ZIGZAG DIFFUSION SAMPLING: THE PATH TO SUCCESS IS ZIGZAG

Anonymous authors
Paper under double-blind review

Figure 1: The qualitative results of Z-Sampling demonstrate the effectiveness of our method in various aspects, such as style, position, color, counting, text rendering, and object co-occurrence. We present more cases in Appendix D.2.

ABSTRACT

Diffusion models, the most popular generative paradigm so far, can inject conditional information into the generation path to guide the latent towards desired directions. However, existing text-to-image diffusion models often fail to maintain high image quality and high prompt-image alignment for those challenging prompts. To mitigate this issue and enhance existing pretrained diffusion models, we mainly made three contributions in this paper. First, we theoretically and empirically demonstrate that the conditional guidance gap between the denoising and inversion processes captures prompt-related semantic information. Second, motivated by theoretical analysis, we derive Zigzag Diffusion Sampling (Z-Sampling), a novel sampling method that leverages the guidance gap to accumulate semantic information step-by-step throughout the entire generation process, leading to improved sampling results. Moreover, as a plug-and-play method, Z-Sampling can be generally applied to various diffusion models (e.g., accelerated ones and Transformer-based ones) with very limited coding costs. Third, extensive experiments demonstrate that Z-Sampling can generally and significantly enhance generation quality across various benchmark datasets, diffusion models, and performance evaluation metrics. Particularly, Z-Sampling is good at handling those challenging fine-grained prompts, such as style, position, counting, and multiple objects, due to its guidance-gap-based information gain. Moreover, Z-Sampling can even further enhance existing diffusion models combined with other orthogonal methods, including Diffusion-DPO.

1 INTRODUCTION

Diffusion models, known for its powerful generative capabilities and diversity, have become a mainstream generation paradigm in images (Podell et al., 2023; Lin et al., 2024b), videos (Ho et al., 2022; Blattmann et al., 2023), and 3D objects (Luo & Hu, 2021; Voleti et al., 2024) and beyond. One key ability of diffusion model is to guide the sampling path based on additional conditions (e.g., text prompts), leading to conditional or controllable generation (Ho & Salimans, 2022).

However, while strong guidance may improve semantic alignment to those challenging prompts, it often causes significant decline in image fidelity, leading to mode collapse, and resulting inevitable accumulation of errors during the sampling process (Chung et al., 2024). To mitigate this issue, some studies apply additional manifold constraints to the sampling paths (Chung et al., 2024; Yang et al.; He et al.), which compromises the diversity of generated outputs. Others design varying guidance scales across different denoising regions to mitigate this issue (Shen et al., 2024), but such explicit strategies often lead to unnatural outputs. Thus, enhancing high generation quality while maintaining prompt alignment effectively during sampling remains a crucial challenge, especially for those challenging prompts. This challenge may require more controllable prompt guidance beyond classical guidance like classifer-free guidance (Ho & Salimans, 2022).

Fortunately, we discover that semantic information may be inherently embedded in the random latent space, influencing the quality of image generation (Xu et al., 2024b; Po-Yuan et al., 2023; Mao et al., 2023b; Wu et al., 2023c). In Figure 2, we demonstrate the following phenomenon: if a latent can generate images aligned with a specific concept c under no conditional prompt, it will generate high-quality results with c as the conditional prompt. This implies that the latent naturally carries relevant semantic information and can align with relevant semantic prompts very well. Figure 3 intuitively illustrates that the green initial point with certain semantic information is usually superior to the red initial point for the prompts associated with the semantic information.

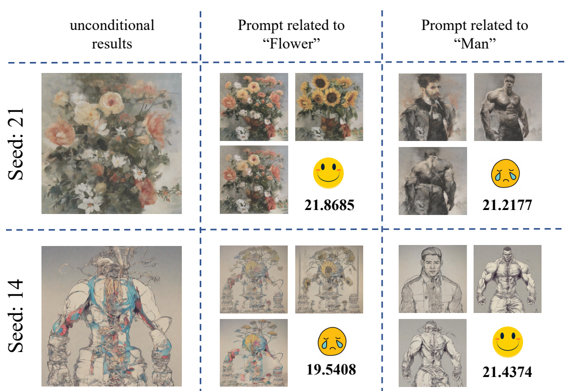

Figure 2: Semantic-rich latents effectively generate images aligned with intended semantics. For instance, the random latent (seed 21) is better suited for generating images related to the concept of “flowers”. We present more cases in Appendix C.1.

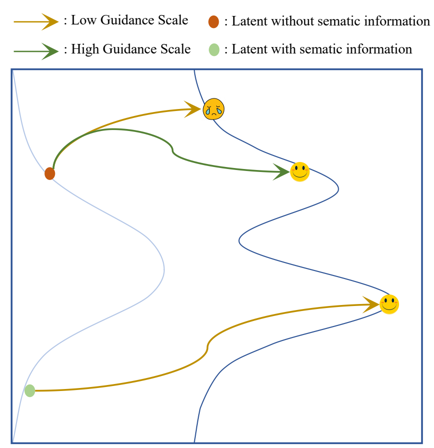

Figure 3: If the latent carries semantic information, we can obtain prompt-related results from this latent even without conditional guidance.

Is it possible to leverage this insight for improved sampling methods? Fortunately, we discover that employing strong guidance during denoising process and employing weak guidance during inversion process establishes a guidance gap that can inject prompt semantic information to the latent. Accumulating or enlarging this guidance gap allows the latent to encode more semantic information, aligning more closely with the properties of the green point in Figure 3. We present more examples and discussion in Appendix C.2.

Just as “the path to success is zigzag”, past experience during zigzag processes can teach people to learn and succeed. Inspired by the wisdom, we let a latent denoise in a zigzag manner, namely a denoising step and a inversion step, step-by-step along the sampling path, which can accumulate semantic information as “past experience”. As Figure 4 illustrates, we propose Zigzag Diffusion Sampling, or Z-Sampling, which can capture semantic information with such repeated zigzag steps and move to more desirable results along the sampling path. Through each zigzag step, the latent accumulates additional semantic information.

The contributions of this work can be summarized as follows.

First, we theoretically and empirically demonstrate that the guidance gap between denoising and inversion processes can capture the semantic information embedded in the latent space, which matters to generation quality and prompt-image alignment.

Second, motivated by the theoretical results, we derive Z-Sampling, a novel sampling method that can leverage the guidance gap to accumulate semantic information through each zigzag step and generate more desirable results. It allows flexible control over the injection of semantic information and is applicable across various diffusion architectures with very limited coding costs. To the best of our knowledge, Z-Sampling is the first method that successfully improve generation via leveraging semantic information from the guidance gap.

Third, extensive experiments demonstrate the effectiveness and generalization of Z-Sampling across various benchmark datasets, diffusion models, and evaluation metrics. As theoretical analysis suggests, Z-Sampling especially excels in challenging complex or fine-grained prompts, such as position, counting, color-attribution, and multi-object, breaking through the performance limit of pretrained diffusion models. Moreover, orthogonal methods, such as Diffusion-DPO (Wallace et al., 2024), can be further enhanced by Z-Sampling. Importantly, as a training-free method, Z-Sampling can still exhibits significant improvements over the baselines with limited computational cost, which suggests its efficiency and practical value. In the efficiency study, even with 36% less computational time, Z-Sampling can reach the best performance of standard sampling.

Algorithm 1 Z-Sampling

Input: Denoising at timestep t: Φ^t, Inversion at timestep t: Ψ^t, text prompt: c, denoising guidance: γ_1, inversion guidance: γ_2, inference steps: T, zigzag optimization steps: λ
Output: Clean image x_0
Sample Gaussian noise x_T
for $t = T$ **to** 1 **do**
    $x_{t-1} = \Phi^t(x_t|c, \gamma_1)$
    if $t > T - \lambda$ **then**
        #Inversion by equation 4
        $\tilde{x}_t = \Psi^t(x_{t-1}|c, \gamma_2)$
        #Denoising by equation 2
        $x_{t-1} = \Phi^t(\tilde{x}_t|c, \gamma_1)$
    end if
end for
return x_0

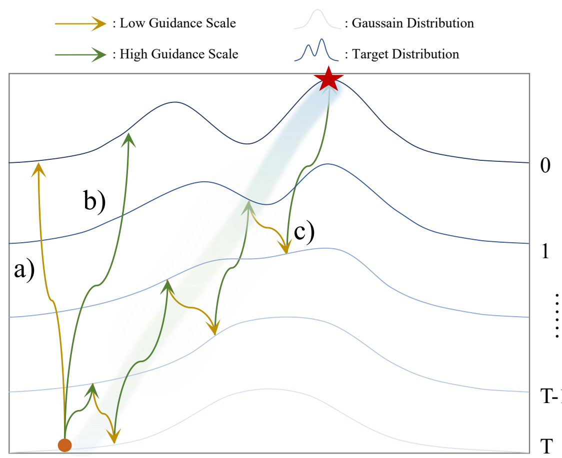

Figure 4: The illustration of our method. (a) weak guidance sampling. (b) strong guidance sampling. (c) Z-Sampling.

2 PRELIMINARIES

In this section, we formally introduce prerequisites and background.

Diffusion Model. We define the total numeber of denoising steps T and conditional prompt c. Given the denoising procss $\Phi : \mathcal{N} \times \mathcal{C} \to \mathcal{D}$ and guidance scale γ_1, starting from $x_T \in \mathcal{N}$, we can generate $x_0 = \Phi(x_T|c, \gamma_1) \in \mathcal{D}$, where $\mathcal{N}$ represents the distribution of Gaussian and $\mathcal{D}$ represents the distribution of target data. We note that the mapping function Φ corresponds to the probability $P(x_0|c, \gamma_1, x_{1:T})$. For simplicity, we simplify only the initial input x_T in Φ. Similarly, we can also reverse this process, given the inversion process $\Psi : \mathcal{D} \times \mathcal{C} \to \mathcal{N}$ under guidance scale γ_2, we obtain inverted data $\tilde{x}_T = \Psi(\tilde{x}_0|c, \gamma_2) \in \mathcal{N}$ from $\tilde{x}_0 \in \mathcal{D}$.

Following Ho et al. (2020), we treat diffusion model as a Monte Carlo process and decompose Φ into T times single-step denoising mappings as

$$\Phi(x_T|c, \gamma_1) = \underbrace{\Phi^T(x_T|c, \gamma_1) \circ \Phi^{T-1}(x_{T-1}|c, \gamma_1) \circ \cdots \circ \Phi^2(x_2|c, \gamma_1) \circ \Phi^1(x_1|c, \gamma_1)}_{T \times \text{Times}}. \quad (1)$$

And we define Φ^t as

$$x_{t-1} = \Phi^t(x_t|c,\gamma) = \sqrt{\alpha_{t-1}}\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^t(x_t)}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\epsilon_\theta^t(x_t), \tag{2}$$

where $a_t := \prod_{i=1}^{t}(1-\beta_i)$ and β_t are the pre-defined parameters for scheduling the scales of adding noises in DDIM scheduler (Song et al., 2020). we denote ϵ_θ^t as the predicted score by the denoising network θ at timestep t, with further details provided in the next paragraph.

Similarly, for the inversion process Ψ, we can also perform this decomposition as

$$\Psi(\tilde{x}_0|c,\gamma_2) = \underbrace{\Psi^1(\tilde{x}_0|c,\gamma_2) \circ \Psi^2(\tilde{x}_1|c,\gamma_2) \circ \cdots \circ \Psi^{T-1}(\tilde{x}_{T-2}|c,\gamma_2) \circ \Psi^T(\tilde{x}_{T-1}|c,\gamma_2)}_{T\times\text{Times}}, \tag{3}$$

where we obtain $\tilde{x}_{t-1}$ via Ψ^t as

$$\tilde{x}_t = \Psi^t(\tilde{x}_{t-1}|c,\gamma_2) = \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\tilde{x}_{t-1} + \sqrt{\alpha_t}\left(\sqrt{\frac{1}{\alpha_t}-1} - \sqrt{\frac{1}{\alpha_{t-1}}-1}\right)\epsilon_\theta^t(\tilde{x}_{t-1}). \tag{4}$$

In equation 4 we approximate the score predicted at timestep t with timestep $t-1$ along the inversion path, i.e, set $\epsilon_\theta^t(\tilde{x}_{t-1}) \approx \epsilon_\theta^t(\tilde{x}_t)$. If this approximation error is negligible, Φ and Ψ can be proven to be inverse functions (Mokady et al., 2023), meaning that $\Psi = \Phi^{-1}$.

Classifier free guidance. Controllable generation typically involves guiding or constraining the semantic representation. In classifier free guidance (Ho & Salimans, 2022), a score prediction network u_θ is trained both conditionally and unconditionally. During inference, denoising scores are computed by interpolating between conditional and unconditional scores predicted by u_θ, thus enabling the adjustment of guidance scale across various levels.

