

# Navigating with Annealing Guidance Scale in Diffusion Space

SHAI YEHEZKEL\*, Tel Aviv University, Israel  
 OMER DAHARY\*, Tel Aviv University, Israel  
 ANDREY VOYNOV, Google DeepMind, Israel  
 DANIEL COHEN-OR, Tel Aviv University, Israel



Fig. 1. Our annealing guidance scheduler significantly enhances image quality and alignment with the text prompt.

Denoising diffusion models excel at generating high-quality images conditioned on text prompts, yet their effectiveness heavily relies on careful guidance during the sampling process. Classifier-Free Guidance (CFG) provides a widely used mechanism for steering generation by setting the guidance scale, which balances image quality and prompt alignment. However, the choice of the guidance scale has a critical impact on the convergence toward a visually appealing and prompt-adherent image. In this work, we propose an annealing guidance scheduler which dynamically adjusts the guidance scale over time based on the conditional noisy signal. By learning a scheduling policy, our method addresses the temperamental behavior of CFG. Empirical results demonstrate that our guidance scheduler significantly enhances image quality and alignment with the text prompt, advancing the performance of text-to-image generation. Notably, our novel scheduler requires no additional activations or memory consumption, and can seamlessly replace the common classifier-free guidance, offering an improved trade-off between prompt alignment and quality.

## 1 Introduction

Denoising diffusion models [Ho et al. 2020; Nichol and Dhariwal 2021; Sohl-Dickstein et al. 2015; Song et al. 2020a; Song and Ermon 2019; Song et al. 2020b] have shown outstanding abilities in text-based generation of images [Dhariwal and Nichol 2021; Podell et al. 2023; Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022]. At training, these models learn to denoise a noisy signal  $z_t$ , e.g. an image latent, based on its existing lower frequency structure, and a text prompt  $c$ . However, while these models are tasked with iteratively pushing the signal towards the conditional distribution  $p(z|c)$ , in practice, step corrections must be applied to sample high-quality results.

The widely used approach, classifier-free guidance (CFG) [Ho and Salimans 2022], suggests corrections by extrapolating predictions away from the unconditional distribution  $p(z)$ . In practice, this is performed by guiding the latent in the direction of difference between the conditional and unconditional predictions  $\delta_t(z_t) \equiv \epsilon_t^c(z_t) - \epsilon_t^\emptyset(z_t)$ , using a step size of  $w$ . This mechanism requires

\* Denotes equal contribution.

special care setting the guidance scale  $w$ , which directly affects image quality, diversity and prompt alignment of the generated image. Selecting a proper guidance scale is extremely challenging. The VAE latent space, which we refer to as *diffusion space*, has a complex high-dimensional landscape with non-uniform densities. Properly navigating through this landscape requires skipping low-likelihood regions towards a nearby mode which aligns well with the prompt.

From this perspective, we can think of CFG as a tool that assists navigating in diffusion space. CFG applies correction steps, by which the latent is iteratively refined to agree with both the prompt and the prior distribution of the diffusion model. The size  $w$  of the correction steps fundamentally affects the success of proper convergence to an image which is both visually appealing and adheres to the prompt.

Recent works have attempted to address the instability of CFG by proposing schedulers for the guidance scale  $w$ , typically defined as functions of the timestep  $t$ . However, these schedules are often manually designed and based on opposing heuristics. Crucially, such methods do not adapt to the initial noise or the evolving denoising trajectory—factors that are essential for navigating the diffusion space effectively.

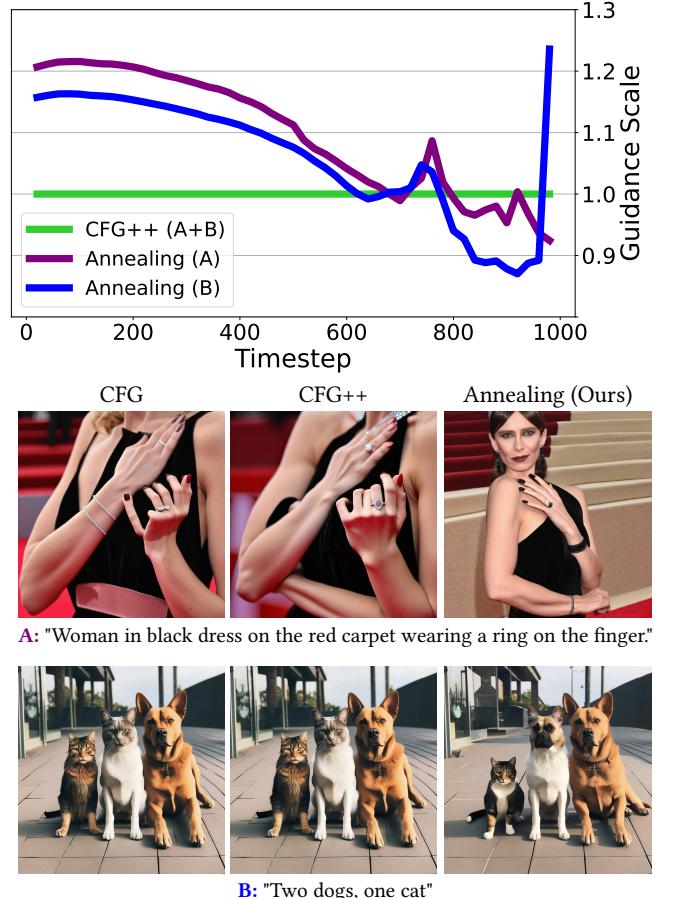
To address this limitation, we propose a learning-based scheduler that adapts the guidance scale throughout the generation process. Our approach leverages the signal  $\delta_t = \epsilon_t^c - \epsilon_t^\emptyset$ , which captures the discrepancy between the model’s conditional and unconditional predictions at each step. We train a lightweight MLP to predict  $w$  as a function of both the timestep  $t$  and  $\|\delta_t\|$ , enabling trajectory-aware, sample-specific guidance.

Our method builds upon CFG++ [Chung et al. 2024], an improved variant of CFG that casts the sampling process as an optimization problem. Specifically, it views guidance as a gradient descent step that minimizes the Score Distillation Sampling (SDS) loss [Poole et al. 2022], which measures the model’s accuracy in predicting the true noise based on the prompt. In this framework, the signal  $\delta_t$  naturally emerges as a proxy for the gradient of the SDS loss, providing a principled way to steer the denoising trajectory toward prompt-consistent samples.

Fig. 1 presents examples where our annealing scheduler enhances prompt alignment and corrects generation artifacts, resulting in visually pleasing images that more accurately reflect the user’s intent.

Fig. 2 illustrates the behavior of our annealing scheduler. As shown in the plots, the predicted scale  $w$  evolves differently across two generations (**A** and **B**), exhibiting non-monotonic fluctuations that adapts to each denoising trajectory. This adaptive behavior contrasts with the fixed guidance scales used in CFG and CFG++, which cannot account for such variations. For scene **A**, our scheduler corrects artifacts present in the baselines, most notably the distorted anatomy of the woman’s hands, resulting in a higher-quality image. For scene **B**, our method produces an image that is more faithfully aligned with the prompt, accurately capturing the specified number of objects, unlike the generations produced by the baselines.

We further explore the behavior of the annealing scheduler over a toy example, and demonstrate quantitatively and qualitatively that our navigation scheme improves the quality and prompt alignment



**Fig. 2. Guidance Scale Over Time.** Top: Guidance scale trajectories for two prompts: **A** and **B**. CFG++ uses a constant scale for both prompts, while our annealing scheduler dynamically adapts the scale per prompt. CFG is omitted from the plot for clarity but uses a fixed scale of  $w = 10$ . Bottom: Comparison of generations from CFG (left), CFG++ (center) and our method (right). Our scheduler improves both quality and alignment: resolving visual artifacts (distorted hands, scene **A**) and correcting object counts (scene **B**).

of generated images. Notably, our scheduler achieves state-of-the-art performance on FID/CLIP and FD-DINOv2/CLIP when evaluated on MSCOCO17 [Lin et al. 2014], outperforming prior methods by a considerable margin.

## 2 Related works

### 2.1 Guidance in Diffusion Models

Diffusion-based models have emerged as the driving force behind advanced generative modeling, defining the state-of-the-art in the synthesis of high-quality, diverse, and coherent data across various domains. A significant aspect of diffusion-based generative models is their ability to perform sampling guided by specific conditions, with text-based conditioning being the most commonly employed.

The conditioning mechanism in diffusion-based generative models can be implemented in various ways, with classifier-free guidance (CFG) [Ho and Salimans 2022] emerging as a foundational and widely adopted technique. CFG replaces the use of external

gradients [Dhariwal and Nichol 2021] by combining conditional and unconditional model outputs in a linear manner, offering a powerful and flexible method for controlling generation. This approach has become a standard in most modern sampling algorithms, significantly enhancing both the quality and controllability of generated outputs. Additionally, other approaches extend conditioning through internal feature corrections [Voynov et al. 2023], domain-specific architectural adaptations [Ye et al. 2023; Zhang et al. 2023], and alternative strategies [Liu et al. 2023; Tumanyan et al. 2023], further enriching the capabilities of diffusion-based models.

## 2.2 Advanced Sampling

Classifier-Free Guidance (CFG) sampling with a simple solver produces plausible results; however, models often struggle to generate complex scenes, such as those with intricate compositions or multiple elements [Chefer et al. 2023; Dahary et al. 2025]. Despite its widespread use, CFG introduces an inherent tradeoff between faithfulness to the desired prompt and diversity, where increasing the guidance scale enhances alignment with the conditioning but reduces output variability. Moreover, simply increasing the guidance scale is not always effective, as it can result in unnatural artifacts or over-saturated images that compromise realism. Additionally, certain seeds have been shown to consistently produce low-quality images [Xu et al. 2024].

Several works have proposed improved sampling techniques to address these challenges. One approach considers various non-learnable hyperparameter configurations for the noise scheduler and guidance scales [Karras et al. 2022a]. Another method introduces guidance distillation, enabling the use of a single model to streamline the sampling process [Meng et al. 2023]. To mitigate issues at higher guidance scales, some techniques suggest clipping the guidance step size to prevent over-saturation [Lin et al. 2024; Sadat et al. 2024], while others propose controlling the step size using empirically designed schedulers [Kynkänniemi et al. 2024; Sadat et al. 2023; Wang et al. 2024].

Other studies have proposed modifications to CFG to address its limitations. Some approaches restrict the guidance to the image manifold, ensuring more coherent outputs [Chung et al. 2024], while others redefine the guidance process by introducing a new basis that better separates the denoising and prompt-guidance components [Sadat et al. 2024]. More relevant to our work are techniques that employ non-constant guidance, such as adjusting the steps at which guidance is applied [Dinh et al. 2024; Kynkänniemi et al. 2024], modifying guidance based on segmentation of generated objects [Shen et al. 2024], or altering the unconditional component in the CFG formulation [Karras et al. 2024].

