Optimizing for the Shortest Path in Denoising Diffusion Model

Ping Chen[1,2], Xingpeng Zhang[4], Zhaoxiang Liu[1,2*], Huan Hu[1,2], Xiang Liu[1,2],
Kai Wang[1,2], Min Wang[4], Yanlin Qian[3], Shiguo Lian[1,2*]

[1] Data Science & Artificial Intelligence Research Institute, China Unicom
[2] Unicom Data Intelligence, China Unicom, [3] DJI Technology Co.,Ltd.
[4] School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu, China
{chenp181, liuzx178, huh30, liux750, wangk115, liansg}@chinaunicom.cn
xpzhang@swpu.edu.cn, wangmin80616@163.com, honza.qian@dji.com

Abstract

In this research, we propose a novel denoising diffusion model based on shortest-path modeling that optimizes residual propagation to enhance both denoising efficiency and quality. Drawing on Denoising Diffusion Implicit Models (DDIM) and insights from graph theory, our model, termed the Shortest Path Diffusion Model (ShortDF), treats the denoising process as a shortest-path problem aimed at minimizing reconstruction error. By optimizing the initial residuals, we improve the efficiency of the reverse diffusion process and the quality of the generated samples. Extensive experiments on multiple standard benchmarks demonstrate that ShortDF significantly reduces diffusion time (or steps) while enhancing the visual fidelity of generated samples compared to prior arts. This work, we suppose, paves the way for interactive diffusion-based applications and establishes a foundation for rapid data generation. Code is available at https://github.com/UnicomAI/ShortDF.

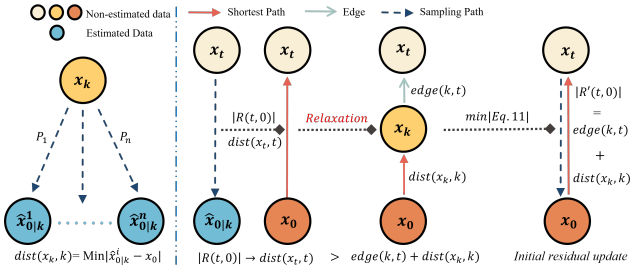

Figure 1. Modeling each reverse step involves identifying a data node x_t that minimizes cumulative transition costs $dist(x_t, t)$ (akin to a shortest path P_i in a weighted graph). By relaxing the strict 'straight-line' analogy to consider minimal-cost paths in the diffusion graph, we treat the initial residual $|R(t,0)|$ as an alternative path candidate. This relaxation is embedded in the loss function, enabling the reverse graph to iteratively refine the residual toward a shorter path (e.g., collapsing $x_0 \to x_k \to x_t$ into $x_0 \to x_t$), thus optimizing the reconstruction path through dynamic path compression.

1. Introduction

Recent years have seen significant advancements in generative models, in which Denoising Diffusion Models demonstrates exceptional performance across various domains, e.g. speech synthesis, 2D or 3D visual assets generation, and other applications [2, 4, 26, 27]. These models capture the inherent data distribution by gradually adding random noise to the data and learning the reverse process.

While the quality and diversity of the generation have been massively improved, concerns still exist on intensive computational load and time consumption [28, 34]. These burdens the further application of diffusion models in real-time scenarios, especially in cases where quick response time is in need, such as interactive graphic editing and real-time video processing (even we seldom consider) [1, 12]. To address these concerns, researchers have been actively investigating more efficient strategies. Traditional diffusion models typically require hundreds of iterations to achieve high-quality results, which leads to considerable computational costs [8, 23]. Recent efforts have focused on reducing the complexity through techniques such as knowledge distillation, which packs the multi-step diffusion sampling procedure into a single-step student network [15, 19, 28]. However, accurately approximating this high-dimensional and complex sampling process remains a challenging task.

Coupled with distillation-based approaches, other researchers have proposed faster numerical solvers, which rely on the current step size to estimate the next solution [34]. While the number of diffusion steps is decreased from

*Corresponding author.

1000 to fewer than 20, these methods [18, 22, 30] are facing a dilemma that generation quality sacrifices. On top of that, further reducing the sampling steps often leads to a decline in performance [3]. In a nutshell, a good trade-off between fast sampling and high generation quality is not easy to achieve, which is we are working on in this paper.

In this paper, we present ShortDF, which successfully achieves the aforementioned trade-off by utilizing a novel optimization framework that propagates residuals while maintaining high generation quality. Building on the foundation of Denoising Diffusion Probabilistic Models (DDPM) and Denoising Diffusion Implicit Models (DDIM), we integrate graph-theoretic techniques to reduce reconstruction errors. By framing the denoising process as an optimization problem for the shortest path in graph theory, we can enhance the efficiency of the procedure without compromising the quality of generation. Fig. 1 provides a clear illustration of the concept presented in this paper.

