outside the target regions, successfully mitigating quality degradation and ensuring high-fidelity generation.

Understanding Behavior of FDS To further investigate the mechanism, we visualize the intermediate refinement process at $t_k = 0.6$ in Fig. 2b. All samples are color-coded by their initial divergence at x_{t_k}. As shown, FDS effectively steers high-divergence samples (orange) toward low-divergence regions (purple). Following the solver step to $t_{k+1} = 1.0$, the unrefined orange samples stray off the checkerboard. In contrast, the refined samples correctly align with the target distribution. Collectively, these observations demonstrate that our enhanced generation quality directly stems from mitigating the discrepancy between marginal and sample-wise velocities on high-divergence region.

Divergence as a Reliable Surrogate While empirical results support that our proposed surrogate (Eq. (8)) serves as a highly reliable discrepancy indicator during inference, we further validate its accuracy by comparing the true discrepancy (Eq. (6)) with our estimation (Eq. (7)) measured with the surrogate. To this end, we visualize both discrepancy maps in Fig. 2c. The top panel illustrates the true discrepancy explicitly measured by the training data distribution, whereas the bottom panel depicts our data-free estimation via flow model divergence. The strong structural alignment between the two maps implies that our divergence-based estimation effectively captures the underlying true discrepancy, thereby demonstrating the empirical soundness of our approach.

5.2 Main Experiments

Experimental Setup To validate the applicability of FDS on pre-trained models, we conduct experiments on CIFAR-10 [19] and ImageNet 256×256 [4]. We apply FDS to well-established flow-based models: EDM [17] for CIFAR-10 and JiT [21] for ImageNet. To evaluate the efficacy of FDS, we construct compute-matched baselines in Tab. 1 (marked with †) by allocating additional steps to match the wall-clock time required by FDS. This ensures a fair comparison under an equivalent computational budget. The exact computing procedure is detailed in App.C and Tab. 6, reflecting a roughly $1.5\times$ increase in wall-clock time for FDS on the same NFEs. For evaluation, we report Fréchet Inception Distance (FID) [12] and Inception Score (IS) [35].

Results As detailed in Tab. 1, FDS consistently improves the generation quality across various datasets, models, and ODE solvers, in terms of FID, demonstrating the robust applicability of our framework. For compute-matched baselines, while increasing integration steps generally enhances performance, these gains quickly saturate with advanced solvers like Heun. In some cases, such as ImageNet 256×256 with the Heun solver, this even leads to performance stagnation or diminishing returns. Conversely, allocating this same computational budget to FDS yields substantial enhancements in image fidelity across all settings.

As discussed in Sec. 3.2, this performance gap arises because FDS operates on a fundamentally different principle. Rather than solely relying on finer temporal integration, our method introduces targeted spatial updates to the

Table 1: Performance comparison on CIFAR-10 and ImageNet 256×256. FDS consistently improves generation quality in terms of FID across all configurations in the main experiments. † denotes a base solver with an increased NFEs to match the wall-clock time of our framework.

Solver	NFE	CIFAR-10				ImageNet 256×256			
		EDM Cond.		EDM Uncond.		JiT-B/16		JiT-L/16	
		FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID ↓)	IS (↑)	FID (↓)	IS (↑)
Euler	50	3.003	9.576	3.034	9.371	4.151	280.07	3.859	277.09
Euler†	77	2.515	**9.671**	2.550	**9.464**	4.061	**287.15**	3.857	**278.75**
Euler + **FDS**	50	**2.319**	9.660	**2.440**	9.387	**3.799**	278.33	**3.519**	278.16
Heun	99	1.904	9.829	2.021	9.615	3.637	270.18	2.713	330.23
Heun†	153	1.910	9.820	2.034	9.606	3.815	**275.10**	2.886	**333.10**
Heun + **FDS**	99	**1.786**	**9.875**	**1.953**	**9.641**	**3.394**	269.09	**2.496**	329.70

