# ProReflow: Progressive Reflow with Decomposed Velocity

Lei Ke[1] Haohang Xu[3] Xuefei Ning[1] Yu Li[1]
Jiajun Li[4] Haoling Li[1] Yuxuan Lin[1] Dongsheng Jiang[3]
Yujiu Yang[1†] Linfeng Zhang[2†]
[1]Tsinghua University [2]Shanghai Jiao Tong University
[3]Huawei Inc. [4]University of Electronic Science and Technology of China
kl23@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn, zhanglinfeng@sjtu.edu.cn

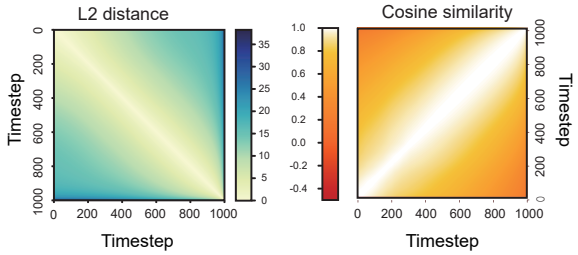
## Abstract

*Diffusion models have achieved significant progress in both image and video generation while still suffering from huge computation costs. As an effective solution, rectified flow aims to rectify the diffusion process of diffusion models into a straight line for few-step and even one-step generation. However, in this paper, we suggest that the original training pipeline of reflow is not optimal and introduce two techniques to improve it. Firstly, we introduce progressive reflow, which progressively reflows the diffusion models in local timesteps until the whole diffusion progresses, reducing the difficulty of flow matching. Second, we introduce aligned v-prediction, which highlights the importance of direction matching in flow matching over magnitude matching. Experimental results on SDv1.5 and SDXL demonstrate the effectiveness of our method, for example, conducting on SDv1.5 achieves an FID of 10.70 on MSCOCO2014 validation set with only 4 sampling steps, close to our teacher model (32 DDIM steps, FID = 10.05). Our codes will be released at Github.*

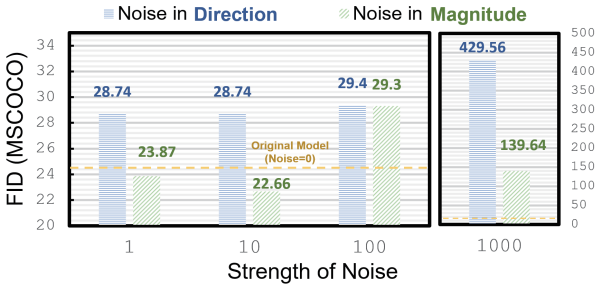
## 1. Introduction

Diffusion models have achieved significant breakthroughs in image and video generation, boosting various downstream applications such as text-to-image generation [27, 32] and image editing [2, 3, 10]. However, compared with traditional generation models such as GANs [9], the sampling process of diffusion models is formulated to include multiple timesteps, which severely harms its computation efficiency, hindering its application in edge devices and real-time applications. To solve this problem, abundant methods have been proposed to reduce the number of sampling steps such as step distillation [24, 34], consistency models [23, 38] and flow matching [16, 17].

†Corresponding author.




(a) Velocity differences across timesteps in diffusion model.




(b) Flow model's performance degrades with increased noise.

Figure 1. (a) L2 distance and Cosine similarity across velocities at different timesteps, the velocity discrepancy between timesteps increases with their distance in timesteps. (b) The consistently larger FID degradation under directional noise demonstrates that velocity direction is more critical for generation quality.

Among them, flow matching has gained popularity due to its simplicity and effectiveness. By re-flowing the pretrained diffusion models into a line, few steps and even 1-step generation can be achieved with tolerant loss in generation quality. The training process of reflow usually contains two steps. Firstly, the pretrained diffusion model generates abundant (noise, image) pairs. Then, the diffusion model is trained to make the velocity at different timesteps to be identical, indicating that the trajectory is rectified. However, in this paper, we suggest that such a training strategy has not fully unleashed the potential of rectified flow and introduced two training techniques referred to as *progressive reflow* and *aligned v-prediction* to further improve it.

**Progressive Reflow:** Traditional Reflow usually starts from a pretrained diffusion model and directly trains it to have a consistent prediction of velocity in all timesteps, which is theoretically correct but introduces difficulty in the optimization process. As shown in Figure 1 (a), the pretrained diffusion model has significantly different velocities at different timesteps, and directly eliminating these differences raises challenges in the training process.

Fortunately, Figure 1 (a) also shows that the pretrained diffusion model exhibits similar velocity in the adjacent timesteps, which provides the possibility to first reflow the model in a local window, and then reflow it in the whole training process. Such a progressive reflow pipeline allows the model to first learn to solve an easy problem and then extend to a difficult problem, which implies curriculum learning in generative models and thus facilitates the training process. Based on this observation, we propose progressive reflow, which firstly divides the whole diffusion process into $N$ windows, and then progressive reflow $N$ windows into $N/2$, $N/4$, $N/2$, $N/8$ until very few and even one window.

**Aligned V-Prediction:** Flow matching aims to match the velocity in different timesteps to achieve the target that the whole diffusion process is a straight line. However, we suggest that such a velocity matching is not optimal for the target of a "straight line" as the velocity can be further decomposed into its direction and magnitude, where the direction is more crucial for straightness. In other words, matching the direction of the velocity should have a higher priority than matching the magnitude, which has been ignored in previous works. Based on this observation, we propose to modify the original training loss of flow matching by introducing direction matching to solve this problem.

