A. Appendix

A.1. The SDE Limit of the Overshooting Sampler

In this section, we derive the asymptotic limit of the overshooting sampler's update as a stochastic differential equation (SDE) by considering the infinitesimal step size $\epsilon \to 0$ in the definitions of s and o. Recall that

$$s = t + \epsilon, \quad o = s + c\epsilon = t + (1 + c)\epsilon, \tag{12}$$

where c is a constant parameter. Combining the update equations (see Equation (1) and Equation (2)), we obtain

$$\begin{aligned} \tilde{\boldsymbol{Z}}_s &= a\hat{\boldsymbol{Z}}_o + b\boldsymbol{\xi} \\ &= a\tilde{\boldsymbol{Z}}_t + a(o-t)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t) + b\boldsymbol{\xi} \\ &= \tilde{\boldsymbol{Z}}_t + \underbrace{(a-1)\tilde{\boldsymbol{Z}}_t + a(o-t)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t)}_{\text{Drift}} + \underbrace{b\boldsymbol{\xi}}_{\text{Diffusion}}, \end{aligned} \tag{13}$$

We aim to express the update in the form

$$\tilde{\boldsymbol{Z}}_{t+\epsilon} \approx \tilde{\boldsymbol{Z}}_t + \boldsymbol{v}^{\text{adj}}(\tilde{\boldsymbol{Z}}_t, t)\epsilon + \sigma_t\sqrt{\epsilon}\boldsymbol{\xi}_t, \tag{14}$$

which corresponds to the Euler–Maruyama discretization of the SDE

$$d\boldsymbol{Z}_t = \boldsymbol{v}^{\text{adj}}(\boldsymbol{Z}_t, t)dt + \sigma_t d\boldsymbol{W}_t, \tag{15}$$

with $\boldsymbol{W}_t$ denoting a standard Wiener process.

To this end, we perform a first-order Taylor expansion assuming $\epsilon \to 0$: First, we compute $a - 1$:

$$a - 1 = \frac{s}{o} - 1 = \frac{s-o}{o} = \frac{-c\epsilon}{o} \approx -\frac{c\epsilon}{t}, \tag{16}$$

where we use the approximation $o \approx t$ for small ϵ. Next, we compute $a(o - t)$:

$$a(o-t) = \frac{s}{o}(o-t) = \frac{t+\epsilon}{t+(1+c)\epsilon}(1+c)\epsilon \approx (1+c)\epsilon, \tag{17}$$

Now, we compute b^2:

$$\begin{aligned} b^2 &= (1-s)^2 - s^2\left(\frac{1-o}{o}\right)^2 \\ &= s^2\left(\left(\frac{1-s}{s}\right)^2 - \left(\frac{1-o}{o}\right)^2\right) \\ &= s^2\left(f(s) - f(o)\right), \end{aligned} \tag{18}$$

where $f(x) = \left(\frac{1-x}{x}\right)^2$. Using a first-order Taylor expansion of $f(x)$ around $x = s$, we have

$$\begin{aligned} f(o) &\approx f(s) + f'(s)(o-s) \\ &= f(s) + f'(s)c\epsilon \\ &= 2\frac{1-t}{t}c\epsilon, \end{aligned} \tag{19}$$

Combining the above results, the update equation becomes

$$\tilde{\boldsymbol{Z}}_{t+\epsilon} \approx \tilde{\boldsymbol{Z}}_t + \boldsymbol{v}^{\text{adj}}(\tilde{\boldsymbol{Z}}_t, t)\epsilon + \sigma_t\sqrt{\epsilon}\boldsymbol{\xi}_t, \tag{20}$$

where the adjusted velocity is

$$\begin{aligned} \boldsymbol{v}^{\text{adj}}(\tilde{\boldsymbol{Z}}_t, t) &= \left(\frac{a-1}{\epsilon}\right)\tilde{\boldsymbol{Z}}_t + \left(\frac{a(o-t)}{\epsilon}\right)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t) \\ &= -\frac{c}{t}\tilde{\boldsymbol{Z}}_t + (1+c)\boldsymbol{v}(\tilde{\boldsymbol{Z}}_t, t), \end{aligned} \tag{21}$$

