AMO Sampler: Enhancing Text Rendering with Overshooting

Xixi Hu[1,2]*, Keyang Xu[1]*, Bo Liu[2], Qiang Liu[2]† and Hongliang Fei[1]†
[1]Google, [2]University of Texas at Austin
{hxixi,bliu,lqiang}@cs.utexas.edu; {keyangxu,hongliangfei}@google.com

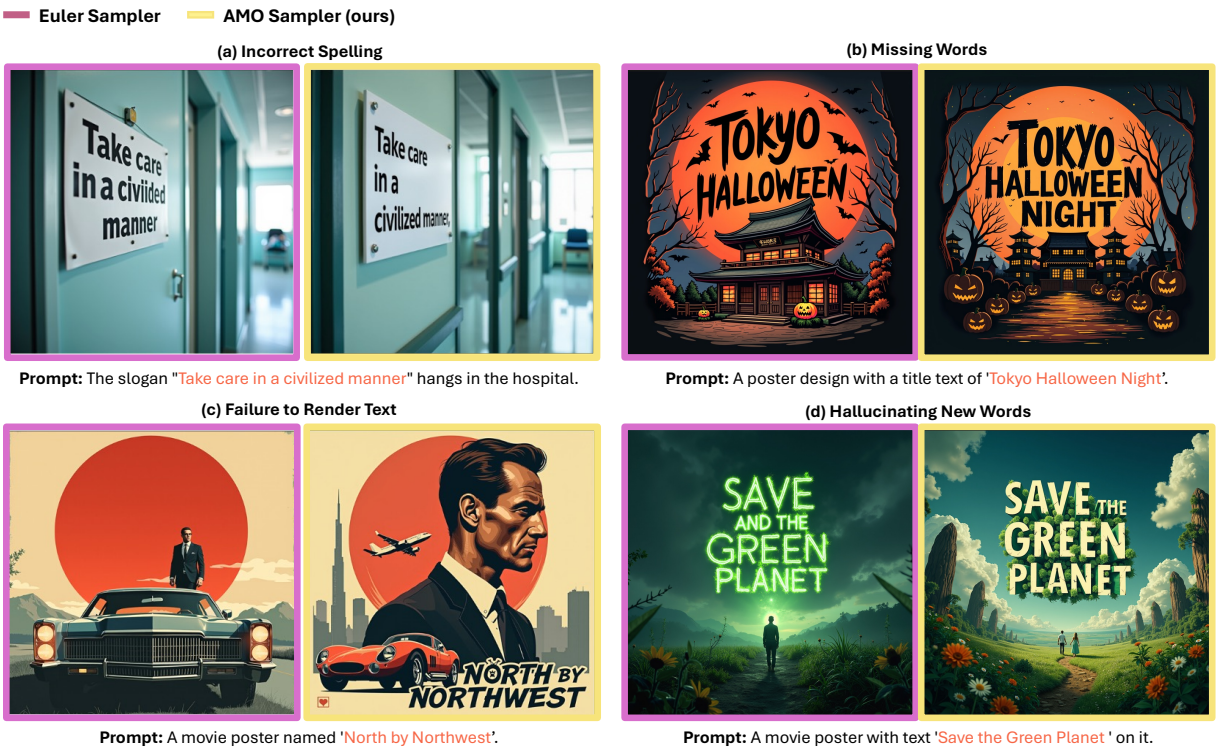


Figure 1. **Improved Text Rendering. (a)-(d)** illustrate four common text rendering mistakes in text-to-image generations. Compared to the standard Euler sampler **(purple)**, our Attention Modulated Overshooting sampler (AMO) **(yellow)** produces accurate and complete text *without* additional training, and remains as computationally efficient as the Euler sampler.

Abstract

Achieving precise alignment between textual instructions and generated images in text-to-image generation is a significant challenge, particularly in rendering written text within images. Sate-of-the-art models like Stable Diffusion 3 (SD3), Flux, and AuraFlow still struggle with accurate text depiction, resulting in misspelled or inconsistent text. We introduce a training-free method with minimal *computational overhead that* significantly *enhances text rendering quality. Specifically, we introduce an overshooting sampler for pretrained rectified flow (RF) models, by alternating between over-simulating the learned ordinary differential equation (ODE) and reintroducing noise. Compared to the Euler sampler, the overshooting sampler effectively introduces an extra Langevin dynamics term that can help correct the compounding error from successive Euler steps and therefore improve the text rendering. However, when the overshooting strength is high, we observe over-smoothing artifacts on the generated images. To address this issue, we propose an* Attention Modulated Overshooting sampler *(*AMO*), which adaptively control the strength of overshooting for each image patch according to their attention score with the text content.* AMO *demonstrates a* ***32.3%*** *and* ***35.9%*** *improvement in text rendering accuracy on SD3 and Flux without compromising overall image quality or increasing inference cost.*

∗ Equal contribution.
† Qiang Liu and Hongliang Fei jointly advised this work.

1. Introduction

Recent advances in diffusion models [18, 19, 31–33] have enabled high-quality image and video generation. Text-to-image generation, where neural networks create images from natural language prompts, has emerged as a transformative application of AI. Despite significant progress, a key challenge remains in precisely aligning generated images with given text instructions, especially in text rendering tasks, where models often struggle to display specific text accurately. This misalignment results in errors like misspellings or incorrect wording (see Figure 1 for examples), limiting the models' utility in fields like graphic design, advertising, and assistive technologies [7, 30].

Although fine-tuning with curated text data can improve text rendering [4, 5], it requires additional data collection and computationally expensive retraining, making it impractical for many applications. Furthermore, such fine-tuning may inadvertently compromise the model's overall image-generation capabilities. In this work, we investigate methods to enhance text rendering quality. We focus on rectified flow (RF) [19] models, which have emerged as a compelling alternative to conventional diffusion models due to their conceptual simplicity, ease of implementation, and improved generation quality [7]. Specifically, we introduce a lightweight, training-free sampling approach that significantly improves text rendering accuracy in generated images.