Our experiments have validated the effectiveness of the two improved training techniques. For instance, on MSCOCO-2017, 10.94 and 21.73 FID reduction can be observed with our ProReflow-II compared to rectified flow (2-ReFlow [18]) at 4 steps and 2 steps respectively, demonstrating improvements in generation quality

In summary, our contributions are as follows.

- We propose progressive reflow, which progressive reflow the diffusion model in local timesteps until the whole diffusion process. Progressive reflow implies the curriculum learning in flow matching and facilitates model training.
- Based on the observation that the direction of velocity is more crucial than the magnitude for straightness, we introduce velocity direction matching as an additional target for flow matching to facilitate model training.
- Extensive experiments demonstrate that both components are effective individually, and their combination achieves state-of-the-art performance with only 4 sampling steps.

## 2. Related Work

### 2.1. Text-to-Image Generation

Diffusion models(DMs) learn the mapping from noise to images by fitting marginal probability distributions at each timestep [8, 37]. It works well because the forward diffusion process, which progressively adds noise to images, maintains the same marginal distributions as the sampling process [22]. Combined with some technologies like classifier-free guidance and text encoder [7, 26, 33], DMs have surpassed GANs [4, 31] and VAEs [12, 29] not only in generation quality but also in training stability. Besides applied in pixel space, DMs can be effectively applied in latent space as well, which significantly reduces computational complexity [32]. Despite achieving impressive generation quality, the iterative nature of DMs impacts its generation efficiency. Consequently, accelerating inference of diffusion models has emerged as an avtive research topic.

### 2.2. Efficient Diffusion

Existing approaches for accelerating DMs can be predominantly classified into two categories: efficient diffusion samplers and step distillation [45].

The former category incorporates differential equation solvers into inference without requiring additional training. DDIM [36] enables step skipping in the reverse process by introducing a non-Markovian sampling strategy. DPM-Solver [21] reformulates the reverse diffusion process into an ODE system and solves it with high-order numerical methods, achieving superior sampling efficiency. Sampler-based methods enable diffusion models to maintain satisfactory generation quality with 20 steps; however, performance deteriorates significantly when further reducing the step count (such as below 10).

The second category methods enhance few-step inference performance through another step distillation process. Progressive Distillation(PD) [34] adopts a staged approach, iteratively halving the student model's sampling steps. Adversarial Diffusion Distillation(ADD) [35] leverages adversarial training for improved supervision, while Consistency Distillation (CD) [38] enforces output convergence toward the target image across the sampling trajectory.

### 2.3. Rectified Flow

Flow matching has emerged as a kind of advanced diffusion model [16, 17]. It reformulates the forward process as a linear interpolation between noise and images, thereby proposing to predict a consistent velocity $v$ across the entire sampling trajectory. Thus, the sampling process is simplified to a temporal integration of the velocity field $v$.

Similarly, ReFlow was proposed as a technique applying flow matching to pretrained diffusion models, enabling the adaptation of existing architectures without retraining from

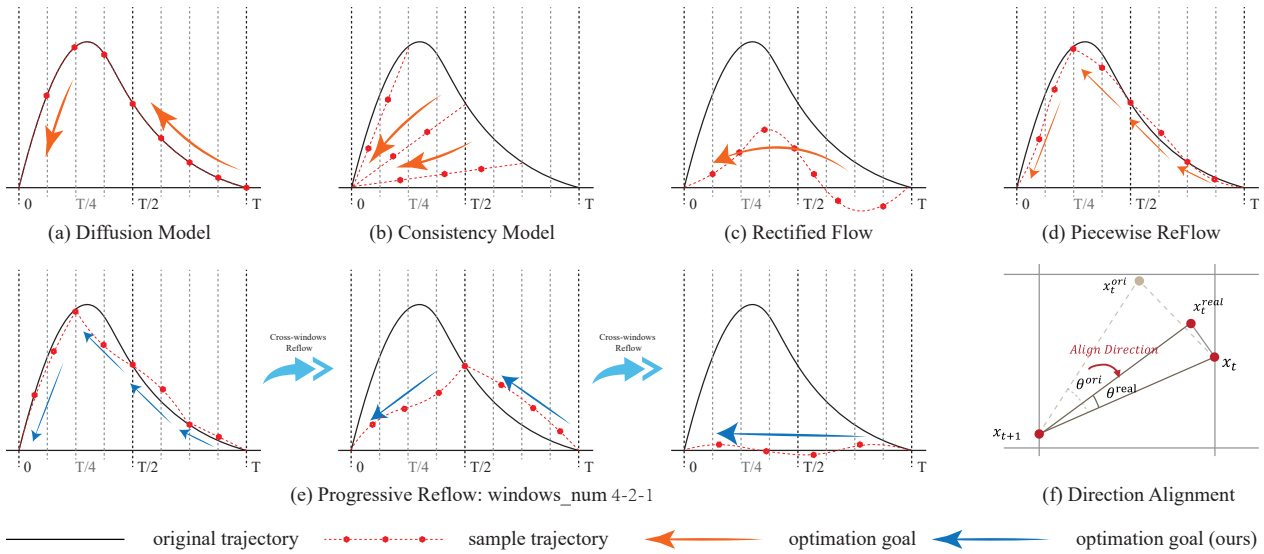



Figure 2. Conceptual illustration of different methods. (a)–(e) compare training objectives and sampling trajectories across different methods. Arrows show optimization targets, and red dashed lines represent actual sampling trajectories, which are curved due to the optimization not achieving the theoretical optimum. (e) shows our progressive reflow method achieves better approximation. (f) presents how our proposed aligned v-prediction works between timesteps $[t, t+1]$, it reduces prediction deviation with velocity direction correction.