Thus, the limit SDE is

$$d\boldsymbol{Z}_t = \boldsymbol{v}^{\text{adj}}(\boldsymbol{Z}_t, t)dt + \sigma_t d\boldsymbol{W}_t = \left((1+c)\boldsymbol{v}(\boldsymbol{Z}_t, t) - \frac{c}{t}\boldsymbol{Z}_t\right)dt + \sqrt{\frac{2c(1-t)}{t}}d\boldsymbol{W}_t. \tag{22}$$

This provides the SDE limit of the overshooting sampler as $\epsilon \to 0$.

A.2. Stochastic Sampler by Fokker Planck Equation

As mentioned in Section 3.2, according to the Fokker-Planck Equation, for an ODE $d\boldsymbol{Z}_t = \boldsymbol{v}(\boldsymbol{Z}_t, t)dt$, we can construct a family of SDEs that share the same marginal law as the ODE at all t:

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

Now, we only need to figure out $\log \rho_t(\boldsymbol{Z}_t)$ and then we can find the corresponding σ_t^2 that matches with the limiting SDE of the overshooting algorithm. To this end, we present the next two lemmas before presenting the equivalence.

Lemma A.1. *Assume random variables $\boldsymbol{X} = \boldsymbol{Y} + \boldsymbol{Z}$, where $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are independent, then*

$$\nabla_x \log \rho_{\boldsymbol{X}}(x) = \mathbb{E}[\nabla_{\boldsymbol{y}} \log \rho_{\boldsymbol{Y}}(\boldsymbol{Y}) \mid \boldsymbol{X} = x] = \mathbb{E}[\nabla_z \log \rho_{\boldsymbol{Z}}(\boldsymbol{Z}) \mid \boldsymbol{X} = x],$$

where $\rho_{\boldsymbol{Z}}$ and $\rho_{\boldsymbol{Y}}$ are the density functions of $\boldsymbol{Z}$ and $\boldsymbol{Y}$, respectively.

Proof.

$$\begin{aligned}
\nabla_x \log \rho_{\boldsymbol{X}}(x) &= \frac{\nabla_x \rho_{\boldsymbol{X}}(x)}{\rho_{\boldsymbol{X}}(x)} \\
&= \frac{\nabla_x \int_z \rho_{\boldsymbol{X},\boldsymbol{Z}}(x, z)dz}{\rho_{\boldsymbol{X}}(x)} \\
&= \frac{\int_z \rho_{\boldsymbol{Z}}(z)\nabla_x \rho_{\boldsymbol{Y}}(x-z)dz}{\rho_{\boldsymbol{X}}(x)} \qquad \text{// } \boldsymbol{Y} \text{ and } \boldsymbol{Z} \text{ are independent} \\
&= \int_z \frac{\nabla_x \rho_{\boldsymbol{Y}}(x-z)}{\rho_{\boldsymbol{Y}}(x-z)} \frac{\rho_{\boldsymbol{Z}}(z)\rho_{\boldsymbol{Y}}(x-z)}{\rho_{\boldsymbol{X}}(x)}dz \\
&= \int_z \nabla_x \log \rho_{\boldsymbol{Y}}(x-z) \frac{\rho_{\boldsymbol{Z}}(z)\rho_{\boldsymbol{Y}}(x-z)}{\rho_{\boldsymbol{X}}(x)}dz \\
&= \mathbb{E}[\nabla_x \log \rho_{\boldsymbol{Y}}(\boldsymbol{X} - \boldsymbol{Z}) \mid \boldsymbol{X} = x] \\
&= \mathbb{E}[\nabla_{\boldsymbol{y}} \log \rho_{\boldsymbol{Y}}(\boldsymbol{Y}) \mid \boldsymbol{X} = x].
\end{aligned}$$