## 3 Overview

The Classifier-Free Guidance (CFG) sampling equation in the simplest case is given by:

$$\hat{\epsilon}_t = \epsilon_t^\emptyset + w \cdot (\epsilon_t^c - \epsilon_t^\emptyset), \quad (1)$$

where  $\hat{\epsilon}_t$  is the guided noise prediction at time step  $t$ ,  $\epsilon_t^\emptyset$  is the unconditional model output,  $\epsilon_t^c$  is the conditional model output, and  $w$  is the guidance scale that controls the extent to which we

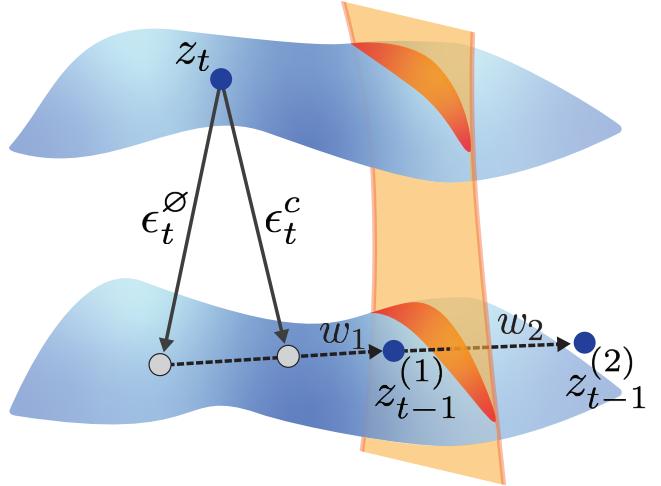


Fig. 3. Classifier-Free Guidance step. The denoising step of a sample  $z_t$  is illustrated as a linear combination of the unconditional noise prediction  $\epsilon_t^\emptyset$  and the conditional noise prediction  $\epsilon_t^c$ . The dashed line represents possible  $z_{t-1}$  predictions using CFG, for the figure simplicity, we don't depict the rescaling of  $z_t$ , which is performed at each denoising step.  $z_{t-1}^{(1)}$  and  $z_{t-1}^{(2)}$  denote predictions corresponding to two different guidance scales,  $w_1$  and  $w_2$ , respectively. The blue manifold represents the density  $p_t(z)$ , while the orange manifold illustrates the conditional distribution density  $p_{t-1}(z|c)$ .

extrapolate from the unconditional to the conditional outputs (see supplement for a detailed algorithm).

The guidance scale  $w$  determines the strength of alignment with the conditioning input, with higher values improving alignment but potentially reducing diversity or introducing artifacts. This is illustrated in Fig. 3, which depicts a denoising step of  $z_t$  over the density manifold  $p_t(z)$  toward the density manifold  $p_{t-1}(z)$ . We show the unconditional noise direction  $\epsilon_t^\emptyset$  and the conditional noise direction  $\epsilon_t^c$ .

The CFG operation aims to increase the probability  $p_t(c | z)$  while staying on the manifold defined by natural images by extrapolating between  $\epsilon_t^c$  and the unconditional prediction  $\epsilon_t^\emptyset$ , weighted by a factor  $w$ . The figure illustrates extrapolations with two scales: one with  $w_1$ , which undershoots the target distribution, and another with  $w_2$ , which overshoots.

Determining the optimal size of the guidance scale  $w$  is a non-trivial task, as it depends on the distribution's local geometry, the target prompt, the initial noise, and the model itself.

The commonly used approach is to keep  $w$  constant throughout the generation process. While other works have explored relations between  $w$  and timesteps— we argue that  $w$  should also depend on the difference defined as

$$\delta_t = \epsilon_t^c - \epsilon_t^\emptyset. \quad (2)$$

Specifically,  $\delta_t$  is affected by the model's predictions on the current noisy latent in relation to the prompt, and thus encapsulates information specific to the denoising trajectory. This dependency suggests that a fixed or simplistic scheduling of  $w$  may not be sufficient for achieving optimal results, stressing the need for more adaptive approaches.

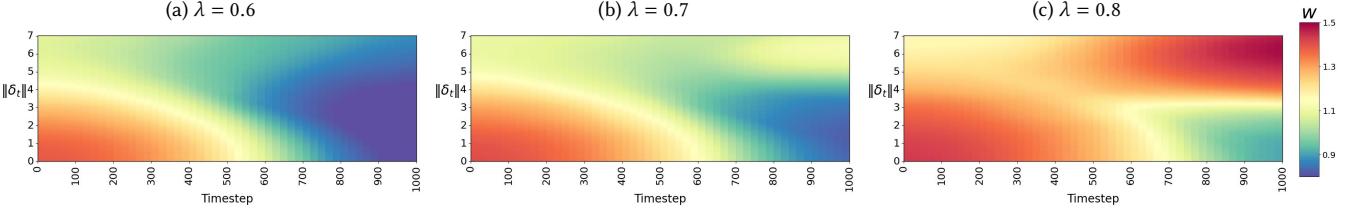


Fig. 4. Heatmaps showing the predicted guidance scale  $w_\theta$  as a function of timestep  $t$  and  $\|\delta_t\|$ , for three values of  $\lambda$ . The color represents the value of  $w_\theta(t, \|\delta_t\|, \lambda)$ , with the colormap shown on the right. Larger  $t$  corresponds to earlier diffusion steps, with  $t = 0$  marking the end of denoising. At each step,  $\|\delta_t\|$  is recomputed and used to dynamically predict the guidance scale, forming a trajectory over time as demonstrated in Fig. 2.

Given the temperamental behavior of  $w$ , we propose a learning-based approach to determine its optimal value. Specifically, we learn  $w$  as a function of the timestep  $t$  and  $\|\delta_t\|$ , enabling a more adaptive and context-aware guidance scale.

### 3.1 SDS and CFG++

Score Distillation Sampling (SDS) [Poole et al. 2022] is a technique for aligning input data with a target distribution defined by a pre-trained diffusion model, by leveraging gradients extracted from the model. It operates using the explicit SDS loss [Zhu et al. 2024]:

$$L^{\text{SDS}}(z_0) = \mathbb{E}_{t,\epsilon} \|\epsilon_t^c(z_t) - \epsilon\|_2^2, \quad (3)$$

which encourages the optimized input  $z_0$  to both align with the conditional signal  $c$  and remain consistent with the model distribution. Here,  $z_t$  is the noisy latent corresponding to  $z_0$ ,  $\epsilon$  is the sampled true noise, and  $\epsilon_t^c$  is the conditional prediction of the model.

Recent work [Chung et al. 2024] adopts this loss formulation to reinterpret guidance as a diffusion-based inverse problem [Chung et al. 2022a]. By solving for  $z_0$  under the constraint that it lies on the clean data manifold  $p_0(z)$ , this approach yields a sampling scheme similar to CFG.

This reformulation, termed CFG++, introduces two key modifications to ensure  $z_0$  is on the image manifold: (1) it restricts the guidance scale  $w^*$  to the interval  $[0, 1]$ ; and (2) in contrast to CFG, which uses the guided noise prediction  $\hat{\epsilon}_t$  for both denoising and renoising when computing  $z_{t-1}$  from  $z_t$ , CFG++ uses  $\hat{\epsilon}_t$  for denoising but reintroduces noise using the unconditional prediction  $\epsilon_t^\phi$ . We refer to the supplement for the full algorithm.

With these adjustments, Eq. (1) can be interpreted as a manifold-constrained gradient descent (MCG) step [Chung et al. 2023, 2022b] toward minimizing the SDS loss, thereby enhancing prompt alignment. Notably, the MCG is approximated at each step by

$$\nabla_{z_{0|t}} L^{\text{SDS}} = 2\gamma_t(\epsilon_t^c - \epsilon_t^\phi), \quad (4)$$

where  $z_{0|t}$  is the current estimate of denoised latent, and  $\gamma_t = \sqrt{\alpha_t}/\sqrt{1-\alpha_t}$  is a time-dependent coefficient that scales the current noise level to the latent space.

Substituting Eq. (2) into Eq. (4) reveals that  $\delta_t$  can serve as a time-normalized proxy for the SDS gradients. Consequently, smaller values of  $\|\delta_t\|$  indicate proximity to stationary points of the SDS loss. Intuitively, if  $z_t$  is within the model’s distribution, stronger

alignment between the conditional and unconditional predictions corresponds to better adherence to the prompt.

In the following section, we build upon this insight to design our scheduler. While constraining the guidance scale  $w$  to the interval  $[0, 1]$  is theoretically well-motivated, we argue that this restriction can hinder the guidance mechanism’s ability to explore diverse modes of the conditional distribution  $p(z | c)$ , ultimately limiting prompt adherence. To overcome this limitation, we lift the constraint on  $w$  and instead train our scheduler to robustly balance between mode exploration and fidelity to the data manifold.

### 4 Annealing Scheduler

Building upon our insight that  $\delta_t$  captures trajectory-specific information and that its norm is representative of the SDS convergence, we propose a learnable model  $w_\theta(t, \|\delta_t\|, \lambda)$  that maps the timestep  $t$  and the magnitude  $\|\delta_t\|$  to a guidance scale. The scalar  $\lambda \in [0, 1]$  serves as a user-defined input that controls the trade-off between image quality and prompt alignment, offering an interpretable alternative to manually selecting a fixed guidance scale  $w$ . Instead of directly tuning  $w$  as in vanilla CFG and CFG++, the user specifies a high-level preference via  $\lambda$ , and the scheduler adaptively determines the optimal  $w$  throughout the generation process. Through experimentation (Sec. 6), we show that this formulation yields more consistent and controllable outcomes.

During inference, we incorporate our scheduler to the CFG++ sampling mechanism by replacing the constant guidance scale  $w$  in Eq. (1) to achieve:

$$\hat{\epsilon}_t = \epsilon_t^\phi + w_\theta(t, \delta_t, \lambda) \cdot (\epsilon_t^c - \epsilon_t^\phi). \quad (5)$$

We implement  $w_\theta$  as a lightweight MLP and train it with a subset of the LAION-POP dataset [Schuhmann et al. 2022], which was curated for high resolution and high prompt-aligned images. We provide full implementation details in the supplement.

During training, the pre-trained diffusion model is kept frozen. At each iteration, we sample an image with its corresponding caption  $c$ , together with a random timestep  $t$  and noise  $\epsilon$  to compute  $z_t$ . The guided noise prediction  $\hat{\epsilon}_t$  is obtained from Eq. (5). The parameter  $\lambda \in [0, 1]$  is sampled uniformly.

Our training loss balances between two objectives, as governed by  $\lambda$ :

$$\mathcal{L} = \lambda L_t^\delta + (1 - \lambda) L_t^\epsilon. \quad (6)$$

Here,  $L_t^\delta$  and  $L_t^\epsilon$  are loss terms that promote prompt alignment and image quality, respectively. We now turn to formally define these losses, and refer to the supplement for the full training algorithm.

\*For simplicity, we use  $w$  to interchangeably denote the guidance scale of both CFG and CFG++.

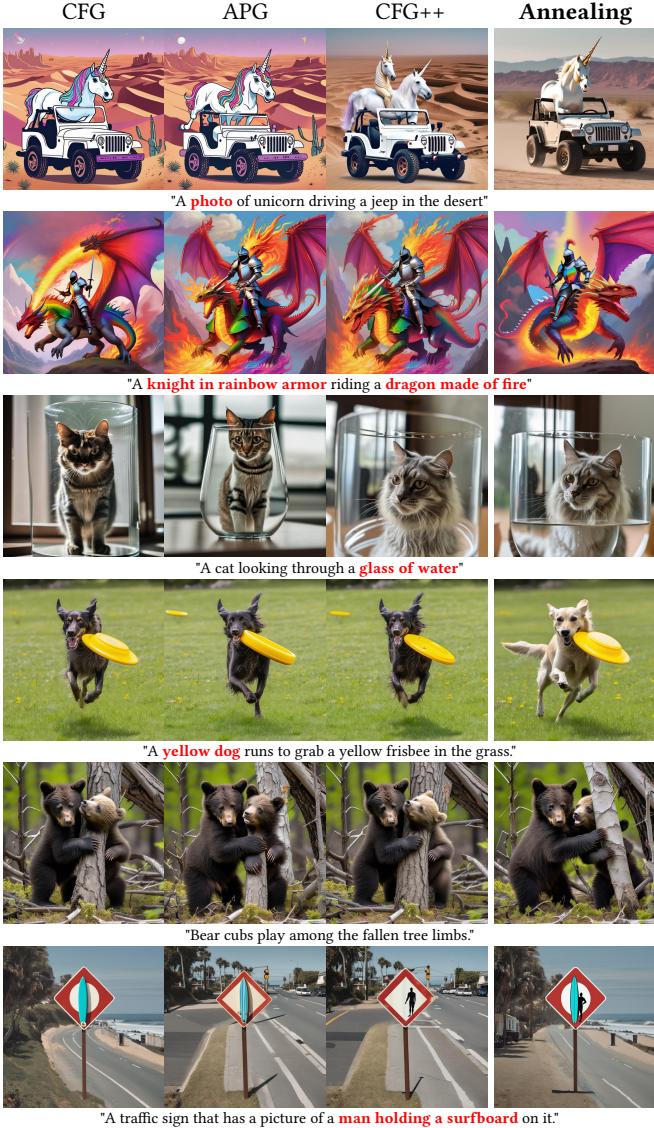


Fig. 5. Qualitative comparison of our Annealing method  $\lambda = 0.8$  (right column) vs. three guidance methods: CFG ( $w = 15$ ), APG ( $w = 20$ ) and CFG++ ( $w = 1.2$ ).

