Diffusion Sampling Path Tells More: An Efficient Plug-and-Play Strategy for Sample Filtering

Supplementary Material

6. A posterior-based motivation for ASD

In this appendix, we provide a posterior-based motivation for the Accumulated Score Difference (ASD) defined in Eq. (6). Our goal is not to claim that ASD is exactly equivalent to the pathwise increment of posterior log-probability, but rather to show that it is closely related to how strongly the conditional signal drives a denoising trajectory toward regions of higher posterior support.

Step 1: Score difference as posterior-score gradient. By Bayes' rule,

$$\log p_t(x \mid c) = \log p_t(c \mid x) + \log p_t(x) - \log p(c). \quad (8)$$

Taking the gradient with respect to x and rearranging the terms gives

$$\nabla_x \log p_t(c \mid x) = \nabla_x \log p_t(x \mid c) - \nabla_x \log p_t(x). \quad (9)$$

Under accurate score estimation, the network outputs $S_\theta(x_t; \sigma_t, c)$ and $S_\theta(x_t; \sigma_t, \emptyset)$ approximate the conditional and unconditional score functions, respectively. Therefore,

$$S_\theta(x_t; \sigma_t, c) - S_\theta(x_t; \sigma_t, \emptyset) \approx \nabla_x \log p_t(c \mid x_t). \quad (10)$$

This shows that the score difference corresponds to the gradient of the posterior log-probability, often referred to as the *posterior-score gradient* (or classifier gradient in classifier-guided diffusion). Its norm therefore reflects the strength of the conditional signal acting on the current sample.

Step 2: Pathwise increase in posterior log-probability. For intuition, we first consider a fixed density and suppress the time dependence. For a scalar field $\phi(x)$ and a path Γ from x_T to x_0, the fundamental theorem of line integrals gives

$$\phi(x_0) - \phi(x_T) = \int_\Gamma \nabla \phi(x) \cdot dx. \quad (11)$$

Let $\phi(x) = \log p(c \mid x)$, then

$$\log p(c \mid x_0) - \log p(c \mid x_T) = \int_\Gamma \nabla_x \log p(c \mid x)^\top dx. \quad (12)$$

This identity shows that the increase in posterior log-probability along a trajectory is determined by how well the trajectory aligns with the posterior-score gradient. In the diffusion setting, x_T is close to pure noise, so $\log p(c \mid x_T)$ is typically small, while a desirable sample x_0 should yield a large $\log p(c \mid x_0)$, corresponding to strong semantic alignment with the condition c. Therefore, the integral on the right-hand side provides a natural proxy for evaluating trajectory quality.

Step 3: Connection to ASD. Consider a discrete denoising trajectory $\mathcal{T} = \{x_T, x_{T-1}, \ldots, x_0\}$ and define $\Delta x_t = x_{t-1} - x_t$. Motivated by Eq. (12), we assume that the update direction at each step is approximately aligned with the posterior-score gradient:

$$\Delta x_t \approx \nabla_x \log p(c \mid x_t)\, \Delta_t, \quad (13)$$

where $\Delta_t > 0$ is a scalar step size.

Under this alignment assumption, the line integral can be approximated as

$$\sum_{t=1}^{T} \|\nabla_x \log p_t(c \mid x_t)\|_2^2 \Delta_t = \sum_{t=1}^{T} \|S_\theta(x_t; \sigma_t, c) - S_\theta(x_t; \sigma_t, \emptyset)\|_2^2 \Delta_t. \quad (14)$$

This motivates the ASD statistic

$$\mathcal{E}_\mathcal{T}(c) = \sum_{t=1}^{T} \|S_\theta(x_t; \sigma_t, c) - S_\theta(x_t; \sigma_t, \emptyset)\|_2^2, \quad (15)$$

which serves as an energy-like proxy for the cumulative influence of the conditional signal along the denoising trajectory.

Remark. The above derivation is intended as a heuristic motivation rather than a strict identity. In particular, ASD is not equivalent to the exact pathwise increase in posterior log-probability, since the underlying distributions $p_t(x)$ vary across noise levels and the alignment assumption in Eq. (13) is only approximate. Nevertheless, this formulation provides useful intuition for why accumulated score differences can effectively identify trajectories that remain strongly guided by the condition.