Specifically, Denoising Diffusion Probabilistic Models (DDPM) generate samples by gradually adding Gaussian noise to the original data and then applying reverse denoising. In contrast, Denoising Diffusion Implicit Models (DDIM) accelerate this process by introducing more flexible sampling paths [8, 22]. However, these methods often expose inefficiencies and residual accumulation in managing the multi-step reverse process. In contrast, our approach presents a novel optimization perspective by framing the denoising process as residual propagation. This method optimizes the initial residuals, thereby enhancing the overall efficiency of the reverse diffusion path. Till our grasp, we are the first to analyze the dynamics of residual propagation in the denoising process and to construct a reverse-step graph, enabling optimal transitions between any arbitrary pair of steps. This graph-based modeling enables us to capture error propagation more effectively. Through iterative relaxation, represented as a loss function based on shortest-path modeling, we gradually reduce the residuals. Our method accelerates the denoising process and ensures high reconstruction fidelity at each step, thereby overcoming the quality degradation observed in recent fast sampling methods. Another novelty is that, while perseving the sampler's task-agnostic nature, our method allows end-to-end optimizaiton of the generator and the sampler and obtaining robustness across domains like text-to-image geneartion.

To summarize the contributions of this work, we highlight the following three points:

• Proposes a novel denoising method (ShortDF) that achieves a balance between fast sampling and high-quality generation through a shortest-path optimization framework.

• Integrates graph-theoretic techniques into the denoising process, reducing reconstruction errors and enhancing efficiency without compromising quality.

• Enables end-to-end optimization of the generator and sampler, and extensive experiments have demonstrated its robustness and effectiveness across various domains.

2. Related Work

Despite SOTA results, the inherently iterative procedure of diffusion models entails a high and often prohibitive computational cost for real-time applications [28]. The inference process of accelerating diffusion models has been a key focus in the field, and there are currently two main research directions: fast diffusion sampler and diffusion distillation [28, 34].

Diffusion distillation. These methods treat diffusion distillation as a form of knowledge distillation [28], where a student model is trained to condense the multi-step outputs of the original diffusion model into a single step. One straightforward approach involves calculating the denoising trajectory in advance and subsequently training the corresponding student model in pixel space [32]. However, this method faces the significant challenge of the high computational cost associated with calculating and fitting the complete denoising trajectory. The Progressive Distillation (PD) model [19, 20] effectively reduces the number of sampling steps by half. InstaFlow [15, 16] progressively learns straighter flows, ensuring that the one-step prediction maintains accuracy over a larger distance. Additionally, Consistency Distillation (CD) [24], TRACT [3], and BOOT [6] aligned the different time steps of the ODE stream with its output, thereby achieving efficient diffusion acceleration. The Variational Score Distillation (VSD) can leverage a pretrained text-to-image diffusion model as a distribution matching loss [25]. Based on VSD, Yin et al. [28] proposed a method called distributed matched distillation to generate highly photorealistic images from complex datasets.

Accelerating diffusion sampling. A substantial body of research on faster diffusion samplers is grounded in solving the probability flow ordinary differential equation (ODE) [26]. Denoising Diffusion Implicit Models (DDIM) [22] were among the first initiatives to accelerate sampling in diffusion models, extending the original Denoising Diffusion Probabilistic Model (DDPM) to non-Markovian scenarios. Building on this foundation, Generalized Denoising Diffusion Implicit Models (gDDIM) [31] introduce a modified parameterization of the scoring network, facilitating deterministic sampling for a broader range of diffusion processes. The Efficient Denoising Model (EDM) [9] presents a design framework that delineates specific design choices to optimize the diffusion process. Liu et al. proposed Pseudo Numerical Diffusion Models (PNDM) [14], which employ a numerical solver with a nonlinear transfer component to address a differential equation on manifolds. The Diffusion Exponential Integrator Sampler [30] and DPM-Solver [18] capitalize on the semi-linear structure of the probability flow ODE to create specialized ODE solvers that

outperform general-purpose Runge-Kutta methods. It has been observed in the Approximate Mean-Direction Solver (AMED-Solver) [34] that nearly every sampling trajectory resides within a two-dimensional subspace of the embedded environment space. This approach eliminates truncation error by directly learning the mean direction, enabling rapid diffusion sampling. The Residual Denoising Diffusion Models (RDDM) [13] decouple the traditional single denoising diffusion process into residual diffusion and noise diffusion. The residual diffusion component represents directional diffusion from the target image is compared to the degraded input image, explicitly guiding the reverse generation process for image restoration. These numerical methods inherently introduce a degree of approximation error, which can impact the quality of image generation.

3. Proposed Method

Building on DDPM and DDIM, we employ graph theory methods to minimize reconstruction error by finding the shortest path in the diffusion model.