$\delta$ -loss. Following our observation in section (3.1), we introduce a novel loss, leveraging  $\|\delta_t\|$  as a proxy value that aims to reflect prompt alignment. This loss is designed to encourage the scheduler to select guidance scales that move the denoising trajectory toward regions where the model’s conditional and unconditional predictions begin to agree, indicating proximity to a prompt-consistent stationary point of the SDS loss.

In practice, for a given  $z_t$ , we perform denoising with  $\hat{\epsilon}_t$  and renoising with  $\epsilon_t^\varnothing$  to obtain  $z_{t-1}$ . By evaluating  $\|\delta_{t-1}\|$  at this point, we introduce our  $\delta$ -loss:

$$L_t^\delta = \|\delta_{t-1}\|_2^2. \quad (7)$$

This loss leverages the diffusion model’s prior of the alignment with the target prompt. However, solely optimizing on  $L_t^\delta$  results in very

high guidance scales, leading to out-of-distribution samples, similar to  $z_{t-1}^{(2)}$  in Fig. 3 (see Sec. 5.6 for further analysis). Therefore, we opt to maintain fidelity to the data manifold using the second loss term  $L_t^\epsilon$ .

$\epsilon$ -loss. To ensure that the predicted guided noise  $\hat{\epsilon}_t$  from Eq. (5) matches the sampled noise  $\epsilon$ , we introduce a denoising objective, namely, the reconstruction loss:

$$L_t^\epsilon = \|\hat{\epsilon}_t - \epsilon\|_2^2. \quad (8)$$

This loss resembles the standard denoising diffusion objective, but instead of applying to the conditional model prediction, it operates on the guided prediction  $\hat{\epsilon}_t$ , which combines both conditional and unconditional signals. Its primary role is to regularize the  $\delta$ -loss by preventing the guidance scale from pushing the generation toward implausible regions. By encouraging  $\hat{\epsilon}_t$  to remain close to the true noise  $\epsilon$ , this loss helps preserve visual quality and ensures that the denoising trajectory remains within realistic bounds.

*Prompt Perturbation.* During training, each latent  $z_t$  is paired with a prompt  $c$  that closely matches the corresponding image. Even after applying noise to obtain  $z_t$ , semantic information about the prompt remains encoded in the latent [Lin et al. 2024], preserving alignment throughout the denoising trajectory. In contrast, inference begins from pure noise, and the prompt is injected through the denoising process. As shown by prior work [Ma et al. 2025; Samuel et al. 2024; Singhal et al. 2025], the alignment of complex prompts remains highly sensitive to the initial seed, often leading to greater variability at inference time.

To simulate this mismatch, we inject Gaussian noise into the prompt embeddings during training (see supplement for details). This exposes the scheduler to imperfect prompt-image alignment, improving its robustness.

Our approach was motivated by CADS [Sadat et al. 2023], where noise is injected into the prompt embeddings *during inference* to encourage mode diversity. Their analysis showed that this perturbation smooths the conditional score  $\nabla_{z_t} \log p(c | z_t)$ , acting as a regularizer that prevents the model from collapsing onto dominant modes. In contrast, we apply this principle *during training* to enhance robustness, enabling the scheduler to generalize across a range of prompt-image alignment scenarios.

This technique improves the scheduler’s behavior across different guidance regimes. When  $\lambda$  is low and  $L_t^\epsilon$  dominates, it promotes the generation of high-quality images even under imprecise alignment. When  $\lambda$  is high and  $L_t^\delta$  dominates, it helps the scheduler adaptively shift toward nearby modes that better satisfy the prompt.

*Predicted guidance scales.* We present the learned guidance scales predicted by the trained scheduler in Figure 4. As shown, the scheduler adapts its annealing strategy based on different values of the user-specified parameter  $\lambda$ .

## 5 2D Toy Example

We now turn to investigating the behavior of our annealing scheduler in a controlled and interpretable setting using a 2D toy example.

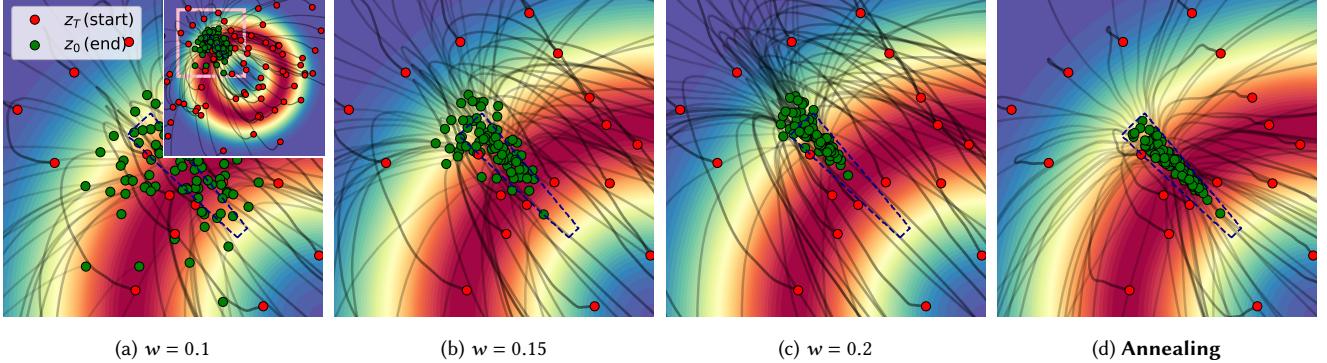


Fig. 6. A 2D diffusion toy example with a distribution density shaped as a wide ring. Random seeds conditioned on  $c = 3\pi/4$  are plotted, with their denoising trajectories shown in gray. The dashed section highlights a region within a tolerance of  $\pm\pi/64$  from  $c = \frac{3\pi}{4}$  where the manifold density is high. (a)  $w = 0.1$ : Sampling with low guidance scale shows sub-optimal condition adherence. (b)  $w = 0.15$ : Moderate guidance improves alignment, though some samples remain out of distribution. (c)  $w = 0.2$ : Stronger guidance overfits the condition, at the expense of the sample quality. (d) Our Annealing scheduler achieves better condition alignment while remaining on the sample manifold.

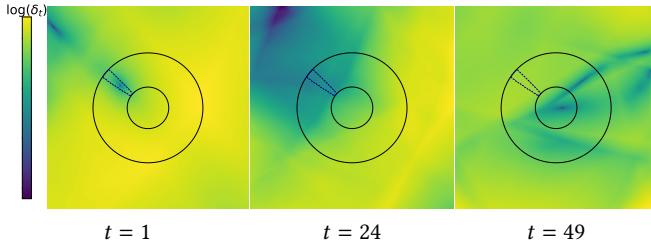


Fig. 7.  $\log \|\delta_t\|$  heatmap for  $c = 3\pi/4$ . This measures the alignment between conditional and unconditional predictions across timesteps. The region between black circles indicates high sample density; the blue dashed line marks the target condition.  $t = 49$ : Noise dominates, and predictions cluster near the source distribution center.  $t = 24$ : Alignment improves near the target, though lower values persist off-distribution.  $t = 1$ : A local minimum emerges at the target location on the ring.

In Fig. 6, we illustrate the behavior of a diffusion model trained to approximate a target distribution shaped as a wide ring. The conditional distribution is defined over the angular variable  $c \sim U(0, 2\pi)$ , while the initial noise samples  $z_T$  are drawn from a standard normal distribution. We condition generation on  $c = \frac{3\pi}{4}$  and visualize the denoising trajectories under different constant guidance scales with CFG++, as well as our adaptive scheduler.

The formed trajectories demonstrate both the strengths and limitations of classifier-free guidance. Increasing the guidance scale enforces stronger adherence to the conditioning signal but also pushes samples away from the data manifold. In contrast, our annealing scheduler (Fig. 12d) adaptively modulates the guidance strength during denoising, resulting in improved condition alignment while preserving fidelity to the data manifold. As can be seen, our scheduler also achieves better coverage of the conditional distribution, reflecting more diverse and representative generations.

To support our insight into the  $\delta_t$ -loss (Eq. 7), we display the norm  $\|\delta_t\|$  across different denoising steps in Fig. 7, for the same conditioning value  $c = \frac{3\pi}{4}$ . As  $t$  decreases, we observe that  $\|\delta_t\|$  becomes small near the correct region of the ring, indicating that the conditional and unconditional predictions are well aligned and the

sample is approaching the target mode. This implies that promoting low  $\|\delta_t\|$  throughout the denoising process can lead to better alignment with the conditioning signal.

However,  $\|\delta_t\|$  also tends to have low values away from the ring (e.g., Fig. 7,  $t = 1$ ), suggesting that minimizing  $\|\delta_t\|$  alone may guide samples off the data manifold. This highlights the need for an additional regularization term, such as the  $\epsilon$ -loss (Eq. (8)), to ensure that generations remain faithful to the data manifold.

## 6 Experiments and Results

To evaluate our annealing guidance scheduler, we conduct a comprehensive set of experiments, including qualitative comparisons, quantitative evaluations, and ablation studies. We compare our method against existing guidance scheduling approaches, including APG [Sadat et al. 2024], CFG++ [Chung et al. 2024], and the commonly used CFG [Ho and Salimans 2022] baseline. All experiments are performed using SDXL [Podell et al. 2023]. In the supplementary material, we provide additional experiments demonstrating the effectiveness of our scheduler when applied with different solvers and noise schedules, as well as its extension to flow matching models, further highlighting its generalizability.

### 6.1 Qualitative comparisons

We compare our annealing guidance scheduler qualitatively in Fig. 5. As shown, our method consistently delivers superior results both in image quality and prompt alignment.

In the first row, where the prompt specifies a photo of a unicorn driving a jeep, baseline methods produce cartoonish results or introduce visual artifacts, and none correctly place the unicorn inside the jeep. Our method, by contrast, generates a photo-realistic image that is both prompt-aligned and compositionally accurate.

In the second row, our approach is the only one to correctly render the knight in rainbow armor. Other methods leak the rainbow onto the dragon's torso, with CFG and APG even hallucinating an extra dragon head.

In the third row, only our scheduler generates the water in the glass, and in the fourth row, generates a dog with yellowish fur.