7. Illustration on toy example

In this section, we detail the construction of the 2D toy example and the procedure to quantify the geometric relationship between accumulated score differences (ASD) and sample density. We further verify that this relationship remains stable across varying guidance strengths ω.

We follow the implementation of [19] for both dataset construction and diffusion model training. Each class $\mathbf{c}$ is modeled as a Gaussian mixture $\mathcal{M}_\mathbf{c} = \big(\{\phi_i\}, \{\mu_i\}, \{\mathbf{\Sigma}_i\}\big)$, where ϕ_i, μ_i and $\mathbf{\Sigma}_i$ denote the mixture weight, mean, and 2×2 covariance matrix of component i, respectively. This formulation allows for closed-form computation of ground-truth scores and densities:

$$p_{\text{data}}(x|c) = \sum_{i\in\mathcal{M}_\mathbf{c}} \phi_i \mathcal{N}(x; \mu_i, \mathbf{\Sigma}_i). \tag{16}$$

To construct a tree-like structure, we follow the open-source implementation provided by [19]. Specifically, we begin with a main branch and recursively subdivide it into finer branches. Each branch is represented by 8 anisotropic Gaussian components, and the subdivision is repeated 6 times. All settings, including initialization, branching rules, and Gaussian configurations, strictly follow the open code given by [19].

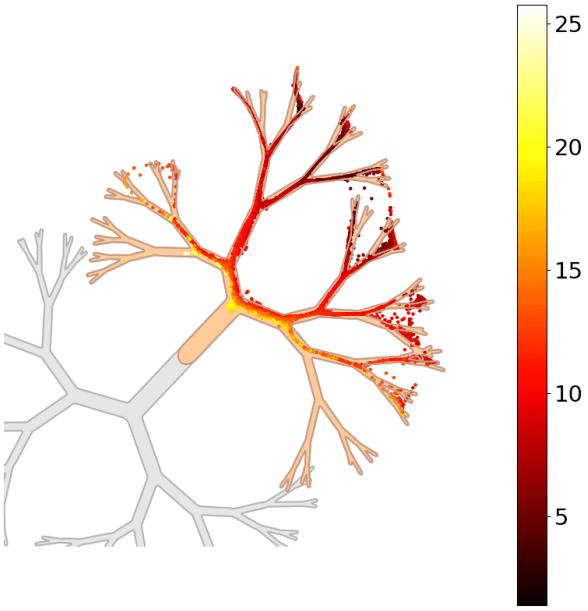
To track the accumulated score differences (ASD), we take advantage of the fact that this is a 2D toy example, which allows us to compute the score difference in closed form using the ℓ_2-norm. Specifically, we directly evaluate $\mathcal{G}_t(c) = \|S_\theta(\mathbf{x}_t; \sigma_t, \mathbf{c}) - S_\theta(\mathbf{x}_t; \sigma_t, \emptyset)\|_2$ following Equation 5 at each timestep and accumulate it across the entire denoising trajectory. Note that the model used here differs from our pre-defined $S_\theta(\mathbf{x}_t; \sigma_t, \mathbf{c})$; thus, we multiply the raw model outputs by σ_t to obtain the correct scaled scores for computing ASD. All results are based on 32-step inference using the default Heun's second-order solver.

Figures 6 present observations with $\omega = 3$. Empirically, we find the relationship between ASD and sample density to be largely invariant to the guidance strength. Samples with high ASD values ($\mathcal{E}_T(c) > \gamma$) predominantly concentrate within the high-density trunk regions, while samples with smaller ASD ($\mathcal{E}_T(c) < \gamma$) primarily populate the low-density branches, with extreme cases ($\mathcal{E}_T(c) \approx 0$) corresponding to degenerate outputs that violate label constraints.

