Improving Diffusion Generalization with Weak-to-Strong Segmented Guidance

Supplementary Material

Contents

A. Error correction analysis on ImageNet

A.1. Derivation of optimal conditional velocity

Previous work has derived the optimal denoiser $\mathbf{D}(\mathbf{x}_t, t)$ [16] and score-matching objective $\mathbf{s}(\mathbf{x}_t, t)$ [9]. Here, we provide a detailed derivation of the optimal conditional velocity, $\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c})$, given a state $(\mathbf{x}_t, t)$ and a condition $\mathbf{c}$.

We adopt the flow matching (OT) [2, 20, 21] schedule, where $\alpha_t = 1-t$, $\sigma_t = t$, and the state is $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\epsilon$. The corresponding true velocity is $\mathbf{u} = \epsilon - \mathbf{x}_0$.

The optimal conditional velocity $\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c})$ is the function that minimizes the mean-squared error. This is achieved by the conditional expectation of the true velocity, given the current state and condition:

$$\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot|\mathbf{c}), \epsilon}[\epsilon - \mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}] \tag{1}$$

We can simplify this expression. Given $\epsilon = (\mathbf{x}_t - (1-t)\mathbf{x}_0)/t$, the true velocity $\mathbf{u}$ becomes:

$$\mathbf{u} = \epsilon - \mathbf{x}_0 = \frac{\mathbf{x}_t - (1-t)\mathbf{x}_0}{t} - \mathbf{x}_0 = \frac{\mathbf{x}_t - \mathbf{x}_0}{t} \tag{2}$$

Substituting this back into the expectation, and noting that $\mathbf{x}_t$ and t are given:

$$\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot|\mathbf{c})}\left[\frac{\mathbf{x}_t - \mathbf{x}_0}{t} \mid \mathbf{x}_t, t, \mathbf{c}\right] \tag{3}$$

$$= \frac{1}{t}\left(\mathbf{x}_t - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}]\right) \tag{4}$$

The problem thus reduces to finding the posterior mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}]$. We find the posterior distribution $p(\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c})$ using Bayes' rule. Note that the perturbation kernel p_{0t} is independent of $\mathbf{c}$:

$$p(\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}) = \frac{p_{0t}(\mathbf{x}_t \mid \mathbf{x}_0)p(\mathbf{x}_0 \mid \mathbf{c})}{p_t(\mathbf{x}_t \mid \mathbf{c})} \tag{5}$$

We now assume a finite dataset. Let the subset of data points belonging to condition $\mathbf{c}$ be a finite set of N samples, $\{\mathbf{x}_0^i\}_{i=1}^N$. The conditional data distribution $p(\mathbf{x}_0 \mid \mathbf{c})$ can be expressed as a sum of Dirac delta functions:

$$p(\mathbf{x}_0 \mid \mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_0 - \mathbf{x}_0^i) \tag{6}$$

The denominator, $p_t(\mathbf{x}_t \mid \mathbf{c})$, is the conditional marginal probability:

$$p_t(\mathbf{x}_t \mid \mathbf{c}) = \int p_{0t}(\mathbf{x}_t \mid \mathbf{x}_0)p(\mathbf{x}_0 \mid \mathbf{c})\mathrm{d}\mathbf{x}_0 \tag{7}$$

$$= \int \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0, t^2\mathbf{I})\left(\frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_0 - \mathbf{x}_0^i)\right)\mathrm{d}\mathbf{x}_0 \tag{8}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^i, t^2\mathbf{I}) \tag{9}$$

With the numerator and denominator defined, the full posterior $p(\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c})$ is a weighted sum of Dirac deltas:

$$p(\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}) = \frac{\sum_{i=1}^{N}\mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^i, t^2\mathbf{I})\delta(\mathbf{x}_0 - \mathbf{x}_0^i)}{\sum_{j=1}^{N}\mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^j, t^2\mathbf{I})} \tag{10}$$

The posterior mean $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}]$ is therefore the weighted average of the conditional data points $\{\mathbf{x}_0^i\}$:

$$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}] = \int \mathbf{x}_0 p(\mathbf{x}_0 \mid \mathbf{x}_t, t, \mathbf{c}) \mathrm{d}\mathbf{x}_0 \tag{11}$$

$$= \frac{\sum_{i=1}^N \mathbf{x}_0^i \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^i, t^2\mathbf{I})}{\sum_{j=1}^N \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^j, t^2\mathbf{I})} \tag{12}$$

Finally, we substitute this posterior mean (Eq. 12) back into our velocity expression (Eq. 4):

$$\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c}) = \frac{1}{t}\left(\mathbf{x}_t - \frac{\sum_{i=1}^N \mathbf{x}_0^i \mathcal{N}(\cdot)}{\sum_{j=1}^N \mathcal{N}(\cdot)}\right) \tag{13}$$

$$= \frac{1}{t}\left(\frac{\mathbf{x}_t \sum_{j=1}^N \mathcal{N}(\cdot) - \sum_{i=1}^N \mathbf{x}_0^i \mathcal{N}(\cdot)}{\sum_{j=1}^N \mathcal{N}(\cdot)}\right) \tag{14}$$

$$= \frac{\sum_{i=1}^N (\mathbf{x}_t - \mathbf{x}_0^i)\mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^i, t^2\mathbf{I})}{t\sum_{j=1}^N \mathcal{N}(\mathbf{x}_t; (1-t)\mathbf{x}_0^j, t^2\mathbf{I})} \tag{15}$$

Given the set of data points $\{\mathbf{x}_0^i\}_{i=1}^N$ corresponding to condition $\mathbf{c}$, this equation provides the exact velocity target.

A.2. Experiments configurations

Pretraind model configuration. For the comparison of CFG and AG against optimal velocity, we pre-trained a SiT-B/2 model for 400k iterations to serve as the strong model. We also pre-trained a SiT-S/2 model for 100k iterations to serve as the weak model required for AutoGuidance (AG) [17].