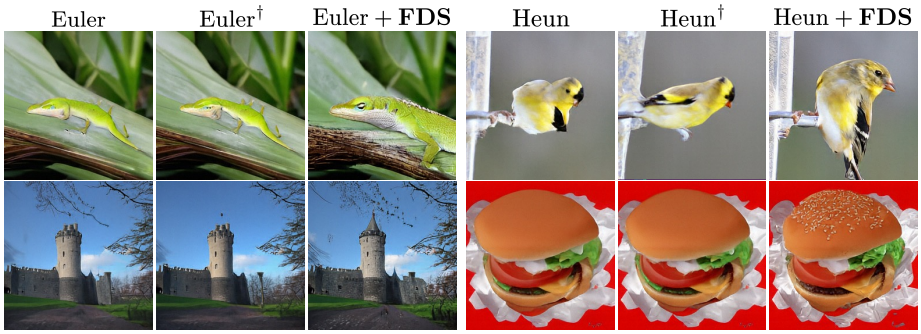


Fig. 3: Qualitative results on ImageNet 256 × 256 with JiT-L/16. Compared to the compute-matched baseline(†), FDS effectively enhances the generation quality.

generation process at fixed timestep t. By steering generative trajectories away from areas of high-discrepancy region, FDS effectively prevents the model from traversing regions that typically cause degraded or blurry outputs. Fig. 3 (right) directly illustrates this advantage: while simply increasing the NFE yields only subtle localized changes, FDS successfully synthesizes fine-grained details—such as distinct sesame seeds on a hamburger—that a naive high-NFE baseline cannot produce. Ultimately, this active spatial refinement at inference-time achieves consistent improvements in fidelity that remain unattainable through the temporal scaling of existing samplers. Additional experimental results, including those on SiT [28], can be found in App. E, while further qualitative results are presented in App. F.

5.3 FDS on Various Tasks

To further validate the robustness of our framework, we provide additional quantitative and qualitative analyses that extend beyond our primary evaluations. Specifically, we explore the versatility of FDS across diverse yet widely adopted

Table 2: Quantitative results for text-to-image synthesis on the DrawBench dataset. FDS consistently improves upon the baselines across most metrics, demonstrating its robustness across different model backbones.

(a) Experiment results on SD3-M

Method	IR	HPSv2	Aes.	CLIP
SD3-M	82.36	27.72	5.70	28.47
SD3-M+**FDS**	**89.33**	**27.76**	**5.72**	**28.76**

(b) Experiment results on FLUX.1

Method	IR	HPSv2	Aes.	CLIP
FLUX.1	92.95	29.37	**6.18**	27.33
FLUX.1+**FDS**	**94.63**	**29.39**	6.14	**27.46**

Fig. 4: Qualitative results on text-to-image synthesis. Images generated with FDS exhibit improved overall quality, characterized by better spatial coherence. More qualitative results can be found in App. F.

guidance settings. This includes evaluating its behavior in text-to-image generation under varying guidance scales, testing its compatibility with standard test-time guidance mechanisms frequently utilized for inverse problems, and assessing its integration with other advanced inference-time methods. The following evaluations demonstrate that FDS can be seamlessly plugged into various generation pipelines, consistently maintaining high fidelity across different configurations

Text-to-Image Generation We evaluate our framework on two different text-to-image generation backbones, SD3-Medium [6] and FLUX.1 [3], using the DrawBench [34] dataset. To comprehensively assess the overall sample quality, we utilize four established reward models widely adopted in text-to-image generation tasks. Specifically, we measure human preference using ImageReward (IR) [42] and HPSv2 [40], evaluate visual appeal via the Aesthetic Predictor [36], and quantify text-image alignment using the CLIP Score [11].

Table 3: Quantitative evaluation on inverse problems. Baselines equipped with FDS demonstrate consistent improvements across all metrics. All reported values are lower-is-better.

Task	Method	FID	LPIPS
Deblur	TFG	64.02	15.50
	TFG + **FDS**	**63.17**	**14.93**
SR ×4	TFG	65.54	18.70
	TFG + **FDS**	**63.14**	**16.23**

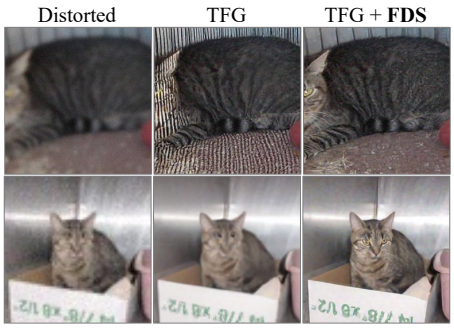


Fig. 5: Qualitative results on inverse problems. (Top) Deblurring. (Bottom) Super-resolution (×4).

Table 4: FDS on diverse inference-time methods. We compare the performance of FDS when integrated with (a) an advanced solver, (b) a higher-order solver, and (c) an advanced CFG framework. The results consistently demonstrate the effectiveness of FDS across all configurations.

(a) FDS on advanced solver.