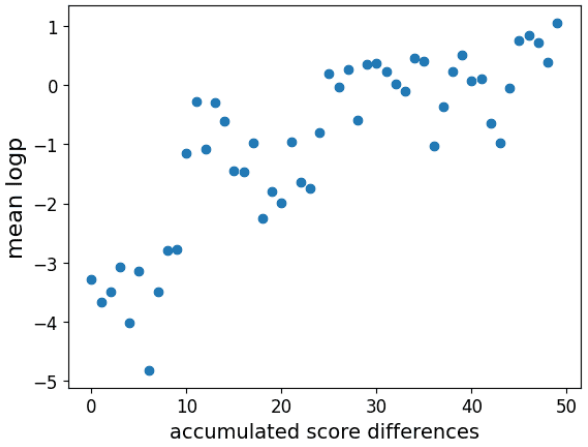
For a quantitative analysis, we divide ASD into 50 bins and plot the mean log-probability density of samples in each bin. Both Figure 6b and Figure 3 reveal a consistent log-linear relationship between local density and ASD across different ω, supporting the use of ASD as a self-supervised signal for filtering.

8. More results on ImageNet

We adopt the pre-trained EDM2-S model provided by [19], using default inference settings with 32 denoising steps and Heun's second-order sampler. The classifier-free guidance weight is set to $\omega = 1.4$. ImageNet samples are generated at 512×512, with denoising carried out in the latent space of shape $4\times64\times64$.



(a) Samples with CFG



(b) Positive relationship

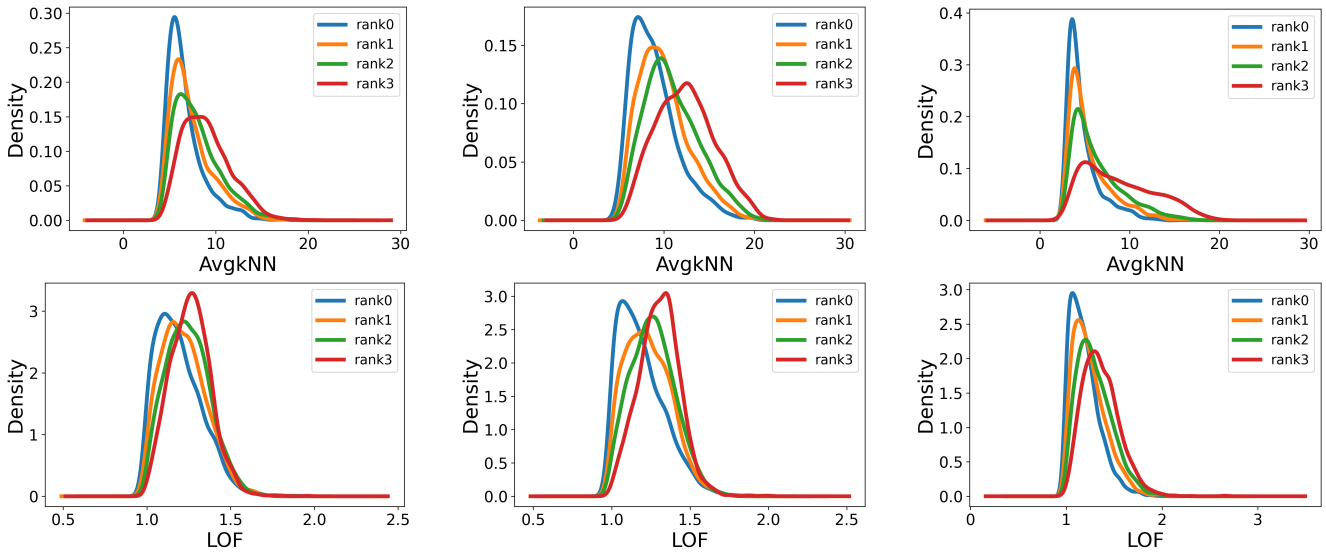
Figure 6. Illustration on toy example with $\omega = 3$.

To compute our metric $\mathcal{G}_t(c)$, we calculate the matrix two-norm of the score difference at each step and average it over the channel dimension. Unlike the 2D toy setting, there is no need to scale the output by σ_t here. All experiments are conducted on a single RTX 4090 GPU.

The following subsections provide additional results: Subsection 8.1 presents extended density visualizations, Subsection 8.2 includes more qualitative comparisons, and Section 8.3 reports quantitative results across baseline methods.

8.1. Manifold density analysis

Figure 7 shows additional density plots across various class labels. Consistent with our observations in the toy setting,



(a) Crib (b) Fountain (c) Parachute

Figure 7. Density estimation curves for generated samples from the classes Crib, Fountain, and Parachute.