We introduce a novel and straightforward stochastic sampling approach on top of RF models, named the Overshooting sampler, that iteratively adds noise to the Euler sampler while preserving the marginal distribution. In particular, the Overshooting sampler alternates between over-simulating the learned ordinary differential equation (ODE) and re-introducing the noise (See Section 3). As we will show in Section 3.1, Overshooting sampler effectively introduces an extra Langevin dynamics term that can help correct the compounding error from the successive applications of the Euler sampler, therefore enhancing the text generation quality. The overshooting strength is controlled by a hyper-parameter $c > 0$, corresponding to the magnitude of the Langevin step. When c is large, the Langevin term becomes inaccurate and can introduce error itself. To mitigate this problem, we propose a targeted use of the Overshooting sampler for text rendering, by adaptively controlling its strength on different patches of the image according to their attention scores with the text content in the prompt. We name the combined approach as Attention Modulated Overshooting sampler (AMO).

We validate the AMO sampler on state-of-the-art RF-based text-to-image models, including SD3, Flux, and AuraFlow. Our experiments demonstrate a *significant* improvement in text rendering accuracy, with correct text generation rates increasing by **32.3%** on SD3 and **35.9%** on Flux, *without* compromising overall image quality.

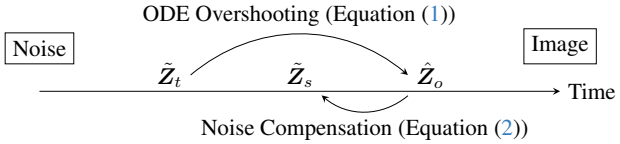


Figure 2. Visualization of the Overshooting Sampler. Given $\tilde{\boldsymbol{Z}}_t$ at time t, we first over-simulate the learned ODE to $\hat{\boldsymbol{Z}}_o$, and then add noise and return to $\tilde{\boldsymbol{Z}}_s$. The noise is carefully selected such that $\tilde{\boldsymbol{Z}}_s$ matches $\boldsymbol{X}_s$'s marginal distribution.

2. Background on Rectified Flow

This section provides a brief introduction to Rectified Flow (RF) [19]. RF seeks to learn a mapping from an easy-to-sample initial distribution $\boldsymbol{X}_0 \sim \pi_0$, which we assume to be the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, to a target data distribution $\boldsymbol{X}_1 \sim \pi_1$. This is achieved by learning a velocity field $\boldsymbol{v}$ that minimizes the following objective:

$$\min_{\boldsymbol{v}} \int_0^1 \mathbb{E}_{(\boldsymbol{X}_0, \boldsymbol{X}_1) \sim \pi_0, \times \pi_1} \left[\left\| v(\boldsymbol{X}_t, t) - \dot{\boldsymbol{X}}_t \right\|^2 \right] dt.$$

In RF, $\boldsymbol{X}_t$ is defined as a time-differentiable interpolation between $\boldsymbol{X}_0$ and $\boldsymbol{X}_1$, i.e., $\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$ and $\dot{\boldsymbol{X}}_t = \boldsymbol{X}_1 - \boldsymbol{X}_0$. Once $\boldsymbol{v}$ is learned, it induces an ordinary differential equation (ODE):

$$\frac{d}{dt}\boldsymbol{Z}_t = \boldsymbol{v}(\boldsymbol{Z}_t, t), \quad \forall t \in [0,1], \quad \boldsymbol{Z}_0 = \boldsymbol{X}_0.$$

It can be shown that $\boldsymbol{Z}_t$ and $\boldsymbol{X}_t$ share the same marginal law [19] if $\boldsymbol{v}$ is learned well, and therefore simulating this ODE with $\boldsymbol{Z}_0 = \boldsymbol{X}_0$ results in $\boldsymbol{Z}_1$ being samples from the target distribution π_1. In practice, this ODE can be discretized using the Euler method by selecting N time steps $t_0 = 0 < t_1 < \cdots < t_N = 1$, and iteratively updating:

$$\tilde{\boldsymbol{Z}}_{t_{k+1}} = \tilde{\boldsymbol{Z}}_{t_k} + (t_{k+1} - t_k)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_{t_k}, t_k), \quad \tilde{\boldsymbol{Z}}_0 \sim \pi_0.$$

We use $\tilde{\boldsymbol{Z}}$ to differentiate the discretized ODE trajectory from its ideal continuous time limit $\boldsymbol{Z}$. Note the entire process is deterministic once $\tilde{\boldsymbol{Z}}_0$ is chosen.

3. Attention Modulated Overshooting Sampler

In this section, we derive the Overshooting sampler that adds stochastic noise to the Euler sampler while preserving the marginal distribution (Section 3.1). Then we illustrate this is equivalent to adding a Langevin dynamics term at each Euler step (Section 3.2). Importantly, the extra Langevin term can help correct the compounding error from successive Euler steps. When the overshooting strength is high, the stochastic sampler can introduce artifacts. To mitigate this problem, we propose a straightforward attention modulation method that adaptively controls the strength of the overshooting for each image patch based on its attention score with the text content in the prompt (Section 3.3).

Algorithm 1 Attention-Modulated Overshoot Sampling for Rectified Flow.

procedure OVERSHOOTINGSAMPLER($v,\ \{t_i\}_{i=0}^{N},\ c \in \mathbb{R}^+$)
  Initialize $\tilde{\boldsymbol{Z}}_0 \sim \mathcal{N}\big(\mathbf{0},\ \boldsymbol{I}\big)$
  for $i \in \{0, \dots, N-1\}$ **do**
    Calculate velocity $\boldsymbol{v}_i = v(\tilde{\boldsymbol{Z}}_{t_i}; t_i)$; get the cross attention mask $\boldsymbol{m}_i$
    Overshooted ODE update:

$$\hat{\boldsymbol{Z}}_{\boldsymbol{o}} = \tilde{\boldsymbol{Z}}_{t_i} + (\boldsymbol{o} - t_i) \circ \boldsymbol{v}_i, \quad \text{with} \quad \boldsymbol{o} = \min\big(t_{i+1} + c(t_{i+1} - t_i)\boldsymbol{m}_i,\ 1\big).$$