3.1. Background: DDPM and DDIM

Diffusion vs. Inverse Diffusion: Based on the Markov chain, DDPM progressively adds noise to the clean data x_0 in the forward process, resulting in a noisy sample x_t at timestep t. In the reverse process, the noise in x_t is gradually eliminated to reveal x_0. The forward process is defined as described in [8]:

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon, \quad \epsilon \sim N(0, \mathbf{I}) \tag{1}$$

where $t \in [1, T]$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and α_t is a predefined variance schedule parameter controlling the strength of noise added at time step t. In the reverse order, the model estimates noise using a neural network $\epsilon_\theta(x_t, t)$, and the estimated clean sample $\hat{x}_0$ at timestep t is given by:

$$\hat{x}_{0|t} = \frac{x_t - \sqrt{1-\bar{\alpha}_t} \cdot \hat{\epsilon}_t}{\sqrt{\bar{\alpha}_t}} \tag{2}$$

Where $\hat{\epsilon}_t = \epsilon_\theta(x_t, t)$ denotes the network's prediction of the noise. The noise loss:

$$L_\epsilon = \mathbb{E}_{t, x_0, \epsilon} \left[\|\epsilon - \hat{\epsilon}_t\|_2 \right] \tag{3}$$

This loss term ensures that the model accurately predicts the noise, allowing it to effectively denoise the sample over multiple steps throughout the entire reverse path.

Accelerating Sampling with Flexible Paths: DDIM accelerates the sampling process by introducing flexibility in the reverse diffusion paths [22]. Unlike DDPM, which employs a fully stochastic reverse process, DDIM enables deterministic or partially stochastic sampling, governed by the variance term σ_n. The DDIM sampling equation is:

$$\begin{aligned} \hat{x}_k &= DDIM(\hat{x}_{0|t}, \hat{\epsilon}_t, k, \sigma_n) \\ &= \sqrt{\bar{\alpha}_k} \cdot \hat{x}_{0|t} + \sqrt{1-\bar{\alpha}_k - \sigma_n^2} \cdot \hat{\epsilon}_t + \sigma_n \cdot \epsilon \end{aligned} \tag{4}$$

where $k \leq t$. DDIM uses the estimate $\hat{x}_{0|t}$ as a replacement for the clean data x_0, reducing the number of reverse diffusion steps.

3.2. Residual Propagation as Path Representation

In DDIM, when $\sigma_n = 0$, the reverse process becomes fully deterministic, allowing us to analyze the residuals along the reverse diffusion path. We assume that the sampling path is $k_1 \to \cdots \to k_n \to 0$, where $k_i \in [1, T]$ and $k_i \geq k_{i+1}$.

Residual Propagation: Let $R(k_i, k_j)$ denote the residual change from the step k_i to k_j. At timestep k_1, the initial residual $R(k_1, 0)$ is the one-step sampling estimation error from k_1 to 0. It quantifies the difference between x_0 and its direct estimate $\hat{x}_{0|k_1}$. Refering to Eq. (1) and Eq. (2), we get:

$$R(k_1, 0) = x_0 - \hat{x}_{0|k_1} = \frac{\sqrt{1-\bar{\alpha}_{k_1}}}{\sqrt{\bar{\alpha}_{k_1}}} (\epsilon_\theta(x_{k_1}, k_1) - \epsilon) \tag{5}$$

When transitioning from the step k_i to a smaller step k_j, the residual propagation, based on Eq. (4) and Eq. (5), is defined as:

$$R(k_i, k_j) = \frac{\sqrt{1-\bar{\alpha}_{k_j}}}{\sqrt{\bar{\alpha}_{k_j}}} (\epsilon_\theta(\hat{x}_{k_j}, k_j) - \epsilon_\theta(\hat{x}_{k_i}, k_i)) \tag{6}$$

Where $\hat{x}_{k_j} = DDIM(\hat{x}_{0|k_i}, \epsilon_\theta(\hat{x}_{k_i}, k_i), k_j, 0)$, $\hat{x}_{k_1} = x_{k_1}$.

Path Residual: The cumulative residual along the entire reverse diffusion path (from $k_1 \to \cdots \to k_n \to 0$), is given by:

$$R(k_n, 0) = R(k_1, 0) - \sum_{i=1}^{n-1} R(k_i, k_{i+1}) \tag{7}$$

Based on Eq. (7), optimizing the path residual requires balancing the cumulative residuals along the path while also offsetting the initial residual. However, directly optimizing the path residual is not feasible due to the complexity of managing residuals across multiple steps. Instead, we focus on a target proxy that optimizes the initial residual. In the reverse process, transitioning from a later step to an earlier step refines the estimate of x_0 incrementally. *A smaller initial residual at k_1 leads to smaller residuals throughout the entire path, as each subsequent timestep builds upon this initial estimate*. Thus, optimizing the initial residual $R(k_1, 0)$ can significantly reduce the cumulative residual, enhancing the overall traceability of the optimization.

3.3. Optimization for Shortest-Path

Building on the insights from Sec. 3.2, directly optimizing cumulative residuals along the path brings significant challenges. However, if the optimal solution at an earlier step k is known, it can be propagated backward to obtain the optimal solution at a later step t. This process enhances the initial residual, thereby optimizing the entire reverse diffusion path. To achieve this, we construct a *step-reverse graph*, where edges are initialized between arbitrary steps k and t with $k < t$. An edge connecting k to t, and shortest-path optimization is employed as a target proxy to minimize the cumulative residual error along the reverse diffusion path.