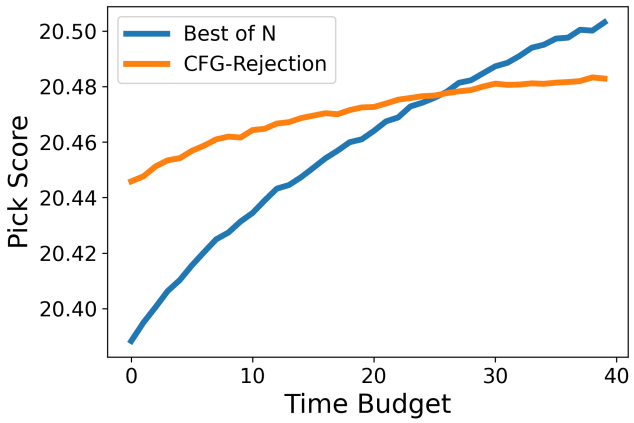


Figure 8. Performance comparison under limited inference budget. CFG-Rejection outperforms Best-of N-method under constrained computational budgets in practical usage.

samples with lower accumulated score differences $\mathcal{E}_T(c)$ tend to exhibit higher AvgkNN and LOF scores, indicating a stronger presence in low-likelihood regions of the data manifold. These results provide further empirical support for the positive correlation between ASD and sample density, reinforcing the foundation of our filtering approach.

8.2. Qualitative comparison

We present additional qualitative results from Figure 9 to Figure 11 , showcasing samples with the highest and lowest accumulated score differences $\mathcal{E}_T(c)$ across various labels. As illustrated, samples with low $\mathcal{E}_T(c)$ often exhibit severe artifacts or semantic inconsistencies, while those with high $\mathcal{E}_T(c)$ maintain high visual fidelity. These results further highlight the effectiveness of our method in filtering out degenerate generations while retaining high-quality outputs.

8.3. Quantitative comparison

We further compare CFG-Rejection ($\tau = 5$) with the Best-of-N strategy under identical time budgets to show the efficiency of our method under stricter constraints. To mitigate selection overfitting, we use Aesthetic Score [28] for selection, then evaluate performance using PickScore and HPSv2 on the selected outputs.

To control inference time, we vary the number of initial candidates in CFG-Rejection from 6,000 to 2,000 (in steps of 100), and adjust Best-of-N accordingly to match the time budget. The x-axis in Figure 8 reflects this normalized computational cost. The comparison metric is the average score of 1,000 selected samples.

As shown in Fig. 8, although Best-of-N improves with more time, it initially underperforms due to the need to fully denoise all candidates before selection. In contrast, CFG-Rejection benefits from early-stage filtering, achieving higher scores in low-time regimes. These findings reinforce the effectiveness and efficiency of our method under realistic computational constraints.

9. Memorization experiment

Following Scenario 1 from [16], we fine-tune SDv2.1 on 10k LAION images to track the similarity to the training set. The model is known to induce memorization. We then use memorization suppression method from [16] to suppress overfitted patterns in the model and conduct the com-