scratch [17]. InstaFlow [18] first extended ReFlow to large-scale text-to-image models through consecutive ReFlow to straighten the ODE trajectory, followed by distillation to achieve single-step sampling. Subsequently, some works explored improving ReFlow's effectiveness or simplifying its training[13, 14, 42, 43]. While ReFlow showed promise for single-step generation, its few-step sampling performance lagged behind state-of-the-art methods [30, 34, 39]. To address this limitation, PeRFlow [44] proposed trajectory partitioning into time windows, achieving competitive few-step sampling through localized straightening within each temporal segment.

### 2.4. Privileged Information in Distillation

Although knowledge distillation has been proven effective as a model compression technique and further extended successfully to diffusion model acceleration, the theoretical explanation for its efficacy has remained elusive. How 'dark knowledge' is effectively captured from teacher models and utilized to guide student learning remains a fundamental theoretical question [6].

Lopez-Paz *et al.* [19] presented a unified theoretical framework that connects distillation with privileged information, establishing a generalized framework for understanding machine-to-machine knowledge transfer. Viewing distillation as a transfer of privileged information, TAKD [25] showed that an assistant model of intermediate capacity could more effectively mediate the knowledge flow between teacher and student models.

## 3. Methods

We present ProReflow, a more robust flow model training method. Our approach is motivated by the observation that training efficient few-step flow models faces two main challenges: (1) the significant trajectory approximation gap between teacher and student models, and (2) the difficulty in achieving accurate velocity prediction across large time intervals. To address these challenges, we propose progressive reflow for stable optimization of sample trajectory and aligned v-prediction for achieving precise velocity prediction, respectively, shown in Fig. 2.

### 3.1. Temporal Segmentation for ReFlow

**Rectified Flow** ReFlow aims to achieve temporally consistent velocity predictions across all timesteps. Given initial Gaussian distribution $\pi_0$ and target image distribution $\pi_1$, where $X_1 \sim \pi_1$ and $X_0 \sim \pi_0$. Reflow defines a linear process from $\pi_0$ to $\pi_1$, where the corresponding sampling process follows the ODE:

$$dX_t = v(X_t, t)dt, \quad t \in [0, 1], \tag{1}$$

Then, it formulates a least-squares optimization problem to ensure the predictions consistency:

$$\min_{\theta} \int_0^1 \mathbb{E}[\|X_1 - X_0 - v_\theta(X_t, t)\|^2], \tag{2}$$

where $X_t = tX_1 + (1-t)X_0$.

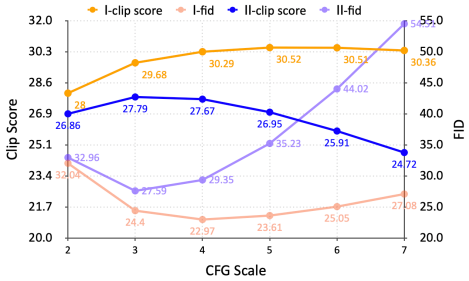



Figure 3. Performance of our models under different factors of classifier-free guidance (CFG) on COCO-2017. CFG scale ranges from 2 to 7. I and II stands for ProReflow-I with 4 steps and ProReflow-II with 2 steps, respectively.

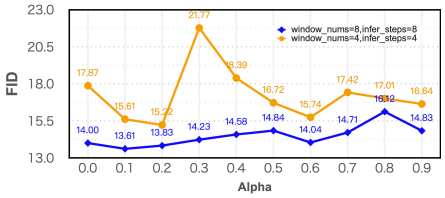



Figure 4. FID on COCO-30K. The yellow curve shows results trained with 4 windows and evaluated using 4 inference steps, while the blue curve represents the model trained with 8 windows and evaluated using 8 inference steps. Both configurations are compared against their baselines where $\alpha = 0$ (MSE loss only). Each model is trained for 10,000 iterations with batch size 32.

**Piecewise ReFlow** Aimed at improving the few-step generation, PeRFlow divides the sampling trajectory into multiple time windows, defined by endpoints $1 = t_K > \cdots > t_k > t_{k-1} > \cdots > t_0 = 0$. Within each time window $[t_k, t_{k-1})$ formed by adjacent endpoints, PeRFlow assumes a linear process to straighten the trajectory, thus eq.(2) can be reformulated as:

$$\min_{\theta} \sum_{k=1}^{K} \mathbb{E}_{z_{t_k}\sim\pi_k} \left[ \int_{t_{k-1}}^{t_k} \| \frac{z_{t_{k-1}} - z_{t_k}}{t_{k-1} - t_k} - v_\theta(z_t, t) \|^2 dt \right], \tag{3}$$

where $z_t = \alpha_t z_{t_k} + (1-\alpha_t) z_{t_{k-1}}, \alpha_t = \frac{t - t_{k-1}}{t_k - t_{k-1}}$. Finally, PeRFlow results in a piecewise linear trajectory composed of multiple segments.