    Backward update by adding noise:

$$\tilde{\boldsymbol{Z}}_{t_{i+1}} \leftarrow \boldsymbol{a}\hat{\boldsymbol{Z}}_{\boldsymbol{o}} + \boldsymbol{b}\boldsymbol{\xi}_i, \quad \text{where } \boldsymbol{\xi}_i \sim \mathcal{N}\big(\mathbf{0}, \boldsymbol{I}\big), \text{ and } \boldsymbol{a} = \frac{s}{\boldsymbol{o}} \text{ and } \boldsymbol{b} = \sqrt{(1-s)^2 - (\boldsymbol{a}(1-\boldsymbol{o}))^2}.$$

  return $\tilde{\boldsymbol{Z}}_{t_N}$

3.1. Stochastic Sampling via Overshooting

This section provides a derivation of a stochastic sampling method, Overshooting sampler, for RF-trained models. The main idea is to overshoot the forward Euler step and subsequently compensate with backward noise injection. In the limit of small step sizes, this process converges to a stochastic differential equation (SDE), and we will show rigorously in the subsequent section that the resulting SDE ensures the marginal preserving property according to the Fokker-Planck equation.

Following our notation in the previous section, let $\tilde{\boldsymbol{Z}}_0 = \boldsymbol{X}_0 \sim \pi_0$ be a sample from the initial noise distribution, and assume we have obtained $\tilde{\boldsymbol{Z}}_t$ at time t, and want to get a $\tilde{\boldsymbol{Z}}_s$ for the next time point $s = t + \epsilon$, where $\epsilon > 0$ is the step size when denoising. Note that for standard Euler sampler, $\tilde{\boldsymbol{Z}}_s$ is obtained from $\tilde{\boldsymbol{Z}}_t + \epsilon \boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t)$. In comparison, to introduce stochastic noise, we propose the overshooting sampler which consists of the following two steps (See Figure 2):

1. ODE Overshooting First, we temporarily advance $\tilde{\boldsymbol{Z}}_t$ to $\hat{\boldsymbol{Z}}_o$ where $o = s + c\epsilon$ (with $c > 0$) denotes an overshooting time point that is larger than s. Specifically, we conduct the following the forward Euler step:

$$\hat{\boldsymbol{Z}}_o = \tilde{\boldsymbol{Z}}_t + \boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t)(o - t) = \tilde{\boldsymbol{Z}}_t + (1 + c)\epsilon \boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t). \quad (1)$$

We use $\hat{\boldsymbol{Z}}_o$ to emphasize that it is reached via overshooting.

2. Noise Compensation Next, we want to revert from time o to time s by *noising* $\hat{\boldsymbol{Z}}_o$. Assume we achieve this by computing

$$\tilde{\boldsymbol{Z}}_s = a\hat{\boldsymbol{Z}}_o + b\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

Then, the goal is to determine the coefficients a and b here. If we assume the overshooting step is accurate, then $\hat{\boldsymbol{Z}}_o \overset{\text{Law}}{\approx} \boldsymbol{Z}_o \overset{\text{Law}}{=} o\boldsymbol{X}_1 + (1-o)\boldsymbol{X}_0$, where $\overset{\text{Law}}{=}$ denotes equality in distribution. Therefore,

$$\begin{aligned} \tilde{\boldsymbol{Z}}_s &\overset{\text{Law}}{=} a(o\boldsymbol{X}_1 + (1-o)\boldsymbol{X}_0) + b\xi \\ &= ao\boldsymbol{X}_1 + \sqrt{a^2(1-o)^2 + b^2}\xi', \end{aligned} \quad (3)$$

where $\xi' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the last line is derived as both ξ and $\boldsymbol{X}_0$ are i.i.d Gaussian noises. On the other hand, we know that $\boldsymbol{Z}_s \overset{\text{Law}}{=} s\boldsymbol{X}_1 + (1-s)\boldsymbol{X}_s$. Hence, by matching the coefficients, we get

$$a = \frac{s}{o}, \quad b = \sqrt{(1-s)^2 - s^2\frac{(1-o)^2}{o^2}}. \quad (4)$$

Recall that $s = t + \epsilon$ and $o = t + (1+c)\epsilon$, as $\epsilon \to 0$, the above process (steps 1 and 2) will approach the following limiting stochastic different equation (SDE) (See Appendix A.1 for the derivation):

$$d\boldsymbol{Z}_t = \left((1+c)\boldsymbol{v}(\boldsymbol{Z}_t, t) - \frac{c}{t}\boldsymbol{Z}_t\right)dt + \sqrt{\frac{2(1-t)}{t}c}\, d\boldsymbol{W}_t, \quad (5)$$

where $\boldsymbol{W}_t$ denotes the Brownian motion.

3.2. Overshooting ≈ Euler + Langevin Dynamics

Equation (5) can also be derived directly from the Fokker-Planck equation following similar ideas from [33]. Let $d\boldsymbol{Z}_t = \boldsymbol{f}_t(\boldsymbol{Z}_t)dt + \sigma_t d\boldsymbol{W}_t$ be a SDE where $\sigma_t \geq 0$ is a diffusion coefficient independent of $\boldsymbol{X}_t$. (This is true in almost all contemporary diffusion/flow models.) Denote by ρ_t the density of $\boldsymbol{Z}_t$. According to the Fokker-Planck equation:

$$\begin{aligned} \dot{\rho}_t &= -\nabla \cdot \big(\boldsymbol{f}_t\rho_t\big) + \frac{1}{2}\nabla^2\big(\sigma_t^2\rho_t\big), \\ &= -\nabla \cdot \big((\boldsymbol{f}_t - \frac{\sigma_t^2}{2}\nabla \log \rho_t)\rho_t\big). \end{aligned} \quad (6)$$