Specifically, for denoising and inversion process, we use guidance scales γ_1 and γ_2, with the corresponding scores as

$$\begin{aligned} \epsilon_\theta^t(x_t) &= (1+\gamma_1)u_\theta(x_t,c,t) - \gamma_1 u_\theta(x_t,\varnothing,t), \\ \epsilon_\theta^t(\tilde{x}_t) &= (1+\gamma_2)u_\theta(\tilde{x}_t,c,t) - \gamma_2 u_\theta(\tilde{x}_t,\varnothing,t), \end{aligned} \tag{5}$$

where u_θ is the noise predictor, and $\varnothing$ is the null prompt, representing the denoising result under unconditional settings.

3 METHODOLOGY

In this section, we discuss how to encode semantic information into latents through the guidance gap and derive Z-Sampling according to theoretical analysis.

3.1 LATENTS WITH RELEVANT SEMANTIC INFORMATION

Our inspiration stems from the question: what makes a good latent in the diffusion process? As Figure 3 illustrates, we argue that a latent with relevant semantic information (green point) can align with the prompt under weak or sometimes even negative conditional guidance. In contrast, a latent lacking semantic information (red point) necessitates strong conditional guidance to attain comparable alignment and may remain unaligned under unconditional generation.

To verify this, we generate images using different latents (seeds) under unconditional settings, shown in Figure 2. We observe that if a latent can generate a image of a certain concept c unconditionally, then, under certain prompt guidance, this latent usually performs higher in generating images related to c compared to other latents. For example, in Figure 2, if the latent (seed 21) generates the images of flowers unconditionally, it yields higher-quality images when used with flower-related prompts in conditional generation. Previous studies also argued that the properties of latents partially predetermine image composition or contents during generation, affecting object position, size, and depth (Wu et al., 2023c; Guttenberg, 2023; Lin et al., 2024a; Xu et al., 2024b; Mao et al., 2023b). However, they did not formally explore how to encode semantic information into the latents.

3.2 Capture semantic information from the guidance gap

Considering a denoising process $\Phi : \mathcal{N} \times \mathcal{C} \to \mathcal{D}$, under text condition $c \in \mathcal{C}$, we sample a initial latent $x_T \in \mathcal{N}$, and obtain the generated data x_0 as

$$x_0 = \Phi(x_T|c, \gamma_1), \tag{6}$$

where γ_1 is condition guidance scale during denoising. Now, we further perform inversion operation on x_0 under the guidance scale of γ_2 as

$$\tilde{x}_T = \Psi(x_0|c, \gamma_2). \tag{7}$$

If the approximation error in the inversion process is negligible, meaning $\Psi^{-1} = \Phi$, then equation 7 can be equivalently inverted as

$$x_0 = \Psi^{-1}(\tilde{x}_T|c, \gamma_2) = \Phi(\tilde{x}_T|c, \gamma_2). \tag{8}$$

Generally, the denoising guidance scale γ_1 is set to a common value (e.g., $\gamma_1 = 5.5$) to maintain standard generation and alignment to the prompt (Ho & Salimans, 2022). Conversely, the inversion guidance scale γ_2 is usually set to a small value (e.g., $\gamma_2 = 0$) to achieve inversion with weak guidance (Mokady et al., 2023). By comparing equation 6 and equation 8, we note that starting from $\tilde{x}_T$, we can generate x_0 under weak or even unconditional guidance scale $\gamma_2 = 0$. In contrast, starting from x_T requires strong conditional guidance scale $\gamma_1 = 5.5$ to produce similar results.

According to the insight discussed in Section 3.1, if a initial latent can generate results related to prompt c under weak guidance, it indicates this latent contains more semantic information related to c. Since guidance scale γ_2 is less than γ_1, we argue that the corresponding inverted latent $\tilde{x}_T$ contains more semantic information compared to x_T. We present more empirical evidence in Appendix C.2,

3.3 Zigzag Diffusion Sampling

Now we know that the guidance gap can capture additional semantic information. The next question is how to effectively leverage this property to inject semantic information into the sampling process.

Vanilla Inversion A vanilla way is to use the inverted latent $\tilde{x}_T$ in place of x_T as the starting point to generate semantically aligned results in the denoising process (see Algorithm 2). We provide Theorem 1 and show that the difference between the original x_T and the inverted $\tilde{x}_T$, namely $\delta_{end2end} = (x_T - \tilde{x}_T)^2$, may reveal how significant the vanilla end-to-end information injection is. An illustrative diagram of the latents' difference is provided in Figure 26 (a) of Appendix F.

Theorem 1 (See the proof in Appendix F.1) *For a random latent $x_T \in \mathcal{N}$ and an inverted latent $\tilde{x}_T$ given by equation 7, the latent difference $\delta_{end2end}$ between x_T and $\tilde{x}_T$ is*

$$\delta_{end2end} = (x_T - \tilde{x}_T)^2 = \alpha_T\Big(\sum_{t=1}^{T} h_t\big(\underbrace{\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_t)}_{\tau_1(t):\textit{semantic information gain term}} + \underbrace{\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1})}_{\tau_2(t):\textit{approx error term}}\big)\Big)^2, \tag{9}$$

where $h_t = \sqrt{1/\alpha_t - 1} - \sqrt{1/\alpha_{t-1} - 1}$, and $\epsilon_\theta^t(\cdot)$ is the predicted score given by equation 5.

Here, $\tau_1(t)$ represents the semantic information gain induced by the guidance gap at timestep t, whereas $\tau_2(t)$ represents the approximation error inherent in the inversion process, which may be neglected for semantic information. We note that in equation 9, the end-to-end aggregation may let the sum of the semantic information τ_1 over each step be small and fail to accumulate the desired semantic information gain step-by-step.

Z-Sampling To let τ_1 of each step be accumulated step-by-step instead of being canceled out in the vanilla sum, we decompose Φ into $\{\Phi^1, \Phi^2, \cdots, \Phi^T\}$, as defined in equation 1. We first denoise x_t to obtain $x_{t-1} = \Phi^t(x_t|c, \gamma_1)$ and then we invert x_{t-1} to get $\tilde{x}_t = \Psi^t(x_{t-1}|c, \gamma_2)$ for each timestep $t \in [T, 1]$. The proposed Z-Sampling method is presented in Algorithm 1 and illustrated in Figure 4. Note that Z-Sampling injects semantic information by replacing x_t with $\tilde{x}_t$ at each timestep. We prove Theorem 2 and demonstrate the cumulative latent difference $\delta_{Z-Sampling} = \sum_{t=1}^{T}(x_t - \tilde{x}_t)^2$, depicted in Figure 26 (b) of Appendix F.

Figure 5: The cross-attention map highlights the interaction between the entity token (red color) and latent variables. Z-Sampling optimizes the latent so that it is more suitable for generating concepts in the related-prompt. For example, in the zigzag path of the second column, semantically injected latents exhibit sharper attention on "dog" with relatively clear boundaries.

Theorem 2 (See the proof in Appendix F.2) *Suppose x_t is the denoised latent at step t, and $\tilde{x}_t$ be the corresponding inverted latent given by equation 4. Then the cumulative latent difference in Z-Sampling can be written as*

$$\delta_{\textit{Z-Sampling}} = \sum_{t=1}^{T}(x_t - \tilde{x}_t)^2 = \sum_{t=1}^{T}\alpha_t h_t^2(\underbrace{\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_t)}_{\tau_1(t):\textit{semantic information gain term}} + \underbrace{\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1})}_{\tau_2(t):\textit{approx error term}})^2, \quad (10)$$

where h_t and $\epsilon_\theta^t(\cdot)$ are consistent with Theorem 1.

Again, focusing on the semantic information gain term, we report that $\delta_{end2end} \propto (\sum_1^T \tau_1(t))^2$ holds for vanilla inversion and $\delta_{Z-Sampling} \propto \sum_1^T (\tau_1(t))^2$ holds for Z-Sampling. Given the Jensen's inequality, we have $\sum_1^T (\tau_1(t))^2 \geq (\sum_1^T \tau_1(t))^2$, showing that the cumulative semantic information gain $\delta_{\text{Z-Sampling}}$ is larger than the end-to-end semantic information gain δ_{end2end}. The semantic information gain induced by the guidance gap in Z-Sampling can be effectively accumulated, solving the previous issue of the semantic information gain cancellation.

We further prove Theorem 3 and show the significant impact of the guidance gap δ_γ on $\delta_{Z-Sampling}$.

Theorem 3 (See the proof in Appendix F.3) *Under the conditions of Theorem 2, the cumulative semantic information gain in Z-Sampling can be written as*

$$\delta_{\textit{Z-Sampling}} = \sum_{t=1}^{T}\alpha_t h_t^2\left(\delta_\gamma\left(u_\theta(x_t, c, t) - u_\theta(x_t, \varnothing, t)\right)\right)^2, \quad (11)$$

where the guidance gap is defined as $\delta_\gamma = \gamma_1 - \gamma_2$.

We note that the larger the δ_γ, the more pronounced the effect of Z-Sampling. When $\delta_\gamma = 0$, it is approximately equivalent to standard sampling. This is also empirically verified in Figure 8.

In Figure 5, we visualize the cross-attention map of Z-Sampling during the early stages (i.e, $t/T = 49/50$) of the generation process. And we observe that Z-Sampling indeed makes the attention regions corresponding to entity tokens more semantically focused, further illustrating the effectiveness of Z-Sampling on the semantic information gain. Mao et al. (2023b) reported that certain regions in random latents can induce objects representing specific concepts, which aligns with our observation that Z-Sampling enhances the association of certain regions with the prompt. Additionally, we discuss the impact of the approximation error τ_2 in Appendix E.2 and E.3.

4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our method, and perform robustness analysis for a more detailed investigation.

4.1 EXPERIMENTS SETTING

Datasets Pick-a-Pic (Kirstain et al., 2023), DrawBench dataset (Saharia et al., 2022), and GenEval (Ghosh et al., 2024). We leave more details in Appendix A.1.

Metrics We use multiple evaluation metrics, including HPS v2 (Wu et al., 2023c), PickScore (Kirstain et al., 2023), and ImageReward (Xu et al., 2024a). They are trained on large-scale human preference datasets, providing a reliable indication of genuine human preferences. Furthermore, we also employ the traditional metric AES (Schuhmann et al., 2022), which purely evaluate image quality. More details are found in Appendix A.2.

Diffusion Models We use various diffusion models as the generation backbone in main experiments. For SD2.1 (Rombach et al., 2022), SDXL (Podell et al., 2023), and Hunyuan-DiT (Li et al., 2024), we perform 50 denoising steps. For DreamShaper-xl-v2-turbo, which achieves efficient and high-quality generation by fine-tuning SDXL Turbo (Sauer et al., 2023), we set denoising step T only to 4. And we set $\gamma_1 = 5.5$ in SDXL/SD2.1, $\gamma_1 = 6.0$ in Hunyuan-DiT, and $\gamma_1 = 3.5$ in DreamShaper-xl-v2-turbo, all to the recommended default values. For all diffusion models, we set the zigzag operation to be executed throughout the entire path ($\lambda = T - 1$) and inversion guidance scale γ_2 as zero.

Baselines We validate the effectiveness of Z-Sampling and compare it against the following baseline: **(a) standard sampling**, we use the Multistep DPM Solver (Lu et al., 2022) for DreamShaper-xl-v2-turbo and Hunyuan-DiT, and DDIM (Song et al., 2020) for the SD-2.1 and SDXL. **(b) Resampling** (lug, 2022), repeatedly performs denoising at the same timestep by adding random noise to maintain the latent on the data manifold. Moreover, due to the page limit, we discuss related works and how they differs from Z-Sampling in Appendix A.3.

4.2 MAIN EXPERIMENTS

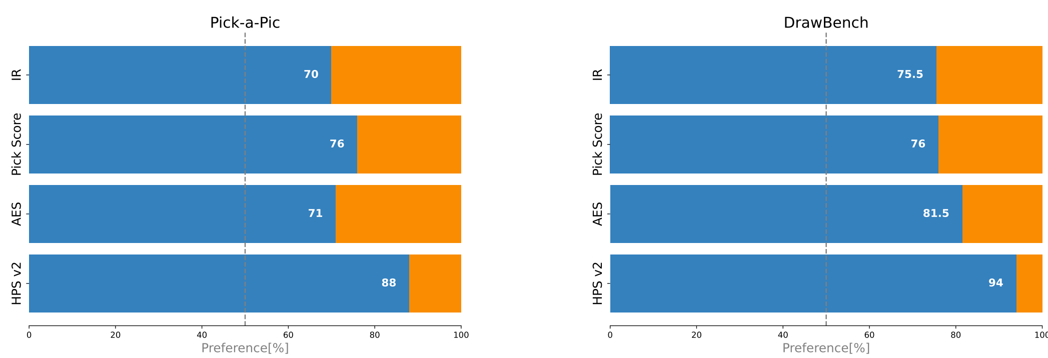

Figure 6: The winning rates of Z-Sampling over standard sampling. The blue bars represent the side of our method. The orange bars represent the side of the standard sampling. Model: DreamShaper-xl-v2-turbo. We present more results in Appendix D.3

In Table 1, we evaluate our method against standard sampling and Resampling across various diffusion architectures, including U-Net, DiT, and distillation architectures. Z-Sampling achieves top performance across nearly all metrics and Figure 6 shows the winning rates across these two benchmarks, exceeding 80% on HPS v2. Furthermore, for a more detailed comparison, we present results on GenEval (Ghosh et al., 2024), which serves as a challenging benchmark. As Table 2 show, Z-Sampling significantly enhances alignment in aspects such as counting, two-object relations, and color attribution, further demonstrating the effectiveness of our method.