### 3.2. Progressive ReFlow

PeRFlow originally sets the number of time windows to 4. Despite achieving improvement in few-step inference, PeRFlow faces a significant optimization challenge: it attempts to approximate the teacher model's irregular trajectory using four linear intervals within a single training stage.

We propose a multi-stage progressive training scheme to tackle this challenge. Rather than directly mapping the original trajectory to four time windows, our method first obtains an eight-window approximation from the original trajectory, and subsequently apply *Cross-windows ReFlow* to refine this eight-window representation into the target four-window configuration.

**Cross-windows ReFlow** Consider three consecutive time points $t_{k-1}, t_k, t_{k+1}$. The optimization objectives in first training stage can be formulated as:

$$\min_{\theta} \left( \mathbb{E}_{z_{t_k}\sim\pi_k} \int_{t_{k-1}}^{t_k} \| \frac{z_{t_{k-1}} - z_{t_k}}{t_{k-1} - t_k} - v_\theta(z_t, t) \|^2 dt + \mathbb{E}_{z_{t_{k+1}}\sim\pi_{k+1}} \int_{t_k}^{t_{k+1}} \| \frac{z_{t_k} - z_{t_{k+1}}}{t_k - t_{k+1}} - v_\theta(z_t, t) \|^2 dt \right), \tag{4}$$

where $z_t = \begin{cases} \alpha_t z_{t_k} + (1-\alpha_t) z_{t_{k-1}}, & t \in [t_{k-1}, t_k) \\ \beta_t z_{t_{k+1}} + (1-\beta_t) z_{t_k}, & t \in [t_k, t_{k+1}) \end{cases}$ with $\alpha_t = \frac{t - t_{k-1}}{t_k - t_{k-1}}$ and $\beta_t = \frac{t - t_k}{t_{k+1} - t_k}$. In adjacent time windows, trajectories evolve from $z_{t_{k-1}}$ to $z_{t_k}$ in $[t_{k-1}, t_k]$, and from $z_{t_k}$ to $z_{t_{k+1}}$ in $[t_k, t_{k+1}]$.

Cross-windows ReFlow aligns the optimization direction by guiding trajectories in both intervals to progress from $z_{t_{k-1}}$ towards $z_{t_{k+1}}$, thus eq.(4) can be reformulated as:

$$\min_{\theta} \mathbb{E}_{z_{t_{k+1}}\sim\pi_{k+1}} \int_{t_{k-1}}^{t_{k+1}} \| \frac{z_{t_{k-1}} - z_{t_{k+1}}}{t_{k-1} - t_{k+1}} - v_\theta(z_t, t) \|^2 dt, \tag{5}$$

where $z_t = \alpha_t z_{t_{k+1}} + (1-\alpha_t) z_{t_{k-1}}, \alpha_t = \frac{t - t_{k-1}}{t_{k+1} - t_{k-1}}$.

**Theoretical Explanation** Based on the theoretical framework of knowledge distillation [19], we can explain the effectiveness of Progressive ReFlow. Consider three key functions: the teacher function $f_t \in \mathcal{F}_t$ representing the original diffusion trajectory, an intermediate function $f_a \in \mathcal{F}_a$ for the 8-segment approximation, and the student function $f_s \in \mathcal{F}_s$ for the target 4-segment representation. According to the VC theory [41], when the student learns directly from the teacher, the learning rate is bounded by:

$$\mathcal{R}(f_s) - \mathcal{R}(f_t) \leq \mathcal{O}\left( \frac{|\mathcal{F}_s|_C}{n^\beta} \right) + \varepsilon_l, \tag{6}$$

where $\beta \in [\frac{1}{2}, 1]$ denotes the learning rate associated with task difficulty, $\varepsilon_l$ represents the approximation error, $\mathcal{R}$ represents the error, $\mathcal{O}(\cdot)$and $\epsilon$ represent the estimation error and approximation error, respectively.. The challenge lies in the significant capacity gap between the complex trajectory and the 4-segment approximation, resulting in a small $\beta$ that indicates difficult learning.

Progressive ReFlow decomposes this challenging pro-

cess into two stages:

$$\text{Stage 1: } \mathcal{R}(f_a) - \mathcal{R}(f_t) \leq \mathcal{O}\left(\frac{|\mathcal{F}_a|_C}{n^{\beta_1}}\right) + \varepsilon_{at}, \quad (7)$$

$$\text{Stage 2: } \mathcal{R}(f_s) - \mathcal{R}(f_a) \leq \mathcal{O}\left(\frac{|\mathcal{F}_s|_C}{n^{\beta_2}}\right) + \varepsilon_{sa}. \quad (8)$$

The effectiveness is theoretically guaranteed when:

$$\mathcal{O}\left(\frac{|\mathcal{F}_a|_C}{n^{\beta_1}} + \frac{|\mathcal{F}_s|_C}{n^{\beta_2}}\right) + \varepsilon_{at} + \varepsilon_{sa} \leq \mathcal{O}\left(\frac{|\mathcal{F}_s|_C}{n^{\beta}}\right) + \varepsilon_s, \quad (9)$$

this inequality is satisfied in practice due to two key principles: (1) 8-segment allows for better fitting of the teacher's complex sampling trajectory,leading to smaller combined approximation error ($\varepsilon_{at} + \varepsilon_{sa} < \varepsilon_l$), (2) Enhanced optimization efficiency through the progressive process, where each stage solves a simpler problem compared to direct optimization,resulting in $\beta_1, \beta_2 > \beta$.