□

Lemma A.2. *Given the linear interpolation in Rectified Flow $\boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0$, where $\boldsymbol{X}_0 \sim \mathcal{N}(0, I)$, we have*

$$\nabla_x \log \rho_t(x) = \frac{t\boldsymbol{v}(x, t) - x}{1-t}. \tag{23}$$

Proof. As $\boldsymbol{X}_0$ and $\boldsymbol{X}_1$ are independent since $\boldsymbol{X}_0$ is the standard multivariant Gaussian and $\boldsymbol{X}_1$ is the data distribution, take $\boldsymbol{Y} = t\boldsymbol{X}_1$ and $\boldsymbol{Z} = (1-t)\boldsymbol{X}_0$. According to Lemma A.1, we have

$$\begin{aligned}
\nabla_x \log \rho_t(x) &= \mathbb{E}[\nabla_z \log \rho_{\boldsymbol{Z}}(\boldsymbol{Z}) \mid \boldsymbol{X}_t = x] \\
&= -\frac{1}{(1-t)^2}\mathbb{E}[\boldsymbol{Z} \mid \boldsymbol{X}_t = x] \qquad \text{// } \boldsymbol{Z} \sim \mathcal{N}(0, (1-t)^2 I) \\
&= -\frac{1}{1-t}\mathbb{E}[\boldsymbol{X}_0 \mid \boldsymbol{X}_t = x] \qquad \text{// } \boldsymbol{Z} = (1-t)\boldsymbol{X}_0 \\
&= \frac{1}{1-t}\mathbb{E}[t(\boldsymbol{X}_1 - \boldsymbol{X}_0) - \boldsymbol{X}_t \mid \boldsymbol{X}_t = x] \qquad \text{// } \boldsymbol{X}_t = t\boldsymbol{X}_1 + (1-t)\boldsymbol{X}_0 \\
&= \frac{t\boldsymbol{v}(x, t) - x}{1-t} \qquad \text{// } E[\boldsymbol{X}_1 - \boldsymbol{X}_0 \mid \boldsymbol{X}_t] = \boldsymbol{v}(\boldsymbol{X}_t, t).
\end{aligned}$$

□

Plugging in Equation (23) to the SDE, we have

$$d\boldsymbol{Z}_t = \left(\boldsymbol{v}(\boldsymbol{Z}_t, t) + \frac{\sigma_t^2}{2}\frac{t\boldsymbol{v}(\boldsymbol{Z}_t, t) - \boldsymbol{Z}_t}{1-t}\right)dt + \sigma_t d\boldsymbol{W}_t.$$

If we choose $\sigma_t^2 = 2c\frac{1-t}{t}$, then we get

$$\begin{aligned} d\boldsymbol{Z}_t &= \left(\boldsymbol{v}(\boldsymbol{Z}_t, t) + c\frac{1-t}{t}\frac{t\boldsymbol{v}(\boldsymbol{Z}_t, t) - \boldsymbol{Z}_t}{1-t}\right)dt + \sqrt{2c\frac{1-t}{t}}d\boldsymbol{W}_t \\ &= \left((1+c)\boldsymbol{v}(\boldsymbol{Z}_t, t) - \frac{c}{t}\boldsymbol{Z}_t\right)dt + \sqrt{2c\frac{1-t}{t}}d\boldsymbol{W}_t, \end{aligned} \tag{24}$$

which matches Equation (5) exactly.

A.3. Experiment Details

Model Configurations and Hyperparameter Settings. The hyperparameter settings for the Flux (FLUX.1-dev), Stable Diffusion 3 Medium, and AuraFlow models are summarized in Table 4. Unless stated otherwise, all experiments are conducted with a default inference step count of 100.

Hyperparameters	FLUX.1-dev	Stable Diffusion 3 Medium	AuraFlow
Image size	1024×1024	1024×1024	1024×1024
CFG scale	3.5	7.0	3.5
Model Precision	BFloat16	Float32	Float16
Overshooting Strength c	2.0	1.0	1.0

Table 4. Hyperparameter settings for our experiments.