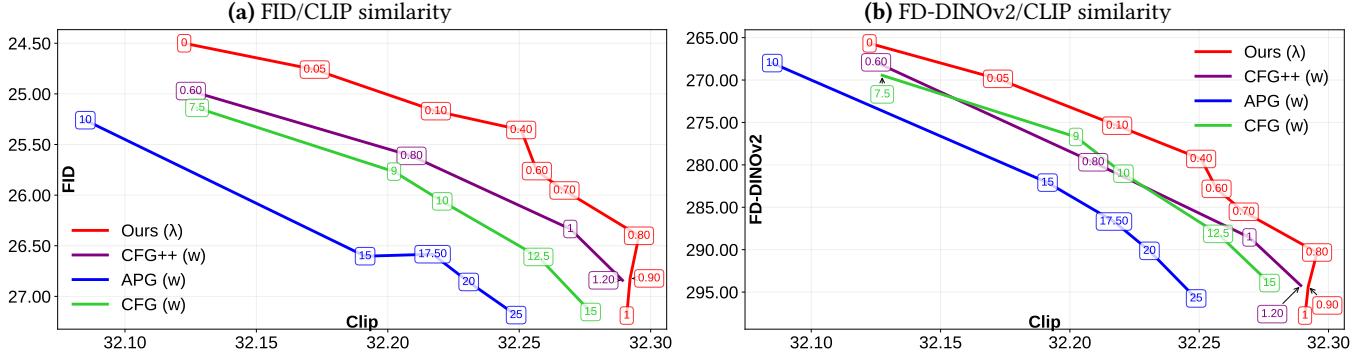


Fig. 8. Quantitative Metrics. (a) FID versus CLIP. (b) FD-DINOv2 versus CLIP.

In the fifth row, our method successfully separates the bear from the tree, whereas other methods blend the two together, producing unrealistic results.

Additional qualitative comparisons are shown in Figs. 9 and 10, against CFG++ and CFG, respectively, with more results provided in the supplementary material.

## 6.2 Quantitative comparisons

We conduct a quantitative evaluation on the MSCOCO 2017 validation set by generating 5,000 images per model using identical seeds. Image quality is assessed using FID [Heusel et al. 2018] and the recently proposed FD-DINOv2 [Oquab et al. 2024], while prompt alignment is measured via CLIP similarity [Radford et al. 2021].

To visualize the trade-off between image quality and prompt alignment at commonly used CFG guidance scales ( $w \geq 7.5$ ), we plot FID vs. CLIP and FD-DINOv2 vs. CLIP in Figure 8. As shown, APG does not improve over CFG across these metrics, while CFG++ provides gains only in the FID/CLIP space. In contrast, our method consistently enhances both alignment and image quality, outperforming all baselines across both evaluation criterias.

For direct comparison, we select multiple operating points of our scheduler by varying  $\lambda$ , and match each to the closest configuration of CFG, CFG++, or APG in terms of FD-DINOv2. Table 1 reports the corresponding FID, CLIP similarity, and additionally ImageReward [Xu et al. 2023] for human-preference, and precision and recall [Kynkänniemi et al. 2019] for quality and diversity respectively. Across all settings, our scheduler achieves the lowest FID, the highest CLIP similarity, and consistently outperforms baselines in recall at higher guidance strengths. Notably, it also attains the highest ImageReward in two out of four matched configurations. A full table including FD-DINOv2 scores, and implementation details for evaluation is provided in the supplementary material.

## 6.3 Ablation Studies

To understand the contribution of each component in our method, we conduct ablation studies by retraining the scheduler from scratch under different configurations. In all cases, we fix the prompt alignment parameter to  $\lambda = 0.8$  during evaluation, and report FID, CLIP, and ImageReward to assess the trade-off between visual quality and prompt alignment. Table 2 summarizes the results.

Method	FID ↓	CLIP ↑	IR ↑	P ↑	R ↑
CFG ( $w = 7.5$ )	25.13	32.12	0.817	<b>0.863</b>	0.630
APG ( $w = 10$ )	25.25	32.08	<b>0.818</b>	0.862	<b>0.631</b>
CFG++ ( $w = 0.6$ )	24.97	32.12	0.808	0.859	0.629
Ours ( $\lambda = 0.05$ )	<b>24.76</b>	<b>32.16</b>	0.809	0.860	0.620
CFG ( $w = 10$ )	26.06	32.22	0.859	0.859	0.594
APG ( $w = 15$ )	26.60	32.19	<b>0.865</b>	<b>0.864</b>	0.592
CFG++ ( $w = 0.8$ )	25.61	32.20	0.857	0.855	0.601
Ours ( $\lambda = 0.4$ )	<b>25.35</b>	<b>32.25</b>	<b>0.865</b>	0.859	<b>0.606</b>
CFG ( $w = 12.5$ )	26.61	32.25	0.881	0.850	0.570
APG ( $w = 17.5$ )	26.58	32.21	<b>0.887</b>	<b>0.861</b>	0.586
CFG++ ( $w = 1$ )	26.33	<b>32.26</b>	0.882	0.848	0.570
Ours ( $\lambda = 0.7$ )	<b>25.95</b>	<b>32.26</b>	0.884	0.852	<b>0.594</b>
CFG ( $w = 15$ )	27.15	32.27	0.883	0.844	0.570
APG ( $w = 20$ )	26.85	32.23	0.893	<b>0.855</b>	0.577
CFG++ ( $w = 1.2$ )	26.84	32.28	0.894	0.847	0.551
Ours ( $\lambda = 0.8$ )	<b>26.40</b>	<b>32.29</b>	<b>0.898</b>	0.846	<b>0.586</b>

Table 1. Comparison of CFG, APG, CFG++, and our method across FID, CLIP similarity, Image Reward (IR), Precision (P), and Recall (R). Arrows indicate whether higher (↑) or lower (↓) values are better.

We assess the role of inputs to the scheduler. Omitting timestep information ( $w/o t$ ) or the alignment signal ( $w/o \delta_t$ ) inputs leads to lower performance in all metrics, indicating that both inputs contribute to the overall effectiveness of our scheduler.

Dropping CFG++’s renoising step ( $w/o$  *CFG++ Renoise*) results in a significant drop in CLIP and ImageReward.

Removing prompt perturbation during training ( $w/o$  *Perturbation*) degrades performance across all metrics, indicating its importance for robustness, and constraining the predicted guidance scale  $w$  to the range  $[0, 1]$ , as done in CFG++ (*Constrained w*), achieves the lowest FID, but at the cost of reduced alignment and reward. Given this trade-off, we deliberately opt to leave  $w$  unconstrained, as it enables a better overall balance across metrics—maintaining strong prompt alignment and perceptual quality.

## 7 Conclusions

We have presented an annealing guidance scheduler that adaptively adjusts the guidance scale throughout the denoising process. Unlike the widely used CFG, which relies on a fixed guidance scale, and

Configuration	FID ↓	CLIP ↑	ImageReward ↑
Annealing ( $\lambda = 0.8$ )	26.40	<b>32.29</b>	<b>0.898</b>
w/o $t$	26.86	32.27	0.896
w/o $\delta_t$	26.97	32.28	0.896
w/o CFG++ Renoise	26.34	32.18	0.831
w/o Perturbation	27.01	32.25	0.884
Constrained w	<b>26.15</b>	32.25	0.880

Table 2. Ablation study results. Each variant removes or modifies a key component of our model.

its improved variant CFG++, our method dynamically determines step sizes based on the evolving structure of the latent space. This approach is grounded in viewing guidance as an optimization problem aimed at minimizing the SDS loss, steering latents to better match the prompt while remaining faithful to the model’s prior distribution.

We find that this adaptive strategy is particularly beneficial for complex prompts, where balancing prompt fidelity and sample quality is most challenging. Nonetheless, our results highlight a fundamental trade-off between strict adherence to the prompt and staying within the data manifold.

Navigating the high-dimensional diffusion space remains inherently difficult due to its intricate and multimodal structure. Nevertheless, our work opens the door to future exploration of more principled, context-aware guidance mechanisms that better adapt to the geometry of the denoising trajectory.

## Acknowledgments

We would like to thank Oren Katzir, Daniel Garibi, Or Patashnik, and Shelly Golan for their early feedback and insightful discussions. We also thank the anonymous reviewers for their thorough and constructive comments, which helped improve this work.

## References

- Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–10.
- Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. 2022a. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687* (2022).
- Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. 2024. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070* (2024).
- Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. 2023. Decomposed diffusion sampler for accelerating large-scale inverse problems. *arXiv preprint arXiv:2303.05754* (2023).
- Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. 2022b. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems* 35 (2022), 25683–25696.
- Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. 2025. Be yourself: Bounded attention for multi-subject text-to-image generation. In *European Conference on Computer Vision*. Springer, 432–448.
- Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.
- Anh-Dung Dinh, Daochang Liu, and Chang Xu. 2024. Compress Guidance in Conditional Diffusion Sampling. *arXiv preprint arXiv:2408.11194* (2024).
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv:1706.08500 [cs.LG]* <https://arxiv.org/abs/1706.08500>.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022a. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* 35 (2022), 26565–26577.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022b. Elucidating the Design Space of Diffusion-Based Generative Models. *arXiv:2206.00364 [cs.CV]* <https://arxiv.org/abs/2206.00364>.
- Tero Karras, Miika Aittala, Tuomas Kynkänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. 2024. Guiding a Diffusion Model with a Bad Version of Itself. *arXiv preprint arXiv:2406.02507* (2024).
- Tuomas Kynkänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. 2024. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724* (2024).
- Tuomas Kynkänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems* 32 (2019).
- Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. 2024. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 5404–5411.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v* 13. Springer, 740–755.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747* (2022).
- Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. 2023. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.
- Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. 2025. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732* (2025).
- Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. 2023. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14297–14306.
- Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*. PMLR, 8162–8171.
- Maxime Oquab, Timothée Darcret, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *arXiv:2304.07193 [cs.CV]* <https://arxiv.org/abs/2304.07193>.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs.CV]* <https://arxiv.org/abs/2103.00020>.
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. 2023. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. *arXiv preprint arXiv:2310.17347* (2023).
- Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. 2024. Eliminating Oversaturation and Artifacts of High Guidance Scales in Diffusion Models. *arXiv preprint arXiv:2410.02416* (2024).
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamayra Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.
- Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. 2024. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4695–4703.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation

- image-text models. *Advances in neural information processing systems* 35 (2022), 25278–25294.
- Dazhong Shen, Guanglu Song, Zeyue Xue, Fu-Yun Wang, and Yu Liu. 2024. Rethinking the Spatial Inconsistency in Classifier-Free Diffusion Guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9370–9379.
- Raghav Singh, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. 2025. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848* (2025).
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020a. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
- Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020b. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).
- Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1921–1930.
- Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. 2023. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–11.
- Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abreva, David Picard, and Vicky Kalogeiton. 2024. Analysis of Classifier-Free Guidance Weight Schedulers. *arXiv preprint arXiv:2404.13040* (2024).
- Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2023. ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. *arXiv:2304.05977 [cs.CV]* <https://arxiv.org/abs/2304.05977>
- Katherine Xu, Lingzhi Zhang, and Jianbo Shi. 2024. Good Seed Makes a Good Crop: Discovering Secret Seeds in Text-to-Image Diffusion Models. *arXiv preprint arXiv:2405.14828* (2024).
- Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. (2023).
- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models.
- Junzhe Zhu, Peiyi Zhuang, and Sanmi Koyejo. 2024. HiFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. *arXiv:2305.18766 [cs.CV]* <https://arxiv.org/abs/2305.18766>