Method	FID(↓)	IS(↑)
UniPC [46]	6.21	270.8
UniPC+**FDS**	**5.59**	**272.4**

(b) FDS on high-order solver.

Method	FID(↓)	IS(↑)
RK4	3.77	**264.40**
RK4+**FDS**	**3.45**	261.9

(c) FDS on advanced CFG.

Method	FID(↓)	IS(↑)
CFG-Zero* [7]	4.10	**276.8**
CFG-Zero*+**FDS**	**3.78**	276.5

Quantitative results in Tab. 2 and qualitative samples indicate that FDS can be effectively applied to text-to-image generation tasks, offering improvements in visual quality and showing robustness across different backbones. Specifically, integrating our framework helps produce images with enhanced fidelity, as demonstrated by the accurate rendering of fine-grained details like the clock faces and numbers in Fig. 4 (bottom left). Overall, these enhancements support that FDS is a beneficial inference-time module that consistently boosts the performance of advanced text-to-image models. Additional quantitative results for text-to-image generation, including experiments with varying backbone architectures and evaluations on the MS-COCO [22] dataset, are provided in App. E.

Inverse Problem We further demonstrate the plug-and-play capability of FDS by integrating it with inference-time guidance frameworks to enhance generation quality. We assess this validation on image deblurring and super-resolution tasks following the setup of TFG [43] using the Cat [5] dataset, with detailed settings provided in the App. C.

As demonstrated quantitatively in Tab. 3, FDS consistently improves upon the baseline by generating substantially more detailed and realistic images. Notably, it achieves lower FID and LPIPS [44], indicating a superior capacity to capture both global structures and fine-grained details. This quantitative superiority is directly corroborated by the visual comparisons in Fig. 5. While the

baseline often yields blurry or less-realistic samples, our proposed framework successfully retains intricate detailed features. This is clearly evidenced by the rendering of the cat, where fine details such as eye pupils, complex fur textures, and individual whiskers are preserved with enhanced fidelity.

FDS on Advanced Inference-time Methods To comprehensively validate the broad compatibility and plug-and-play capability of FDS, we integrate it into several established methodologies following the settings in Sec. 5.2. We empirically evaluate this adaptability across distinct generative paradigms: an advanced solver (UniPC [46], 20 NFEs), a high-order solver (RK4), and an advanced classifier-free guidance technique (CFG-Zero* [7]). As demonstrated in Tabs. 4a, 4b, and 4c, our framework consistently enhances generation quality in every scenario. These results demonstrate that rather than competing with existing enhancements, FDS serves as a highly compatible module capable of seamlessly elevating performance across diverse inference-time settings.

6 Analysis

6.1 Comparison with Training-Based Frameworks

Besides the benefits of FDS being a training-free and plug-and-play method, we also observe that such approach can outperform training-based methods in terms of generation quality. The comparison in Tab. 5 shows that FDS, when used with the EDM, surpasses the training-based method for resolving crossing (*i.e.*, HRF [45]) by a large margin in a parameter-count matched setup.

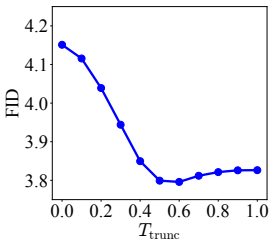
We conjecture that the performance gap stems from the discrepancy during training induces a severe capacity bottleneck for these methods as they model the complex distribution of individual sample-wise velocities, rather than simply regressing their marginal velocity. Therefore, the baselines demonstrate degraded performance compared to EDM, which retains the original training objective. Since FDS bypasses this issue and operates entirely at inference-time, it can be applied to strong pre-trained flow models without any modifications or re-training. By steering generation away from high-discrepancy regions on-the-fly, FDS can effectively resolves the discrepancy and mitigates the performance limitations in the training-based approaches, solidifying its practical advantage.

Table 5: Comparison with training-based frameworks. For VRFM*, we report the original values from the paper as the source code is not publicly available. All configurations utilize Euler solver (50 NFEs) on CIFAR-10.