### 3.3. Aligned V-prediction

We analyzed approximate error in the optimization process and found that directional errors lead to more significant performance degradation compared to magnitude errors, shown in Fig.1 (b). We then propose *aligned v-prediction*, which emphasizes direction alignment in training.

**Direction Matters** Consider two arbitrary points $z_{t_{i-1}}$ and $z_{t_i}$ along the trajectory. Given the target vector $v = z_{t_i} - z_{t_{i-1}}$ and the model prediction $p$. According to the law of cosines, the error between $v$ and $p$ can be expressed as:

$$r = |p|^2 + |v|^2 - 2|p||v|\cos\theta, \quad (10)$$

where $\theta$ denotes the angle between $p$ and $v$. We analyze two extreme cases:

• Misaligned, accurate magnitude ($|p| = |v|, \theta = \epsilon$):

$$r_1 = 2|v|^2(1 - \cos\epsilon); \quad (11)$$

• Aligned, inaccurate magnitude ($\theta = 0, |p| = |v| + \epsilon$):

$$r_2 = (|v| + \epsilon)^2 + |v|^2 - 2|v|(|v| + \epsilon) = \epsilon^2. \quad (12)$$

Let $y = r_1 - r_2$. Using Taylor expansion for small $\epsilon$:

$$y = (|v|^2 - 1)\epsilon^2. \quad (13)$$

Our empirical measurements using real image-noise pairs during training show that $|v|$ typically ranges from 70 to 120, yielding $y > 0$ with a substantial margin. This indicates that directional errors lead to significantly larger performance degradation than magnitude errors.

**Directional Alignment** Our analysis reveals that directional components of $v$ play a more crucial role in generation quality than magnitude. Based on this, we proposed aligned v-prediction in flow matching, which incorporates directional alignment through cosine similarity measurements. Specifically, we propose a novel flow matching loss function that places greater emphasis on directional alignment:

$$L = (1-\alpha)\cdot \text{MSE}(v, pred) + \alpha \cdot (1 - \cos(v, pred)), \quad (14)$$

where the first term provides basic magnitude consistency, the second term enforces explicit directional alignment via cosine similarity. The hyperparameter $\alpha$ balances the relative importance between magnitude and direction.

**Algorithm 1:** ProReflow Algorithm

**Input:** $\mathcal{D}$: dataset, $f_\phi$: teacher model, $K$: list of window numbers (e.g., [8,4,2]), $\alpha$: loss weight (default=0.1)

Initialize student model $f_\theta \leftarrow f_\phi$;
**for** *k in K* **do**
  Split time $[0, 1]$ into $k$ windows;
  **while** *not converged* **do**
    Sample $x$ from dataset $\mathcal{D}$;
    Sample $\epsilon \sim N(0, 1)$;
    Sample timestep $t$ and locate window $[t_1, t_2]$ s.t. $t \in [t_1, t_2]$;
    $z_{t_2} = t_2 * x + (1 - t_2)\epsilon$;
    Compute $z_{t_1}$ using $f_\phi$;
    $z_t = t * x + (1 - t)\epsilon$;
    Compute target velocity $v = x - \epsilon$;
    Predict $v_\theta = f_\theta(z_t, t)$;
    $\mathcal{L} = (1 - \alpha)\text{MSE}(v, v_\theta) + \alpha\text{Dir}(v, v_\theta)$;
    Update parameters $\theta$;
  **end**
**end**

**Output:** Trained model $f_\theta$

**Hyperparameter Configuration** Increasing the value of $\alpha$ enhances the directional supervision in the optimization objective. When $\alpha = 0$, the loss function degenerates to the conventional MSE loss. To determine the optimal hyperparameter configuration, we systematically evaluated different settings: We randomly sampled 0.8M images from LAION-art as our training set and fine-tuned SDv1.5 with different $\alpha$ values while maintaining the same number of windows. We computed FID on coco-30k to evaluate these models.

As shown in Fig. 4, the choice of $\alpha$ significantly impacts the model performance. Among the evaluated $\alpha$ values, more positive gains were observed with windows = 4 compared to windows=8, which may be attributed to the increased importance of directional consistency at larger window spans. Our experiment results show that $\alpha$=0.1 works well in all experiment settings, thus we maintained $\alpha$=0.1 for subsequent experiments.

Combining *progressive reflow* and *aligned v-prediction*, we present ProReflow, as shown in Algorithm 1.