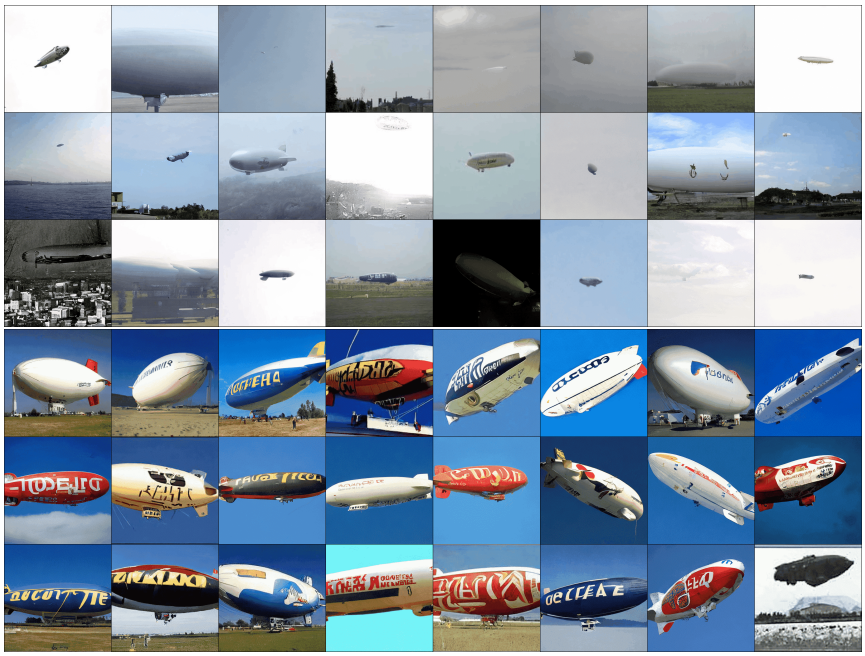


Figure 9. Qualitative comparison of Airship. The top row contains samples with the lowest accumulated score difference (ASD) $\mathcal{E}_T(c)$, where the airships appear faint, occupy minimal space in the image, or are indistinguishable from the sky background—suggesting uncertainty in semantic focus. In contrast, the bottom row showcases high-ASD $\mathcal{E}_T(c)$ samples, where airships are prominently rendered with clear visual traits such as elongated fuselages, tail fins, and gondolas suspended beneath the envelope. This comparison highlights how high-ASD images better align with the semantic concept of the "Airship" class, confirming the effectiveness of ASD as a density-aware signal for filtering.

Figure 10. Qualitative comparison of Crib. Top-row images depict distorted infants in cluttered bedding, lacking structural clarity. Bottom-row samples present recognizable cribs with wooden railings and coherent furniture structure.

Figure 11. Qualitative comparison of Beacon. Top-row images show tiny beacons embedded in varied backgrounds like seascapes and rocky terrain, often difficult to identify. In contrast, bottom-row samples feature prominent, clearly defined beacons with distinctive tower structures.

parisons on this cleaned generation pool. Specifically, we apply opposite guidance during the first 17 denoising steps (the transition point chosen by [16]), and use ASD computed from the remaining 33 steps for filtering (one sample from four generations for each prompt). In Table 4, we report the 95 percentile of similarity scores to reflect worst-case memorization behavior. All the experiments are conducted on a single RTX 4090 GPU.

Table 6. Similarity Scores for CFG-Rejection vs. Random Selection (Overfitted Models)

Method	Mean	Std	75% Percentile	Ratio
Random	0.3735	0.1424	0.4691	0.1974
1-from-2	0.3717	0.1420	0.4665	0.1926
1-from-3	0.3699	0.1415	0.4643	0.1870
1-from-4	0.3690	0.1414	0.4633	0.1859

Table 7. Similarity Scores for CFG-Rejection vs. Random Selection (Memorization-Mitigation Applied)

Method	Mean	Std	75% Percentile	Ratio
Random	0.2302	0.0734	0.2547	0.0099
1-from-2	0.2329	0.0751	0.2584	0.0114
1-from-3	0.2342	0.0759	0.2607	0.0120
1-from-4	0.2348	0.0763	0.2616	0.0118

To further demonstrate that our method does not result in significant memorization, we perform a comprehensive similarity comparison, as shown in Tables 6 and 7. The Mean and Std represent the average and variance of similarity scores, while the 75% percentile highlight the tail behavior. The Ratio indicates the percentage of samples with a similarity score greater than 0.5, serving as an approximation of percentage of potentially memorized samples. We compare CFG-Rejection with varying filtering ratios against random selection in both the overfitted model and memorization-mitigation scenarios, providing both indirect and direct evidence.

As shown in Table 6, despite the presence of memorization in the underlying model, CFG-Rejection does not increase the similarity to training images when compared to random selection. In Table 7, after the application of memorization mitigation, similarity scores across all selection schemes remain very close to random selection. The slight differences observed are within a reasonable range and do not indicate reliance on memorization. Furthermore, the quality of generated samples improves using our method, as indicated by Table 4. These results collectively demonstrate that ASD filtering does not rely on memorized training samples, but rather enhances the overall quality of the generated images by aligning them more closely with the desired conditions.