Inception distance between guided velocity and optimal velocity. For the inception distance [11] analysis, we use the standard extrapolation guidance formula $\mathbf{v}_w = \mathbf{v}_{\text{cond}} + w \cdot (\mathbf{v}_{\text{cond}} - \mathbf{v}_{\text{weak}})$. For CFG [12], we use unconditional output $\mathbf{v}(\mathbf{x}_t, t, \emptyset)$ as $\mathbf{v}_{\text{weak}}$. For AG [17], we use the condition-aligned output $\tilde{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c})$ from SiT-S/2 as $\mathbf{v}_{\text{weak}}$. We tested extrapolation scales of $w \in \{1.0, 1.2, 1.4, 1.6\}$. The $w = 1.0$ case represents the unguided conditional output ($\mathbf{v}_w = \mathbf{v}_{\text{cond}}$) and serves as our baseline for comparison. For a given class $\mathbf{c}$, we calculate the following distance objective:

$$\mathbb{E}_{t,\mathbf{x}_t} \|\dot{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}_w(\mathbf{x}_t, t, \mathbf{c})\|_2^2 \tag{16}$$

We sample 100000 samples to calculate the corresponding state $(\mathbf{x}_t, t)$ given timestamp t.

B. Toy experiment implementation

B.1. Network architectures

For all 2D toy experiments we train a class-conditional diffusion model with a velocity parameterization

$$\mathbf{v} : \mathbb{R}^2 \times [0,1] \times \mathcal{C} \to \mathbb{R}^2, \qquad (\mathbf{x}_t, t, \mathbf{c}) \mapsto \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}), \tag{17}$$

where $\mathbf{x}_t \in \mathbb{R}^2$ is the noisy state, $t \in [0,1]$ is the continuous time index, and $\mathbf{c} \in \mathcal{C} = \{1, \ldots, \text{CLS}\}$ is the class label.

For network architectural design, we follow the score network in [17] but apply the following modification to enable conditional and unconditional prediction on the same model architecture. We introduce a learnable class-embedding table

$$E : \mathcal{C} \cup \{\emptyset\} \to \mathbb{R}^{d_c}, \qquad \mathbf{e}_c = E(\mathbf{c}), \tag{18}$$

where $\mathbf{c} = \emptyset$ denotes the unconditional (null) condition and d_c is the embedding dimension. Let $\text{Enc}(\mathbf{x}_t, t) \in \mathbb{R}^{d_h}$ denote the standard feature encoding of the noisy state and time as in [17]. We then form the input to the backbone as

$$\mathbf{h}_0 = \left[\text{Enc}(\mathbf{x}_t, t)\,;\, \mathbf{e}_c\right] \in \mathbb{R}^{d_h + d_c}, \tag{19}$$

and use the same network weights for all $\mathbf{c} \in \mathcal{C} \cup \{\emptyset\}$. This design allows us to obtain both

$$\mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) \quad \text{and} \quad \mathbf{v}(\mathbf{x}_t, t, \emptyset) \tag{20}$$

from a single model, thereby enabling classifier-free guidance [12] and our segmented guidance without changing the backbone.

Training follows the flow-matching parameterization adopted in the main paper. We sample $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{c})$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, and construct the noisy state via the stochastic interpolant

$$\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\epsilon, \qquad t \sim p(t), \tag{21}$$

where $p(t)$ is the lognormal time sampling distribution used throughout the paper. The network is trained with the standard velocity regression objective

$$\mathcal{L}_{\text{toy}}(\theta) = \mathbb{E}_{t,\mathbf{c},\mathbf{x}_0,\epsilon}\left[\left\|\mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) - (\epsilon - \mathbf{x}_0)\right\|_2^2\right]. \tag{22}$$

B.2. Construction of the toy dataset

Our toy dataset construction shares the principle of using a mixture of Gaussians as the building block as in [17], but explicitly exposes the *granularity of the condition* via the number of classes and the recursive depth. The dataset returns a Gaussian mixture distribution constructed from a collection of leaf- and branch-like components.

We denote the class set by

$$\mathcal{C} = \{1, \ldots, \text{CLS}\}, \tag{23}$$

corresponding to the `num_classes` argument. Internally, the function assembles a list of Gaussian components indexed by $i = 1, \ldots, K$, each with a weight $\phi_i > 0$, mean $\boldsymbol{\mu}_i \in \mathbb{R}^2$, covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{2\times 2}$, and a discrete class label $c_i \in \mathcal{C} \cup \{c_{\text{base}}\}$. After selecting a subset of labels through the `classes` argument, the final distribution is the normalized Gaussian mixture

$$p_{\text{data}}(\mathbf{x}_0, \mathbf{c}) = p(\mathbf{c})\, p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{c}), \qquad p(\mathbf{c}) = \frac{1}{|\mathcal{C}_{\text{sel}}|}, \tag{24}$$

where $\mathcal{C}_{\text{sel}} \subseteq \mathcal{C}$ is the set of selected class labels and

$$p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{c}) = \sum_{i \in I_c} \pi_i^{(\mathbf{c})} \mathcal{N}\big(\mathbf{x}_0;\, \boldsymbol{\mu}_i,\, \boldsymbol{\Sigma}_i\big), \qquad \sum_{i \in I_c} \pi_i^{(\mathbf{c})} = 1. \tag{25}$$

Here $I_c = \{i : c_i = \mathbf{c}\}$ collects all components assigned to class $\mathbf{c}$, and the mixture weights $\pi_i^{(\mathbf{c})}$ are obtained by normalizing the raw branch weights ϕ_i within the class.

The geometry of the mixture components is determined by a recursive branching construction. A single "main branch" of length

$$L_{\text{main}} = 0.4\,\big(1 + 0.1 \cdot \texttt{num\_classes}\big) \tag{26}$$

is grown from a base point $\mathbf{x}_{\text{base}} \in \mathbb{R}^2$ with initial angle $\alpha_{\text{main}} \approx 85^\circ$. This branch is split into `num_classes` segments, and each segment serves as the attachment point for a class-specific subbranch.