Table 1. Performance comparison on COCO-2017 validation set, following the evaluation setup in [40]. Our method outperforms existing flow-based approaches.

| Method | Step | FID (↓) | CLIP Score(↑) |
|---|---|---|---|
| 2-Reflow [18] | 2 | 49.32 | 27.36 |
| 2-Reflow [18] | 4 | 32.97 | 28.93 |
| Instaflow-0.9B [18] | 2 | 71.54 | 26.07 |
| Instaflow-0.9B [18] | 4 | 102.41 | 24.39 |
| PeRFlow [44] | 4 | 23.81 | 30.24 |
| **ProReflow-**I (ours) | 4 | **22.97** | **30.29** |
| **ProReflow-**II (ours) | 2 | 27.59 | 27.79 |
| **ProReflow-**II (ours) | 4 | **22.03** | 29.95 |

# 4. Experiments

## 4.1. Experiment Configuration

**Model and Dataset** We evaluate our proposed method primarily on Stable Diffusion v1.5 and Stable Diffusion XL. During training, we freeze all modules except the UNet and employ BF16 mixed precision training. For SDv1.5, we initialize our training process with windows numbers=8 and progressively apply our method to derive ProReflow-I (4 windows), which subsequently serves as the basis for developing ProReflow-II. For SDXL, we adopt training configurations established in ProReflow-I on SDXL to develop ProReflow-SDXL, achieving four-steps sampling.

As for multi-stage training, we maintain consistency in the teacher model's sampling trajectory across different training stages by fixing the total DDIM steps to 32. Specifically, when windows = 8, we use 4 DDIM steps within each window to derive the endpoint from the starting point. For windows = 4, we use 8 DDIM steps per window. This ensures that the teacher's sampling trajectory remains identical across different training stages, allowing for fair comparisons and stable optimization.

SDv1.5 is trained on the LAION-Art dataset, with all images center-cropped to $512 \times 512$ resolution following its default setup. For SDXL, we fine-tune the model using a combination of LAION-Art and 1.5 million samples from the laion2B-en-aesthetic dataset, with all images center-cropped to $1024 \times 1024$ resolution. All experiments were conducted on 8 NVIDIA H20 GPUs.

**Evaluation Setting** Following common practice in text-to-image generation, we adopt two widely-used quantitative metrics: Fréchet Inception Distance (FID) [5] and clip score [28]. The evaluation is mainly conducted on two standard benchmarks: MS COCO 2014 validation dataset [15] and MS COCO 2017 validation dataset [15].

Table 2. Performance comparison on COCO-2014 validation set, following the evaluation setup in [11].

| Method | Time (↓) | Step | FID (↓) |
|---|---|---|---|
| **ODE-solver based methods** | | | |
| DPMSolver [21] | 0.88s | 25 | 9.78 |
| DPMSolver [21] | 0.34s | 8 | 22.44 |
| DPMSolver++ [20] | 0.26s | 4 | 22.36 |
| DDIM(our teacher) [36] | – | 32 | 10.05 |
| **Distillation-based methods** | | | |
| LCM-LoRA [23] | 0.12s | 2 | 24.28 |
| LCM-LoRA [23] | 0.19s | 4 | 23.62 |
| UniPC [46] | 0.19s | 4 | 23.30 |
| Flash Diffusion [1] | 0.19s | 4 | 12.41 |
| PCM [39] | 0.19s | 4 | 11.70 |
| **Flow-based methods** | | | |
| Instaflow-0.9B [18] | 0.13s | 2 | 24.61 |
| Instaflow-0.9B [18] | 0.21s | 4 | 44.01 |
| 2-ReFlow [18] | 0.13s | 2 | 20.17 |
| 2-ReFlow [18] | 0.21s | 4 | 15.32 |
| PeRFlow [44] | 0.21s | 4 | 12.01 |
| ProReflow-I (ours) | 0.21s | 4 | 11.16 |
| ProReflow-II (ours) | 0.13s | 2 | 15.44 |
| **ProReflow-**II (ours) | 0.21s | 4 | **10.70** |

## 4.2. Quantitative Results

We first compare our method with other flow-based acceleration approaches on COCO-2017 validation set, as shown in Table 1. With 4 inference steps, ProReflow-II achieves an FID of 22.03 and a CLIP score of 29.95, showing significant improvements over 2-ReFlow, Instaflow and PeRFlow.Even with only 2 steps, ProReflow-II maintains competitive performance. ProReflow-I also demonstrates strong performance with an FID of 22.97 and the highest CLIP score of 30.29. Table 2 summarizes the comprehensive evaluation results Result on COCO-2014 valdation dataset with other diffusion acccelaration methods. With 32-step DDIM serving as our teacher model, ProReflow-II achieves a competitive FID of 10.70 using only 4 steps.

Table 3 presents a comprehensive comparison of our method with advanced acceleration approaches on SDXL. Our method achieves state-of-the-art performance while maintaining the same inference cost.

## 4.3. Qualitative Comparison

We compared our method against leading flow-based approaches (Rectified Flow, InstaFlow, and PerFlow) as shown in Figure 5. Our method demonstrates superior performance across multiple aspects: it achieves more faithful detail preservation, renders more coherent global structures, and produces sharper textures with fewer artifacts.