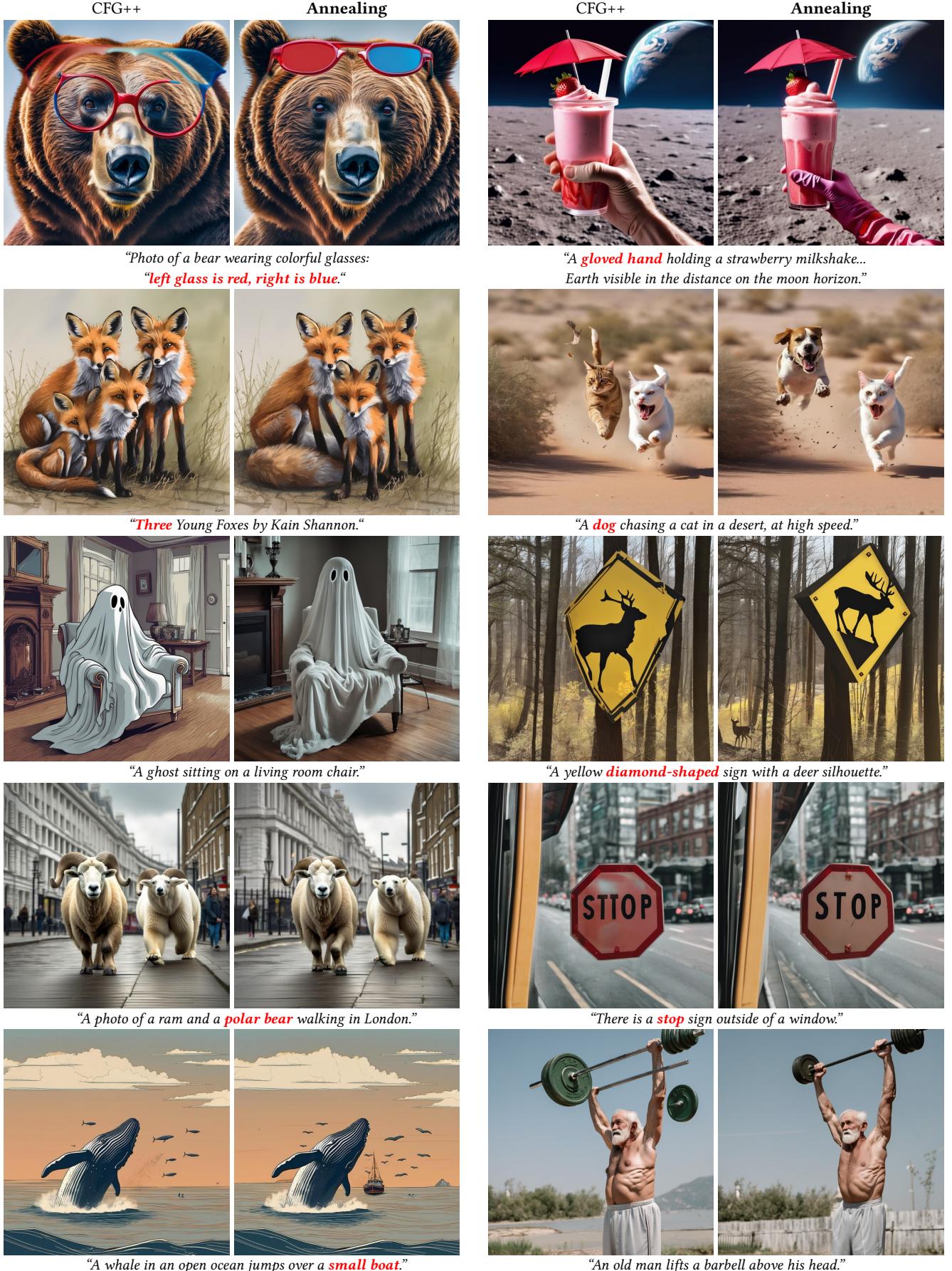
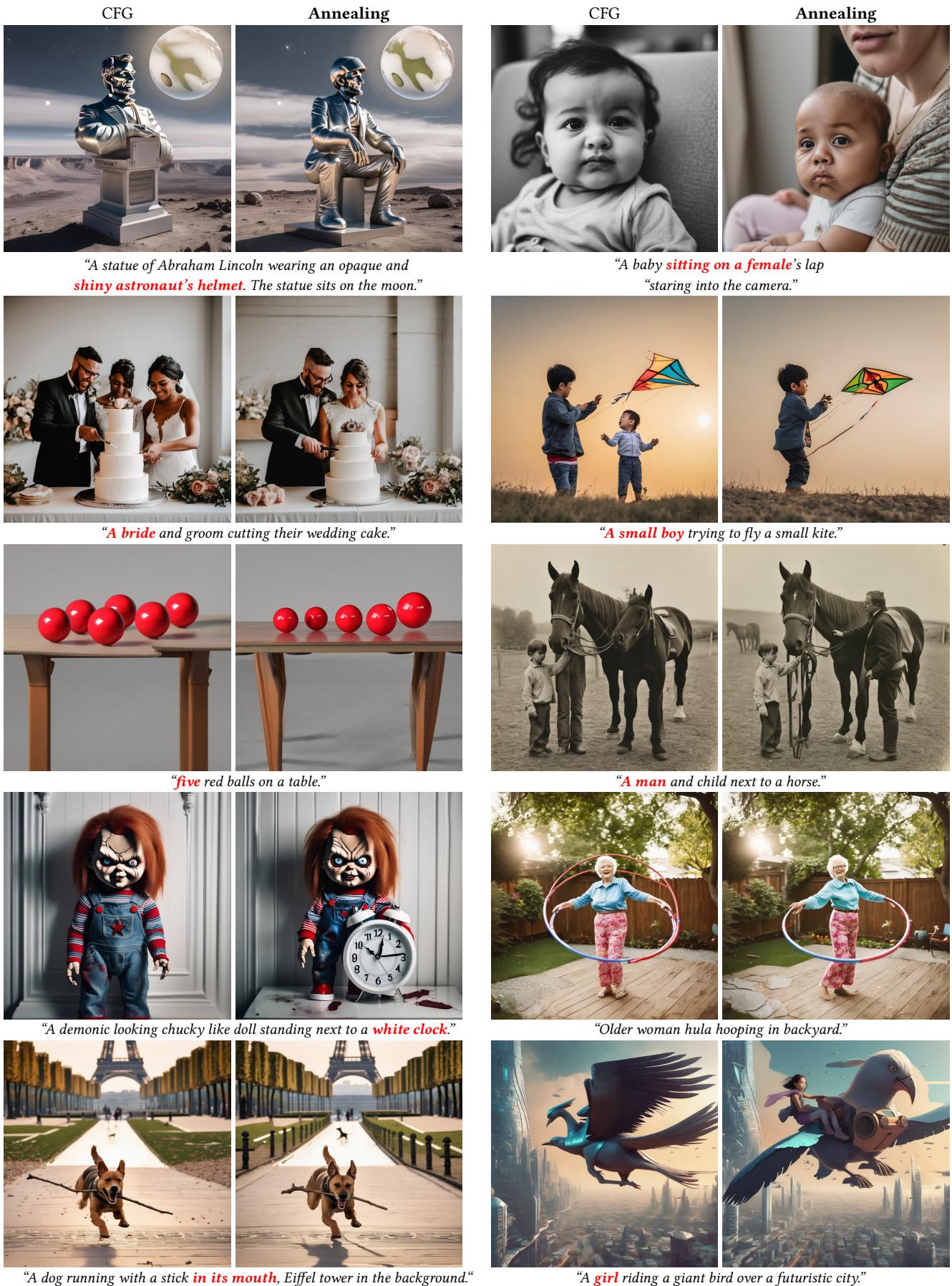


Fig. 9. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG++  $w = 0.8$  (left).

Fig. 10. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG  $w = 10$  (left).

## A Supplementary Material

### A.1 Implementation Details

We provide our annealing scheduler training algorithm in Alg.1, and its inference algorithm in Alg.4. Our annealing scheduler  $w_\theta(t, z_t, \lambda)$  is implemented as a lightweight MLP with three hidden layers of dimension 128, resulting in a total of 52K trainable parameters. The model takes as input sinusoidal embeddings of three features: (1) the normalized timestep  $t/T$ , (2) the normalized guidance magnitude  $\|\delta_t\|/\|\delta\|_{\max}$ , where  $\|\delta\|_{\max}$  corresponds to the typical maximum norm of  $\delta_t$  observed empirically across the training set and set to 5.0 in SDXL, and (3) the prompt-alignment parameter  $\lambda$ . Each embedding is 4-dimensional, and the three embeddings are concatenated before being passed through the first layer. ReLU activations are applied after each layer. The network outputs a single scalar corresponding to the predicted guidance scale. When we constrain the guidance scale  $w$  in the ablation to  $[0, 1]$  we add a sigmoid layer at the output. Training is performed for a maximum of 20,000 steps using the AdamW optimizer with a learning rate of  $1e-3$  and weight decay of 0.01. We train with a per-device batch size of 2 and accumulate gradients for 8 steps before performing an optimizer update. We use the default Kaiming-uniform initialization. All training runs complete within approximately 4.5 hours on a single NVIDIA A6000 GPU (48GB).

---

#### ALGORITHM 1: Annealing Scheduler - Training

---

**Require:**

- $w_\theta$ : trainable guidance scale model;
- $\epsilon_t^{(\cdot)}$ : frozen noise predictor, accepts  $\emptyset$  or a condition, at timestep  $t$ ;
- $T$ : total number of denoising steps

**repeat**

- // – Sample data and noise –
- Sample  $(z_0, c) \sim p(z_0, c)$ ,  $t \sim U[1, T]$ ,  $\epsilon \sim \mathcal{N}(0, I)$ ,  $\lambda \sim [0, 1]$ ;
- $z_t \leftarrow \text{AddNoise}(z_0, t, \epsilon)$ ;
- $\tilde{c} \leftarrow \text{Perturb}(c)$ ;
- // – Step at time  $t$  –
- $\delta_t \leftarrow \epsilon_t^{\tilde{c}}(z_t) - \epsilon_t^\emptyset(z_t)$ ;
- $\hat{\epsilon}_t \leftarrow \epsilon_t^\emptyset(z_t) + w_\theta(t, \delta_t, \lambda) \cdot (\epsilon_t^{\tilde{c}}(z_t) - \epsilon_t^\emptyset(z_t))$ ; // CFG
- $z_{0|t} \leftarrow (z_t - \sqrt{1 - \bar{\alpha}_t} \cdot \hat{\epsilon}_t) / \sqrt{\bar{\alpha}_t}$ ; // Denoise
- $z_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot z_{0|t} + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon_t^\emptyset(z_t)$ ; // Renoise
- // – Step at time  $t - 1$  –
- $\delta_{t-1} \leftarrow \epsilon_t^{\tilde{c}}(z_{t-1}) - \epsilon_t^\emptyset(z_{t-1})$ ;
- // – Compute loss and update –
- $\mathcal{L} \leftarrow \lambda \|\delta_{t-1}\|^2 + (1 - \lambda) \|\epsilon - \hat{\epsilon}_t\|^2$ ;
- Take gradient step on  $\nabla_\theta \mathcal{L}$ ; // Update scheduler

**until** converged;

---

### A.2 Training Data

We use the LAION-POP subset of LAION-5B dataset [Schuhmann et al. 2022] with high-resolution images with detailed descriptions, and selected 20,000 images based on the highest similarity scores.