Each subbranch is generated by a recursive procedure with maximum depth, branching factor, and curvature. At recursion depth $d \in \{0, \ldots, \texttt{max\_depth} - 1\}$, a subbranch located at position $\mathbf{p}^{(d)}$ with direction $\mathbf{u}^{(d)} \in \mathbb{R}^2$ and overall size $s^{(d)}$ generates a sequence of Gaussian components

$$\boldsymbol{\mu}_i = \big(\mathbf{p}^{(d)} + \lambda\, \mathbf{u}^{(d)}\big) \odot \mathbf{s}, \qquad \boldsymbol{\Sigma}_i = \mathbf{R}^{(d)}\, \mathbf{D}^{(d)}\, \mathbf{R}^{(d)\top}, \tag{27}$$

for several values of $\lambda \in (0, 1)$ along the branch; here $\mathbf{s} = \texttt{scale} \in \mathbb{R}^2$ scales the coordinates, $\mathbf{R}^{(d)}$ is the 2×2 rotation induced by the current branch angle, and $\mathbf{D}^{(d)}$ is a diagonal matrix encoding the anisotropic thickness of the branch. The raw weight of each component is proportional to a depth-dependent factor $\phi_i \propto s^{(d)}(0.6)^d$, which causes branch segments closer to the root to receive higher total mass.

B.3. Guidance baselines and configurations

We evaluate four guidance configurations on the above toy datasets: unguided sampling, condition-dependent guidance (CDG, instantiated by CFG [12]), condition-agnostic guidance (CAG, instantiated by AG [17]), and our proposed segmented guidance (SGG). All methods act on the same strong model $\mathbf{v}(\mathbf{x}_t, t, \mathbf{c})$, and differs only in how they construct the weak signal $\mathbf{v}_{\text{weak}}$ within the weak-to-strong extrapolation.

Unguided. We use the strong model directly,

$$\mathbf{v}^{\text{ung}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}), \tag{28}$$

corresponding to $w = 1$.

CDG: CFG. Here the weak signal is the unconditional prediction $\mathbf{v}_{\text{weak}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \emptyset)$ and the strong signal is the class-conditional prediction $\mathbf{v}_{\text{strong}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \mathbf{c})$. This yields the usual classifier-free guidance form

$$\mathbf{v}_w^{\text{CFG}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) + (w - 1)\big(\mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}(\mathbf{x}_t, t, \emptyset)\big). \tag{29}$$

CAG: AG Following autoguidance [17], we construct a weaker but condition-aligned model $\tilde{\mathbf{v}}_\theta(\mathbf{x}_t, t, \mathbf{c})$ by reducing capacity or early stopping. In this case the weak signal is $\mathbf{v}_{\text{weak}}(\mathbf{x}_t, t, \mathbf{c}) = \tilde{\mathbf{v}}_\theta(\mathbf{x}_t, t, \mathbf{c})$, which leads to

$$\mathbf{v}_w^{\text{AG}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) + (w - 1)\big(\mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) - \tilde{\mathbf{v}}(\mathbf{x}_t, t, \mathbf{c})\big). \tag{30}$$

Segmented guidance (SGG, ours). SGG uses a time-dependent segmentation between CDG and CAG. For a switching time $\tau \in (0, 1)$ we define

$$\mathbf{g}(\mathbf{x}_t, t, \mathbf{c}) = \begin{cases} \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}(\mathbf{x}_t, t, \emptyset), & t \geq \tau, \\ \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) - \tilde{\mathbf{v}}_\theta(\mathbf{x}_t, t, \mathbf{c}), & t < \tau, \end{cases} \tag{31}$$

and set

$$\mathbf{v}_w^{\text{SGG}}(\mathbf{x}_t, t, \mathbf{c}) = \mathbf{v}(\mathbf{x}_t, t, \mathbf{c}) + (w - 1)\, \mathbf{g}(\mathbf{x}_t, t, \mathbf{c}). \tag{32}$$

To simulate the interplay between condition granularity and model fitting capacity, we vary the tuple

$$(\text{CLS}, \text{Depth}, B)$$

where CLS is the number of classes, Depth is the maximum recursion depth and B is the number of branches per split, together with the training budget T of the strong model. We consider three representative configurations:

- **Config A (blurry condition, complex in-class).** This regime uses a small number of classes but a deep recursive structure,

  $$\text{CLS} = 4, \quad \text{Depth} = 3, \quad B = 2, \quad T = 2^{15},$$

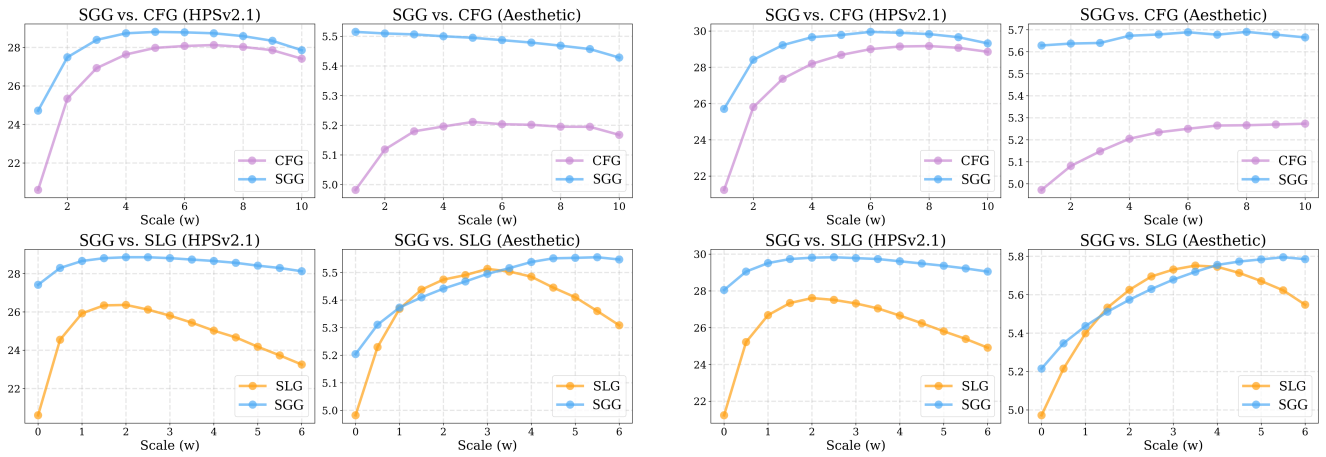
  which yields *blurry* conditions and highly intricate within-class manifolds (well-fitted but hard to disambiguate at the label level).
- **Config B (sharp condition, simple in-class).** This regime uses many classes but a shallow recursive structure,