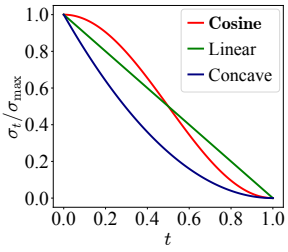
Model	FID	IS	Param.
VRFM* [9]	5.27	-	37.2M
HRF [45]	4.96	8.98	56.0M
EDM [17]	3.04	9.37	55.7M
EDM + **FDS**	**2.44**	**9.39**	55.7M

6.2 Ablation Study

To provide a deeper understanding of our proposed framework and to justify our default hyperparameter settings, we conduct comprehensive ablation studies that



(a) Ablation on T_{trunc}



(b) Visualization of σ_t schedule

σ_t schedule	FID (↓)
Baseline	4.151
Cosine	3.799
Linear	3.823
Concave	3.893

(c) Ablation on σ_t schedule

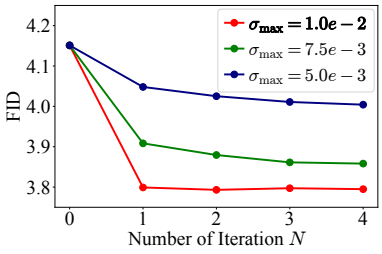
Fig. 6: Ablation on early-stage refinement. Applying FDS during earlier denoising stages effectively improves generation quality. **Bold text** indicates the default configuration (T_{trunc}=0.5, cosine schedule).

validate the core design choices abstracted in Sec. 3.2. All ablation experiments are conducted on ImageNet 256 × 256 with JiT-B/16 backbone and an Euler solver with 50 NFEs.

Efficacy of Early-Stage Interventions To analyze the impact of FDS across different generation stages, we investigate restricting the refinements exclusively to timesteps $t \leq T_{\text{trunc}}$. As shown in Fig. 6a, the FID improves as T_{trunc} increases but saturates around $T_{\text{trunc}} = 0.5$. This suggests that FDS yields the most substantial benefits during the initial stages of flow integration rather than in later phases. Furthermore, Fig. 6c indicates that adopting a cosine schedule for the perturbation scale σ_t, which allocates larger weights to earlier timesteps, surpasses both linear and concave schedules, further emphasizing the critical role of early-stage refinement. Taken together, these findings imply that the refinements during the early phases of generation are particularly effective in steering the samples away from high-discrepancy regions.

Refinement Iteration N While increasing the number of iterative updates for x_t yields progressive improvements in visual fidelity, we observe that the performance gains eventually plateau. Guided by this empirical observation (Fig. 7a), we opt to prioritize computational efficiency at inference time. Specifically, our results indicate that a single update step is sufficient to achieve highly competitive generation quality. Consequently, we adopt $N = 1$ and $\sigma_{\max} = 0.01$ as our default experimental setting, finding that this configuration strikes a practical balance between the representational benefits of FDS and minimal computational overhead.

Number of Candidates M To further explore the operational dynamics of FDS, we investigate the impact of varying the number of candidates M introduced in Sec. 3.2. Our empirical evaluations suggest that a single particle is generally sufficient to minimize discrepancy, offering a favorable balance between computational efficiency and generative quality. As shown in Fig. 7b, while increasing the candidate count M yields subtle but consistent improvements, the



(a) Ablation on number of iteration

M	FID (↓)
0 (Baseline)	4.151
1	3.799
2	3.795
8	3.785
20	3.765

(b) Ablation on number of perturbation candidates

Fig. 7: Effect of iterations N and candidates M. While increasing N and M generally improves performance, setting $N = M = 1$ offers the best trade-off.

overall performance quickly plateaus. We hypothesize that this occurs because a single perturbation, when applied consistently at every timestep, cumulatively guides the representation toward a stable, adjacent conditional mode. Admittedly, evaluating multiple candidates presents a distinct computational advantage: unlike increasing sequential NFEs, these evaluations can be fully parallelized without incurring additional latency overhead. Nevertheless, as scaling to multiple candidates yields only marginal generative benefits, the single-particle configuration remains a highly practical and efficient default choice.

7 Conclusion

In this work, we introduced a novel, training-free inference framework designed to mitigate the discrepancy between marginal and instantaneous sample-wise velocities in flow matching. Unlike existing solvers that primarily focus on minimizing temporal discretization errors, our approach actively steers the trajectory away from high-discrepancy regions to enhance generation quality. Extensive evaluations demonstrate that FDS is highly robust, yielding consistent performance improvements across various generation tasks. Ultimately, we believe our exploration of spatial trajectory refinement establishes a strong foundation for future research, offering a new perspective on reliable generation.

Acknowledgements

This work was in part supported by the National Research Foundation of Korea (RS-2024-00351212 and RS-2024-00436165), the Institute of Information & communications Technology Planning & Evaluation (IITP) (RS-2024-00509279, RS-2022-II220926, and RS-2022-II220959, RS-2019-II190075), and the "HPC support" project funded by the Korea government. This work was supported by the InnoCORE program of the Ministry of Science and ICT (AI Meta-Scientist, N10260110).