We also compare our method with a recent sampling technique designed to enhance semantic injection. Shen et al. (2024) proposed Semantic-aware CFG, dividing the latent into independent semantic regions at each denoising step and adaptively adjusting their guidance, thereby unifying

Table 1: The quantitative results of Z-Sampling on Pick-a-Pic and DrawBench.

Method		Pick-a-Pic					DrawBench				
		HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Average ↑	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Average ↑
SD-2.1	Standard	0.2305	5.2778	19.0793	-0.4366	6.0376	0.2390	5.2006	20.4970	-0.4434	6.3733
	Resampling	0.2446	5.4620	19.5135	**-0.1807**	6.2598	0.2394	5.0838	20.4031	-0.3090	6.3543
	Z-Sampling(ours)	**0.2453**	**5.4704**	**19.5144**	-0.1862	**6.2609**	**0.2467**	**5.2891**	**20.8238**	**-0.2361**	**6.6814**
SDXL	Standard	0.2989	6.0870	21.6353	0.5865	7.1520	0.2881	5.5595	22.3086	0.6075	7.1909
	Resampling	0.3054	6.0395	21.7256	0.7860	7.2141	0.2962	5.5797	**22.5178**	0.7269	7.2802
	Z-Sampling(ours)	**0.3128**	**6.1302**	**21.8477**	**0.7922**	**7.2682**	**0.3050**	**5.6739**	22.4581	**0.7997**	**7.3092**
DreamShaper -xl-v2-turbo	Standard	0.3004	5.9355	21.5899	0.6618	7.1219	0.2685	5.2846	21.7861	0.4022	6.9354
	Resampling	0.3142	6.0416	21.9517	0.8243	7.2829	0.2855	5.3912	22.3292	0.6469	7.1632
	Z-Sampling(ours)	**0.3238**	**6.1542**	**22.1025**	**0.9087**	**7.3723**	**0.2990**	**5.6433**	**22.3485**	**0.7351**	**7.2565**
Hunyuan-DiT	Standard	0.3082	6.20461	21.8851	0.9422	7.3350	0.3022	5.7033	22.2926	0.8263	7.2811
	Resampling	0.3110	6.1932	21.8745	0.9551	7.3334	**0.3072**	5.6763	22.3175	0.9582	7.3148
	Z-Sampling(ours)	**0.3112**	**6.3071**	**21.8982**	**0.9788**	**7.3738**	0.3053	**5.7525**	**22.3988**	**0.9613**	**7.3545**

Table 2: The quantitative results of Z-Sampling on GenEval. Model: SDXL

Method	Single object ↑	Two object ↑	Counting ↑	Colors ↑	Position ↑	Color attribution ↑	Overall ↑
Standard	97.50%	69.70%	33.75%	86.71%	10.00%	18.00%	52.52%
Resampling	98.75%	**76.77%**	38.75%	**88.30%**	5.00%	20.00%	54.594%
Z-Sampling(ours)	**100.00%**	74.75%	**46.25%**	87.23%	**10.00%**	**24.00%**	**57.04%**

Table 3: The quantitative results of Z-Sampling and Semantic-CFG. Model: SD-2.1. For fairness, we follow the default settings of Semantic-CFG with the 768×768 resolution and SD-2.1.

Method	Pick-a-Pic					DrawBench				
	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑
Standard	0.2567	5.6579	20.2041	0.0053	6.5310	0.2598	5.3707	21.3889	0.0797	6.7747
Semantic-aware CFG	0.2602	5.6512	20.2818	0.0203	6.5534	0.2603	5.3729	21.3754	0.0939	6.7756
Z-Sampling(ours)	**0.2705**	**5.7423**	**20.4113**	**0.3689**	**6.6983**	**0.2671**	**5.4349**	**21.5466**	**0.2542**	**6.8757**

Table 4: Z-Sampling can enhance the training-free AYS. Model: DreamShaper-xl-v2-turbo.

Method	Pick-a-Pic					DrawBench				
	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑
Standard	0.3280	6.0493	22.3139	0.9148	7.4015	0.3094	5.5738	22.6760	0.7744	7.3334
Z-Sampling(ours)	**0.3353**	**6.1614**	**22.4479**	**1.0395**	**7.4960**	**0.3192**	**5.7145**	**22.7786**	**0.9582**	**7.4427**
AYS	0.3278	6.0523	22.3174	0.9188	7.4041	0.3095	5.5709	22.6798	0.7785	7.3347
AYS + Z-Sampling(ours)	**0.3357**	**6.1528**	**22.4463**	**1.0422**	**7.4942**	**0.3193**	**5.7152**	**22.7524**	**0.9482**	**7.4338**

Table 5: Z-Sampling can enhance the training-based Diffusion-DPO. Model: SDXL.

Method	Pick-a-Pic					DrawBench				
	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑
Standard	0.2989	6.0870	21.6353	0.5865	7.1520	0.2881	5.5595	22.3086	0.6075	7.1909
Z-Sampling(ours)	**0.3128**	**6.1302**	**21.8477**	**0.7822**	**7.2682**	**0.3050**	**5.6739**	**22.4581**	**0.7997**	**7.3092**
Diffusion-DPO	0.3141	5.9997	22.0070	0.9028	7.3059	0.2980	5.6604	**22.4695**	0.8594	7.3218
DPO + Z-Sampling(ours)	**0.3160**	**6.0836**	**22.1784**	**0.9448**	**7.3807**	**0.3035**	**5.6731**	**22.4673**	**0.9334**	**7.3443**

the effects across regions. While the setting is different from previous experiments, this results still underscore the effectiveness of Z-Sampling remains unaffected. As shown in Table 10, we observe that Z-Sampling demonstrates a higher improvement.

Moreover, we present more quantitative experimental results in Appendix D.1 and more qualitative comparison across various dimensions (e.g, color, style, and etc.) in Appendix D.2.

Specifically, we also discuss the effect of Z-Sampling under extremely high CFG guidance in Appendix D.4, demonstrating its ability to achieve a favorable balance between image quality and prompt adherence, suppressing artifacts and oversaturation.

Orthogonal Methods Z-Sampling can be combined with other orthogonal methods to further enhance diffusion models. In Table 4, Z-Sampling further enhances AYS-Sampling, a sampling

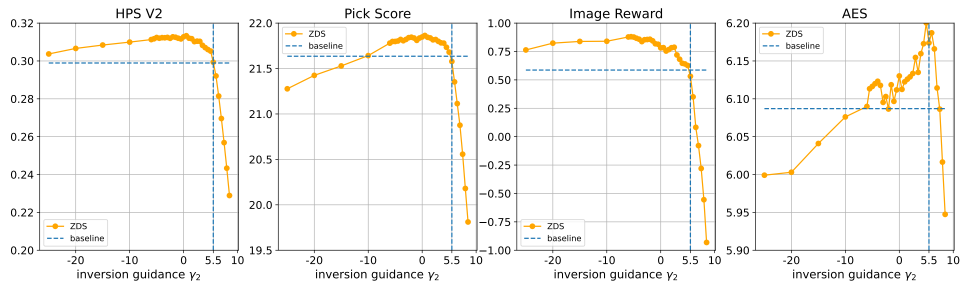

Figure 7: Robustness to the inversion guidance scale. When the gap is zero, i.e., the inversion guidance equals the denoising guidance (e.g. $\gamma_1 = \gamma_2$), the positive gains almost disappear.

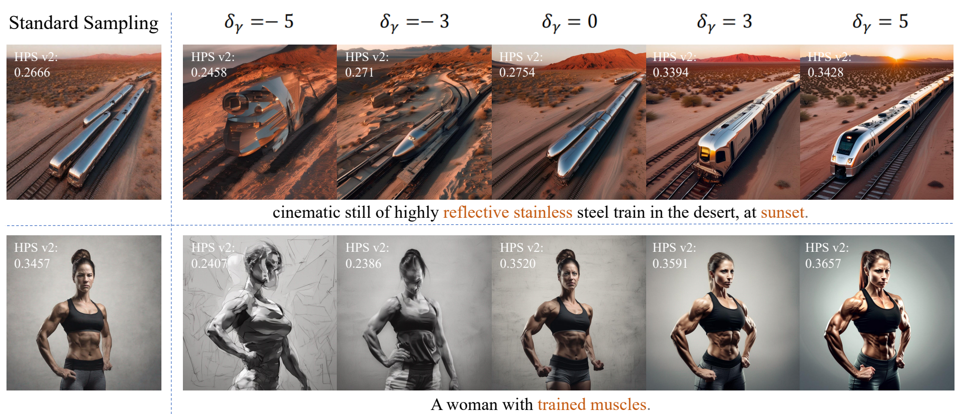

Figure 8: The guidance gap δ_γ between γ_1 and γ_2 influences both the magnitude and direction of semantic injection. When δ_γ is larger (δ_γ=5), the gain of Z-Sampling becomes pronounced. Conversely, when δ_γ is zero or even negative, it approximately degenerates into standard sampling or significantly break generation.

strategy that optimizes the denoising scheduler, leading to improved overall performance. Note that AYS-Sampling only released the 10-step scheduler, which is more applicable to DreamShaper-v2-turbo. Additionally, Table 5 shows that Z-Sampling can also be combined with training-based methods, further enhancing the generation quality of Diffusion-DPO. We leave more quantitative results of enhancing orthogonal methods in Table 8.

The Guidance Gap We first examine the impact of guidance scale. In Section 3.1, we show that the guidance gap between denoising and inversion dictates the degree of semantic information gain. To further verify this, we fix the guidance scale γ_1 as 5.5 following standard sampling. By varying γ_2, we control the guidance gap $\delta_\gamma = \gamma_1 - \gamma_2$ to observe its impact. As shown in Figure 7, when γ_2 increases and the guidance gap δ_γ narrows, the benefits of Z-Sampling diminish. According to the theoretical results of semantic information gain, a zero guidance gap can approximately lead to standard sampling. When the gap is below zero ($\gamma_2 > \gamma_1$), it can result in a negative gain. In Figure 8, we present a qualitative analysis showing that when the zero guidance gap indeed yields very similar results to standard sampling.

Zigzag Diffusion Steps We note that λ indicates the first λ steps using the zigzag operation. For example, when λ is 0, it reverts to standard SDXL. When λ is 25, it means the first 25 steps of the denoising process use the zigzag operation. We conducted experiments on Pick-a-Pick using SDXL (50 steps), as shown in Figure 9, when λ increases from 0 to 25, the winning rate rises from 50% to 75%. However, when λ increases from 25 to 50 steps, it only rises from 75% to 80%. This indicates that the **zigzag operation** is more effective during the early stages of denoising process.

Time Efficiency Comparison When the denoising steps are fixed (e.g., T=50), Z-Sampling naturally incurs additional time consumption due to the zigzag step. To facilitate a fairer comparison

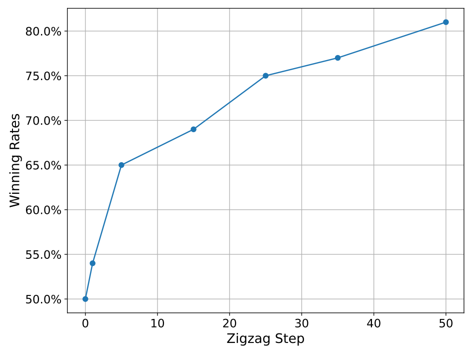

Figure 9: Robustness to the zigzag diffusion steps λ. The horizontal axis shows the number of zigzag operations, and the vertical axis represents the winning rate over HPS v2 on Pick-a-Pic. As λ increases, generation quality improves, indicating effective semantic information gain throughout the whole path.