Human Evaluation Setup. We conducted human evaluations to assess text rendering quality and image fidelity. The details of the human evaluation setup are as follows:

- **Text Rendering Evaluation:** The evaluation includes a total of 100 prompts, each consisting of 5–8 words, which are provided in the supplementary material (`prompts_human_eval.txt`). Participants are presented with a text prompt and an image generated by one of the models. They are tasked with assessing the correctness of the rendered text in the image. Each image is evaluated by at least two participants. In total, this evaluation involved 72 unique participants.
- **Comparative Evaluation of Text and Image Quality:** To compare text rendering quality and overall image quality, 100 samples were selected. These include 25 prompts each from the DrawTextCreative, ChineseDrawText, and TMDBEval500 benchmarks, as well as the primary human evaluation prompts. This evaluation was conducted with 15 participants.

This comprehensive evaluation ensures a robust assessment of the model's ability to generate high-quality images and accurately render text.

A.4. Problem in evaluating OCR

While OCR tools provide an automatic method for assessing the correctness of rendered text in images, our experiments reveal limitations in existing OCR systems when evaluating state-of-the-art text-to-image models such as FLUX. Specifically, we employed Mask TextSpotter v3 [17] and found that it struggles to accurately detect and recognize text generated by FLUX. As illustrated in Figure 9, Mask TextSpotter performs better when evaluating models like TextDiffuser and GlyphControl, which tend to generate text with simpler layouts and standard fonts. These characteristics align more closely with the training data of the OCR model, making detection easier. In contrast, FLUX-generated text exhibits greater stylistic flexibility and diversity, posing significant challenges for existing OCR tools despite the text being rendered correctly. We provide examples in Figure 9, highlighting the OCR performance disparity. The detected text boxes and predictions are shown in red. These results underscore the need for improved OCR systems capable of handling the creative and flexible text styles generated by advanced text-to-image models.

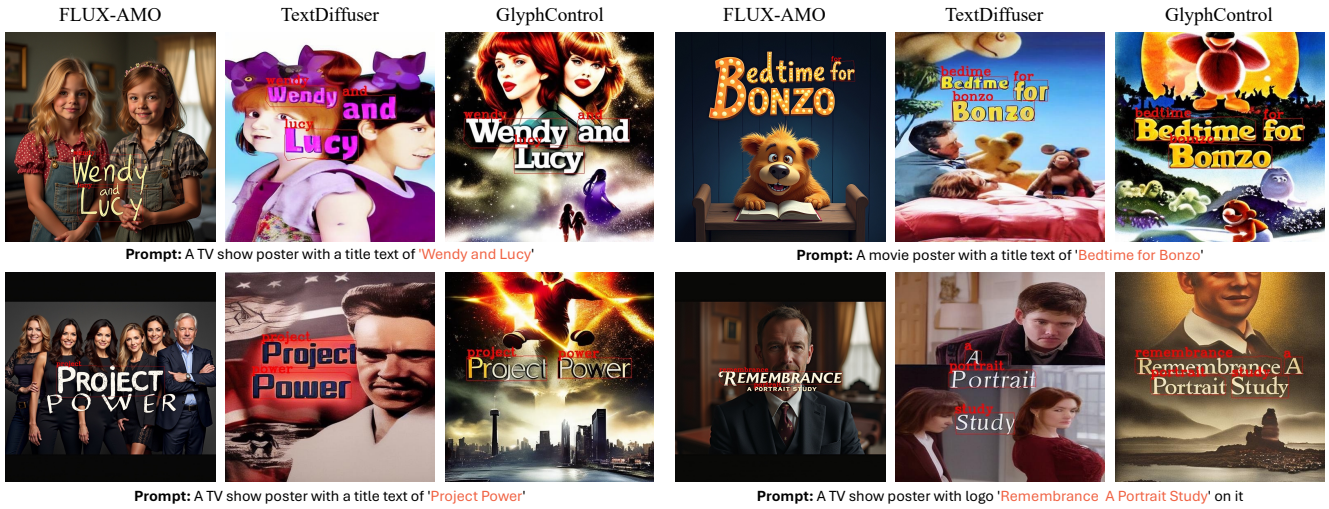


Figure 9. Examples of OCR model performance. Detected text boxes and prediction results are shown in red. The OCR model fails to detect text generated by the FLUX model effectively, even though the text is rendered correctly.