The last line follows $\nabla \rho_t / \rho_t = \nabla \log \rho_t$. Now, let $\boldsymbol{f}_t(\boldsymbol{Z}_t) = \boldsymbol{v}(\boldsymbol{Z}_t, t) + \frac{\sigma_t^2}{2} \nabla \log \rho_t(\boldsymbol{Z}_t)$, one can check that for any function σ_t (according to Equation (6)), the SDE

$$d\boldsymbol{Z}_t = \left(\boldsymbol{v}(\boldsymbol{Z}_t, t) + \frac{\sigma_t^2}{2} \nabla \log \rho_t(\boldsymbol{Z}_t) \right) dt + \sigma_t d\boldsymbol{W}_t, \quad (7)$$

shares the *same* marginal law at any time t as the ODE

$$d\boldsymbol{Z}_t = \boldsymbol{v}(\boldsymbol{Z}_t, t) dt.$$

This is because the $\frac{\sigma_t^2}{2} \nabla \log \rho_t$ cancels out in Equation (6) and it reduces to the continuity equation $\dot{\rho}_t = -\nabla \cdot (\boldsymbol{v}_t \rho_t)$ for the original ODE. Note that for RF models, $\nabla \log \rho_t(x) = (t\boldsymbol{v}(x,t) - x)/(1-t)$. Hence, by choosing $\sigma_t^2 = 2(1-t)c/t$, one exactly recovers Equation (5). The full derivation is provided in Appendix A.2.

Langevin Dynamics Corrects Marginals Note that Equation (7) can be equivalently viewed as the ODE combined with one step of Langevin dynamics:

$$d\boldsymbol{Z}_t = \boldsymbol{v}(\boldsymbol{Z}_t, t) dt + \underbrace{\left(\alpha_t \nabla \log \rho_t(\boldsymbol{Z}_t) dt + \sqrt{2\alpha_t} d\boldsymbol{W}_t \right)}_{\text{Langevin Dynamics}}, \quad (8)$$

where $\alpha_t = \sigma_t^2/2$ is the step size for the Langevin dynamics. As $\tilde{\boldsymbol{Z}}_t$ comes from successive Euler steps, it does not necessarily follow ρ_t of $\boldsymbol{Z}_t$. In this case, the Langevin term in Equation (8) helps move $\tilde{\boldsymbol{Z}}_t$ towards the desired marginal ρ_t (see e.g., [14]). In Figure 3, we visualize this correction effect on a 2D toy problem.

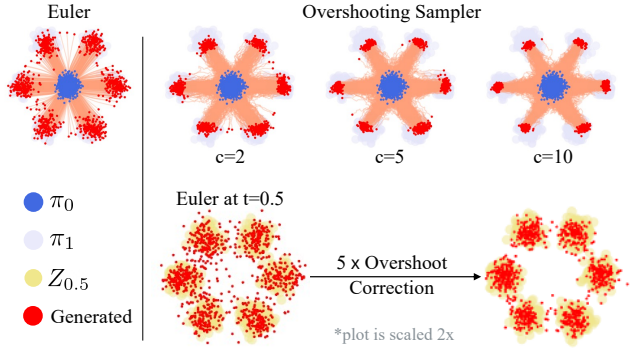


Figure 3. **Euler versus Overshooting on a toy dataset.** The noise (π_0) and data (π_1) distributions are shown as blue and light-purple dots. **Top:** The samples from Euler deviate from π_1. Overshooting sampler helps correct the marginal. As c increases, the correction effect is stronger, but it also introduces smoothing artifacts. **Bottom:** Starting with $\tilde{\boldsymbol{Z}}_t$ ($t = 0.5$) from the Euler sampler, if we apply 5 times of (Overshooting - Euler), i.e., the Langevin dynamics part in Equation (8)), the samples align better with $\boldsymbol{Z}_{0.5}$.

Remark It is worth noting that the above equivalence (between Equation (5) and Overshooting) only holds in the limit of infinitely small step sizes. Compared to applying the Euler discretization of the SDE in Equation (5), we empirically found that the Overshooting sampler tends to be more stable and yields better text rendering on real images (See Table 1).

	20 Steps	50 Steps	100 Steps
Discretize SDE in Equation (5)	51.5 %	71.0 %	76.0 %
Overshooting	**68.5 %**	**81.5 %**	**81.5 %**

Table 1. **Text Rendering Accuracy between Discretize SDE in Equation (5) and Overshooting**. We conduct a human study on the text rendering accuracy and compare discretizing SDE in Equation (5) and Overshooting. We compare the two samplers for 20, 50, and 100 inference steps, and found that with fewer steps, the Overshooting sampler demonstrates a more significant improvement over applying the discretized SDE. This is because when ϵ (the step size) is large, the Euler discretization becomes inaccurate and unstable.

3.3. Attention Modulation

In practice, while increasing c can enhance text rendering quality, it may also introduce artifacts (see Figure 6 for examples). This is because, with larger values of c, the single-step Langevin correction in Equation (8) becomes less accurate. To address this issue, we propose dynamically adjusting the overshooting strength for different image patches based on their attention scores to the text content in the prompt. Simply put, this approach increases overshooting for areas related to the text while applying less to the rest of the image.

More concretely, assume the image consists of $h \times w$ patches, where h and w denote the height and width dimensions of the 2D image tokens (e.g., 64×64 in our experiment). Let $\boldsymbol{x}_{h,w}^{\texttt{image}}$ be the (h, w)-th image patch token. Let $\{\boldsymbol{x}_i^{\texttt{text}}\}_{i=1}^n$ denote the set of tokens within the language instruction prompt for generating text and n is the total number of text-related tokens (e.g., so $\{\boldsymbol{x}^{\texttt{text}}\}$ can be the 'Tokyo Halloween Night' in the prompt "A poster design with a title text of 'Tokyo Halloween Night'). Then, we construct a mask $\boldsymbol{m} \in \mathbb{R}^{h \times w}$ in the following way:

$$\boldsymbol{m}_{h,w} = \sum_{i=1}^{n} \frac{\exp(Q(\boldsymbol{x}_i^{\texttt{text}})^\top K(\boldsymbol{x}_{h,w}^{\texttt{image}}))}{\sum_{h',w'} \exp(Q(\boldsymbol{x}_i^{\texttt{text}})^\top K(\boldsymbol{x}_{h',w'}^{\texttt{image}}))}, \quad (9)$$