---

#### ALGORITHM 2: CFG - Inference (DDIM)

---

**Require:**

- $T$ : total number of denoising steps;
- $w$ : guidance scale;
- $\epsilon_t^{(\cdot)}$ : frozen noise predictor, accepts  $\emptyset$  or a condition, at timestep  $t$ ;
- $c$ : condition;

$z_T \sim \mathcal{N}(0, I)$ ;

**for**  $t = T$  **to** 1 **do**

- $\hat{\epsilon}_t \leftarrow \epsilon_t^\emptyset(z_t) + w \cdot (\epsilon_t^c(z_t) - \epsilon_t^\emptyset(z_t))$ ; // CFG
- $z_{0|t} \leftarrow (z_t - \sqrt{1 - \bar{\alpha}_t} \cdot \hat{\epsilon}_t) / \sqrt{\bar{\alpha}_t}$ ; // Denoise
- $z_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot z_{0|t} + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \hat{\epsilon}_t$ ; // Renoise

**end**

**return**  $z_0$

---



---

#### ALGORITHM 3: CFG++ - Inference (DDIM)

---

**Require:**

- $T$ : total number of denoising steps;
- $w \in [0, 1]$ : guidance scale;
- $\epsilon_t^{(\cdot)}$ : frozen noise predictor, accepts  $\emptyset$  or a condition, at timestep  $t$ ;
- $c$ : condition;

$z_T \sim \mathcal{N}(0, I)$ ;

**for**  $t = T$  **to** 1 **do**

- $\hat{\epsilon}_t \leftarrow \epsilon_t^\emptyset(z_t) + w \cdot (\epsilon_t^c(z_t) - \epsilon_t^\emptyset(z_t))$ ; // CFG
- $z_{0|t} \leftarrow (z_t - \sqrt{1 - \bar{\alpha}_t} \cdot \hat{\epsilon}_t) / \sqrt{\bar{\alpha}_t}$ ; // Denoise
- $z_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot z_{0|t} + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon_t^\emptyset(z_t)$ ; // Renoise

**end**

**return**  $z_0$

---



---

#### ALGORITHM 4: Annealing Scheduler - Inference (DDIM)

---

**Require:**

- $\lambda \in [0, 1]$ : prompt alignment weighting parameter;
- $T$ : total number of denoising steps;
- $w_\theta$ : trained guidance scale model;
- $\epsilon_t^{(\cdot)}$ : frozen noise predictor, accepts  $\emptyset$  or a condition, at timestep  $t$ ;
- $c$ : condition;

$z_T \sim \mathcal{N}(0, I)$ ;

**for**  $t = T$  **to** 1 **do**

- $\delta_t \leftarrow \epsilon_t^{\tilde{c}}(z_t) - \epsilon_t^\emptyset(z_t)$ ;
- $\hat{\epsilon}_t \leftarrow \epsilon_t^\emptyset(z_t) + w_\theta(t, \|\delta_t\|, \lambda) \cdot (\epsilon_t^c(z_t) - \epsilon_t^\emptyset(z_t))$ ; // CFG
- $z_{0|t} \leftarrow (z_t - \sqrt{1 - \bar{\alpha}_t} \cdot \hat{\epsilon}_t) / \sqrt{\bar{\alpha}_t}$ ; // Denoise
- $z_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot z_{0|t} + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon_t^\emptyset(z_t)$ ; // Renoise

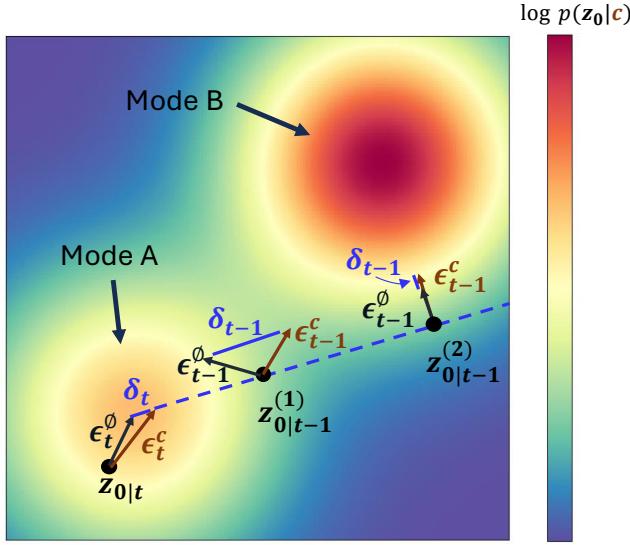
**end**

**return**  $z_0$

---

### A.3 Memory and Time Consumption

We evaluated the inference time of our lightweight model, which has a footprint of only 700KB. Running the model 10,000 times on a NVIDIA RTX A5000 yielded a mean inference time of 0.001434 seconds with a standard deviation of 0.000123 seconds. Given that the model is activated for 50 timesteps during a typical diffusion process, this results in an additional computational cost of approximately 0.0717 seconds per sample.



**Fig. 11. Intuition behind the  $\delta$ -loss.** A 2D illustration showing how the magnitude of  $\delta_t = \epsilon_t^c - \epsilon_t^\emptyset$  reflects alignment with the prompt. At time  $t$ , the sample  $z_{0|t}$  lies near mode A, which partially aligns with the prompt, resulting in a small  $\|\delta_t\|$ . As the denoising progresses, following the direction of  $\delta_t$  leads toward an augmented mode B that even better reflects the prompt semantics. Among the candidate points,  $z_{0|t-1}^{(2)}$  lies closest to mode B, where the conditional and unconditional predictions are best aligned, yielding a minimal  $\|\delta_{t-1}\|$ . The  $\delta$ -loss encourages such behavior.

#### A.4 Intuition for $\delta$ -loss

To provide further intuition into  $\|\delta_t\|$  as a navigational tool, we present a 2D illustration depicting two modes of the conditional distribution  $p(z_0 | c)$  in Figure 11.

The point  $z_{0|t}$  represents the estimated clean image at time  $t$ , and the vectors  $\epsilon_t^c$  and  $\epsilon_t^\emptyset$  denote the conditional and unconditional noise predictions, respectively (scaling factors omitted for clarity). Their difference,  $\delta_t = \epsilon_t^c - \epsilon_t^\emptyset$ , shown in blue, reflects the guidance direction.

At time  $t$ , the sample  $z_{0|t}$  is close to mode A, which corresponds to a high-quality image that partially matches the prompt  $c$ . As a result,  $\epsilon_t^c$  is only slightly biased toward mode B relative to  $\epsilon_t^\emptyset$ , leading to a small  $\|\delta_t\|$ .

From a navigation perspective, we aim to reach mode B, which even better aligns with the prompt. We consider candidate estimates along the blue dashed line.

The point  $z_{0|t}^{(1)}$  is the clean image estimate at the next step when a small guidance scale  $w$  is used. At this location, there is a larger gap between  $\epsilon_{t-1}^c$  and  $\epsilon_{t-1}^\emptyset$ , indicating misalignment.

In contrast, the point  $z_{0|t}^{(2)}$ , which lies near mode B, represents a more optimal solution. Here, both  $\epsilon_{t-1}^c$  and  $\epsilon_{t-1}^\emptyset$  are already aligned toward mode B, resulting in a minimal  $\|\delta_{t-1}\|$ .

Our  $\delta$ -loss leverages this geometric insight during training by encouraging smaller values of  $\|\delta_{t-1}\|$  through the adaptive selection of the guidance scale  $w$ .

Noise Scale $s$	FID $\downarrow$	CLIP $\uparrow$	ImageReward $\uparrow$
0	27.01	32.25	0.884
0.025	<b>26.40</b>	<b>32.29</b>	<b>0.898</b>
0.1	27.17	32.27	0.880
0.25	28.14	32.27	0.873

**Table 3. Ablation over noise scaling parameter  $s$  in the mode augmentation scheme.**

#### A.5 Prompt Perturbation

We perturb the conditioning signal solely during training, following CADS [Sadat et al. 2023]. In practice, we apply the noise directly to the prompt embedding  $c$ , using the corruption rule:

$$\hat{c} = \sqrt{\gamma(t)} c + s\sqrt{1 - \gamma(t)} n, \quad n \sim \mathcal{N}(0, I),$$

where  $\gamma(t)$  is a schedule and  $s$  controls the noise level. We adopt a linear schedule with  $\tau_1 = 0$ ,  $\tau_2 = T$ , such that  $\gamma(t)$  decays from 1 to 0 over the course of denoising, thus inducing higher corruption in earlier timesteps. To maintain the norm of the noised embedding, we rescale the signal as proposed in CADS:

$$\hat{c}_{\text{rescaled}} = \frac{\hat{c} - \text{mean}(\hat{c})}{\text{std}(\hat{c})} \text{std}(c) + \text{mean}(c), \quad \tilde{c} = \psi \hat{c}_{\text{rescaled}} + (1 - \psi) \hat{c},$$

and set the mixing factor to  $\psi = 1$ .

We set the noise scale  $s$  to 0.025. We ablate this scale by fixing  $\lambda = 0.8$  and reporting the performance of the trained scheduler in terms of FID, CLIP similarity, and ImageReward on the COCO2017 Validation set in Table 3.

#### A.6 Metrics Calculation

For assessment of fidelity and diversity, we report Precision and Recall [Kynkänniemi et al. 2019] in the DINOv2 [Oquab et al. 2024] feature space.

Precision measures the fraction of generated samples that lie within the support of the real image distribution. This is estimated by checking whether each generated sample has a real image among its  $k$  nearest neighbors in feature space. Conversely, recall quantifies the fraction of real images that lie within the support of the generated distribution, also based on their nearest neighbors among generated samples. Higher precision indicates better sample fidelity, while higher recall reflects greater diversity in generation. We used  $k = 5$  in our reports. Additionally, we report FD-DINOv2, a feature distance metric computed in the same feature space for image quality assessment.

To construct the generated dataset, we use the same captions as the COCO 2017 validation set. Each image in this set has five human-provided annotations; we consistently use the first caption per image for generation. We use a unique random seed for each image, setting it to the corresponding `image_id` from the COCO validation set.

Method	FID ↓	FD-DINOv2 ↓	CLIP ↑	Image Reward ↑	Precision ↑	Recall ↑
CFG ( $w = 7.5$ )	25.13	269.44	32.12	0.817	<b>0.863</b>	0.630
APG ( $w = 10$ )	25.25	268.00	32.08	<b>0.818</b>	0.862	<b>0.631</b>
CFG++ ( $w = 0.6$ )	24.97	267.91	32.12	0.808	0.859	0.629
Ours ( $\lambda = 0.05$ )	<b>24.76</b>	<b>267.17</b>	<b>32.16</b>	0.809	0.860	0.620
CFG ( $w = 10$ )	26.06	281.04	32.22	0.859	0.859	0.594
APG ( $w = 15$ )	26.60	282.09	32.19	<b>0.865</b>	<b>0.864</b>	0.592
CFG++ ( $w = 0.8$ )	25.61	279.69	32.20	0.857	0.855	0.601
Ours ( $\lambda = 0.4$ )	<b>25.35</b>	<b>279.30</b>	<b>32.25</b>	<b>0.865</b>	0.859	<b>0.606</b>
CFG ( $w = 12.5$ )	26.61	288.13	32.25	0.881	0.850	0.570
APG ( $w = 17.5$ )	26.58	286.67	32.21	<b>0.887</b>	<b>0.861</b>	0.586
CFG++ ( $w = 1$ )	26.33	288.55	<b>32.26</b>	0.882	0.848	0.570
Ours ( $\lambda = 0.7$ )	<b>25.95</b>	<b>285.52</b>	<b>32.26</b>	0.884	0.852	<b>0.594</b>
CFG ( $w = 15$ )	27.15	293.93	32.27	0.883	0.844	0.570
APG ( $w = 20$ )	26.85	290.93	32.23	0.893	<b>0.855</b>	0.577
CFG++ ( $w = 1.2$ )	26.84	294.22	32.28	0.894	0.847	0.551
Ours ( $\lambda = 0.8$ )	<b>26.40</b>	<b>290.33</b>	<b>32.29</b>	<b>0.898</b>	0.846	<b>0.586</b>

Table 4. Comparison of CFG, APG, CFG++, and our method across FID, FD-DINOv2, CLIP score, Image Reward (IR), Precision, and Recall. Arrows indicate whether higher ( $\uparrow$ ) or lower ( $\downarrow$ ) values are better.

## A.7 Implementation Details for Other Methods

For **CFG++**, we followed the official implementation\* and evaluated the method using  $\lambda$  values ranging from 0.4 to 1.2.

For **APG**, we adopted the settings provided in the original paper [Sadat et al. 2024]. Specifically, we used the recommended hyperparameters for SDXL:  $\eta = 0$ ,  $r = 15$ , and  $\beta = -0.5$ , and varied guidance scales  $w$  from 7.5 through 25.