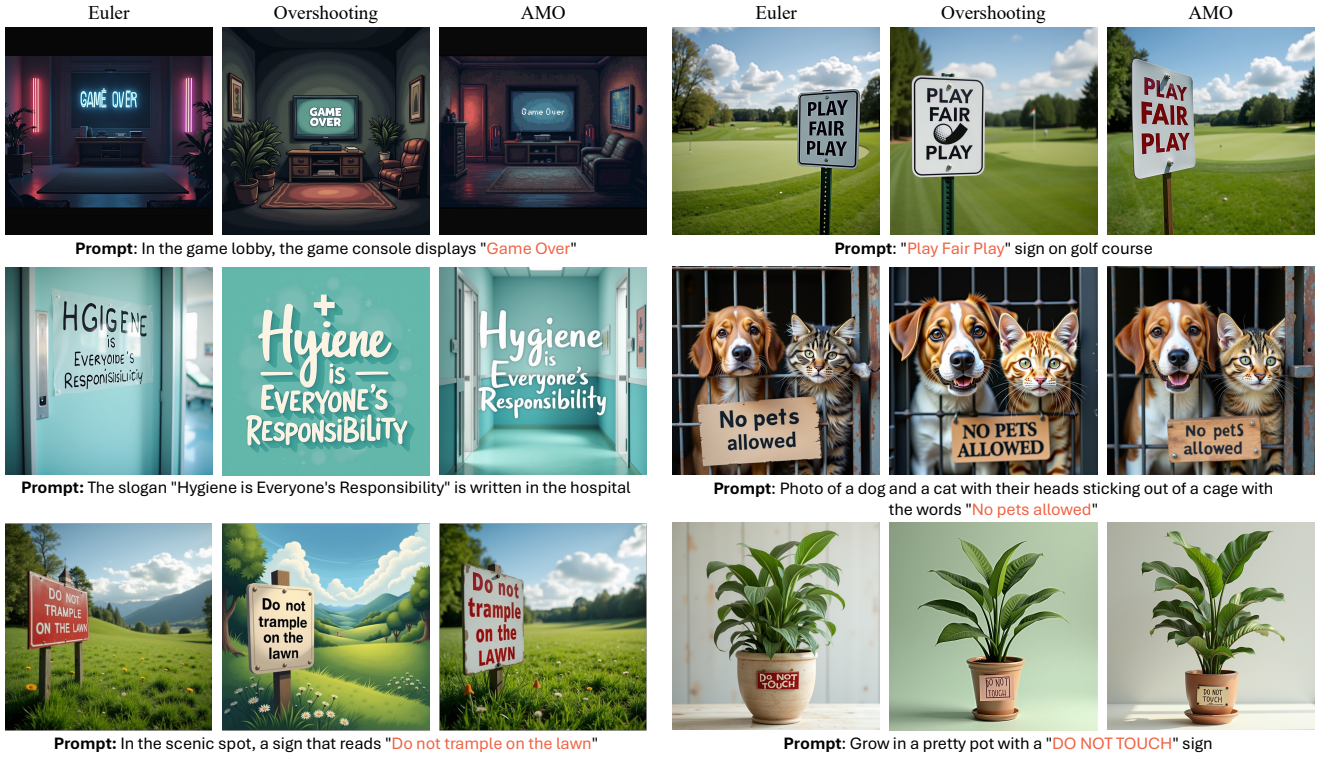


Figure 10. **Image Quality for Euler, Overshooting, and AMO.** Please zoom in for finer details. The Overshooting method shown here employs a one-step overshooting strategy, ensuring the overall computational cost remains comparable across all three methods. The overshooting approach results in cartoonish, over-smoothed outputs that lack high-frequency details. In contrast, Euler and AMO generate images that resemble real-world visuals more closely.