  $$\text{CLS} = 24, \quad \text{Depth} = 1, \quad B = 2, \quad T = 2^{12},$$

  leading to *sharp* conditions with relatively simple manifolds that are harder to fit under the limited training budget.
- **Config C (intermediate, realistic regime).** This regime interpolates between the above two,

  $$\text{CLS} = 12, \quad \text{Depth} = 2, \quad B = 2, \quad T = 2^{15},$$

  producing moderately complex intra-class structure together with non-trivial conditioning. The difficulty of this task is relatively higher than Config A and B, attempting to project to realistic scenarios.



(a) LAION-5B-1K (b) MSCOCO-1K

Figure 1. Quantitative comparison of guidance scale (w) for CFG and SLG with SD3.5 on LAION-5B-1K and MSCOCO-1K datasets.

Setting	T_{main}	T_{weak}	τ	w_{cfg}	w_{ag}
Config A	2^{15}	2^{11}	0.5	2.0	2.0
Config B	2^{12}	2^{10}	0.1	2.0	2.0
Config C	2^{15}	2^{11}	0.3	2.0	2.0

Table 1. Hyperparameter settings for toy experimentation.

Configurations across settings. Here we provide the full setting on toy experiments across Configurations and hyperparameters, as illustrated in Table 1.

Limitations across dimensionality. Toy examples are powerful for visualizing algorithmic behavior at the distribution level, rather than relying solely on aggregate quantitative metrics [3, 6, 7, 17]. However, there is an inherent gap between 2D toy class-conditional tasks, image class-conditional tasks, and image prompt-conditional (text-to-image) tasks, and phenomena observed in the 2D plane are not guaranteed to transfer to high-dimensional image space with 100% accuracy [29, 33]. Our inductive bias in this work is to isolate the interplay between *condition granularity* and *fitting capacity* as the factors influencing the behavior of CDG and CAG. The 2D toy results should therefore be viewed as qualitative insight into these mechanisms, providing explanations for real image-generation setups.

C. Inference implementation and ablations

C.1. Inference time settings

Pretrained models and baseline methods selection. For pre-trained model, we use the SD3-Medium and SD3.5-Medium as base models [4]. We use MS-COCO-1k [19] subset and LAION-1k [27] randomly selected subset for prompt instantiation. We compare our method against several baselines, including standard conditional generation (no guidance), CFG [12], and Skip-Layer Guidance (SLG). We also include comparisons to recent advanced guidance variants, such as S^2-Guidance [3], Guidance Interval [18], CFG+SLG [14], CFG-Zero* [5] and Rectified-CFG++ [25]. We use the standard 28 inference steps throughout experiments. All methods are evaluated using HPSv2.1 Score [32] and Aesthetic Score [26]. We select standard CFG [12] as CDG and SLG [14] as CAG in SGG implementations.

Hyperparameter settings. For standard CFG [12], we performed a grid search for the guidance scale w in the range $[1.0, 9.0]$ with an interval of 0.5, selecting the optimal value of $w = 5$. For Guidance Interval [18], we searched for the optimal interval t with a step of 0.1, finding that removing guidance for the 20% of timestamps closest to the data ($t < 0.2$) yielded the best results. For S^2-Guidance [3], CFG+SLG [14], CFG-Zero* [5], and Rectified-CFG++ [25], we adhered to the recommended

Method	Avg	Aesthetic	Overall Cons.	Imaging Qual.	Subject Cons.	Dynamic Deg.	Motion Smooth.
CFG	<u>0.6877</u>	<u>0.6003</u>	**0.2317**	0.6553	0.9391	<u>0.7222</u>	0.9776
SLG	0.6809	0.5822	0.2026	<u>0.6597</u>	**0.9504**	0.7083	**0.9820**
SGG	**0.7001**	**0.6107**	<u>0.2314</u>	**0.6624**	<u>0.9402</u>	**0.7778**	<u>0.9785</u>

Table 2. Comparison on WAN-1.3B [31] with CFG, SLG and SGG (Ours). Best results are **bolded**, and second-best results are <u>underlined</u>.

hyperparameter settings from their respective papers. For our method, SGG, we set the segmentation timestamp $\tau = 12/28$ ($t_m = 0.69$, SD3.5), $\tau = 16/28$ ($t_m = 0.8$, SD3) and use a scale of $w = 5$ for the CDG (CFG) component and $w = 3$ for the CAG (SLG) component for both models. The skipping layers are the default setting in vanilla SLG with 7,8,9.

C.2. Ablations on inference guidance scale

Besides ablations on the segmentation timestamp τ provided in the main paper, here we provide an additional ablation on the inference-time guidance scale w. We fixed the segmented timestep $\tau = 0.5$ and ablate the guidance scale of CFG and SLG. As illustrated in Figs. 1a and 1b, this analysis highlights a notable trade-off: CFG excels at semantic adherence (measured by HPSv2.1), but its aesthetic scores are comparatively low. Conversely, SLG produces high aesthetic quality but remains less competitive on HPSv2.1. Our method, SGG, successfully synergizes these two, achieving strong, comparable results across both metrics.