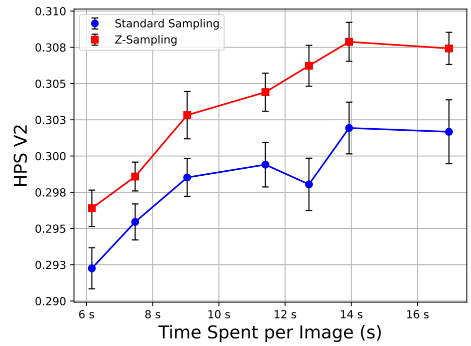

Figure 10: Z-Sampling outperforms standard sampling with the same time consumption and significantly enhance the performance limit of pretrained SDXL. The horizontal axis shows the average time per image, while the vertical axis shows the average HPS v2 on the Pick-a-Pic benchmark.

in terms of computation time, we compare evaluation score under the same generation time consumption per image, where Z-Sampling can maintain high quality with fewer sampling steps than standard methods. Here we apply Z-Sampling to the first half path, namely $\lambda = T/2$. Figure 10, indicates that Z-Sampling outperforms standard sampling and significantly enhance the performance limit of SDXL. Particularly, even with 36% less computational time, Z-Sampling can reach the best performance of standard sampling with HPS v2 ≈ 0.3.

5 Discussion

In this section, we further discuss the limitations and future directions of our work. First, we note that Z-Sampling relies on the semantic information gain through deterministic inversion, limiting its applicability to deterministic samplers, such as DDIM. Extending it to the SDE-based diffusion framework is an important direction for future work (see Appendix E.1). Second, while Z-Sampling exhibits strong generalization, we only studied text-to-image diffusion models in this work. Therefore, exploring its applications to areas such as video generation, 3D generation, and molecular synthesis is naturally another promising research direction. However, due to the different natures of latent space and sampling schedulers, this direction may require further algorithm design and theoretical understanding. Third, Z-Sampling can take more computational time than standard sampling due to its zigzag step given the fixed inference step T. It will be helpful to employ different step sizes for denoising and inversion. It is possibile to accelerate Z-Sampling with less zigzap steps while maintaining the comparable performance.

6 Conclusion

To the best of our knowledge, this work is the first to theoretically and empirically discover that the guidance gap between denoising and inversion can inject semantic information into the latent space, which can lead to improved generation with relevant semantic information as the prompt. By theoretically investigating how the semantic information gain depend on the guidance gap, we naturally derive a novel Z-Sampling method that can accumulate semantic information through each zigzag step and, thus, generate more desirable results. The conducted extensive experiments not only demonstrate that Z-Sampling significant outperforms the baselines in various settings, but also suggest that Z-Sampling can further enhance other orthogonal methods. In summary, Z-Sampling is flexible, additive, and powerful with limited time consumption. Given the theoretical mechanism and empirical success of Z-Sampling, we believe this work will motivate better theoretical understanding of diffusion sampling and inspire more advanced diffusion sampling methods along this approach beyond T2I generation.

Ethics Statement

We propose Z-Sampling, a novel guidance mechanism designed to enhance the quality of diffusion model generation. Although it does not directly involve human subjects or issues related to dataset privacy, we have carefully considered its potential ethical and moral implications. We ensure the transparency of all datasets used for debugging and developing the algorithm, and their randomness guarantees the absence of bias in the ethical domain, which is of utmost importance. Additionally, all models used comply with the terms of open-source licenses. Given Z-Sampling's significant commercial potential, we strive to apply this technology responsibly, ensuring that its applications yield positive societal benefits.

References

Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 843–852, 2023.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070*, 2024.

Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. *arXiv preprint arXiv:2403.14602*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Nicholas Guttenberg. Diffusion with offset noise, 2023.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*.

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

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

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.

Susung Hong. Smoothed energy guidance: Guiding diffusion models with reduced energy curvature of attention. *arXiv preprint arXiv:2408.00760*, 2024.

Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. *arXiv preprint arXiv:2210.00939*, 2022.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:36652–36663, 2023.

Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19948–19960, 2023.

Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024.

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 5404–5411, 2024a.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2837–2845, 2021.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5321–5329, 2023a.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Semantic-driven initial image construction for guided image synthesis in diffusion model. *arXiv preprint arXiv:2312.08872*, 2023b.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Mao Po-Yuan, Shashank Kotyan, Tham Yik Foong, and Danilo Vasconcellos Vargas. Synthetic shifts to initial seed vector exposes the brittle nature of latent-based diffusion models. *arXiv preprint arXiv:2312.11473*, 2023.

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.

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*, pp. 8748–8763. PMLR, 2021.

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

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. *arXiv preprint arXiv:2404.14507*, 2024.

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.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. 2024.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.3.0.

Dazhong Shen, Guanglu Song, Zeyue Xue, Fu-Yun Wang, and Yu Liu. Rethinking the spatial inconsistency in classifier-free diffusion guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9370–9379, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7623–7633, 2023a.

Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023b.

Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *European Conference on Computer Vision*, pp. 378–394. Springer, 2025.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023c.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024a.

Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Good seed makes a good crop: Discovering secret seeds in text-to-image diffusion models. *arXiv preprint arXiv:2405.14828*, 2024b.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. In *Forty-first International Conference on Machine Learning*.

Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *arXiv preprint arXiv:2406.03293*, 2024.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*.

Shenghai Yuan, Jinfa Huang, Yongqi Xu, Yaoyang Liu, Shaofeng Zhang, Yujun Shi, Ruijie Zhu, Xinhua Cheng, Jiebo Luo, and Li Yuan. Chronomagic-bench: A benchmark for metamorphic evaluation of text-to-time-lapse video generation. *arXiv preprint arXiv:2406.18522*, 2024.

A Experimental Details

In this section, we introduce the details of the metrics and benchmarks used in the experiments.

A.1 Datasets

Pick-a-Pic. The Pick-a-Pic dataset (Kirstain et al., 2023) was generated by logging user interactions with the Pick-a-Pic web application for text-to-image generation. Each entry includes a prompt, two generated images, and a label indicating the preferred image or a tie if neither is significantly favored. Here we use only the first 100 prompts as the test set, which is sufficient to reflect the model's capabilities.

Drawbench. DrawBench is a comprehensive and challenging benchmark for text-to-image models, introduced by the Imagen research team (Saharia et al., 2022). It contains 11 categories, including aspects such as color, counting, and text, with approximately 200 text prompts.

GenEval. Geneval (Ghosh et al., 2024) is an object-focused framework designed to evaluate compositional properties of images, including object co-occurrence, position, count, and color. It incorporates 553 prompts, achieving an 83% agreement with human judgments regarding the correctness of the generated images[1].

PartiPrompts. PartiPrompts (Yu et al.) is a collection of over 1,600 diverse prompts in English, designed to assess the capabilities of models across different categories and challenges. The prompts cover a wide range of topics and styles, helping evaluate the strengths and weaknesses of models in areas like language understanding, creativity, coherence. Here we randomly select 100 prompts from Part for evaluation.

A.2 Metrics

AES. Aesthetic score (AES) (Schuhmann et al., 2022) refers to a mechanism for evaluating the visual quality of generated images, which assigns a quantitative score based on attributes like contrast, composition, color, and detail, reflecting alignment with human aesthetic standards.

[1]To ensure consistency with other experiments, we used a denoising guidance scale of 5.5, differing from the default 9.0 in GenEval.

PickScore. Kirstain et al. (2023) developed Pick-a-Pic, a large open dataset consisting of text-to-image prompts and real user preferences for generated images. They then utilized this dataset to train a CLIP-based scoring function, PickScore, for the task of predicting human preferences.

ImageReward. Xu et al. (2024a) developed ImageReward, the first general-purpose text-to-image human preference reward model. which is trained based on systematic annotation pipeline, including rating and ranking and has collected 137,000 expert comparisons to date.

HPS v2. Wu et al. (2023c) first introduced the Human Preference Dataset v2 (HPD v2), a large-scale dataset comprising 798,090 human preference choices on 433,760 pairs of images. By fine-tuning CLIP using HPD v2, they developed the Human Preference Score v2 (HPS v2), a scoring model that more accurately predicts human preferences for generated images.

A.3 BASELINES

Semantic-aware CFG (Shen et al., 2024), adaptively adjust the CFG scales across different semantic regions to mitigate the undesired effects caused by guidance.

Diffusion-DPO (Wallace et al., 2024), finetune a pretrained Diffusion model using carefully curated high quality images and captions to improve visual appeal and text alignment.

AYS-Sampling (Sabour et al., 2024), a strategy for optimizing sampler timesteps, which accounts for the dataset, model, and sampler to enhance image quality.

B RELATED WORKS

In this section, we discuss existing work related to Z-Sampling.

Semantic Information in Latent Space Recent works have shown that the prior information present in the noise latent can significantly impact the quality of image generation (Xu et al., 2024b; Mao et al., 2023a; Samuel et al., 2024). For example, Mao et al. (2023b) found certain regions in random latents can induce objects representing specific concepts. And (Po-Yuan et al., 2023) found slight perturbations can lead to significant changes in the diffusion model's generated results. And injecting semantic information (e.g., low-frequency wavelengths) into Gaussian noise can enhance image quality, particularly improving alignment performance (Wu et al., 2023c; Guttenberg, 2023; Lin et al., 2024a). IRFDS (Yang et al., 2024) utilizes a pretrained rectified flow model to provide a prior, optimizing the initial latent for image editing task. Building on these studies, we investigate semantic information from the guidance perspective, implicitly integrating it into the generation process without requiring explicit reference data.

Sampling Strategies of Diffusion Model To improve the sampling process, lug (2022) proposed Resampling that involves adding random noise and performing multiple back-and-forth samples at each timestep. Subsequent studies adopted this paradigm for tasks such as video generation (Wu et al., 2023b) and universal classifier guidance (Bansal et al., 2023). IRFDS () utilizes a pretrained rectifying flow model to provide a prior, optimizing the initial latent for better image editing. However, they overlooked the importance of inverted latent and simply applied random noise, which does not effectively enhance prompt adherence. In Tune-a-Video, to ensure structural consistency, Wu et al. (2023a) incorporate the denoising-inversion paradigm as a subcomponent. However, their end-to-end approach is not optimal and overlooks the importance of the guidance gap. To reduce spatial inconsistency in different latent regions under the same guidance scale, Shen et al. (2024) developed adaptive guidance based on semantic segmentation. It relies on attention-level changes, limiting adaptability to other algorithms, and its robustness is influenced by semantic segmentation effectiveness. Constraint-based approaches aim to improve sampling, for example, Chung et al. (2024) substitutes conditional noise with unconditional noise to enhance generation quality from an image manifold perspective, though improvements are minimal. Yang et al. applies spherical gaussian constraint during guidance, but it requires a reference data, limiting its applicability. Finally, Garibi et al. (2024) proposed Renoise, which enhances image editing by ensembling latents through inversion operations. However, it focuses on inversion error smoothing after multiple inversions and lacks a thorough investigation of the guidance mechanism.

C Motivation and Phenomena

C.1 Latents with Semantic Information

In Figure 11, we present additional cases illustrating that random latents encode relevant semantic information. For instance, for prompts related to the concept "Jeep Cars", the latent corresponding to seed 20 achieves the highest performance, with PickScore of 23.4784, whereas latents from other seeds fail to exceed PickScore of 23.

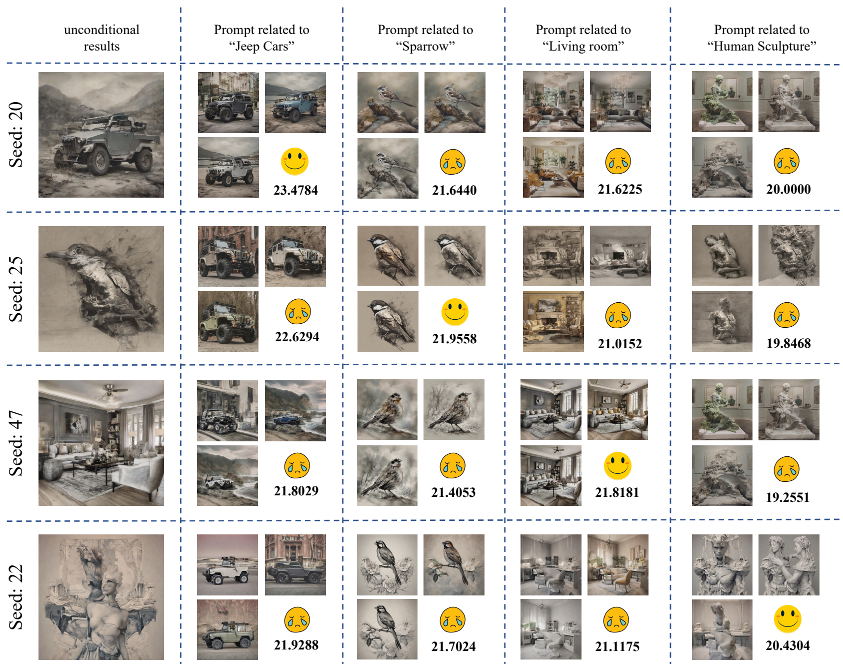

Figure 11: Latents with relevant semantic information about a specific concept can generate images more effectively from prompts related to that concept. Each row shows the results of the same latent across different prompts, while each column shows results from different latents under the same prompts. For each cell, we compute the PickScore. For example, the latent related seed 20 achieves an PickScore of 23.4784 when generating images related to "Jeep Cars".