Residual Elimination and Graph Construction: However, directly defining edge weights based on residual propagation (e.g., $R(t, k)$) is not feasible. Ideally, the optimal solution at step k should not be influenced by residuals at a later step t. Defining edge weights using residuals would introduce interference between steps, allowing optimal residual propagation to become untraceable.

To address this issue, we begin with the clean data x_0 (due to it provides the precise reference) and then transition from t to k using x_0 via Eq.(4) eliminates residuals along the path $t \to k$. Thus, the residuals along $t \to k$ are effectively counteracted.

We define the edge weight between step k and t as the residual difference computed using both the estimated $\hat{x}_0$ and the groundtruth x_0. This edge weight quantifies the error that accumulates when using the noisy estimate $\hat{x}_{0|t}$ compared to the true clean data x_0. Specifically, the edge weight is defined as follows:

$$\text{edge}(k, t) = |x_0 - \hat{x}'_{0|k}| - |x_0 - \hat{x}_{0|k}| \tag{8}$$

Here, $\hat{x}'_{0|k}$ and $\hat{x}_{0|k}$ represent estimates of x_0 at step k, but from two different transformations. $\hat{x}_{0|k}$ is estimated using $\hat{x}_k$, where:

$$\hat{x}_k = \text{DDIM}(\hat{x}_{0|t}, \epsilon_\theta(x_t, t), k, 0)$$

corresponds to the estimate-based transformation from timestep t to k. Similarly, $\hat{x}'_{0|k}$ is computed using x_k, where:

$$x_k = \text{DDIM}(x_0, \epsilon_\theta(x_t, t), k, 0)$$

represents the transformation based on the true clean data x_0. Both $\hat{x}'_{0|k}$ and $\hat{x}_{0|k}$ are then computed using the reverse process described in formula (2).

By defining the edge weight in this manner, we ensure that the residuals are eliminated when using the true x_0 and that the propagation of residuals from an earlier step k does not affect the optimal solution at a later step t. The $edge(k, t)$ represents the residual error from step t to step k, illustrating the additional influence of the estimation error at step t on the residual at step k during the reverse diffusion process.

Relaxation: By utilizing the accurate clean data x_0, we eliminate the residuals along the diffusion path, ensuring that the result at step k is optimal. With this optimal result at k, we can now employ shortest-path optimization to determine the optimal residual at step t.

The relaxation method is introduced to iteratively refine the residual at each step. Given the noisy data x_t, the objective is to minimize the residual error at step t, defined as $dist(x_t, t)$. This is formulated as the absolute value of the initial residual $R(t, 0)$, i.e., $dist(x_t, t) = |R(t, 0)| = |x_0 - \hat{x}_{0|t}|$. This represents a near-optimal error across all possible paths. To ensure better optimization at step t, we propagate the optimal result from step k using the following relaxation condition:

$$cond : dist(x_t, t) > dist(x_k, k) + edge(k, t) \tag{9}$$

When this condition holds, the graph model chooses to transition directly from k to t, as it provides a more optimal path wrt. residual error. Consequently, the residual error at step t is updated: $dist(x_t, t) \leftarrow dist(x_k, k) + edge(k, t)$. Through the relaxation and update process, we iteratively refine the path selection within the graph model, adjusting the residual at each step to minimize the residual error.

3.4. Loss Function

Based on Shortest-Path optimization, we design the total optimization loss as:

$$L = \lambda \cdot L_\epsilon + cond \cdot L_r, \tag{10}$$

$$L_r = \|dist(x_k, k) + edge(k, t) - dist(x_t, t)\|_2 \tag{11}$$

where λ is the noise loss weight, $cond$ the relaxation condition term from Eq.(9), and L_r the relaxation loss, which allows the optimzal solution at step k to guide the optimization at step t. Inherently, this equals to adding a reverse edge in the graph from t to 0, with the weight $edge(t, 0) = dist(x_k, k) + edge(k, t)$.

Optimizing the model parameters at step t effectively adds a new edge to the graph, connecting t directly to 0. This "shortcut" edge incorporates the contribution from the noisy input x_t. Using the optimal solution at step k, the update reduces the residual at step t, thereby enhancing the reconstruction of the clean data x_0. While graph-theoretic optimal solution is well-defined, the randomness and the ubiquitous presence of noise makes explicit graph construction and mathematical derivation of the optimal solution challenging. However, intuitively, network parameters can implicitly build the graph through learning, enabling us to approximate the optimal solution.