C.3. More metrics on condition-adherence

CLIPScore and GenEval. Besides HPSv2.1 [32], we additionally report CLIPScore [10] and GenEval [8] on MS-COCO-1K and LAION-5B-1K for SD3.5-medium. As shown in Table 4, SGG improves Aesthetic while remaining competitive on condition-based metrics.

C.4. Ablations and guidance schedules

Guidance schedules. SGG is orthogonal to scalar guidance scheduling $w(t)$ [23]: it changes the guidance *family* across noise regimes and can be combined with standard schedules. We include linear $w(t)$ variants for CFG and SGG in Table 4, with mild early-time clamping (e.g., starting from a non-trivial scale), increasing schedules do not weaken conditioning.

More ablations. We test (i) swapping CDG/CAG orders, (ii) removing CDG (CFG) from SGG, and (iii) removing CAG (SLG) from SGG. All variants are inferior to standard SGG in Table 4, supporting the intended regime assignment.

C.5. Extension to video generation

To further evaluate the applicability of Segmented Guidance (SGG) principle across modalities, we extend our experiments to video generation using the `Wan2.1-1.3B` [31] model on the subset of VBench [13] prompts corresponding to the metrics. For inference configuration, we adhere to the default setting with 50 sampling steps and 5.0 CFG scale. For SGG, we set the segmented timestamp $\tau = 25$, and SLG scale 3.0. We selected 6 metrics and calculate the average score. The quantitative results in Table 2 demonstrate that SGG manages to generate videos with better Aesthetic and imaging quality, while also remain competitive on physical plausibility.

D. Training implementation and analysis

D.1. Training time settings

Models and metrics selection. We conduct training evaluation mainly on SiT-B/2 model [22] due to computational constraints. We use lognormal-timestep sampling throughout all experiments to boost convergence, following [28]. We perform experiments in both unconditional and conditional settings. CAG methods are applied in both settings, whereas CDG method is naturally applied only in conditional training. For the conditional setting. All models are trained for 400k iterations. The sampling configuration is SDE Euler-Maruyama sampler with steps=250. We report the FID, sFID and Inception Score for all methods.

	Conditional					Unconditional		
Parameter	Baseline	AG	BR	MG	SGG	Baseline	AG	BR
Training configuration								
Generation Type	Cond	Cond	Cond	Cond	Cond	Uncond	Uncond	Uncond
Batch-size	256	256	256	256	256	256	256	256
Num. GPUs (A100)	4	4	4	4	4	4	4	4
Noise Schedules	OT	OT	OT	OT	OT	OT	OT	OT
LR	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
W2S guidance								
Guidance Scale w	/	0.3	0.3	0.5	0.6 / 0.5 (REPA)	/	0.3	0.3
Inferior model	/	SiT-S/2 (T/4)	/	/	/	/	SiT-S/2 (T/4)	/
Output layers	/	/	4 (/12)	/	4 (/12)	/	/	4 (/12)
Total params	137.8M	137.8 (+39.5)M	139.0M	137.8M	139.0M	137.8M	137.8 (+39.5)M	139.0M
time/it	1.00	1.27	1.02	1.23	1.22	1.00	1.26	1.02
Segmented stamp τ	/	/	/	/	0.2	/	/	/

Table 3. Training information for W2S guidance experiments. All models are trained on SiT-B/2. For AG, we train a separate weak model with T/4 iterations, where T is the strong model's iteration.

Method	MS-COCO-1K				LAION-5B-1K			
	HPSv2.1	Aesthetic	CLIP	GenEval	HPSv2.1	Aesthetic	CLIP	GenEval
CFG	29.219	5.279	26.316	0.628	28.234	5.203	27.521	**0.674**
SLG	27.295	5.714	25.145	0.507	26.050	5.512	25.440	0.523
CFG+SLG	28.931	5.678	26.485	0.598	27.246	5.421	27.154	0.638
CFG (linear schedule)	28.833	5.261	26.351	0.633	27.954	5.220	27.777	0.665
SGG (swap orders)	27.393	5.366	25.523	0.534	26.270	5.311	26.283	0.579
SGG (w/o CDG (CFG))	26.050	5.685	24.630	0.497	25.195	5.516	24.800	0.515
SGG (w/o CAG (SLG))	27.832	5.207	26.112	0.602	27.173	5.178	27.422	0.647
SGG	**29.736**	5.717	**26.713**	0.632	**28.687**	5.518	27.649	0.668
SGG (linear schedule)	29.712	**5.752**	26.595	**0.637**	28.564	**5.525**	**27.783**	0.672

Table 4. SD3.5-medium on MS-COCO-1K and LAION-5B-1K

Implementation details of training W2S variants. Here we summarize the training-time implementation of AG, BR, CFG (our reimplementation of MG), and SGG. The objective of the weak model in all variants shares the same base regression target $\mathbf{u} = \epsilon - \mathbf{x}_0$ and differs only in how the weak prediction and the W2S-modified strong target are constructed. Further hyperparameter choices and information are listed in Table 3.

AG. For autoguidance [17], we use a separate weak model $\mathbf{v}_{\theta_w}$ with a smaller backbone (SiT-S/2) and a strong model $\mathbf{v}_\theta$ (SiT-B/2). The weak model is updated once every 4 updates of the strong model.

BR. In BR, the weak prediction is implemented as a shallow branch head $\mathbf{v}_\theta^{\text{br}}(\mathbf{x}_t, t, \mathbf{c})$ (same architecture of `FinalLayer()`) that taps into intermediate features of the same transformer, while the final head $\mathbf{v}_\theta^{\text{full}}(\mathbf{x}_t, t, \mathbf{c})$ serves as the strong output. The extra computational overhead of this model is negligible (2%, *i.e.*, time/it = 1.02)

MG (CFG reimplementation). For MG/CFG, the weak prediction is provided by the unconditional branch $\mathbf{v}_\theta(\mathbf{x}_t, t, \emptyset)$ of the same model, while $\mathbf{v}_\theta(\mathbf{x}_t, t, c)$ is the strong (conditional) output. We keep the default `DROPOUT` rate 0.1 to train the unconditional model, as CFG [12]. This is equivalent to Model Guidance [30] at the parameterization level.