where Q and K denote the query and key vectors in attention. We then average the attention map over different layers and heads and rescale its values between 0 and 1. After that, we apply the resulting attention map, $\boldsymbol{m}$, to give different image patches different amounts of overshooting. Specifically, assume $\boldsymbol{o} \in [0,1]^{h \times w}$, where $\boldsymbol{o}_{h,w} = s + c\epsilon \boldsymbol{m}_{h,w}$, we have

$$\hat{\boldsymbol{Z}}_o = \tilde{\boldsymbol{Z}}_t + \boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t) \circ (\boldsymbol{o} - t) = \tilde{\boldsymbol{Z}}_t + \epsilon(1 + c\boldsymbol{m}) \circ \boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t), \quad (10)$$

where $\circ$ denotes element-wise product. The noise compensation step in Equation (4) is similarly adapted, where a, b, o are replaced by other vector counterparts:

$$\boldsymbol{a} = \frac{s}{\boldsymbol{o}}, \quad \boldsymbol{b} = \sqrt{(1-s)^2 - s^2 \frac{(1-\boldsymbol{o})^2}{\boldsymbol{o}^2}}. \tag{11}$$

In the above, all operations are elementwise. Both $\boldsymbol{a}$ and $\boldsymbol{b}$ have dimensions $\mathbb{R}^{h \times w}$. The entire AMO sampling process is provided in Algorithm 1.

4. Related Work

This section gives an overview of diffusion and flow-based generative models, followed by a discussion on deterministic and stochastic sampling techniques, and recent advances in enhancing text rendering in text-to-image generation.

Diffusion and Flow Models. Diffusion models [12, 32, 33] have emerged as powerful generative frameworks capable of producing high-fidelity data, including images, videos, audio, and point clouds [2, 3, 6, 13, 26, 30]. These models add noise to data in a forward process, then learn to reverse this noise to generate new samples, thereby modeling the data distribution through a progressive denoising process. Recently, Rectified Flow (RF) [1, 10, 18, 21]—also known as Flow Matching, InterFlow, and IADB—has been proposed as a novel approach that leverages an ordinary differential equation (ODE) with deterministic sampling during inference. RF simplifies the diffusion and denoising process, offering computational efficiency while maintaining high-quality generation, and positioning itself as a compelling alternative to traditional diffusion models. The rectified flow model has proven successful in various applications, including image generation[7], sound generation [9, 16] and video generation [27].

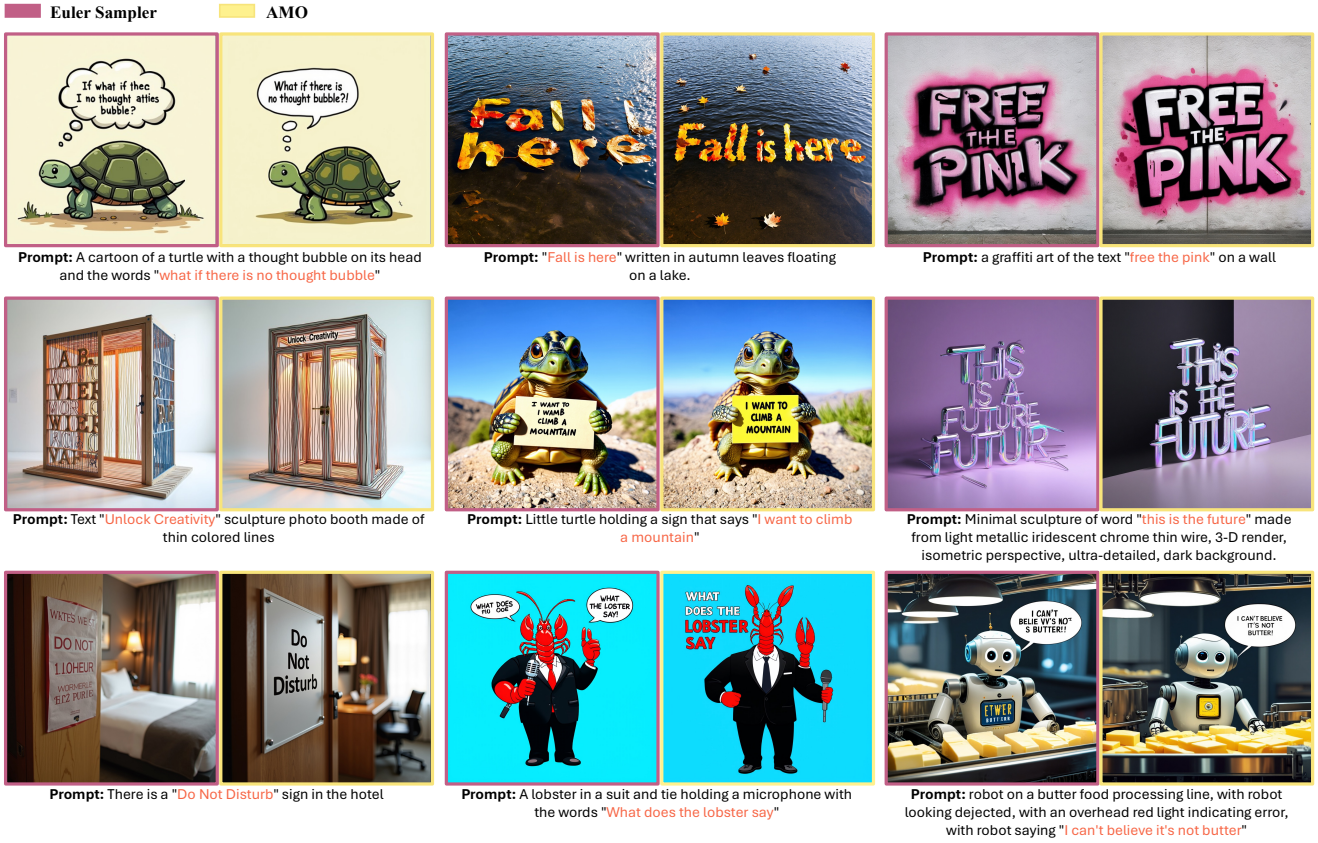
Deterministic and Stochastic Sampling Methods. Sampling strategies in generative modeling are crucial as they influence the quality, diversity, and efficiency of generated samples. Deterministic sampling methods, such as those based on ODE solvers [14, 23, 31, 33], provide computational efficiency and stability but may lead to poorer output quality [14]. On the other hand, stochastic sampling methods introduce randomness into the sampling process, offering an alternative approach with added variability in intermediate steps. Meng et al. [25] introduced an N-step stochastic sampling method for distilled diffusion models, where noise is added at intermediate steps to achieve efficient sampling with as few as 2-4 steps. This approach provides an alternative to deterministic sampling, allowing the model to produce high-quality samples. Karras et al. [14] proposed a hybrid stochastic sampling technique that combines deterministic ODE steps with noise injection. In their method, noise is temporarily added at each step to improve sampling quality, followed by an ODE backward step to maintain the correct distribution. This hybrid approach results in better output quality compared to purely deterministic sampling methods. However, these existing methods are specifically designed for diffusion models. In contrast, we propose a stochastic sampler for rectified flow, providing an alternative solution to the traditional Euler method.