3.5. Optimization Strategy

To optimize the model effectively and stabilize the denoising process, as depicted in Algorithm 1, we employ a multi-state optimization strategy that utilizes three instances of the

Algorithm 1 Multi-State Optimization Training

Initialize ϵ_θ, $\epsilon_{\theta,\text{ema}}$, and $\epsilon_{\theta,\text{graph}}$ with the same set of parameters.
Set moving average decay factor α for updating $\epsilon_{\theta,\text{ema}}$. Set iteration N and interval n for updating $\epsilon_{\theta,\text{graph}}$.
for iteration $i = 1, 2, \ldots, N$ **do**
  Sample real data x_0 and a step t, then generate noisy data x_t via Eq. (1).
  Compute L_ϵ using ϵ_θ via Eq. (3), and $dist(x_t, t) = |R(t, 0)|$ using ϵ_θ via Eq. (2),(5).
  Randomly sample another step $k < t$. Use $\epsilon_{\theta,\text{graph}}$ to obtain $\hat{x}_{0|t}$ via Eq. (2), then transform x_0 and $\hat{x}_{0|t}$ to obtain x_k and $\hat{x}_k$ using Eq. (4).
  Compute the edge weight $edge(k, t)$ using $\epsilon_{\theta,\text{graph}}$ via Eq. (8).
  Use $\epsilon_{\theta,\text{ema}}$ to compute the reconstruction error $dist(x_k, k)$ at step k.
  Compute the relaxation condition via Eq. (9).
  Compute the total loss via Eq. (10):
  Update ϵ_θ wrt. L, Update $\theta_{\text{ema}} \leftarrow \alpha\theta_{\text{ema}} + (1 - \alpha)\theta$
  Every n iterations, $\theta_{\text{graph}} \leftarrow \theta_{\text{ema}}$.
end for

model: the Base Model ϵ_θ, the EMA Model $\epsilon_{\theta,\text{ema}}$, and the Graph Model $\epsilon_{\theta,\text{graph}}$. This strategy improves the stability of training in the presence of random noise.

The Base Model ϵ_θ handles noise prediction and residual updates, serving as the fundamental logic for reconstruction. The EMA Model $\epsilon_{\theta,\text{ema}}$ smooths updates to stabilize training and provides more accurate estimates of the optimal error at timestep k. The Graph Model $\epsilon_{\theta,\text{graph}}$ calculates edge weights for shortest-path optimization by employing delayed updates to capture global information and enhance stability in error accumulation. This combination of models ensures both stability and effectiveness in enhancing reconstruction quality within the shortest-path framework.

The whole optimization is unfolded as: first, $dist(x_t, t)$ is computed using ϵ_θ, while $dist(x_k, k)$ is computed using $\epsilon_{\theta,\text{ema}}$, and the edge weights $edge(k, t)$ are calculated using $\epsilon_{\theta,\text{graph}}$. Then, the model parameters are updated based on the total loss function Eq.10. At intervals the parameters of $\epsilon_{\theta,\text{graph}}$ are synchronized to those of $\epsilon_{\theta,\text{ema}}$ to maintain stability in edge weight calculations, ensuring accurate shortest-path optimization. This multi-state strategy not only improves robustness to random noise but also enhances the accuracy of error propagation across step, resulting in better denoising performance. It is worthy to note that, without graph modeling, ShortDF dengerates to DDIM.

3.6. Insights from Shortest Path Optimization

This section demonstrates how multi-state optimization helps achieve the shortest path in an effective manner. By illustrating a practical generation case, we validate the theoretical foundation of the proposed optimization strategy and provide insights into the operational mechanism.

We study making a case using a few generation steps. For instance, prior to iterations, assume $t = 10$ and $k = 2$: paths $10 \to 0$ and $10 \to 2 \to 0$ represent one-step and two-step generation, respectively.

In iteration 1, if the relaxation condition is satisfied, the reconstruction ability of path $(10 \to 0)$ becomes equivalent to that of path $(10 \to 2 \to 0)$, thus making one-step generation equivalent to the original two-step generation.

In iteration 2, assuming $t = 100$ and $k = 10$, and the relaxation condition still holds, the path $(100 \to 0)$ becomes equivalent to path $(100 \to 10 \to 0)$. Similarly, we can infer that path $(100 \to 0)$ is equivalent to $(100 \to 10 \to 2 \to 0)$, making one-step generation equivalent to the original three-step generation. Continuing this reasoning through iteration n, the path $(T \to 0)$ becomes representative of the optimal error across various routes.

4. Experiments

4.1. Settings

Datasets. We evaluate our method on several widely adopted benchmarks, including CIFAR-10 (32×32) [11], CelebA (CelebFaces Attributes Dataset) (64×64) [17], and LSUN Churches (256×256) [29]. The Fréchet Inception Distance (FID) [7] is utilized to assess image quality.

Our implementation utilizes the official code and configuration of DDIM [22] on a single NVIDIA A100 80GB GPU. We combine the designed shortest path loss with the original loss of DDIM to train the model. In the subsequent experiments conducted on three datasets, we employed the same noise input to generate DDIM samples and the output results of the proposed method, respectively. This approach explains the high degree of similarity observed in the generated samples.

Models Compared We compared with the most accelerated diffusion models available, *e.g.* DDIM [22], DPM-solver [18], DPM-solver++ [33], iPNDM [14], DEIS [30], D-ODE [10], SDDPM [5], Resfusion [21], and gDDIM [31].