## A.8 Other Solvers & Noise Schedules

To evaluate the robustness of our method across different samplers and noise schedules, we adopt the CFG++ renoising step generalized to both the Euler sampler [Karras et al. 2022b] and the Euler Ancestral sampler, following the formulations in CFG++ [Chung et al. 2024]. For each sampler, we fix the learned annealing scheduler and evaluate it with  $\lambda = 0.4$ , comparing against a CFG++ baseline using the same sampler with a fixed guidance weight  $w = 0.8$ . We additionally report results using DDIM for completeness, using a scaled\_linear beta schedule with  $\beta_{\text{start}} = 0.00085$  and  $\beta_{\text{end}} = 0.012$ , while Euler and Euler Ancestral use a linear schedule with  $\beta_{\text{start}} = 0.0001$  and  $\beta_{\text{end}} = 0.02$ .

We report FID, CLIP, and ImageReward in Table 5. Notably, our scheduler outperforms the CFG++ baseline in all metrics across different solvers.

## A.9 Extension to Flow Matching

Our scheduler can be naturally extended to flow-based models by leveraging the continuous-time formulation of Flow Matching [Lipman et al. 2022]. In this setting, we model the trajectory of samples

Method	FID ↓	CLIP ↑	IR ↑
DDIM (CFG++, w=0.8)	25.61	32.20	0.857
DDIM (Ours, $\lambda = 0.4$ )	<b>25.35</b>	<b>32.25</b>	<b>0.865</b>
Euler (CFG++, w=0.8)	26.17	32.23	0.867
Euler (Ours, $\lambda = 0.4$ )	<b>25.92</b>	<b>32.21</b>	<b>0.873</b>
Euler Ancestral (CFG++, w=0.8)	28.57	32.32	0.900
Euler Ancestral (Ours, $\lambda = 0.4$ )	<b>28.09</b>	<b>32.34</b>	<b>0.906</b>

Table 5. Comparison across solvers, CFG++ and ours. We consistently achieve better metrics in terms of FID, CLIP and IR.

$x(t)$  using a learned velocity field  $v_\theta(x, t, c)$ , governed by the ordinary differential equation:

$$\frac{dx}{dt} = v_\theta(x, t, c),$$

where  $c$  denotes the conditioning signal. The model is trained to match the true velocity  $x_1 - x_0$  by sampling intermediate points  $x(t) = x_0 + t(x_1 - x_0)$  and minimizing a velocity prediction loss, analogous to the diffusion-based  $\epsilon$ -prediction loss. Specifically, the equivalent of the  $\epsilon$ -loss becomes:

$$\mathcal{L}_\epsilon = \|v_\theta(x(t), t, c) - (x_1 - x_0)\|^2.$$

Furthermore, we define a  $\delta$ -loss similar to the diffusion model case. After an integration step to a future point  $x(t + \Delta t)$ , we compute the discrepancy between the conditional and unconditional velocity predictions:

$$\delta_{t+\Delta t} = v_\theta(x(t + \Delta t), t + \Delta t, c) - v_\theta(x(t + \Delta t), t + \Delta t, \emptyset),$$

and define the loss as:

$$\mathcal{L}_\delta = \|\delta_{t+\Delta t}\|^2.$$

\*<https://github.com/CFGpp-diffusion/CFGpp>

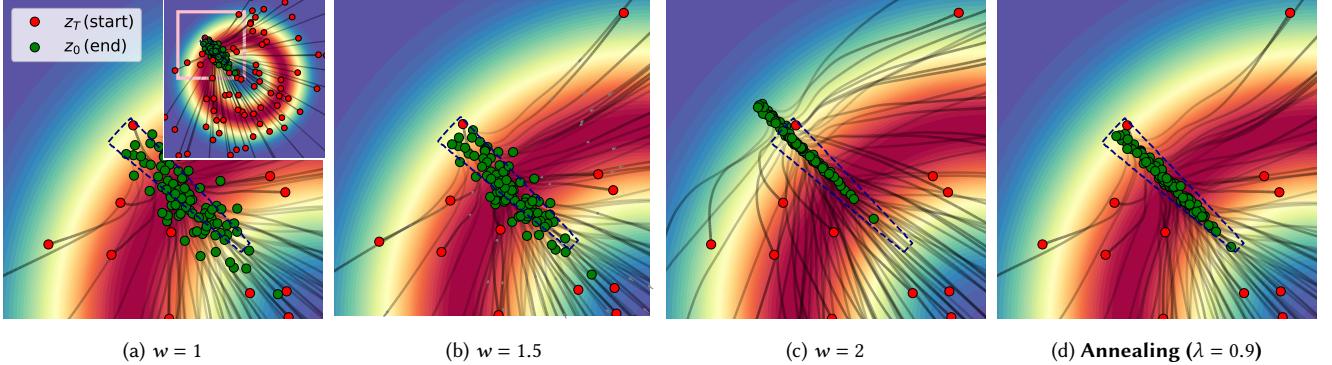


Fig. 12. A 2D flow matching toy example with a target distribution shaped as a wide ring. Random seeds conditioned on  $c = 3\pi/4$  are shown, along with their trajectories in gray. The dashed region indicates a tolerance band of  $\pm\pi/64$  around the target condition  $c$ , where the manifold density is high. (a)  $w = 1.0$ : Low guidance scale results in weak condition adherence. (b)  $w = 1.5$ : Moderate guidance slightly improves alignment, but some samples still deviate from the target region. (c)  $w = 2.0$ : Strong guidance leads to overfitting the condition, pulling samples off the true manifold. (d) Ours: The trained annealing scheduler achieves accurate condition alignment while preserving sample quality.

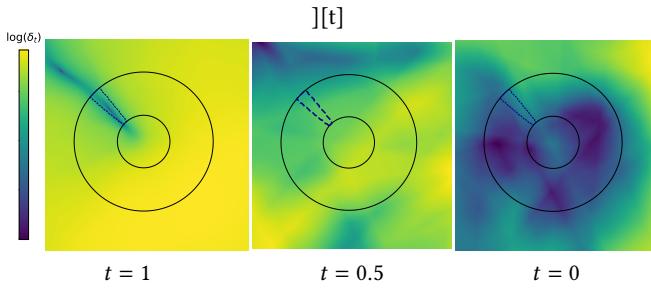


Fig. 13.  $\log \|\delta_t\|$  heatmap for  $c = 3\pi/4$ . This quantity illustrates how well unconditional and conditional predictions align in magnitude and direction across the latent space for varying timesteps. The region between the black circles marks areas of high sample density, while the blue dashed line represents the desired conditional region.  $t = 1$ : At large  $t$ 's, noise dominates the samples, with model predictions tending toward the center of the source distribution, albeit slightly noisy.  $t = 0.5$ : As  $t$  increases, unconditional and conditional predictions align more closely, showing lower values near the desired condition, but even lower values remain farther from the distribution in the direction of  $c$ .  $t = 1$ : By the final timestep, a local minimum is achieved at the target location within the ring.

Similarly to the toy example presented for diffusion models, we train a 2D toy flow matching model that predicts the conditional velocity field  $v_\theta(x, t, \{c, \emptyset\})$ , and afterwards train an annealing scheduler using the equivalent velocity matching objectives described above ( $\mathcal{L}_\delta$  and  $\mathcal{L}_e$ ). At inference time, as a baseline, we apply guidance by combining the conditional and unconditional velocity predictions using a scaled interpolation (namely the guidance scale  $w$ ), and finally present our scheduler guidance. For the target condition  $c = 3\pi/4$ , we visualize the resulting trajectories and sample alignments in Figure 12. The results show that our proposed annealing-based scheduler achieves better condition alignment and sample quality compared to constant guidance scales, confirming the effectiveness of our approach in the flow matching setting as well. To further analyze the behavior of the model, we visualize the quantity

$\|\delta_t\|$  across different timesteps. As shown in Figure 13, we observe a pattern similar to that in the diffusion model, highlighting the desirability of low  $\|\delta_t\|$  values, which indicate the desirability of better agreement between conditional and unconditional predictions.

#### A.10 Additional Results

We present additional qualitative comparisons to further highlight the differences between methods. Figures 14 and 15 show comparisons against CFG, while Figs. 16 and 17 present comparisons against CFG++.

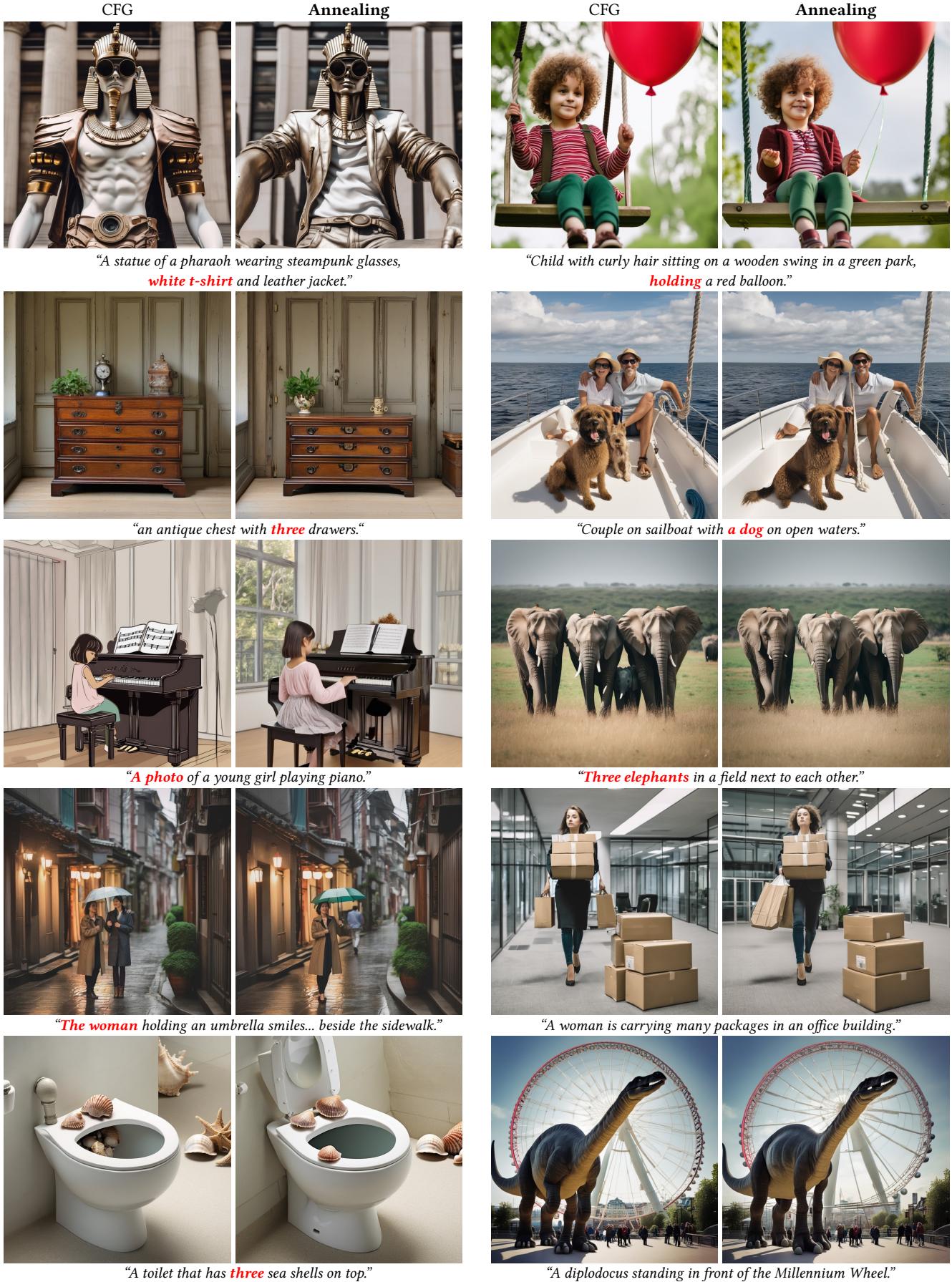
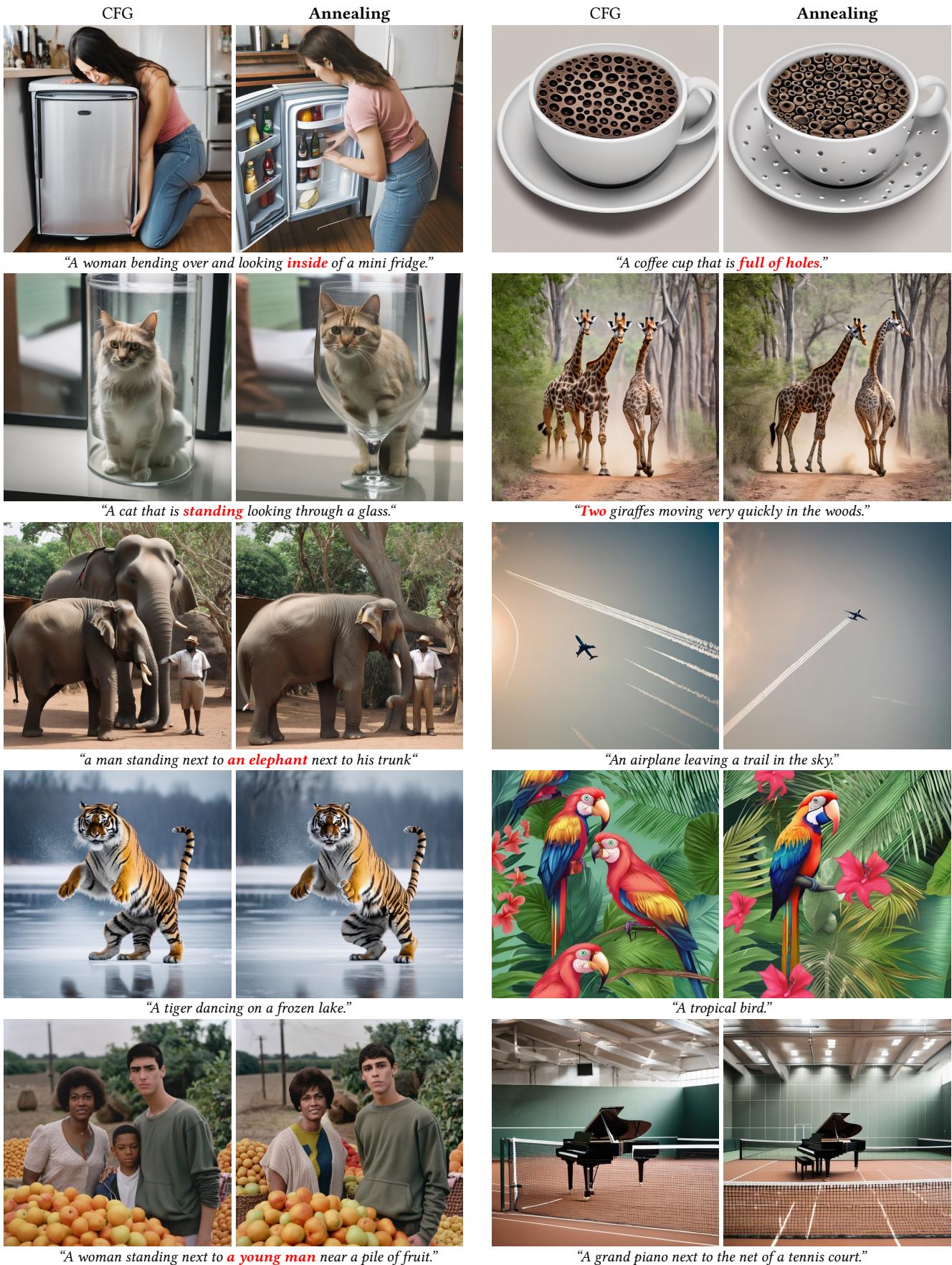