C.2 Inversion process makes good latent

In this section, we show that the inverted latent inherently carries semantic information related to the conditional prompt c. These extra semantic information gain leads to superior generation outcomes.

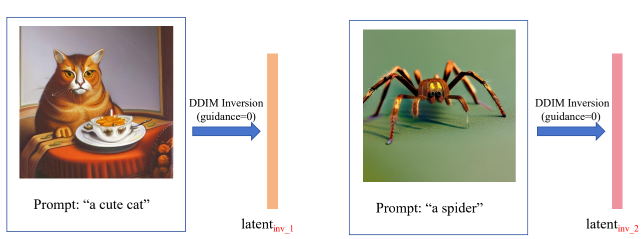

Figure 12: Given two natural images and their corresponding prompts, we perform DDIM inversion to reverse them and obtain the corresponding initial noise latents.

First, we choose images of "cats" and "spiders" as depicted in Figure 12. Employing the DDIM inversion algorithm with guidance scale set to 0, we obtatin latent$_{inv_1}$ and latent$_{inv_2}$. We hypothesize that latent$_{inv_1}$ encapsulates semantic information associated with "cat" whereas latent$_{inv_2}$ inherently relates more closely to "spiders".

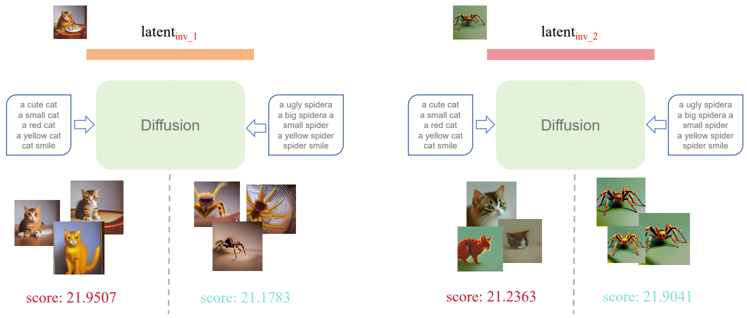

Figure 13: Generate images related to "cat" and "spider" using two latents respectively, and calculate the PickScore.

Next, we use these two latents to generate images conditioned on text prompts "cats" and "spiders" respectively, as illustrated in Figure 13. We observe that latent$_{inv_1}$ performs better when conditioned on text related to "cats" while latent$_{inv_2}$ performs better when conditioned on text related to "spiders". This phenomenon empirically validates our hypothesis that inverted latent does matter.

D Supplementary Experimental Results

In this section, we present more quantitative and qualitative results of Z-Sampling.

D.1 Supplementary Quantitative Results

Results of Z-Sampling in other benchmarks In Table 6, we evaluate 100 randomly selected prompts from PartiPrompts using the SDXL model, with Z-Sampling demonstrating the higher performance. Additionally, we also compare classical metrics such as FID (Seitzer, 2020), IS (Salimans et al., 2016), and clip-score (Radford et al., 2021) on MS-COCO 2014 (Lin et al., 2014). Due to numerous evaluation prompts (30K), we employ the distilled model, DreamShaper-xl-v2-turbo, with 4 denoising steps, showing the higher generation quality in Table 7. We also report additional comparative results on Geneval in Table 8, including Resampling and Diffusion-DPO, showcasing Z-Sampling's superiority in average scores.

Table 6: The quantitative results of Z-Sampling on PartiPrompts. Model: SDXL.

Method	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Average↑
Standard	0.2934	5.8122	22.2719	0.7253	7.2757
Resampling	0.3021	5.7811	22.4247	0.9234	7.3578
Z-Sampling(ours)	**0.3100**	**5.8472**	**22.4317**	**0.9732**	**7.3905**

Table 7: The quantitative results of Z-Sampling on MS-COCO 2014. Model: DreamShaper-xl-v2-turbo.

Method	IS-30K ↑	FID-30K ↓	Clip-Score ↑
Standard	34.0745	24.1420	0.3267
Z-Sampling(ours)	**34.4173**	**23.4958**	**0.3288**

Table 8: The additional quantitative results of Z-Sampling on GenEval. Model: SDXL

Method	Single object ↑	Two object ↑	Counting ↑	Colors ↑	Position ↑	Color attribution ↑	Overall ↑
Standard	97.50%	69.70%	33.75%	86.71%	10.00%	18.00%	52.52%
Diffusion-DPO	100.00%	80.81%	45.00%	88.30%	10.00%	**31.00%**	59.18%
DPO+Z-Sampling(ours)	**100.00%**	**82.83%**	**46.25%**	**89.36%**	**10.00%**	29.00%	**59.57%**

Results of Z-Sampling in other baselines and tasks We also compare Z-Sampling with other methods that improve the effect of guidance. Specifically, Hong et al. (2022) proposed SAG, which employs blur guidance and intermediate self-attention maps to achieve higher quality samples. Furthermore, SEG (Hong, 2024) further optimized SAG from the energy landscape perspective. Here we report the comparison results with SEG in Table 9. Additionally, We have also compared Z-Sampling with CFG++ (Chung et al., 2024), which optimizes the classifier-free guidance mechanism from the perspective of manifold constraints. since it restricts the cfg scale to the range from 0.0 to 1.0, while the classic Z-Sampling is larger, a fair comparison is not possible. Given this, we use $\omega = 0.5$ in CFG++, corresponding to a cfg scale of 5.5 in Z-Sampling.

Table 9: The quantitative results of Z-Sampling and SEG. Model: SDXL.

Method	Pick-a-Pic					DrawBench				
	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑
Standard	0.2989	6.0870	21.6353	0.5865	7.1520	0.2881	5.5595	22.3086	0.6075	7.1909
SEG	0.3053	6.1231	21.4186	0.6157	7.1156	0.2960	5.6596	22.1453	0.6042	7.1763
Z-Sampling(ours)	**0.3128**	**6.1302**	**21.8477**	**0.7822**	**7.2682**	**0.3050**	**5.6739**	**22.4581**	**0.7997**	**7.3092**

Table 10: The quantitative results of Z-Sampling and CFG++. Model: SDXL. It is worth noting that in the official implementation of CFG++, the VAE encoder uses **madebyollin/sdxl-vae-fp16-fix** checkpoint. For fair comparison, we follow this setting, so the results reported for SDXL and Z-Sampling are slightly different from the previous results.

Method	Pick-a-Pic					DrawBench				
	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑	HPS v2↑	AES↑	PickScore↑	IR↑	Average↑
Standard	0.3004	6.1121	21.8053	0.6007	7.2046	0.2885	5.6245	22.4213	0.6761	7.2526
CFG++	0.3028	6.0989	21.8337	0.6730	7.2271	0.2865	5.6174	22.3797	0.6266	7.2275
Z-Sampling(ours)	**0.3124**	**6.1170**	**21.8444**	**0.7855**	**7.2648**	**0.3035**	**5.6594**	**22.4392**	**0.7911**	**7.2983**

Finally, as a general method, we test Z-Sampling's performance on the video generation task. We choose AnimateDiff (Guo et al., 2023) as the baseline model and test it on Chronomagic-Bench-150 (Yuan et al., 2024), and we set $\gamma_1 = 7.5$ and $\gamma_2 = 0$ in Z-Sampling. With the results shown in Table 11, we note that Z-Sampling outperforms both AnimateDiff and another train-free sampling method FreeInit (Wu et al., 2025) in UMT-FVD (Liu et al., 2024), UMT-SCORE (Li et al., 2023), GPT4o-MTSCORE (Achiam et al., 2023).

Table 11: The quantitative results of Z-Sampling on Chronomatic-Bench-150. Model: AnimateDiff.

Method	UMT-FVD ↓	UMT-SCORE ↑	GPT4o-MTSCORE ↑
Standard	275.18	2.82	2.83
FREEINIT	268.31	2.82	2.59
Z-Sampling(ours)	**243.26**	**2.97**	**2.88**

Table 12: The quantitative results of Z-Sampling under different denoising steps k. Model: SDXl.

k	HPS v2 ↑	AES ↓	PickScore ↑	IR ↑
0 (SDXL)	0.2989	6.0870	21.6353	0.5865
1	**0.3128**	**6.1302**	**21.8477**	0.7922
2	0.3111	6.0764	21.7163	**0.8453**
3	0.3075	6.0885	21.4848	0.7854
4	0.3059	6.0940	21.3357	0.7860

Multiple steps of denoising and inversion operation in Z-Sampling We have explored the one-step scenario, i.e, $x_t \rightarrow x_{t-1} \rightarrow \tilde{x}_t$. Here, we extend to multiple steps scenario, i.e., $x_t \rightarrow x_{t-k} \rightarrow \tilde{x}_t$. As shown in Table 12, the best performance is achieved when k=1. As k increases, the performance of Z-Sampling deteriorates, which aligns with the Theorem 1 and Theorem 2, where increasing k gradually brings the step-by-step approach closer to end-to-end, thereby increasing the error term τ_2. Specifically, when k=T-1 and the zigzag operation is only performed on the initial latent, it corresponds to the scenario in Table 16.

D.2 Supplementary Qualitative Results

In Figure 14, we note Z-Sampling can better recognize the stylistic descriptions in prompts. For example, it can generate "Mario characters" that are more realistic and lifelike.

Figure 14: Qualitative comparison in terms of style.

In Figure 15, we note Z-Sampling accurately interprets object positional relationships, e.g., ‘underneath’, ‘on top of’, ‘on the right of’, etc.

Figure 15: Qualitative comparison in terms of position.

In Figure 16, Z-Sampling enhances the binding of color attributes, aligning images more closely with prompts and improving quality.

Figure 16: Qualitative comparison in terms of color.

In Figure 17, we note Z-Sampling demonstrates enhanced capability in understanding quantitative relationships, effectively addressing the persistent challenge in diffusion models. For example, it can effectively understand and generate images such as 'three suitcases', 'four buses', and two beds'.

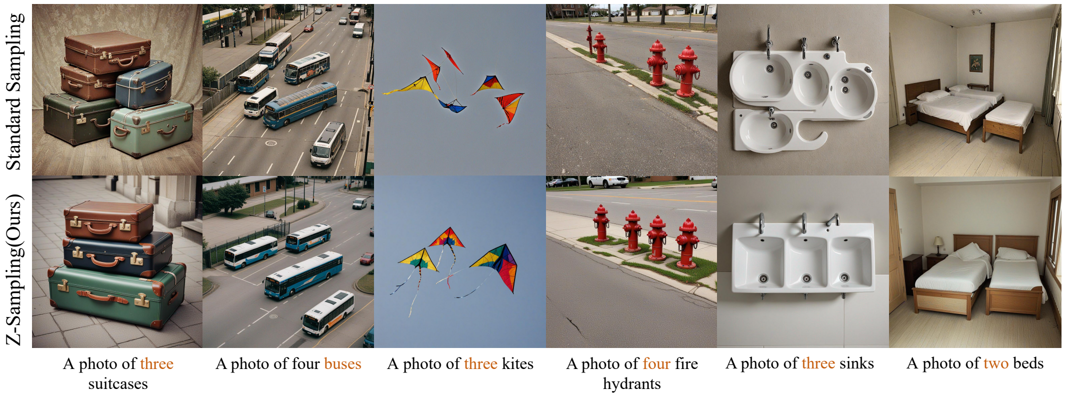

Figure 17: qualitative comparison in terms of counting.

In Figure 18, we find that Z-Sampling aids in generating Multi-object composite (e.g., a mouse and a bowl) or counterfactual (e.g., an elephant in the sea) images, manifested in its enhanced 'co-occurrence' capability.

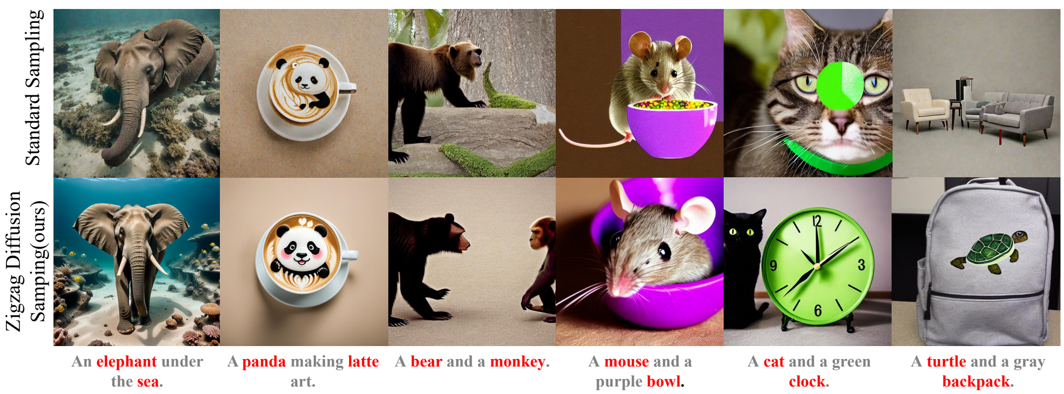

Figure 18: Qualitative comparison in terms of object co-occurrence.