4.2. Results

4.2.1 Quality Analysis on CIFAR-10 Dataset

To evaluate the effectiveness of the proposed method, we conducted a comprehensive analysis of the quality and speed of data generation using the CIFAR-10 dataset. The results are presented in Table 1 and Figure 2.

As shown in Table 1, our proposed ShortDF method achieves significant improvements in both speed and quality compared to the DDIM baseline. Specifically, in Part A of Table 1, ShortDF reduces the number of steps from 10

Figure 2. Sample images generated on the CIFAR-10 dataset at 1, 5, and 10 steps.

Table 1. Performance Metrics for CIFAR-10 on NVIDIA A100

Part A: Standard Evaluation

Method	Steps	FID (↓)	Step Time (ms)	Speed-up	FID Improv.
DDIM	10	11.14	1.2	1.0x	-
ShortDF (Ours)	2	**9.08**	1.2	**5.0x**	**+18.5%**

Part B: Quantitative comparisons across steps (NFE)

Method	1	2	3	4	5	8	10
DDIM	>100	>100	123.54	66.13	26.91	19.06	11.14
DPM-solver	-	-	90.38	33.29	23.31	6.91	5.09
DPM-solver++	-	-	107.02	30.46	18.87	12.58	7.83
iPNDM	-	-	-	>95	70.07	>15	9.36
DEIS3	-	-	-	>75	15.37	>14	4.17
D-ODE	-	>100	-	>40	-	>10	8
SDDPM	-	-	-	19.20	-	16.89	-
ResFusion	-	-	-	-	-	-	28.82
gDDIM (DDPM)	-	-	-	-	-	-	4.17
gDDIM (CLD)	-	-	-	-	-	-	13.41
ShortDF (Ours)	**25.07**	**9.08**	**7.31**	**6.59**	**6.45**	**4.93**	**3.75**

to 2 while achieving a lower FID score of 9.08, compared to DDIM's 11.14. This represents an 18.5% improvement in FID while accelerating the generation process by 5.0x. These results demonstrate the efficiency and effectiveness of ShortDF in generating high-quality images with fewer steps.

In Part B of Table 1, we provide a detailed comparison across different steps (1 to 10). Our method consistently outperforms existing methods at each step level. For example, at 2 steps, ShortDF achieves an FID of 9.08, significantly lower than other methods, while maintaining fast generation speed. At 10 steps, ShortDF reaches an FID of 3.75, outperforming gDDIM (DDPM) which achieved an FID of 4.17. This indicates that ShortDF can generate high-quality images with fewer steps, making it highly efficient for low-resolution image generation tasks.

Figure 2 visually demonstrates the quality of images generated by ShortDF at different steps (1, 5, and 10). The results show that our method produces images with higher authenticity and clarity compared to DDIM, even with fewer steps. For instance, the images generated by ShortDF at 5 steps are comparable in quality to those generated by DDIM at 10 steps. This further highlights the ability of ShortDF to accelerate the diffusion model's data generation process while maintaining or improving image quality.

4.2.2 Quality analysis on CelebA dataset

On a dataset with a larger image size, CelebA, our method achieves excellent data generation results, as shown in Table 2. Due to the larger image size of the CelebA dataset, the FID values under the 2 steps generation strategy are significantly higher than those observed in the CIFAR-10 dataset. The FID value of the proposed approach demonstrates a substantial reduction in the data creation strategy within 10 steps when compared to the baseline method, DDIM. Even with a 20 steps generation process, the FID value achieved

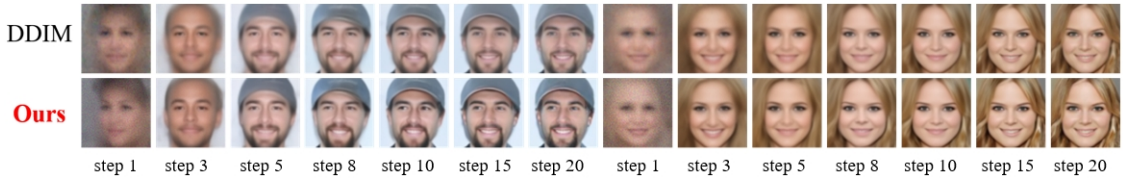

Figure 3. Performance comparison of our method and DDIM on the CelebA dataset at each time node along the same sampling path (with a total of 20 nodes), following the optimization of the initial residuals.

Table 2. Comparison results on the CelebA dataset within different steps (NFE).

Method	2	3	4	5	8	10	20	50
DDIM	>100	57.65	38.01	27.20	13.94	10.59	6.35	4.66
DPM-solver	-	48.61	31.21	21.05	10.89	8.44	5.97	5.43
DPM-solver++	-	61.26	**19.75**	**14.70**	10.24	8.31	5.94	5.40
iPNDM	>100	-	>70	59.87	>20	7.78	-	-
DEIS3	-	-	>40	25.07	>15	6.95	-	-
D-ODE	>100	-	>20	-	>10	>7.5	-	-
SDDPM	-	-	25.09	-	-	-	-	-
RDDM	-	-	-	-	-	23.25	-	-
ShortDF (Ours)	**29.22**	**27.95**	23.37	**13.48**	**7.13**	**5.00**	**3.25**	**2.78**

Table 3. Comparison results on the Churches dataset within different steps (NFE).