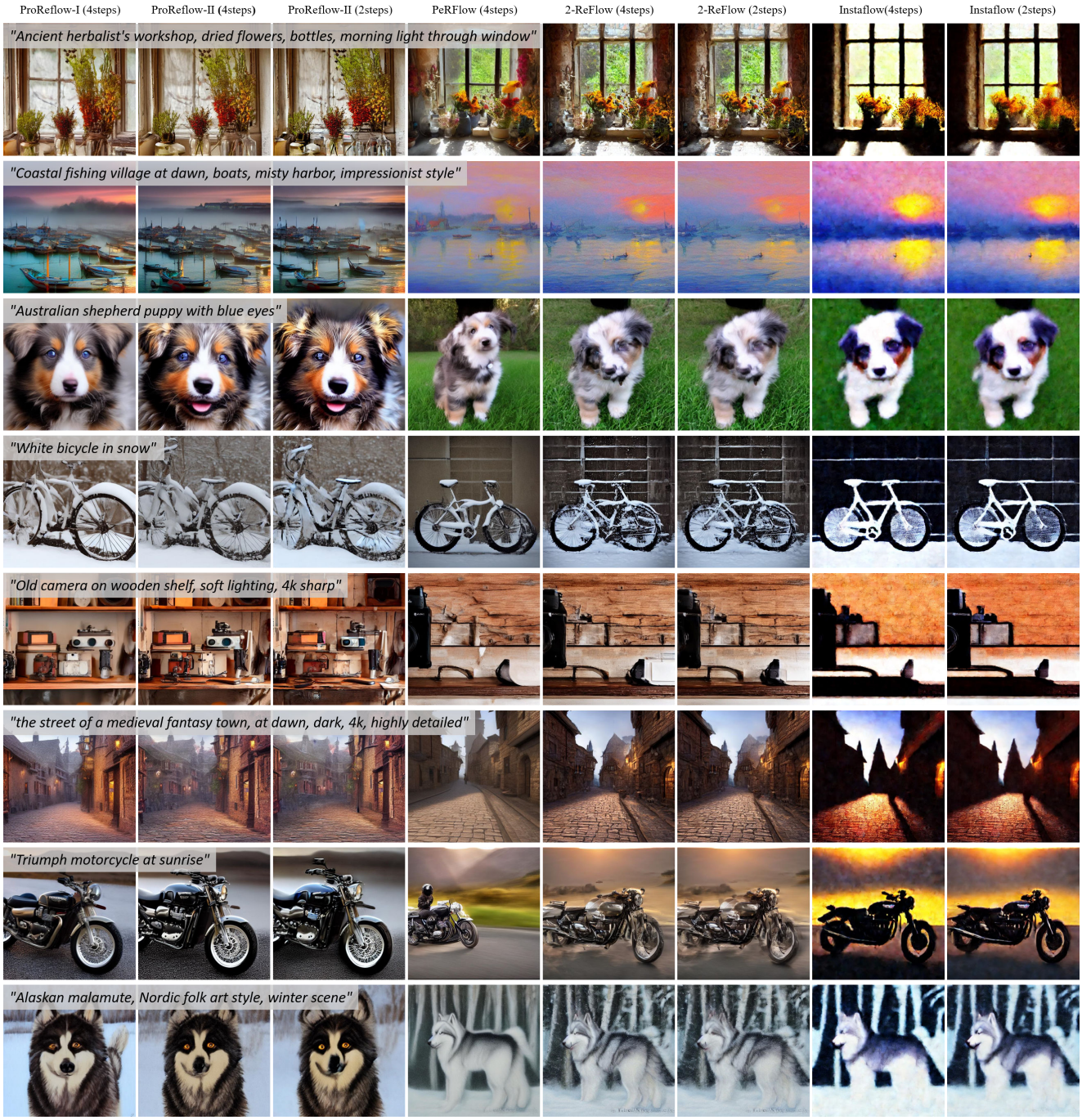



Figure 5. Qualitative comparison of image generation results. Our method demonstrates superior performance in detail rendering compared to other flow-based approaches at both 2-steps and 4-steps sampling.

Specifically, while baseline methods often struggle with detail preservation and suffer from blurry regions or structural distortions, our approach consistently maintains both fine-grained details and global coherence across various scenarios. This comprehensive improvement in generation quality validates the effectiveness of our method.

### 4.4. Training Cost

Our method, though multi-staged, is computationally cheaper compared to traditional approaches. ProReflow-II involves three distinct training stages, starting with windows=8, where each stage is trained for 10,000 iterations at

Table 3. Comparison results on SDXL on COCO2017 validation set and COCO2014-10k validation set with 4 steps, following the evaluation setup in [40].

| Method | Res. | Steps | FID (↓) |
|---|---|---|---|
| **COCO2017** | | | |
| Perflow | 1024 | 4 | 27.06 |
| Rectified Diffusion | 1024 | 4 | 25.81 |
| **ProReflow-SDXL (Ours)** | 1024 | 4 | **25.36** |
| **COCO2014-10k** | | | |
| SDXL-Lightning | 1024 | 4 | 24.56 |
| SDXL-Turbo | 1024 | 4 | 23.19 |
| LCM | 1024 | 4 | 22.16 |
| PCM | 1024 | 4 | 21.04 |
| Perflow | 1024 | 4 | 20.99 |
| Rectified Diffusion | 1024 | 4 | 19.71 |
| DMDv2 | 1024 | 4 | 19.32 |
| **ProReflow-SDXL (Ours)** | 1024 | 4 | **19.10** |

a batch size of 256. Although the number of samples remains consistent across stages, the training time varies due to differences in computational complexity. Following the approach outlined in [44], we randomly sample timesteps per batch, with windows defining the velocity prediction targets: the starting points are derived from noisy real images, while the endpoints are generated by the teacher model. With 32 teacher inference steps uniformly applied across all stages, windows=4 requires double the number of steps compared to windows=8, which directly aligns with their training time ratio. Specifically, ProReflow-I completes its training in 6.5 H20 days, while ProReflow-II extends this by an additional 8.7 days, resulting in a total training duration of 15.2 H20 days.