SGG. SGG combines a condition-agnostic weak signal (BR) and a condition-dependent weak signal (CFG) through a time-dependent switch. With segmented timestamp set to $\tau = 0.2$. (Inspired by the inference time setting of [34] on ImageNet, we also apply guidance interval [18] from $[0.8, 0.2]$ to avoid extreme high noise level experiments for SGG) The overall architecture is identical to BR while keeping the conditional/unconditional training style to create CFG signal. The extra parameter overhead is 0.8% compared to vanilla SiT model. Further details can be referred in Table 3.

D.2. Training instability of SLG

Compared to AG, BR, and MG, we observe that applying Skip Layer Guidance (SLG) during training from scratch exhibits degradation. We attribute this to the high variance of the synthetic weak signal generated by layer perturbation when applied to an *unconverged* model. To address this, we use a warm-up phase utilizing pure regression loss. As illustrated in Fig. 2, extending this warm-up period improves performance, eventually surpassing the baseline at 100k and 200k iterations, likely by ensuring the model is robust enough to provide a stable weak signal. Despite the marginal performance gains, the limitations of this approach are more obvious. Pure SLG [14] necessitates an additional forward pass similar to CFG [12], yet the resulting weak signal is often inferior to the unconditional output. Furthermore, tuning the warm-up hyperparameter becomes computationally prohibitive when scaling to larger tasks. Given these unfavorable trade-offs, we thus exclude SLG from our proposed training framework.

However, for well-trained models at scale [4], the utility of SLG becomes apparent. Training a separate inferior model for AutoGuidance [17] is often impractical for large-scale architectures like Stable Diffusion [4, 24]. In these scenarios, where the primary model is sufficiently robust, self-degradation techniques like SLG provide an efficient mechanism for constructing the Condition-Agnostic Guidance (CAG) signal [1, 14].

D.3. Ablations on training guidance scale

Ablations on segmented timestamp τ of SGG is provided in the main paper, here we provide more ablations on the selection of training time guidance scale w on AG and BR variants on both conditional and unconditional setting, as illustrated in Table 5.

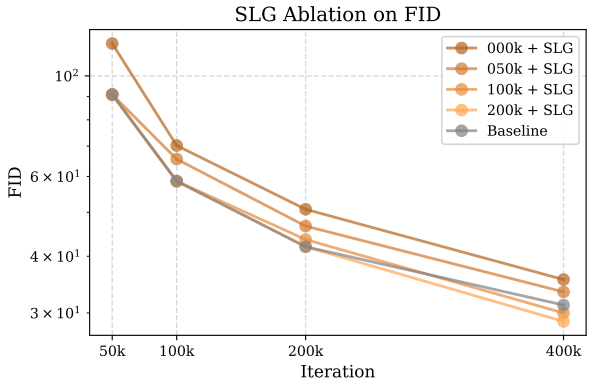


Figure 2. SLG in training, applied in different iterations.

E. Discussion

The generalization trade-off. Theoretically, a velocity predictor $\hat{\mathbf{v}}$ that perfectly minimizes the objective is capable of faithfully reconstructing training data points, a state characterized as memorization [9]. In practice, however, network inductive biases and inevitable approximation errors prevent this, instead enabling the model to generalize to unseen data [15]. Yet, when scaled to complex text-to-image tasks [4, 24], these accumulated errors often cause the unguided generation trajectory to diverge from the perceptually acceptable manifold—a deviation that often persists regardless of the number of sampling steps. Consequently, guidance techniques are required to steer the trajectory back toward perceptually acceptable regions, albeit at the cost of additional computation (*e.g.*, computing an extra weak signal). Thus, the generalization capability of diffusion models presents trade-offs: it is simultaneously enabled by, yet suffers from, the approximation errors accumulated across sampling steps.

More intuition on the proposed method. CDG derives its guidance signal from an *external* semantic discrepancy (i.e., $\mathbf{c}$ vs. $\emptyset$), and thus primarily steers global content such as semantics, coarse structure, and layout. These attributes are largely determined at earlier denoising stages, where the model establishes low-frequency components of the sample. In contrast, CAG is driven by the model's *internal* prediction error under the condition, making its signal inherently condition-aligned and more effective for intra-class refinement, including local details and texture that emerge in later timesteps (high-frequency components) [17]. Consequently, SGG adopts a natural division of labor: it applies CDG in the high-noise regime to quickly locate the correct conditional manifold, and then switches to CAG in the low-noise regime to refine fine-grained details while maintaining prompt consistency.

		Conditional			Unconditional		
Method	Guidance	FID ↓	sFID ↓	Inception Score ↑	FID ↓	sFID ↓	Inception Score ↑
Baseline	$w = 0.0$	31.22	6.41	49.59	61.27	7.00	17.33
BR	$w = 0.2$	18.33	4.71	70.45	46.37	4.93	20.23
	$w = 0.3$	16.02	5.13	76.21	43.25	5.11	20.66
	$w = 0.4$	15.09	7.64	80.29	42.13	7.59	20.43
AG	$w = 0.2$	18.71	4.97	73.30	50.97	5.52	19.40
	$w = 0.3$	13.96	4.68	88.35	45.97	4.94	20.32
	$w = 0.4$	10.91	5.40	102.89	42.44	5.31	21.01

Table 5. Ablation study on guidance scale (w) for W2S training methods (BR and AG) in both conditional and unconditional settings. All models are SiT-B/2 trained on ImageNet 256x256.

F. More qualitative results.

F.1. Qualitative results on text-to-video models

CFG

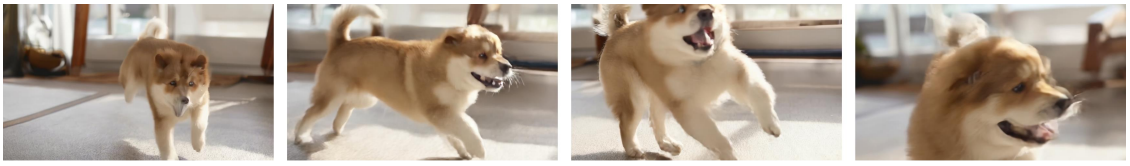
SGG

A high-speed, cinematic drone shot following a sleek, silver peregrine falcon as it weaves and dives through a narrow canyon.