Enhancing Text Rendering in T2I Generation. Accurate text rendering in text-to-image (T2I) generation models remains a significant challenge, as models often struggle to produce text within images that precisely matches the prompts, leading to incoherent or incorrect textual content. Classifier-Free Guidance (CFG)[11] can alleviate this issue by adjusting the influence of the text prompt during sampling, effectively balancing prompt guidance and the diversity of the generated content. Scaling or enlarging the text encoder has been shown to benefit text rendering[26]. Additionally, using a T5 text encoder significantly improves text rendering performance [7, 30]. Specialized fine-tuning approaches have also been explored, where pretrained text-to-image models are adapted with architectures designed specifically for text rendering tasks [4, 5, 20, 22, 24, 29, 34, 35]. These methods enhance the model's ability to generate accurate textual content but typically require extensive retraining or fine-tuning, which can be computationally intensive. In contrast, our work introduces a training-free approach that enhances text rendering during inference. By incorporating stochastic sampling and an attention mechanism into the Rectified Flow framework, we improve text rendering quality without modifying the underlying model or incurring additional training costs.

5. Experiment

We conduct experiments with several open-source text-to-image models based on Rectified Flow, including Stable Diffusion 3 (medium) [7], Flux (dev) [15], and AuraFlow [8]. Image generation was performed using the NVIDIA A40 GPU during inference. To ensure high-quality visual assessment, all output images were generated at a resolution of 1024x1024 pixels. Detailed model configurations and hyperparameter settings can be found in the Appendix A.3.

Evaluation Metrics. We use a combination of automated and human evaluations to assess the performance of our models. For automated evaluation, we adopt benchmarks including DrawTextCreative [20], ChineseDrawText [24] and TMDBEval500 [5], which comprises a total of 893 prompts drawn from various data sources. To assess the correctness of rendered text, we compute **OCR Accuracy** and **OCR F-measure** using a pre-trained Mask TextSpotter



(a) Flux (b) SD3 (c) AuraFlow

Figure 4. **Comparison of text rendering quality between Euler and our stochastic sampling method** across three different text-to-image models: (a) Flux, (b) Stable Diffusion 3 (SD3), and (c) AuraFlow. All results are generated using the same random seed for consistent comparison. Within each pair of images, the left column corresponds to the Euler sampler, while the right column displays the results from our method. Our approach consistently generates clearer and more legible text that closely matches the provided prompts. Additional examples are provided in the Appendix.

	FID (↓)	CLIP (↑)	OCR-A (↑)	OCR-F (↑)	CR (↑)
SD3 (Euler)	81.1	**0.319**	0.256	0.473	32.5 %
SD3 (AMO)	**80.9**	0.317	**0.279**	**0.482**	**43.0 %**
Flux (Euler)	111.8	0.299	0.313	0.458	74.0 %
Flux (AMO)	**108.9**	**0.303**	**0.381**	**0.494**	**82.5 %**
AuraFlow (Euler)	128.4	**0.299**	0.075	0.238	1.0 %
AuraFlow (AMO)	**117.5**	0.298	**0.082**	**0.258**	**3.0 %**

Table 2. Quantitative evaluation of AMO against the Euler sampler. **OCR-A** and **OCR-F** are short for **OCR (Accuracy)** and **OCR (F-Measure)**. **CR** is the correction rate from human evaluation.

v3 model [17]. We evaluate the samples' visual-textual alignment using **CLIP Score**, specifically CLIP L/14 [28], and also compute **FID** for overall visual quality between the CLIP image features and validation set images.

However, automated OCR tools, such as those in MARIO-Eval [4], showed limitations in accurately detecting text from the generated images. This is partly because general-purpose text-to-image models can produce diverse and artistic fonts that humans can readily understand, but OCR models cannot recognize accurately. It is also worth noting that OCR accuracy can be negatively impacted by extraneous content in images, such as posters, hurting recalls but not affecting the overall human perception. This discrepancy is especially pronounced for general-purpose text-to-image models (see Appendix A.4). To address these limitations, we conduct **human evaluation** by manually assessing the correctness of text rendering on samples covering 100 prompts. Further details are provided in the Appendix A.3.

5.1. Comparison with Euler Sampler

In this section, we compare AMO against the Euler sampler both quantitatively and qualitatively. We found that AMO significantly outperforms Euler on text rendering without sacrificing overall visual quality.

Quantitative Results As shown in Table 2, AMO achieves 82.5% accuracy on Flux model in human evaluation, notably surpassing the 74% accuracy of the standard Euler sampler. In addition, AMO improves the OCR metrics across

all three text-to-image models, demonstrating a substantial enhancement in the model's ability to render text accurately. Furthermore, compared to the standard Euler method, AMO yields better FID scores, indicating superior overall image quality. As shown in Figure 5, we evaluated the performance of AMO using 20, 50, and 100 steps. Our results demonstrate that AMO consistently outperforms the deterministic sampler across all step counts, with a better performance improvement in the low-step scenarios.