Method	2	4	5	10	20	50
DDIM	168.72	122.62	69.53	21.52	8.24	7.65
DPM-solver++	-	**50.27**	47.18	38.89	18.48	10.69
PNDM	-	-	**20.50**	11.80	9.20	9.49
ShortDF (Ours)	**134.91**	66.48	40.38	**11.78**	**7.97**	**6.95**

by this research remains only half that of DDIM. The reduction in the FID value for the suggested method is comparable to that observed with DPM-Solver++. In contrast to this study, more advanced techniques such as iPNDM and DEIS exhibit slightly higher performance in the 10 steps generation strategy but show significantly greater values in other step configurations.

The visualization results of several samples are presented in Fig. 3 and Fig. 4.

As shown in Fig. 3, the performance of our method and DDIM on the CelebA dataset is compared at each time node along the same sampling path. The figure clearly indicates that ShortDF exhibits much faster generation speed and higher clarity than DDIM when using the same step generation approach. Specifically, the image quality achieved with the 8th sampling (step 8) in this paper is quite comparable to that of DDIM's 15th and 20th samplings. This demonstrates the efficiency and effectiveness of our method in generating high-quality images with fewer steps.

Fig. 4 presents more high-quality sampling instances generated by our method on the CelebA dataset via 10 steps. These samples further illustrate the superior performance of ShortDF in terms of image quality and generation speed.

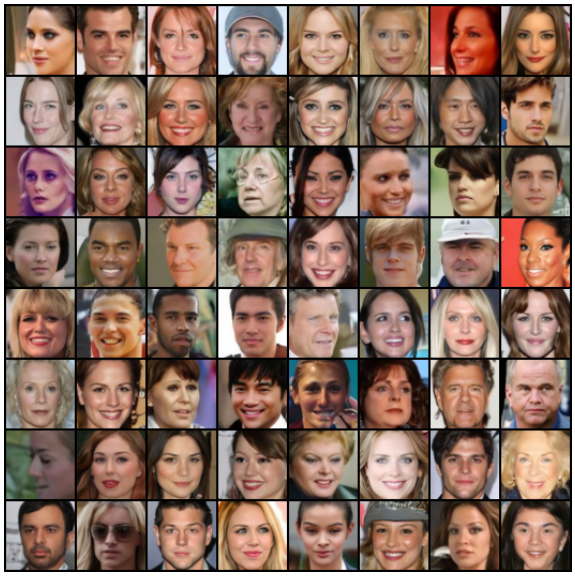

Figure 4. High-quality CelebA image samples generated using our method with only 10 diffusion steps.

4.2.3 Quality analysis on Church dataset

The church dataset is a large collection of images that exceeds the sizes of both CIFAR-10 and CelebA. Its images often include additional elements such as sky and vegetation, which can significantly affect the quality of generated images. Consequently, there has been limited research conducted on this dataset. In this paper, we select the DDIM and PNDM methods for comparative analysis, as illustrated in Table 3. Due to the large image size, all methods yield relatively high FID values in fewer than five steps. When comparing the 2 steps and 4 steps strategies, our proposed method demonstrates a significantly lower FID value than DDIM and is slightly lower than DPM-Solver++ in the 5 steps strategy, although it performs better than the PNDM method. As the number of steps increases, the FID value decreases markedly, with values significantly lower than those of the PNDM method in the 10, 20, and 50 steps.

As illustrated in Fig. 5, the DDIM approach exhibits increased noise at the first sampling node (step 1), rendering the object's contours nearly invisible. Additionally, the