A.5. Additional Qualitative Results

A.5.1 Additional Results on Image Quality for Euler, Overshooting, and AMO

We present additional results in Figure 10 to further illustrate our findings. These results confirm that overshooting (without attention modulation) tends to produce an over-smoothing effect, leading to generated samples lacking high-frequency details.

Figure 11. **Samples generated by varying c.** As c increases, the images gradually lose complexity and fine details due to over-smoothing. For moderate values of c, such as $c = 2$, the results achieve a balance between accurate text rendering and visual quality.

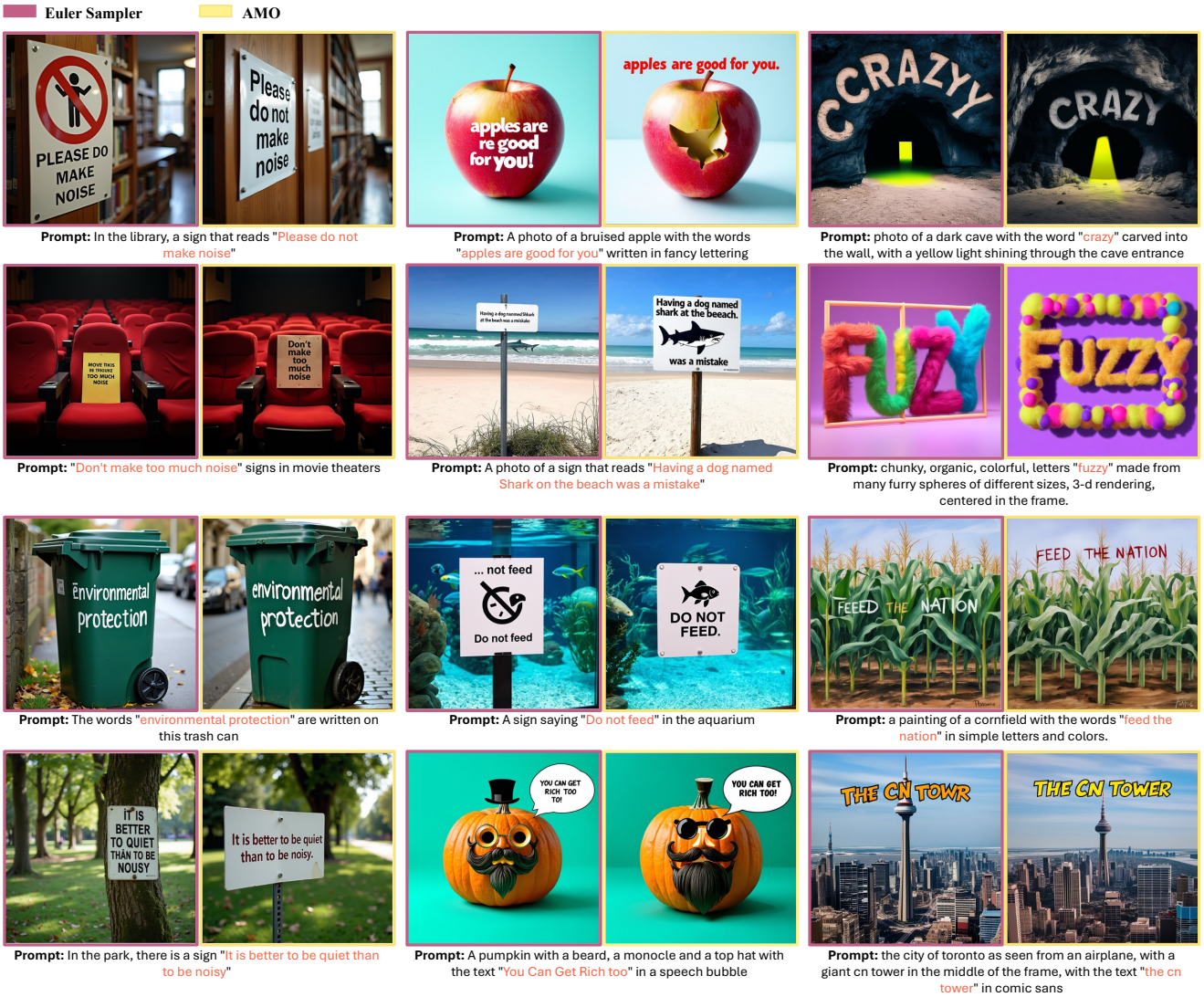
A.5.2 Quantative Results on AMO with Different Overshooting Strength c