Table 8. The quantitative results on GenEval. Model: SDXL

	Method	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
guidance = 5	random	0.9714	0.6987	0.4094	0.8493	**0.1121**	0.1942	0.5393
	4 from 20	**0.9813**	0.7639	0.3703	0.8405	0.1013	0.225	0.5470
	4 from 50	0.9781	**0.7677**	0.3250	**0.8617**	0.0875	0.2225	0.5404
	$\tau = 10$	0.9766	0.7197	**0.425**	0.8484	0.09	0.2125	0.5454
	$\tau = 20$	0.9781	0.7589	0.3844	0.8444	0.0975	0.2313	**0.5491**
	$\tau = 30$	**0.9813**	0.7652	0.3813	0.8378	0.10	**0.2238**	0.5482
	$\tau = 40$	**0.9813**	0.7665	0.3703	0.8405	0.1025	0.225	0.5477
guidance = 9	random	0.9828	0.7399	**0.4349**	**0.8759**	**0.120**	0.2213	0.5625
	4 from 20	0.9891	0.7942	0.3672	0.8631	0.10	0.2538	0.5612
	4 from 50	**0.9938**	0.7828	0.3281	0.8590	0.09	0.2525	0.5510
	$\tau = 10$	0.9906	0.7652	0.4297	0.8684	0.0925	0.2588	0.5675
	$\tau = 20$	0.9907	0.7867	0.4031	0.8644	0.1013	**0.2613**	**0.5679**
	$\tau = 30$	0.9907	**0.7993**	0.3828	0.8590	0.1013	0.2538	0.5645
	$\tau = 40$	0.9891	0.7967	0.3734	0.8604	0.1013	0.255	0.5626

Table 9. The quantitative results on DPG-bench. Model: SDXL

	Method	Global	Entity	Attribute	Relation	Other.	Overall↑
guidance = 5	random	85.46	80.86	79.50	**86.34**	62.6	73.54
	4 from 20	83.44	81.95	79.98	85.87	**64.8**	74.58
	4 from 50	84.50	81.98	79.64	85.58	64.4	74.52
	$\tau = 10$	**85.56**	81.13	79.65	85.83	62.6	73.61
	$\tau = 20$	83.43	81.64	79.85	86.08	63.2	74.3
	$\tau = 30$	83.43	81.82	79.78	85.83	63.8	74.42
	$\tau = 40$	83.28	**81.99**	**79.99**	85.79	64.4	**74.66**
guidance = 9	random	**85.46**	82.27	80.4	87.01	65.67	75.16
	4 from 20	83.89	82.84	**81.42**	87.1	65.8	**75.98**
	4 from 50	82.07	**82.92**	81.22	86.46	65.60	75.71
	$\tau = 10$	84.35	82.05	80.87	86.64	64.2	75.06
	$\tau = 20$	83.59	82.61	81.15	86.97	65.8	75.65
	$\tau = 30$	83.59	82.75	81.26	**87.26**	**66.2**	75.85
	$\tau = 40$	83.89	82.80	81.28	87.14	65.6	75.90

10. More results on Geneval and DPG

In this section, we detail the experimental settings for tracking accumulated score differences using SDv1.5 and SDXL, along with quantitative results for SDXL. All image generations are performed on 5 NVIDIA H100 GPUs, while benchmark evaluations are conducted on a single RTX 4090 GPU. For each experiment, filtering is applied with three distinct initial seeds, and the reported results correspond to the average scores.

For SDv1.5, we employ the default PNDM scheduler at a resolution of 512×512. The latent noise dimension is $4\times64\times64$. To track the score difference $\mathcal{G}_t(c)$, we compute the direct difference between the outputs of the conditional and unconditional models, flatten the resulting tensors into a vector, and measure its ℓ_2-norm.

For SDXL, we utilize the default Flow Match Euler Discrete scheduler with a resolution of 1024×1024. The latent noise dimension is $4\times128\times128$. The procedure for tracking $\mathcal{G}_t(c)$ follows that of SDv1.5, with the key distinction being a scaling by σ_t at each step t due to the way noise estimation is incorporated in different sampling schedulers.

10.1. GenEval experiment

The quantitative results on GenEval using SDXL are presented in Table 8. As discussed in Section 4.3, the performance gain of CFG-Rejection with SDXL is less pronounced compared to SDv1.5. We attribute this discrepancy to SDXL's inherent tendency to generate samples con-

centrated in high-density regions of the latent space, which limits the effectiveness of density-based filtering strategies. Notably, when the guidance scale is set to $\omega = 9$, little improvement is observed. In this case, the excessive guidance leads to further concentration in high-density areas, rendering CFG-Rejection largely ineffective.