# 5. Discussion

## 5.1. Ablation Study

We conduct ablation studies to examine our two core designs: *aligned v-prediction* and *progressive reflow*. Table 4 presents results on COCO-2017 validation set. Both components contribute to model performance, with their combination yielding the best result.

## 5.2. CFG Influence

It is well-established that the classifier-free guidance scale $w$ is a crucial factor affecting the performance of Stable Diffusion. During training, we set $w = 1$ (i.e., without classifier-free guidance) throughout all the stages. To thoroughly understand the model's behavior under different guidance settings, we conducted extensive evaluations across a broad range of $w$ values from 2 to 7, measuring both FID and CLIP score, results are shown in Figure 3.

Table 4. Ablation studies on COCO-2017 validation set. We first show the results of gradually removing progressive reflow, aligned v-prediction, and both components, followed by our full model. We use a guidance scale of 4 for all the models.

| Method | Steps | FID (↓) | CLIP (↑) |
|---|---|---|---|
| w/o progressive reflow | 4 | 23.46 | 30.21 |
| w/o aligned v-prediction | 4 | 23.09 | 30.25 |
| w/o both | 4 | 23.81 | 30.24 |
| **ProReflow-I** | 4 | **22.97** | **30.29** |

## 5.3. Step scalability

Intuitively, for diffusion models, higher sampling steps should lead to better performance. However, this assumption does not always hold in practice, as certain models exhibit counterintuitive behavior. For instance, PeRFlow demonstrates an unexpected increase in FID when increasing sampling steps from 4 to 8 on COCO-2014 [44], which significantly limits its practical applications. We find that our progressive training scheme effectively addresses this limitation by optimizing the model's ability to generalize across varying sampling steps. Although ProReflow-II is trained with fewer window numbers(=2), it achieves lower FID score with 4-steps sampling compared to ProReflow-I, as shown in Table 1 and Table 2, demonstrating its superior performance.

# 6. Conclusion

In this paper, we propose an efficient training framework for flow-based diffusion acceleration. If viewing the optimization process from temporal and spatial dimensions, our method naturally leads to two complementary techniques that correspond to these two dimensions respectively. Temporally, *progressive reflow* bridges the trajectory approximation gap through curriculum learning, enabling gradual adaptation from more windows to fewer windows. Spatially, our velocity decomposition strategy emphasizes directional alignment over magnitude accuracy in velocity prediction. This principled design not only yields superior sampling quality but also brings advantages in optimization stability, training efficiency, and computational costs.

**Limitations** Given promising few-step sampling performance, our method shows potential for one-step generation. However, due to computational constraints, we were unable to train the model with single window to full convergence. Nevertheless, we have validated the effectiveness of velocity decomposition in this challenging setting with the same training cost, only-one-window model equipped with aligned v-prediction demonstrate superior performance compared to the vanilla counterpart. We plan to move to one-step generation when resources allow.

**Acknowledgments** This work was partly supported by the National Key Research and Development Program of China(No.2024YFB2808903), the Shenzhen Science and Technology Program(JCYJ20220818101014030) and the research grant No.CT20240905126002 of the Doubao Large Model Fund.

## References

[1] Clement Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. *arXiv preprint arXiv:2406.02347*, 2024. 6

[2] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR 2023 (Eleventh International Conference on Learning Representations)*, 2023. 1

[3] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7430–7440, 2023. 1

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[5] 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. *Advances in neural information processing systems*, 30, 2017. 6

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *stat*, 1050:9, 2015. 3

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1

[10] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 1

[11] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pages 381–399. Springer, 2024. 6

[12] Diederik P Kingma and Max Welling. Auto-encoding variational {Bayes}. In *Int. Conf. on Learning Representations*, 2014. 2

[13] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ode-based generative models. In *International Conference on Machine Learning*, pages 18957–18973. PMLR, 2023. 3

[14] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in Neural Information Processing Systems*, 37:63082–63109, 2024. 3

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[16] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2

[17] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 3

[18] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023. 2, 3, 6

[19] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015. 3, 4

[20] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 6

[21] 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 2, 6

[22] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 2

[23] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. 1, 6

[24] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 1

[25] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5191–5198, 2020. 3

[26] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion

models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 2

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[29] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2

[30] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, XING WANG, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[31] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[34] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 1, 2, 3

[35] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2025. 2

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 6

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[38] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 1, 2

[39] Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in Neural Information Processing Systems*, 37:83951–84009, 2024. 3, 6

[40] Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. *arXiv preprint arXiv:2410.07303*, 2024. 6, 8

[41] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Kdgan: Knowledge distillation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018. 4

[42] Siyu Xing, Jie Cao, Huaibo Huang, Xiao-Yu Zhang, and Ran He. Exploring straighter trajectories of flow matching with diffusion guidance. *arXiv preprint arXiv:2311.16507*, 2023. 3

[43] Chen Xu, Tianhui Song, Weixin Feng, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Accelerating image generation with sub-path linear approximation model. In *European Conference on Computer Vision*, pages 323–339. Springer, 2024. 3

[44] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Jiashi Feng, et al. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3, 6, 8

[45] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 2

[46] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 6