In the experiment section, we demonstrated that increasing the overshooting strength c improves text rendering accuracy, with performance plateauing at $c \geq 2$ and occasionally declining for very large values of c. Here, we provide visual examples for

varying values of c, as shown in Figure 11. We observe that as c increases significantly, the generated images tend to exhibit simpler structures and fewer details. This behavior is expected because the attention modulation applies a soft overshooting strategy, where excessively large c introduces over-smoothing artifacts. However, these artifacts are significantly mitigated compared to results generated without attention modulation.

A.5.3 Additional Samples on Comparison between Euler and AMO

We provide more results in Figure 12. showcasing the differences between the Euler sampler and our AMO method.



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

Figure 12. **Comparison of text rendering quality between Euler and AMO.** Results are presented across three different text-to-image models: Flux, Stable Diffusion 3, and AuraFlow. All images are generated using the same random seed. In each pair of images, the left column shows the results from the Euler sampler, while the right column displays results generated by our AMO method. AMO consistently produces clearer and more legible text that aligns more closely with the given prompts, demonstrating its superiority in text rendering quality.

A.6. Additional Results on Comparison with Finetuned Text-to-Image Models

We present sample images from the human evaluation study comparing TextDiffuser, GlyphControl, Euler, Overshooting, and AMO. These examples are shown in Figure 13. During the human evaluation, participants were presented with five images generated by the respective methods and asked to answer two questions: **Question 1**: Which of the following images exhibits the highest text rendering quality? (Multiple-choice) **Question 2**: Which of the following images demonstrates the best overall image quality? (Single-choice)