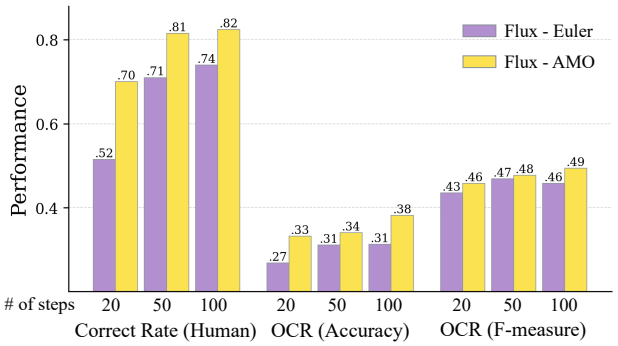


Figure 5. The comparison of Euler sampler and AMO across different sampling steps (20, 50, and 100 steps). AMO consistently outperforms the deterministic sampler on text rendering performance across all step sizes.

Qualitative Results. Figure 4 presents a visual comparison between AMO and the standard Euler sampling applied to Flux, Stable Diffusion 3, and AuraFlow. Images generated by AMO exhibit clear and legible text that closely aligns with the given prompts. In contrast, the Euler method frequently produces misaligned and misspelled text.

5.2. Ablation Studies

In this section, we conduct a detailed ablation study to analyze the impact of different components in AMO on text rendering accuracy and image quality using the Flux model.

Impact of Components in AMO. The AMO sampler essentially consists of three parts: ODE overshooting (**O**), noise compensation (**N**), and attention modulation (**A**). We ablate their individual contribution, by comparing AMO without all of them (✗, ✗, ✗) (i.e., the Euler sampler), with only overshooting (✓, ✗, ✗), without attention modulation (✓, ✓, ✗) and the full AMO. Results are summarized in Table 3. It is observed that *both* overshooting and noise compensation are crucial for achieving accurate text rendering and high image quality. Notably, introducing overshooting alone results in a 0% correct rate, as the marginal law is not preserved. The full AMO results in a similar performance to the Overshooting sampler (AMO without attention modulation), we think this is because metrics like OCR-F do not capture the image generation quality well. Therefore, we provide a visualization of samples from the Overshooting sampler against those from AMO in Figure 6. We observe that Overshooting without attention modulation can result in an over-smoothing effect, making the generated samples lose high-frequency details (See the parrot feather and smoke). This confirms the necessity of attention modulation.

(O, N, A)	FID (↓)	CLIP (↑)	OCR-A (↑)	OCR-F (↑)	CR (↑)
(✗, ✗, ✗)	111.8	0.299	0.313	0.458	74.0 %
(✓, ✗, ✗)	367.8	0.126	0.030	0.000	0.0 %
(✓, ✓, ✗)	109.5	**0.304**	0.368	**0.503**	81.5 %
(✓, ✓, ✓)	**109.0**	0.301	**0.381**	0.494	**82.5** %

Table 3. Ablation of each component in AMO on Flux.

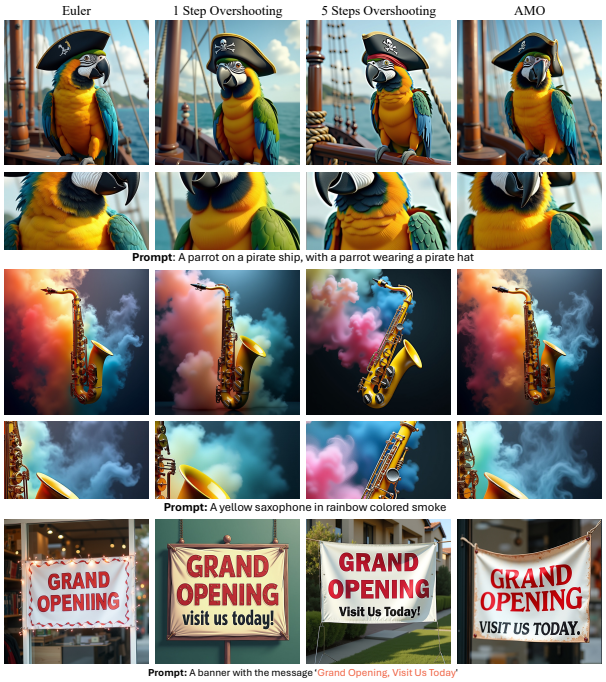


Figure 6. **Image Quality for Euler, Overshooting, and AMO.** Please zoom in for details. **Bottom:** both Overshooting (AMO without attention modulation) and AMO render the correct texts, while Euler renders misspelled texts. **Top:** Looking at the parrot's feather or the smoke behind the saxophone, Euler generates high-fidelity high-frequency details while the Overshooting sampler over-smooths the image (fewer details). AMO preserves the details from the Euler, with attention modulation. In addition, we conduct 5 Steps Overshooting, meaning that we use $c' = c/5$ but apply (Overshoot - Euler) 5 times (i.e., the Langevin step in Equation (8)) followed by 1 Euler step in the end at each time t. We see that with smaller c but more local Langevin steps the smoothing effect also goes away, but in practice, this requires more model evaluations.

Impact of Overshooting Strength. We further examine the effect of c, which governs the maximum overshooting

strength per step. Results are shown in Figure 7. Generally, increasing c enhances text rendering accuracy, with performance plateauing at $c \geq 2$ and occasionally declining for very large values of c. Consequently, we set $c = 2$ as the default in practice. To illustrate the degradation in image quality caused by high c, we direct readers to Appendix A.5.2 for examples.