Fig. 14. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG  $w = 10$  (left).

Fig. 15. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG  $w = 10$  (left).

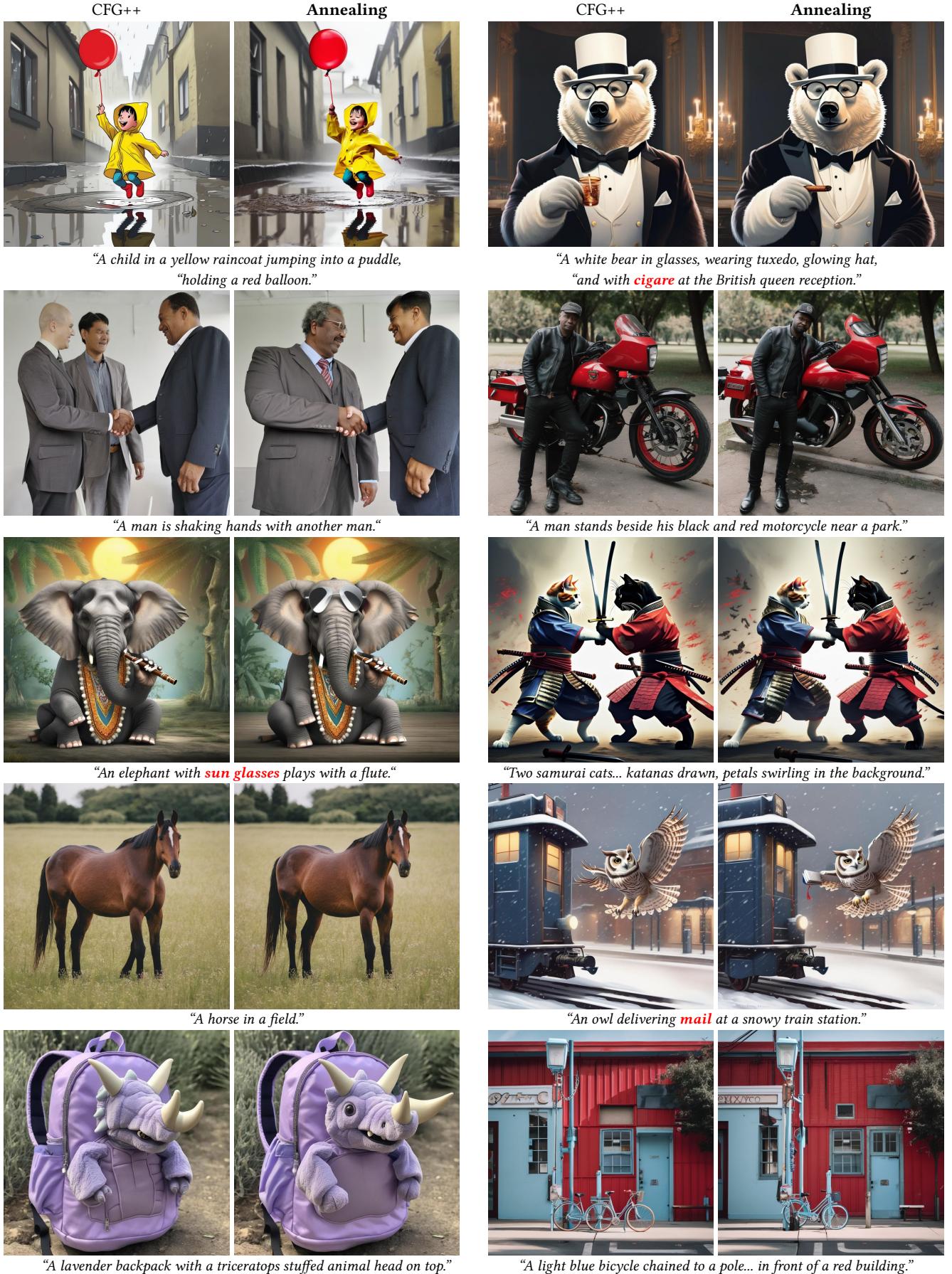
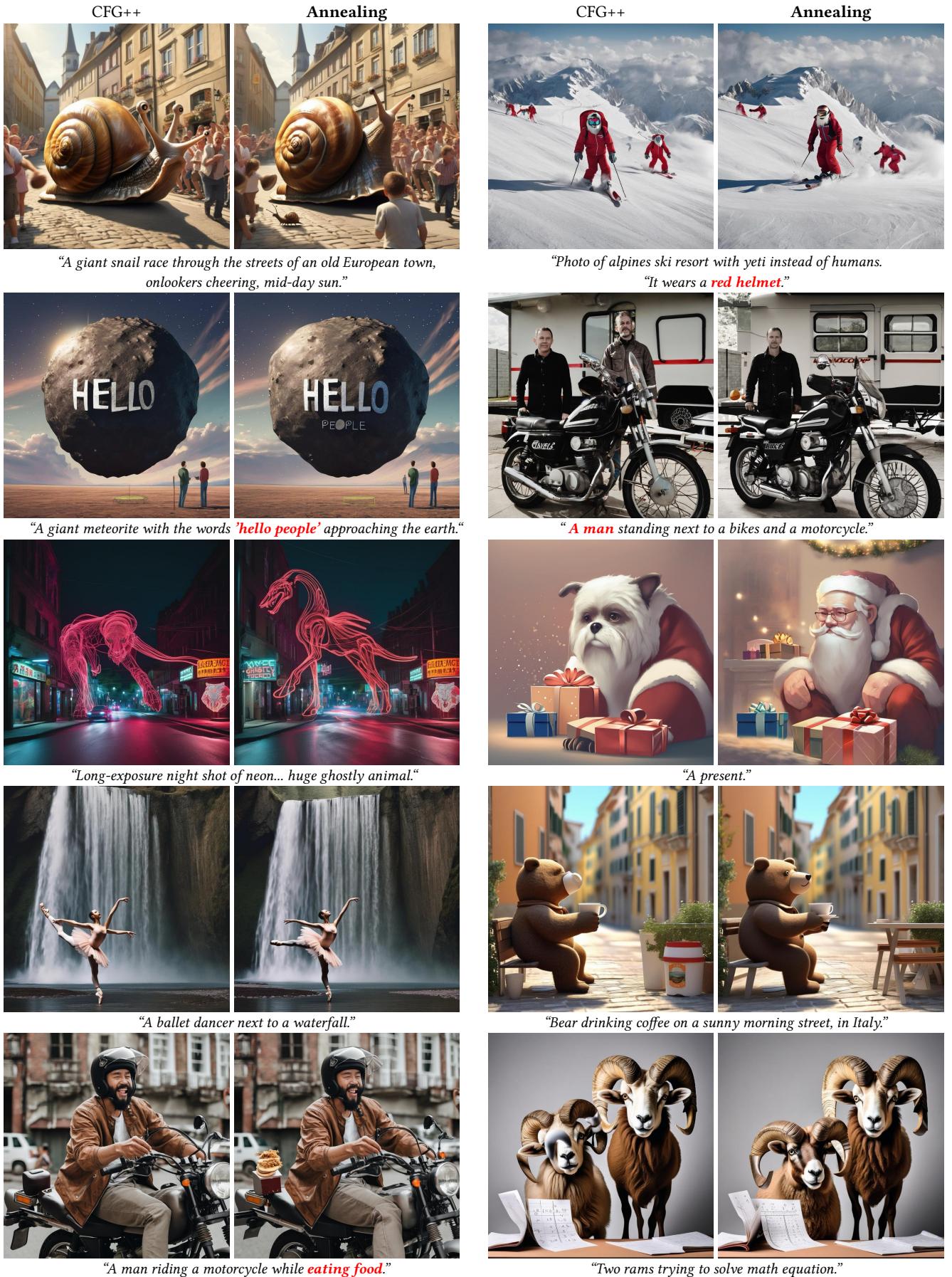


Fig. 16. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG++  $w = 0.8$  (left).

Fig. 17. Qualitative comparison of our Annealing method  $\lambda = 0.4$  (right) vs. CFG++  $w = 0.8$  (left).