D.3 Winning rates comparison

Here, we present a comparative analysis of winning rates under various settings, such as different models and denoising steps. The blue bars represent Z-Sampling (ours), while the orange bars represent the standard sampling method. Winning rates of our method exceeds 50% in all metrics. Especially HPS v2, which is much better than standard method.

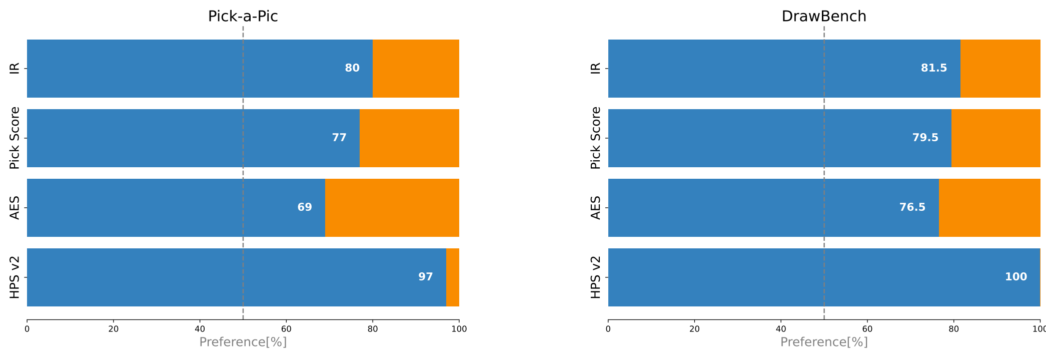

Figure 19: Comparison of Winning Rates with 10 Denoising Steps in the SDXL.

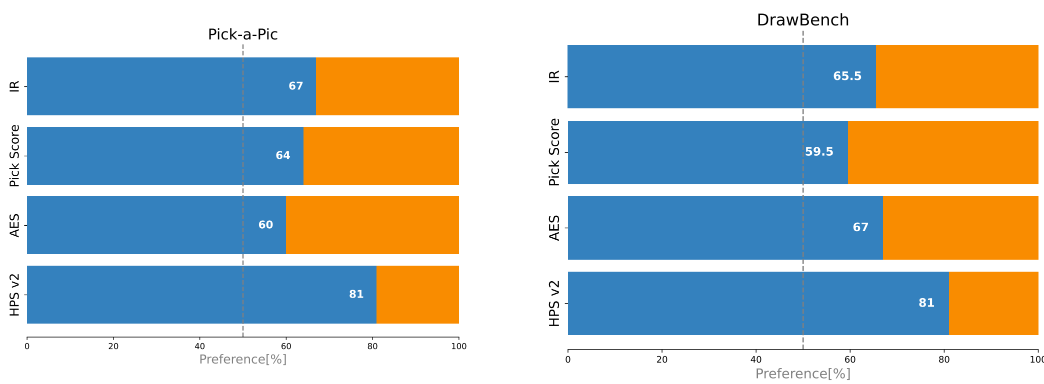

Figure 20: Comparison of Winning Rates with 50 Denoising Steps in the SDXL.

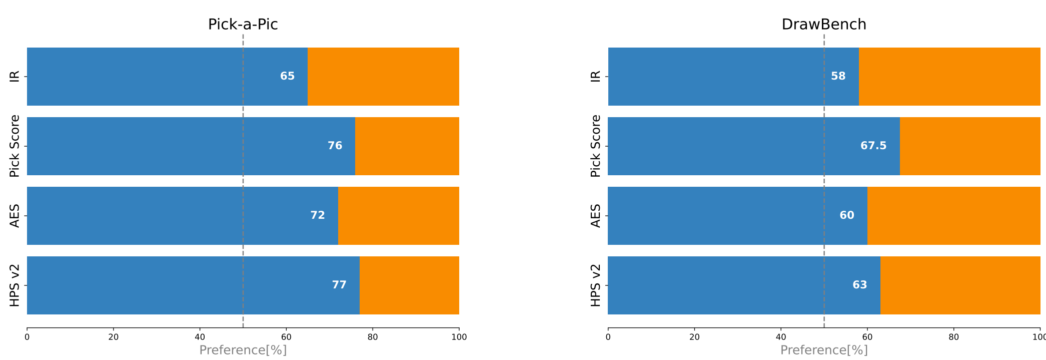

Figure 21: Comparison of Winning Rates with 50 Denoising Steps in the SD 2.1.

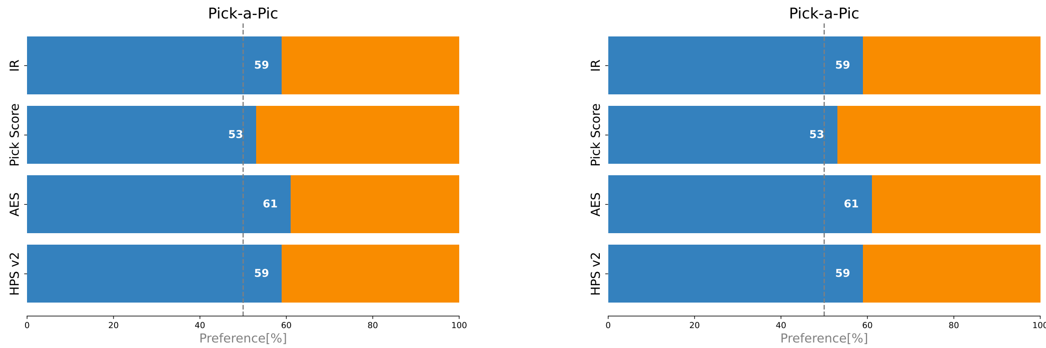

Figure 22: Comparison of Winning Rates with 10 Denoising Steps in the Hunyuan-DiT.

D.4 PERFORMANCE OF Z-SAMPLING UNDER HIGH CFG SCALE

We also report the performance of Z-Sampling under different intensities of classifier free guidance γ_1 during denoising process.

We use DreamShaper-xl-turbo-v2 as the base model. As shown in Table 13, the standard sampling performs best at $\gamma_1 = 3.5$, which is also the official recommended guidance sclae. When $\gamma_1 \geq 3.5$, the standard sampling begins to exhibit issues such as oversaturation and artifacts.

However, Z-Sampling consistently yields positive gains, indicating that our method can still work effectively under high guidance scales. And we present the winning rate of Z-Sampling over Standard sampling on HPS v2 across different guidance sclae γ_1 in Figure 23, further validating this point.

Table 13: Performance of Z-Sampling under different guidance γ_1. Model: DreamShaper-xl-turbo-v2. We note that the official recommended guidance scale $\gamma_1 = 3.5$. When $\gamma_1 > 3.5$, the quality of standard sampling gradually declines, while Z-Sampling still shows improvement on this basis.

Method	γ_1	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Winning Rate↑
Standard Sampling	1.5	0.2851	5.8327	21.3729	0.4325	-
Z-Sampling	1.5	**0.2951**	**6.0143**	**21.6541**	**0.5589**	73%
Standard Sampling	3.5	0.3004	5.9355	21.5899	0.6618	-
Z-Sampling	3.5	**0.3238**	**6.1542**	**22.1025**	**0.9087**	88%
Standard Sampling	5.5	0.2996	5.9668	21.3718	0.6446	-
Z-Sampling	5.5	**0.3142**	**6.0513**	**21.8309**	**0.7600**	85%
Standard Sampling	7.5	0.2910	5.8816	21.0236	0.6026	-
Z-Sampling	7.5	**0.3090**	**5.9537**	**21.5977**	**0.7418**	86%
Standard Sampling	9.5	0.2798	5.7649	20.5981	0.4170	-
Z-Sampling	9.5	**0.2995**	**5.8788**	**21.2806**	**0.6340**	92%
Standard Sampling	11.5	0.2693	5.6030	20.3055	0.3145	-
Z-Sampling	11.5	**0.2897**	**5.7694**	**20.9710**	**0.5569**	91%

Generally, classifier-free guidance serves as a mechanism for semantic control, balancing image quality and prompt adherence, with excessive guidance scale causing deviations and artifacts. Z-Sampling, as a similar semantic enhanced mechanism, employs an iterative approach (unlike the vanilla CFG mechanism, which directly alters the latent distribution) to more effectively explore this balance. And we presents some visual cases in Figure 24, showcasing Z-Sampling's capability to maintain image quality even under high guidance scale.

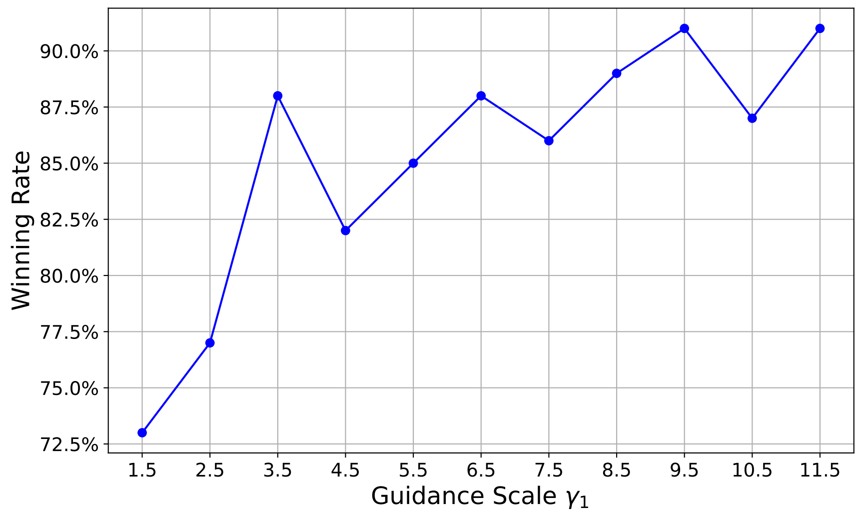

Figure 23: Comparison of Winning Rates under different guidance scale γ_1. Model: DreamShaper-xl-turbo-v2. Horizontal axis: guidance scales γ_1. Vertical axis: Z-Sampling vs Standard Sampling winning rates on Pick-a-Pic.

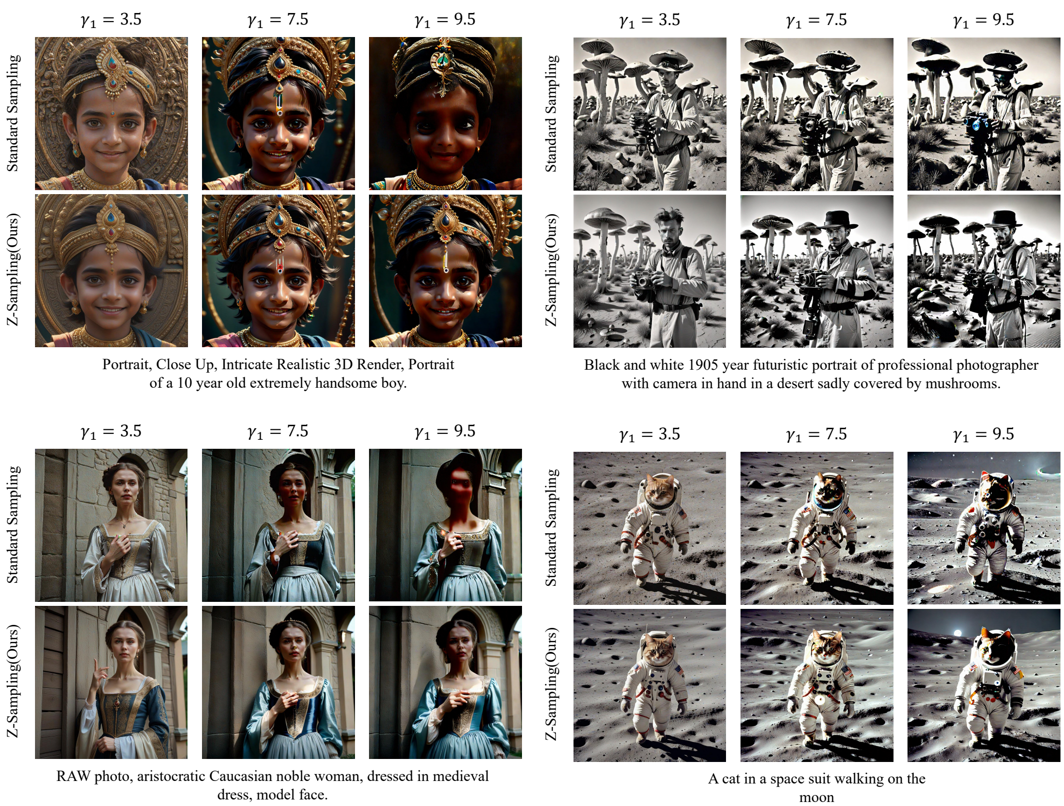

Figure 24: Qualitative comparison under high guidance scale. When $\gamma_1 = 3.5$ (the official recommended guidance scale), both Z-Sampling and Standard exhibit no artifacts or degradation in image quality. As γ_1 increases, standard sampling exhibits artifacts and oversaturation, while Z-Sampling is less affected.