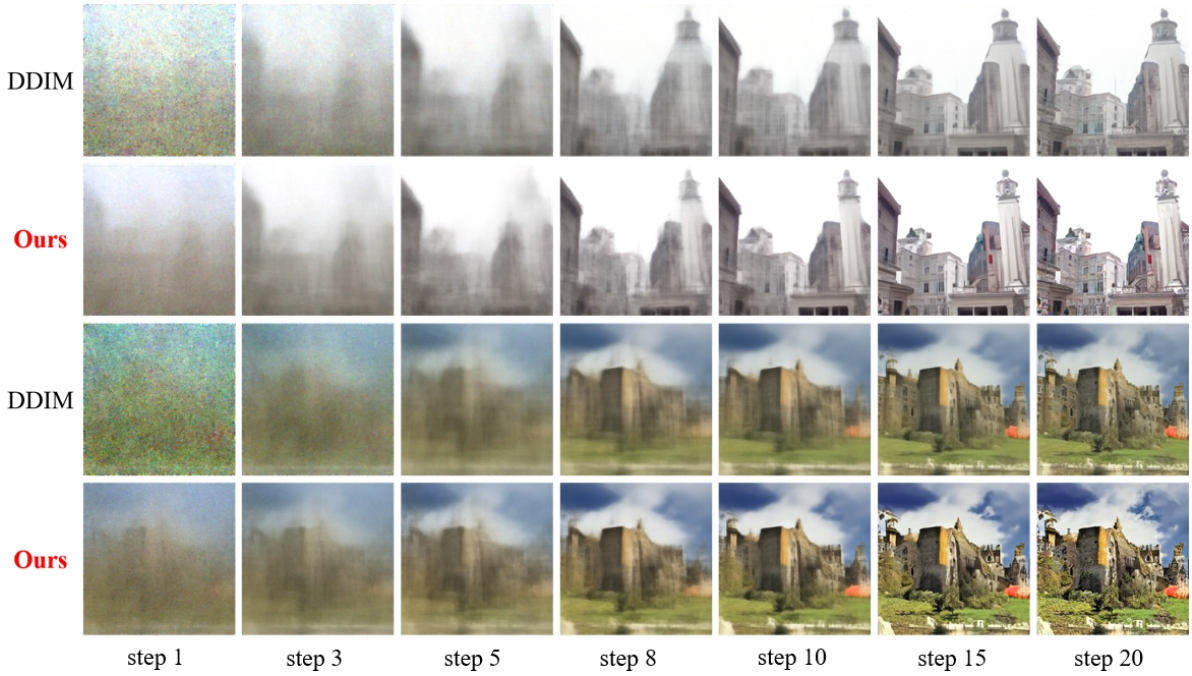

Figure 5. Denoising quality of our method with DDIM on the Church dataset at each time node along the same sampling path, following the optimization of the initial residuals.

Figure 6. High-quality church image samples generated using our method with only 10 diffusion steps.

church casts a subtle shadow in our approach. The image quality of the proposed method at the 8th sampling node is remarkably similar to that of DDIM at the 15th sampling node. It is important to highlight that the procedure from the 10th to the 20th sampling node employed in this article yields superior generation effects in the church background region, including the grass and blue sky.

Similarly, as demonstrated in Fig. 6, ShortDF achieves high-quality church image generation with only 10 diffusion steps, closely aligning the generated samples with the target data. This further highlights the efficiency and effectiveness of our approach in producing visually coherent results with minimal computational overhead.

5. Conclusion

This paper proposes a novel denoising diffusion method based on shortest path modeling, which enhances both the efficiency and quality of the denoising process by optimizing residual propagation. This approach significantly reduces sampling time while maintaining, and in some cases improving, the quality of the generated samples. To validate the method, we conduct extensive experiments on various benchmark datasets, demonstrating its state-of-the-art performance in addressing the computational overhead typically associated with diffusion models. This study demonstrates the effectiveness of incorporating graph theory into diffusion processes, providing valuable insights for future research and development. It not only advances the field of generative modeling but also establishes a solid foundation for real-time applications.

References

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, pages 17981–17993, 2021. 1

[2] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022. 1

[3] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. TRACT: denoising diffusion models with transitive closure time-distillation. *CoRR*, abs/2303.04248, 2023. 2

[4] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *CVPRW*, pages 4174–4185, 2022. 1

[5] Jiahang Cao, Ziqing Wang, Hanzhong Guo, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking denoising diffusion probabilistic models. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4912–4921, 2024. 5

[6] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Josh Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *Int. Conf. Mach. Learn.*, 2023. 2

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 2, 3

[9] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 2

[10] Sanghwan Kim, Hao Tang, and Fisher Yu. Distilling ode solvers of diffusion models into smaller steps. In *CVPR*, pages 9410–9419, 2024. 5

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. 5

[12] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *NeurIPS*, 2022. 1

[13] Jiawei Liu, Qiang Wang, Huijie Fan, Yinong Wang, Yandong Tang, and Liangqiong Qu. Residual denoising diffusion models. In *CVPR*, pages 2773–2783, 2024. 3

[14] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022. 2, 5

[15] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 1, 2

[16] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*, 2024. 2

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 5

[18] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022. 2, 5

[19] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, pages 14297–14306, 2023. 1, 2

[20] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 2

[21] Zhenning Shi, Haoshuai Zheng, Chen Xu, Changsheng Dong, Bin Pan, Xueshuo Xie, Along He, Tao Li, and Huazhu Fu. Resfusion: Denoising diffusion probabilistic models for image restoration based on prior residual noise. In *NeurIPS*, 2024. 5

[22] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 3, 5

[23] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *NeurIPS*, 2020. 1

[24] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Int. Conf. Mach. Learn.*, pages 32211–32252, 2023. 2

[25] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 2

[26] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4):105:1–105:39, 2024. 1, 2

[27] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. In *NeurIPS*, 2023. 1

[28] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *CoRR*, abs/2311.18828, 2023. 1, 2

[29] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[30] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *ICLR*, 2023. 2, 5

[31] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. In *ICLR*, 2023. 2, 5

[32] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *ICLR*, pages 42390–42402, 2023. 2

[33] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In *NeurIPS*, pages 55502–55542, 2023. 5

[34] Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *CVPR*, pages 7777–7786, 2024. 1, 2, 3