CFG

SGG

Anime style girl with pink hair running through a field of sunflowers, bright blue sky, wind blowing hair, Studio Ghibli artistic style.



CFG

SGG

A playful puppy chasing its tail in a sun-drenched living room, blurry motion, happy energy, cute animal behavior.

Figure 3. Qualitative comparison of CFG and SGG (Ours) on video generation

F.2. Qualitative results on text-to-image models

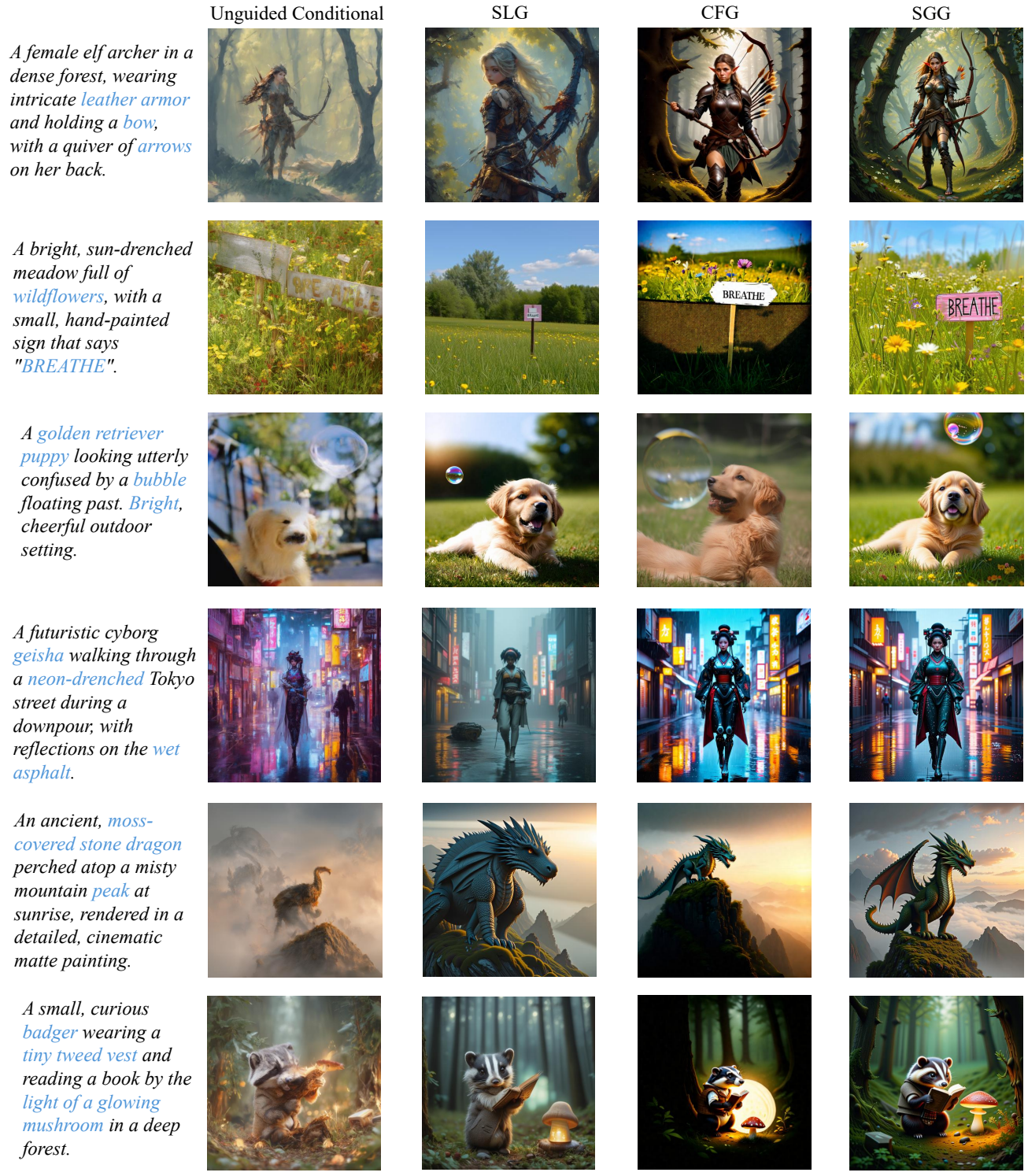


Figure 4. Qualitative Comparison between Unguided Conditional, CFG, SLG and SGG (Ours) (1/2)