E ANALYSIS OF THE APPROXIMATION ERROR TERM

In this section, we undertake a more in-depth analysis of the approximation error term τ_2 within Equation 10. We first demonstrate Z-Sampling's results under the uncertainty scheduler. Then, we analyze how this approximation error affects the performance of Z-Sampling.

E.1 UNCERTAINTY AND STOCHASTIC SAMPLERS

To assess the impact of different inversion algorithms on generation quality, we test various inversion methods. Specifically, we use SDXL-Turbo (4 steps) (Sauer et al., 2023) , an adversarial distillation diffusion model. Notably, SDXL-Turbo's default sampler is an ancestral Euler sampler, which introduces random noise at each denoising step, leading to highly inaccurate inversion.

Table 14: With stochastic samplers (e.g., Euler(a)), inversion inaccuracies reduce Z-Sampling's effectiveness. In contrast, deterministic samplers (e.g., Euler) yield better results with Z-Sampling.

Method	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Average ↑
Standard Sampling$_{\text{Euler(a)}}$	**0.3123**	**5.9561**	21.6364	**0.8224**	**7.1818**
Z-Sampling$_{\text{Euler(a)}}$	0.3078	5.9523	**21.6503**	0.8060	7.1791
Standard Sampling$_{\text{Euler}}$	0.2705	5.6023	20.3643	**0.4144**	6.6628
Z-Sampling$_{\text{Euler}}$	**0.2857**	**5.8482**	**20.9639**	0.3954	**6.8733**

From Table 14, it can be seen that when using the Euler ancestral sampler, e.g., Euler(a), which introduces randomness in the denoising process, most metrics show a decline. This is because Euler(a) leads to inaccuracies in the inversion process, causing the approximation error term in equation 23 to increase significantly. As a result, Z-Sampling diverges from the data manifold, leading to reduced effectiveness.

However, when using deterministic Euler samplers, although the overall performance does not match that of the Euler(a) Sampler—acknowledging that other sampling methods on the turbo model may introduce blurring and related issues—Z-Sampling still demonstrates performance improvements over the corresponding baseline. For example, the PickScore increase from 20.3643 to **20.9639** This highlights the importance of the inversion algorithm and presents opportunities for improving Z-Sampling under stochastic samplers

Corresponding to equation 10, a deterministic sampler implies that the inversion process is imprecise, leading to an increase in $\tau_2(t)$. We note that end-to-end inversion amplifies the approximation error (Mokady et al., 2023), risking latents deviating from the data manifold. Z-Sampling, on the other hand, truncates the error at each step, reducing τ_2, making semantic injection more efficient.

E.2 THE INCREASE IN APPROXIMATION ERROR RESULTS IN NEGATIVE GAINS

To focus solely on the approximation error τ_2 in Equation 10, we need to eliminate the influence of the semantic term τ_1. So we set $\gamma_1 = \gamma_2 = 5.5$, which means $\delta_\gamma = 0$ and $\tau_1 = 0$. Then Equation 10 can be transformed as

$$\delta_{\text{Z-Sampling}} = \sum_{t=1}^{T}(x_t - \tilde{x}_t)^2 = \sum_{t=1}^{T} \alpha_t h_t^2 \underbrace{(\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1}))}_{\tau_2(t)\text{:approx error term}}^2. \tag{12}$$

Similarly, Equation 9 can be transformed as

$$\delta_{end2end} = (x_T - \tilde{x}_T)^2 = \alpha_T(\sum_{t=1}^{T} h_t \underbrace{(\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1}))}_{\tau_2(t)\text{:approx error term}})^2. \tag{13}$$

Since the semantic term τ_1 no longer contributes, only the effect of τ_2 remains, as shown in Table 15 and Figure 25, both the end-to-end and step-by-step approaches result in negative gains. Notably, the approximation error introduced by the end-to-end method is two orders of magnitude higher than that of the step-by-step method, significantly degrading the image quality. This demonstrates that:

- **An increase in the error term τ_2 degrades the sampling effect**.
- **The step-by-step approach helps reduce the error term τ_2, mitigating this negative gain**.

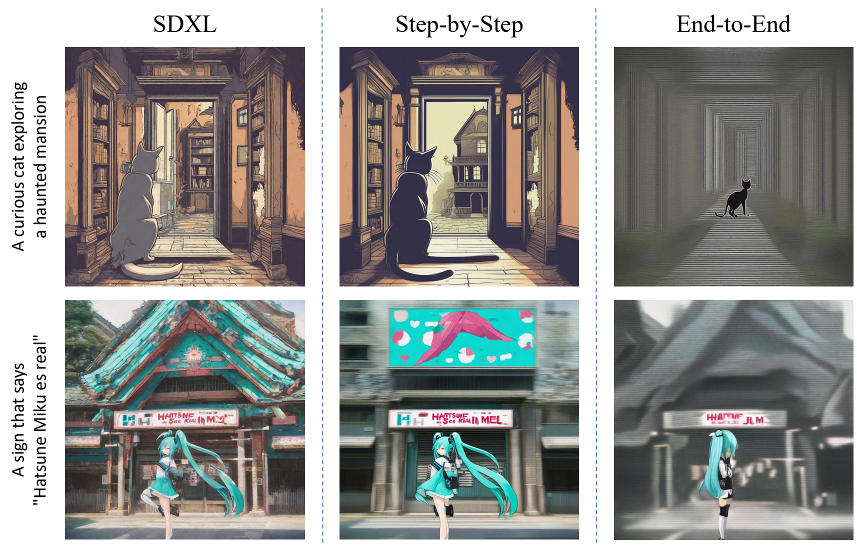

Figure 25: When the semantic term τ_1 is removed (e.g., $\tau_1 = 0$), the presence of only the error term τ_2 degrades the quality of generation results, and this negative gain effect is more pronounced in the end-to-end method.

Additionally, we test the performance of end-to-end and step-by-step methods in the presence of the semantic term τ_1, as shown in Table 16. Since in this case, τ_1 and τ_2 are mixed together, so we only report the PickScore to reflect the quality of the generated results, as we are unable to report the exact Approx Error. It can be observed that with the presence of the semantic term, both methods yield positive gains, and the step-by-step method performs better.

Table 15: The results on Pick-a-Pick, excluding semantic term τ_1. Model: SDXL.

Method	δ_γ	PickScore ↑	Approx Error τ_2
SDXL	-	**21.6353**	0
End-to-End	0	18.8182	160.3313
Step-by-Step	0	21.5257	0.9919

Table 16: The results on Pick-a-Pick, including semantic term τ_1. Model: SDXL.

Method	δ_γ	PickScore ↑
SDXL	-	21.6353
End-to-End	5.5	21.6485
Step-by-Step	5.5	**21.8477**

E.3 Artificially introducing Gaussian error

Specifically, to further illustrate that the approximation error τ_2 leads to negative gains, we consider adding an additional random Gaussian term $\mathbf{error}_{gs}$ to Equation 12, artificially simulating and controlling the inversion approximation error as

$$\delta_{\text{Z-Sampling}} = \sum_{t=1}^{T}(x_t - \tilde{x}_t)^2 = \sum_{t=1}^{T} \alpha_t h_t^2 (\underbrace{\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1})}_{\tau_2(t)\text{:approx error term}} + s * \frac{norm(\epsilon_\theta^t(x_t))}{norm(\mathbf{error}_{gs})}\mathbf{error}_{gs})^2, \quad (14)$$

where s is used to control the magnitude of the error. As seen in Table 17, the larger the value of s, the worse the performance of Z-Sampling, further illustrating that reducing the error term introduced by inversion is a direction that warrants attention.

Table 17: As the coefficient of the Gaussian error term increases, the quality of generation decreases.

s	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Average ↑
0	**0.2995**	**6.1889**	**21.5257**	**0.5112**	**7.1313**
0.5	0.2993	6.1502	21.5139	0.4553	7.1046
1.0	0.2812	6.0076	20.7824	0.2874	6.8396

F Proofs

In this section, we derive the relationship between the end-to-end semantic injection approach and Z-Sampling, proving Z-Sampling's superiority. Then we formalize how Z-Sampling injects semantics via the guidance gap.

Proof F.1 (Theorem 1) *Given inference timesteps of T, from equation 4, we can obtain the inverted latent $\tilde{x}_T$ as*

$$\tilde{x}_T = \sqrt{\frac{\alpha_T}{\alpha_{T-1}}}\tilde{x}_{T-1} + \sqrt{\alpha_T}\left(\sqrt{\frac{1}{\alpha_T} - 1} - \sqrt{\frac{1}{\alpha_{T-1}} - 1}\right)\epsilon_\theta^T(\tilde{x}_{T-1}). \tag{15}$$

For the sake of convenience, we set

$$m_T = \sqrt{\frac{\alpha_T}{\alpha_{T-1}}}, \qquad n_T = \sqrt{\alpha_T}\left(\sqrt{\frac{1}{\alpha_T} - 1} - \sqrt{\frac{1}{\alpha_{T-1}} - 1}\right). \tag{16}$$

So, equation 15 could also be written as

$$\tilde{x}_T = m_T\tilde{x}_{T-1} + n_T\epsilon_\theta^T(\tilde{x}_{T-1}). \tag{17}$$

Through iterative and combinatorial processes in equation 3, $\tilde{x}_T$ could be expressed as

$$\begin{aligned}
\tilde{x}_T &= m_T\tilde{x}_{T-1} + n_T\epsilon_\theta^T(\tilde{x}_{T-1}) \\
&= m_T m_{T-1}\tilde{x}_{T-2} + m_T n_{T-1}\epsilon_\theta^{T-1}(\tilde{x}_{T-2}) + n_T\epsilon_\theta^T(\tilde{x}_{T-1}) \\
&= m_T m_{T-1} m_{T-2}\tilde{x}_{T-3} + m_T m_{T-1} n_{T-2}\epsilon_\theta^{T-2}(\tilde{x}_{T-3}) + m_T n_{T-1}\epsilon_\theta^{T-1}(\tilde{x}_{T-2}) + n_T\epsilon_\theta^T(\tilde{x}_{T-1}) \\
&= \prod_{i=0}^{T} m_i\tilde{x}_0 + \sum_{t=1}^{T} n_t \prod_{k=t+1}^{T} m_k\epsilon_\theta^t(\tilde{x}_{t-1}).
\end{aligned} \tag{18}$$

Similarly, based on equation 1 and equation 2, we can perform iterative derivations to obtain the equivalent form of x_T as

$$x_T = \prod_{i=0}^{T} m_i x_0 + \sum_{t=1}^{T} n_t \prod_{k=t+1}^{T} m_k\epsilon_\theta^t(x_t). \tag{19}$$

We can determine the difference between x_T and $\tilde{x}_T$, representing the gain from end-to-end semantic injection as

$$\begin{aligned}
\delta_{\textit{end2end}} &= \left(x_T - \tilde{x}_T\right)^2 \\
&= \left(\prod_{i=0}^{T} m_i \left(x_0 - \tilde{x}_0\right) + \sum_{t=1}^{T} n_t \prod_{k=t+1}^{T} m_k \left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right)\right)^2 \\
&= \left(\sum_{t=1}^{T} \sqrt{\alpha_T} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1}\right) \left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right)\right)^2 \\
&= \alpha_T \left(\sum_{t=1}^{T} \left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1}\right) \left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right)\right)^2,
\end{aligned} \tag{20}$$

where we set $h_t = \frac{n_t}{\sqrt{\alpha_t}}$, and further refine equation 20 to yield the semantic injection term τ_1 and the approximation error term τ_2 as

$$\begin{aligned}
\delta_{\textit{end2end}} &= \alpha_T \left(\sum_{t=1}^{T} h_t \left(\epsilon_\theta^j(x_t) - \epsilon_\theta^t(\tilde{x}_t)\right)\right)^2 \\
&= \alpha_T \left(\sum_{t=1}^{T} h_t \left(\underbrace{\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_t)}_{\tau_1\textit{:semantic information gain term}} + \underbrace{\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1})}_{\tau_2\textit{:approx error term}}\right)\right)^2.
\end{aligned} \tag{21}$$