Furthermore, we observe that the two filtering configurations, retaining 4 out of 20 and 4 out of 50 candidates, yield nearly identical performance. This observation is consistent with the trend shown in Figures 8, where performance improves only marginally with increasing sampling budget.

These results suggest that CFG-Rejection is particularly well suited for two scenarios: (1) when using high-diversity models where generated images are distributed across regions of varying density, and (2) when operating under constrained inference budgets, where evaluating all candidates for Best-of-N selection is computationally prohibitive.

10.2. DPG-Bench experiment

The quantitative results on DPG-Bench using SDXL are shown in Table 9. Similar trends to those observed on the GenEval benchmark can be found here. First, compared to SDv1.5, the performance gain of CFG-Rejection on SDXL is relatively modest. When varying the guidance scale, a greater improvement is observed at $\omega = 5$ compared to $\omega = 9$, suggesting that our method is more effective in high-diversity generation settings.

Second, the performance under two filtering configurations—selecting 4 out of 20 and 4 out of 50 samples—remains nearly identical, indicating that CFG-Rejection is particularly advantageous in scenarios with limited inference budgets, where full evaluation of all candidates is impractical.

Third, we observe that different generation categories respond differently to changes in the filtering threshold τ. For instance, the Attribute category shows consistent improvement as τ increases from 10 to 40, while the Relation category exhibits a non-monotonic trend: performance first improves and then degrades. This intriguing behavior suggests that the semantic signal captured by the ASD metric may vary depending on the specific demands of the generation task. Exploring this direction further could enable more fine-grained control of generation dynamics using intrinsic signals during inference.

11. Text rendering results with Flux

We evaluate our method on the visual text rendering task using FLUX.1-dev. The inference is performed with $T = 28$ steps under the default Flow Match Euler Discrete scheduler and a guidance scale of $\omega = 6$. The procedure for tracking $\mathcal{G}_t(c)$ follows that of SDXL: we compute the difference between the conditional and unconditional model outputs at each step, flatten the resulting tensor into a vector, and compute its ℓ_2-norm. We further scale this value by σ_t at each step t, consistent with the noise estimation mechanism in the sampling scheduler.

As FLUX is trained with guidance distillation, practical deployment requires an additional model evaluation with $\omega = 1$ to obtain the hidden outputs of the conditional and unconditional branches. The score difference is then tracked using these outputs. All generations and filtering processes are run on a single RTX 4090 GPU.

As illustrated from Figures 12 to 14, our method improves the success rate of text rendering. Samples with higher ASD values are more likely to successfully render the intended visual text, whereas low-ASD samples often result in incomplete or entirely missing text. This indicates that the ASD metric provides a meaningful signal for selecting high-quality generations in this task.

Figure 12. Visual text rendering for the prompt ”A beach with shells organized to form the words ’Every grain of sand holds a universe of endless possibilities’”. The top three rows showcase generations with the lowest accumulated score difference (ASD), where many images either omit parts of the phrase or display illegible, fragmented text. In contrast, the bottom three rows depict high-ASD samples, which more reliably render the full phrase with higher clarity and semantic correctness (e.g., well-formed and complete phrases such as ”Every grain of sand holds a universe of endless possibilities”). This example further supports ASD’s role in enhancing alignment with complex visual text prompts.

Figure 13. Visual text rendering for the prompt “A city skyline at sunset with clouds forming the words ’Together we rise, apart we fall. Embrace unity’”. The top three rows display images with the lowest accumulated score difference (ASD), while the bottom three rows present those with the highest ASD. High-ASD generations exhibit clearer and more complete textual formations embedded in the clouds, faithfully rendering the intended message (e.g., ”TOGETHER WE RISE, apart is FALL”), while low-ASD samples often lack visible or coherent text. This again highlights ASD’s predictive utility in selecting faithful visual text renderings.

Figure 14. Visual poster generation for the prompt ”A poster with a title ’My cute pet cat’”. The top half shows samples with low ASD, where the title is often missing, incomplete, or misaligned (e.g., only “My cute” or “pet cat”). The bottom half presents generations with high ASD, which consistently exhibit well-positioned and legible titles alongside semantically aligned visuals. This reinforces the effectiveness of ASD-based filtering in guiding structured text generation in stylized or design-oriented tasks.