References

1. Bai, L., Shao, S., Zhou, Z., Qi, Z., Xu, Z., Xiong, H., Xie, Z.: Zigzag diffusion sampling: Diffusion models can self-improve via self-reflection. arXiv preprint arXiv:2412.10891 (2024)
2. Bertrand, Q., Gagneux, A., Massias, M., Emonet, R.: On the closed-form of flow matching: Generalization does not arise from target stochasticity (2025)
3. Black Forest Labs: Announcing Black Forest Labs. Blog post on Black Forest Labs website (Aug 2024), `https://blackforestlabs.ai/announcing-black-forest-labs/`, accessed: 2025-05-14
4. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. pp. 248–255. IEEE Computer Society (2009)
5. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a captcha that exploits interest-aligned manual image categorization. In: Conference on Computer and Communications Security (2007)
6. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis (2024)
7. Fan, W., et al.: CFG-Zero*: Improved classifier-free guidance for flow matching models. arXiv preprint (2025)
8. Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367 (2018)
9. Guo, P., Schwing, A.: Variational rectified flow matching. In: Forty-second International Conference on Machine Learning (2025)
10. He, Y., Murata, N., Lai, C.H., Takida, Y., Uesaka, T., Kim, D., Liao, W.H., Mitsufuji, Y., Kolter, J.Z., Salakhutdinov, R., Ermon, S.: Manifold preserving guided diffusion. In: The Twelfth International Conference on Learning Representations (2024)
11. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning (2022)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
14. Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022)
15. Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., Zhao, Z.: Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In: International Conference on Machine Learning. pp. 13916–13932. PMLR (2023)
16. Hutchinson, M.: A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communication in Statistics- Simulation and Computation **18**, 1059–1076 (01 1989)
17. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models (2022)

18. Kim, S., Yoo, J., Kim, J., Cha, Y., Kim, S., Hong, S.: Simulation-free training of neural odes on paired data. Advances in Neural Information Processing Systems **37**, 60212–60236 (2024)
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
20. Lee, S., Kim, B., Ye, J.C.: Minimizing trajectory curvature of ode-based generative models. In: International Conference on Machine Learning. pp. 18957–18973. PMLR (2023)
21. Li, T., He, K.: Back to basics: Let denoising generative models denoise (2026)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
23. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling (2023)
24. Liu, H., Yuan, Y., Liu, X., Mei, X., Kong, Q., Tian, Q., Wang, Y., Wang, W., Wang, Y., Plumbley, M.D.: Audioldm 2: Learning holistic audio generation with self-supervised pretraining. IEEE/ACM Transactions on Audio, Speech, and Language Processing **32**, 2871–2883 (2024)
25. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds (2022)
26. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in neural information processing systems **35**, 5775–5787 (2022)
27. Luo, Y., Huang, H., Zhou, T.Y., Wang, M.: Look-ahead and look-back flows: Training-free image generation with trajectory smoothing (2026)
28. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers (2024)
29. Ma, N., Tong, S., Jia, H., Hu, H., Su, Y.C., Zhang, M., Yang, X., Li, Y., Jaakkola, T., Jia, X., Xie, S.: Inference-time scaling for diffusion models beyond scaling denoising steps (2025)
30. Park, D., Lee, S., Kim, S., Lee, T., Hong, Y., Kim, H.J.: Constant acceleration flow (2024)
31. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
32. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023)
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding (2022)
35. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)
36. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy,

S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: Laion-5b: An open large-scale dataset for training next generation image-text models (2022)
37. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
38. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2022)
39. Wang, K., Mao, J., Wu, T., Xiang, Y.: Towards a golden classifier-free guidance path via foresight fixed point iterations. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
40. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis (2023)
41. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., et al.: Sana: Efficient high-resolution image synthesis with linear diffusion transformers. arXiv preprint arXiv:2410.10629 (2024)
42. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation (2023)
43. Ye, H., Lin, H., Han, J., Xu, M., Liu, S., Liang, Y., Ma, J., Zou, J., Ermon, S.: TFG: Unified training-free guidance for diffusion models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
45. Zhang, Y., Yan, Y., Schwing, A., Zhao, Z.: Towards hierarchical rectified flow. arXiv preprint arXiv:2502.17436 (2025)
46. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models (2023)