Proof F.2 (Theorem 2) *Unlike end-to-end approaches, in Z-Sampling, we focus solely on the local cycle of "$x_t \to x_{t-1} \to \tilde{x}_t$". Substituting equation 2 into equation 4 yields $\tilde{x}_t$ as*

$$\begin{aligned}
\tilde{x}_t &= x_t - \sqrt{1-\alpha_t}\epsilon_\theta^t(x_t) + \sqrt{\frac{(1-\alpha_{t-1})\alpha_t}{\alpha_{t-1}}}\epsilon_\theta^t(x_t) \\
&+ \sqrt{\alpha_t}\left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1}\right)\epsilon_\theta^t(\tilde{x}_{t-1}) \\
&= x_t + \sqrt{1-\alpha_t}\left(\epsilon_\theta^t(\tilde{x}_{t-1}) - \epsilon_\theta^t(x_t)\right) + \sqrt{\frac{(1-\alpha_{t-1})\alpha_t}{\alpha_{t-1}}}\left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right) \\
&= x_t + \left(\sqrt{1-\alpha_t} - \sqrt{\frac{(1-\alpha_{t-1})\alpha_t}{\alpha_{t-1}}}\right)\left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right) \\
&= x_t + \sqrt{\alpha_t}\left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1}\right)\left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right).
\end{aligned} \tag{22}$$

The latent difference of Z-Sampling is accumulated as

$$\begin{aligned}
\delta_{\textit{Z-Sampling}} &= \sum_{t=1}^{T}(x_t - \tilde{x}_t)^2 \\
&= \sum_{t=1}^{T} \alpha_t h_t^2 \left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_{t-1})\right)^2 \\
&= \sum_{t=1}^{T} \alpha_t h_t^2 \left(\underbrace{\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_t)}_{\tau_1\textit{:semantic information gain term}} + \underbrace{\epsilon_\theta^t(\tilde{x}_t) - \epsilon_\theta^t(\tilde{x}_{t-1})}_{\tau_2\textit{:approximation error term}}\right)^2.
\end{aligned} \tag{23}$$

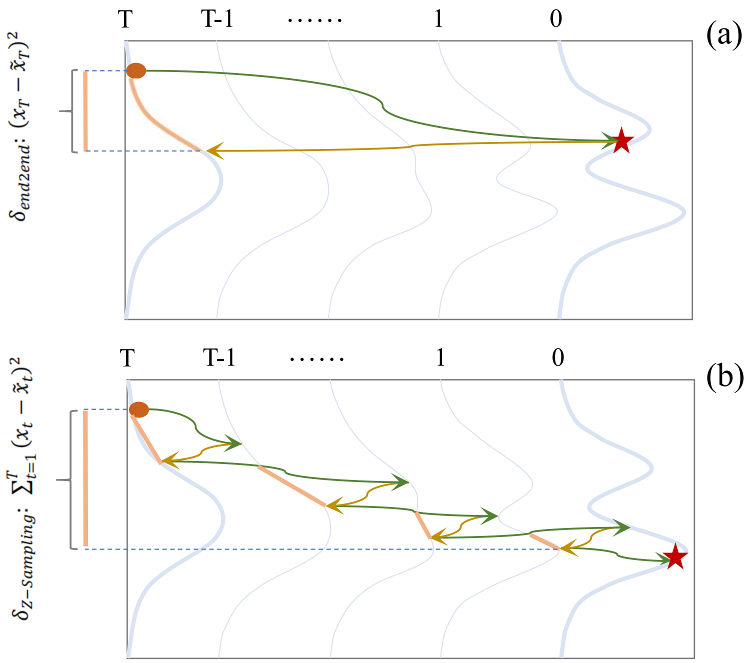

Figure 26: The End-to-End injection risks semantic cancellation across stages, leading to suboptimal results. In contrast, Z-Sampling captures and injects semantic information at each step in a timely manner along the sampling path, resulting in a stronger injection effect.

In Figure 26, we visually represent the effect of equation 21 and equation 23. Z-Sampling clearly injects semantic information at each step in a timely manner, leading to a more pronounced effect and a deeper level of semantic injection.

We note in Equation 24 that $\epsilon_\theta^t(\tilde{x}_t)$ actually represents the denoising result of latent x_t under low guidance γ_2, written this way for consistency with Equation 5. Therefore, the only difference between $\epsilon_\theta^t(\tilde{x}_t)$ and $\epsilon_\theta^t(x_t)$ is the guidance scale: $\epsilon_\theta^t(x_t)$ uses the guidance scale of γ_1, while $\epsilon_\theta^t(\tilde{x}_t)$ uses the guidance scale of γ_2. The latent input to the denoising network is the same for both x_t.

Proof F.3 (Theorem 3) *Excluding the approximation error introduced by inversion algorithm, we can rewrite equation 23 as*

$$\delta_{\textit{Z-Sampling}} = \sum_{t=1}^{T} \alpha_t h_t^2 \left(\epsilon_\theta^t(x_t) - \epsilon_\theta^t(\tilde{x}_t)\right)^2 . \tag{24}$$

Although the step-by-step approach results in x_t and $\tilde{x}_t$ being the same at each timestep t, from equation 5, we note that $\epsilon_\theta^t(x_t)$ and $\epsilon_\theta^t(\tilde{x}_t)$ are obtained under guidance scales γ_1 and γ_2 respectively. Thus, the effect of Z-Sampling is further equivalent as

$$\begin{aligned}
\delta_{\textit{Z-Sampling}} &= \sum_{t=1}^{T} \alpha_t h_t^2 \left((\gamma_1 - \gamma_2)\, u_\theta(x_t, c, t) - (\gamma_1 - \gamma_2)\, u_\theta(x_t, \varnothing, t)\right)^2 \\
&= \sum_{t=1}^{T} \alpha_t h_t^2 \left((\gamma_1 - \gamma_2)\left(u_\theta(x_t, c, t) - u_\theta(x_t, \varnothing, t)\right)\right)^2 \\
&= \sum_{t=1}^{T} \alpha_t h_t^2 \left(\delta_\gamma \left(u_\theta(x_t, c, t) - u_\theta(x_t, \varnothing, t)\right)\right)^2 .
\end{aligned} \tag{25}$$

Here, δ_γ represents the guidance gap between denoising and inversion, i.e., $\gamma_1 - \gamma_2$.

From equation 25, we note that the effectiveness of Z-Sampling primarily depends on:

1. The guidance gap δ_γ, which we can control to regulate the magnitude and intensity of the optimization.
2. The difference between the conditional branch $u_\theta(x_t, c, t)$ and unconditional branch $u_\theta(x_t, \varnothing, t)$, which is determined by the prompt c and the model parameters θ.

As mentioned in the end of Proof F.2, in the absence of inversion approximate errors, the only difference between $\epsilon_\theta^t(x_t)$ and $\epsilon_\theta^t(\tilde{x}_t)$ in Equation 24 is they use the different guidance scale. Therefore, even when $\gamma_2 = 0$, our focus remains on the invariant, which is the difference between the network outputs of the conditional and unconditional branches $u_\theta(x_t, c, t) - u_\theta(x_t, \varnothing, t)$.

G The End-to-End Semantic Injection Algorithm

In this section, we show how to inject semantic information end-to-end as described in Section 3.3.

Algorithm 2 End-to-End Semantic Injection

Input: Denoising Process: Φ, Inversion Process: Ψ, text prompt: c, denoising guidance: γ_1, inversion guidance: γ_2, inference steps: T, zigzag optimization steps: λ
Output: Clean image x_0
Sample Gaussian noise x_T
$x_0 = \phi(x_T|c, \gamma_1)$ #see equation 6
$\tilde{x}_T = \psi(x_0|c, \gamma_2)$ #see equation 7
$x_0 = \phi(\tilde{x}_T|c, \gamma_1)$ #see equation 8
return x_0

H Additional Experiments on Various Guidance Scales

We report more visual cases in Figure 27, showcasing the performance of Z-Sampling in SDXL under different guidance scales γ_1. It can be observed that as the guidance scale increases, the phenomenon of artifacts and oversaturation for standard sampling become more pronounced, while Z-Sampling effectively mitigates these issues. The similar observation also holds in Figure 24 with DreamShaper.

To further investigate the performance improvement with various high CFG scales, we present the conclusive quantitative experimental results of SD 2.1, SDXL, and DreamShaper together in the additional Table 18. We searched the best guidance scales for each model in terms of HPS v2 and present the results. For SDXL/SD2.1, the seached guidance range was set from 3.5 to 25.5, and for DreamShaper-xl-turbo-v2, it was set from 1.5 to 11.5. Note that existing relevant studies commonly do not fine-tune the guidance scale hyperparameter.

The conclusive quantitative results demonstrate that Z-Sampling can significant improve the best performance of all three diffusion models with various choices of the guidance scales. Moreover, the results indicate that the distilled DreamShaper with Z-Sampling can even outperform SDXL, while DreamShaper with standard sampling cannot match SDXL.

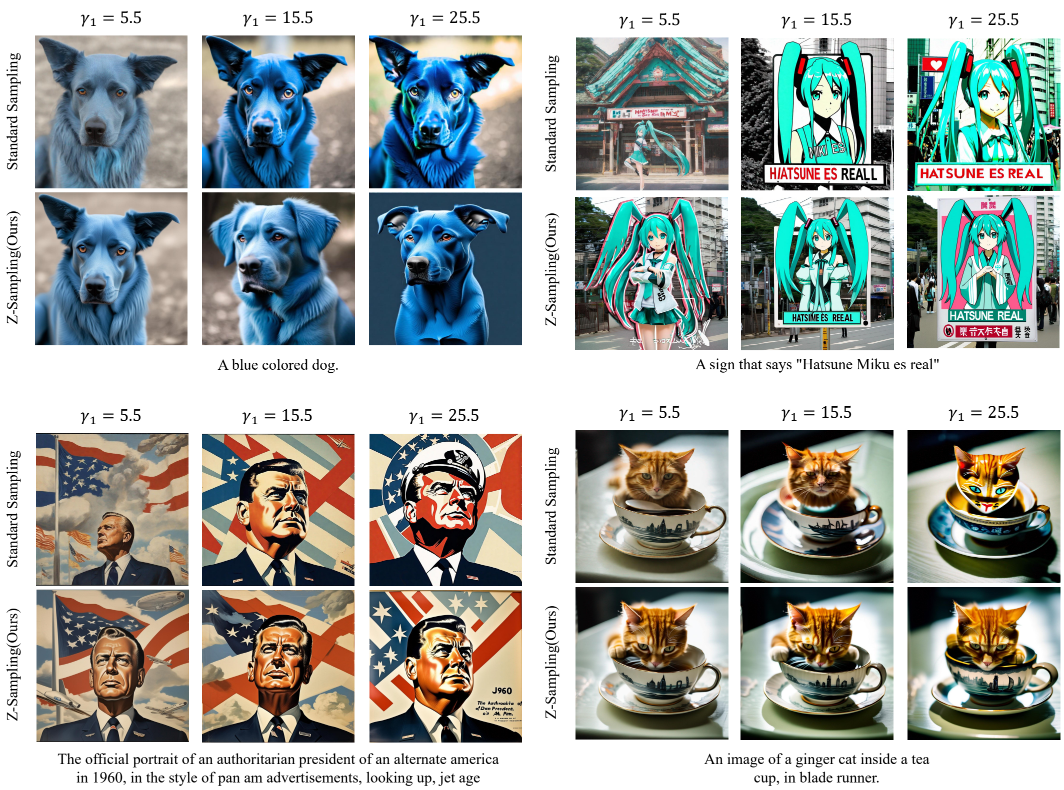

Figure 27: Qualitative comparison under high guidance scales. Standard sampling suffers from supersaturation more significantly than Z-Sampling under the high guidance scales. Model:SDXL.

Table 18: Quantitative comparison of Standard Sampling and Z-Sampling with the best grid searched guidance scale on Pick-a-Pic Dataset. Z-Sampling consistently outperforms across SDXL, SD2.1, and DreamShaper-xl-v2-turbo, indicating significantly better performance, even with grid searched results. Note that previous studies commonly use the defaulted guidance scale instead of fine-tuning it for diffusion models. This further demonstrated that the advantage of Z-Sampling is robust to various settings.

Model	Method	HPS v2 ↑	AES ↑	PickScore ↑	IR ↑	Winning Rate ↑
SD-2.1	Standard	0.2686	5.7014	**20.3946**	0.2387	-
	Z-Sampling(ours)	**0.2729**	**5.7126**	20.3796	**0.2885**	62%
SDXL	Standard	0.3100	6.0985	21.7238	0.7917	-
	Z-Sampling(ours)	**0.3128**	**6.1302**	**21.8477**	**0.7922**	61%
DreamShaper -xl-v2-turbo	Standard	0.3016	5.9898	21.5388	0.6899	-
	Z-Sampling(ours)	**0.3238**	**6.1542**	**22.1025**	**0.9087**	86%