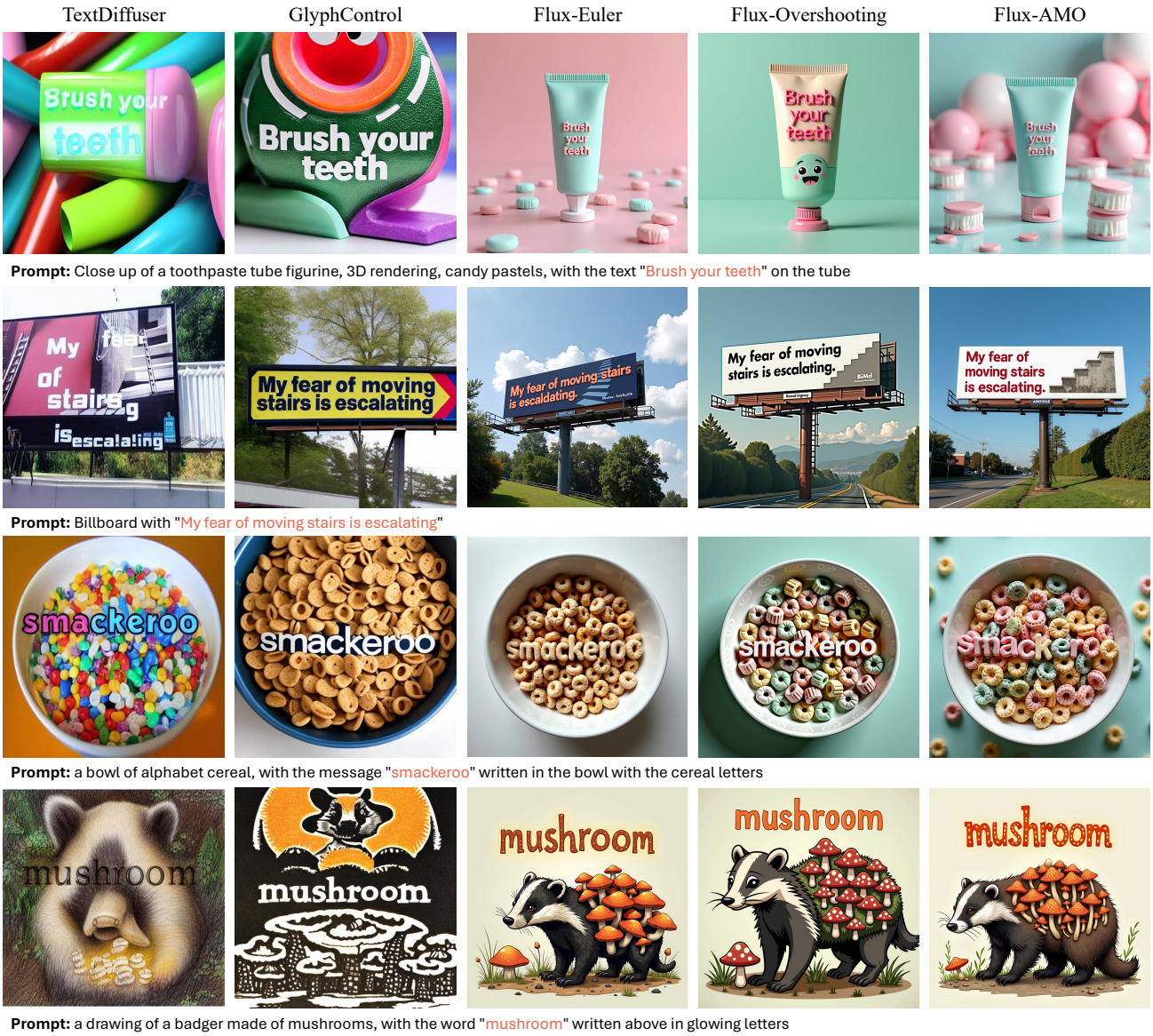


Figure 13. **Comparison of samples generated by different methods**, including TextDiffuser, GlyphControl, Euler, Overshooting, and AMO. During the human evaluation, participants were shown five images for comparison.

A.7. Exploring Tasks Beyond Text Rendering

Our initial exploration shows that overshooting sampler improves the rendering of details such as hands and human body structures (see Fig. 14). However, these improvements are difficult to quantify without extensive human evaluation. Hence, we

focus on text rendering, where OCR-based metrics, supplemented by human evaluation, provide a more direct and affordable evaluation. This work lays the groundwork for future exploration of other tasks.

Figure 14. Correcting hands and body structure using our method.

A.8. OCR-based Comparison with Specialized Text Rendering Models

We present the OCR results in Table 5, but it is crucial to mention that these numbers can be **misleading**. Current OCR tools struggle with the diverse and artistic fonts generated by general-purpose T2I models such as Flux. Specialized text rendering models, on the other hand, tend to produce text in a single, OCR-optimized font. Consequently, the seemingly lower OCR scores for Flux do not necessarily indicate poorer text rendering performance. Thus we prefer human evaluation as shown in Figure 8.

On the TMDBEval500 dataset (500 images), our manual evaluation of the OCR model revealed it has only 54% accuracy in recognizing rendered text from Flux, compared to 92% for TextDiffuser. Furthermore, extraneous text content, often generated by general-purpose T2I models, can negatively influence OCR-A by reducing precision.

	TextDiffuser	GlyphControl	Flux-Euler	Flux-Ours
OCR-A	0.491	0.537	0.313	0.381
OCR-F	0.625	0.591	0.458	0.494

Table 5. OCR-A and OCR-F results. Note that TextDiffuser and GlyphControl were optimized for OCR tools, while Flux's generated text are more diverse, leading to a lower reported score.