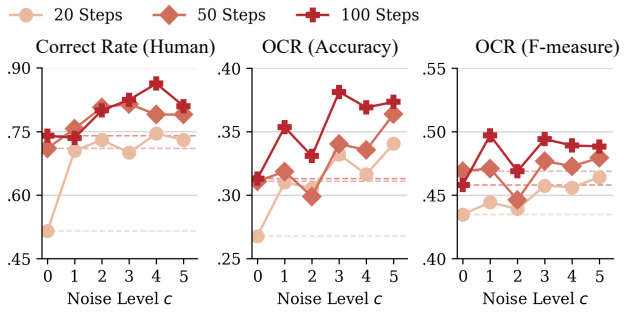


Figure 7. **Impact of overshooting strength on Text Rendering Performance.** This figure illustrates how varying the overshooting strength parameter c in AMO affects the Flux model's text rendering performance. Larger c tends to achieve higher text rendering quality. We observe that $c \geq 2$ is usually good and use $c = 2$ as the default value.

5.3. Comparison with Finetuned T2I models

In this section, we evaluate both text rendering capability and overall quality for two image generation model families for text rendering: **1)** General-purpose T2I models trained on extensive image datasets using rectified flow, such as Stable Diffusion 3, Flux and AuraFlow; and **2)** Task-specific T2I models explicitly trained on datasets of ground-truth written text, such as GlyphControl [22] and TextDiffuser [5].

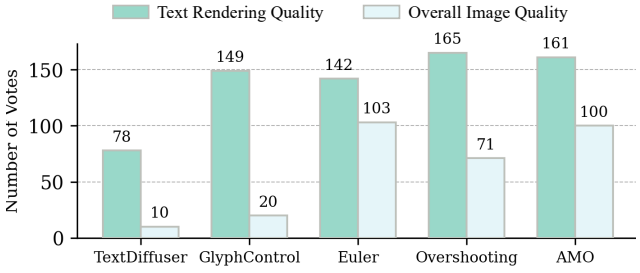


Figure 8. **Results of human evaluation comparing text rendering quality and overall image quality across five methods.** Participants viewed five images, each generated by one of the methods, and were asked to vote for: (1) the models with the best text rendering quality (multiple-choice), and (2) the model with the best overall image quality (single-choice). The chart shows the number of votes received by each method for both criteria.

Specifically, we conducted a human evaluation study to assess image generation and text rendering quality across different methods. The study involved 304 cases (selected randomly from DrawTextCreative, ChineseDrawText, and TMDBEval500) and 15 participants, with each case presenting two questions: (1) "Which of the following images exhibits the highest text rendering quality? (multiple-choice)" and (2) "Which of the following images demonstrates the best overall image quality? (single-choice)".

The results are summarized in Figure 8. As shown, AMO achieves text rendering accuracy on par with the GraphControl model, which is specifically trained for this task, while delivering superior overall image quality due to the advanced capabilities of the Flux model. Crucially, since AMO is a training-free method—unlike approaches such as TextDiffuser or GlyphControl—it can be effortlessly applied to any existing model without requiring further training or risking potential image quality loss from fine-tuning. Additionally, when compared to Overshooting (AMO without attention modulation), AMO demonstrates a clear advantage: while both yield similar text rendering quality, the Overshooting sampler diminishes image quality relative to the Euler sampler, whereas AMO maintains parity with Euler. This underscores the importance of attention modulation for optimal performance. For additional qualitative comparisons, please refer to Appendix A.6.

6. Conclusion and Limitation

Accurate text rendering has been a critical challenge in text-to-image generation. Existing solutions, such as specialized fine-tuning or scaling up the text encoder, often require modifications to the training process, which can be computationally expensive and time-consuming. This work introduces a training-free method, the Overshooting sampler sampler, that enhances text rendering by strategically incorporating curated randomness into the sampling process. Importantly, Overshooting sampler significantly improves text rendering accuracy with almost no overhead compared to the vanilla Euler sampler. In particular, Overshooting sampler alternates between overshooting the learned ODE and re-introducing noise, while ensuring the marginal laws are well-preserved. In addition, we introduce an attention modulation that quantitatively controls the degree of overshooting, concentrating on text regions within the image while minimizing interference with other areas. We validate AMO on popular open-source text-to-image flow models, including Stable Diffusion 3, Flux, and AuraFlow. Results demonstrate that AMO consistently outperforms baseline methods in text rendering accuracy without degrading the image quality of the pretrained model.

One limitation of this work is the lack of systematic evaluation of overshooting's impact on specific aesthetics or its applicability beyond image generation. Future work could explore extending AMO to other domains.

7. Acknowledgment

We thank Yeqing Li, Eugene Ie and Zihui Xue for their constructive feedback, and the anonymous reviewers for their helpful suggestions. This work was supported in part by the National Science Foundation (NSF) under Grant NSF CAREER 1846421, Office of Navy Research, NSF AI Institute for Foundations of Machine Learning (IFML) and a grant from Google.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 5

[2] Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, et al. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024. 5

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 5

[4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing the power of language models for text rendering. *arXiv preprint arXiv:2311.16465*, 2023. 2, 5, 6

[5] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5, 8

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 5

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2, 5

[8] Hugging Face. *Aura Flow Pipeline Documentation*, 2023. Huggingface. 5

[9] Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. Flux that plays music. *arXiv preprint arXiv:2409.00587*, 2024. 5

[10] Eric Heitz, Laurent Belcour, and Thomas Chambon. Iterative α-(de) blending: A minimalist deterministic diffusion model. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–8, 2023. 5

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 5

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 4, 5

[15] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2023. GitHub repository. 5

[16] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024. 5

[17] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020. 6, 13

[18] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2, 5

[19] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022. 2

[20] Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, RJ Mical, Mohammad Norouzi, and Noah Constant. Character-aware models improve visual text rendering. *arXiv preprint arXiv:2212.10562*, 2022. 5

[21] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022. 5

[22] Zeyu Liu, Weicong Liang, Yiming Zhao, Bohan Chen, Ji Li, and Yuhui Yuan. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering. *arXiv preprint arXiv:2406.10208*, 2024. 5, 8

[23] 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. 5

[24] Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation. *arXiv preprint arXiv:2303.17870*, 2023. 5

[25] 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. 5

[26] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 5

[27] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of

media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 5

[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, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3, 2022. 5

[30] 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, 5

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5

[32] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 5

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2, 3, 5

[34] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023. 5

[35] Yiming Zhao and Zhouhui Lian. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models. *arXiv preprint arXiv:2312.04884*, 2023. 5