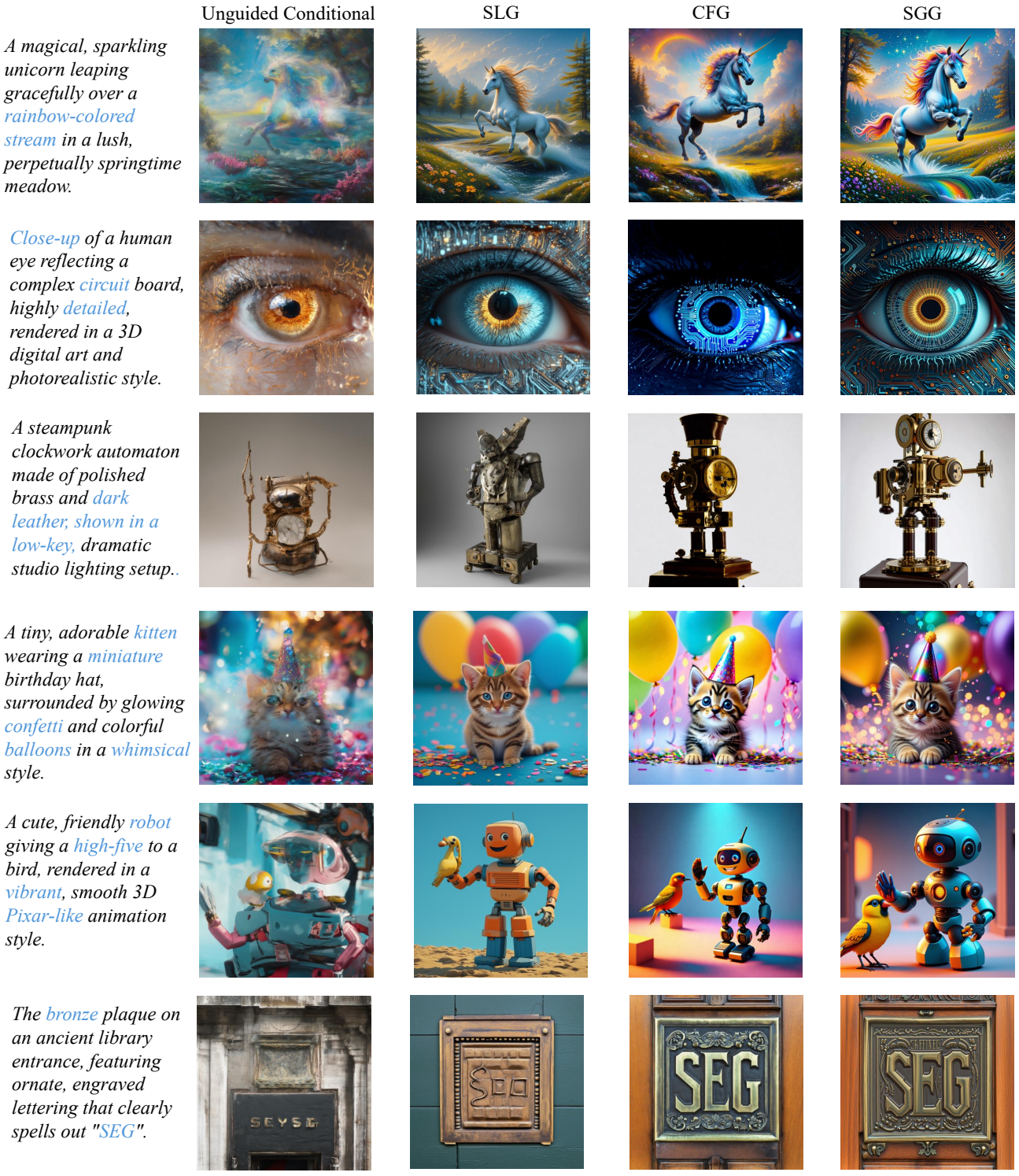


Figure 5. Qualitative Comparison between Unguided Conditional, CFG, SLG and SGG (Ours) (2/2)

F.3. Qualitative results on ImageNet256

SiT-B/2

SiT-B/2 + SEG

SiT-B/2 + REPA

SiT-B/2 + REPA + SEG

Figure 6. Qualitative Comparison between SiT-B/2 (Baseline), SGG (Ours), REPA, REPA+SGG

References

[1] Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *Proc. ECCV*, 2024. 6

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. In *Proc. ICLR*, 2023. 1

[3] Chubin Chen, Jiashu Zhu, Xiaokun Feng, Nisha Huang, Meiqi Wu, Fangyuan Mao, Jiahong Wu, Xiangxiang Chu, and Xiu Li. S^2-guidance: Stochastic self guidance for training-free enhancement of diffusion models, 2025. 4

[4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proc. ICML*, 2024. 4, 6, 7

[5] Weichen Fan, Amber Yijia Zheng, Raymond A. Yeh, and Ziwei Liu. Cfg-zero*: Improved classifier-free guidance for flow matching models, 2025. 4

[6] Alexandre Galashov, Ashwini Pokle, Arnaud Doucet, Arthur Gretton, Mauricio Delbracio, and Valentin De Bortoli. Learn to guide your diffusion model. *arXiv preprint arXiv:2510.00815*, 2025. 4

[7] Zhengqi Gao, Kaiwen Zha, Tianyuan Zhang, Zihui Xue, and Duane S Boning. Reg: Rectified gradient guidance for conditional diffusion models. In *Proc. ICML*, 2025. 4

[8] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Proc. NeurIPS*, 2023. 5

[9] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *TMLR*, 2025. 1, 7

[10] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proc. EMNLP*, 2021. 5

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. NeurIPS*, 2017. 2

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Proc. NeurIPS Workshop*, 2021. 2, 3, 4, 6

[13] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proc. CVPR*, 2024. 5

[14] Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *Proc. CVPR*, 2025. 4, 6

[15] Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *Proc. ICLR*, 2024. 7

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022. 1

[17] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Proc. NeurIPS*, 2024. 2, 3, 4, 6, 7

[18] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Proc. NeurIPS*, 2024. 4, 6

[19] 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 *Proc. ECCV*, 2014. 4

[20] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proc. ICLR*, 2023. 1

[21] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proc. ICLR*, 2023. 1

[22] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *Proc. ECCV*, 2024. 5

[23] Dawid Malarz, Artur Kasymov, Maciej Zieba, Jacek Tabor, and Przemyslaw Spurek. Classifier-free guidance with adaptive scaling. 2025. 5

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. 6, 7

[25] Shreshth Saini, Shashank Gupta, and Alan C. Bovik. Rectified-cfg++ for flow based models. In *Proc. NeurIPS*, 2025. 4

[26] Christoph Schuhmann. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022. 4

[27] 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. In *Proc. NeurIPS*, 2022. 4

[28] Inkyu Shin, Chenglin Yang, and Liang-Chieh Chen. Deeply supervised flow-based generative models. In *Proc. ICCV*, 2025. 5

[29] Kiwhan Song, Jaeyeon Kim, Sitan Chen, Yilun Du, Sham Kakade, and Vincent Sitzmann. Selective underfitting in diffusion models. *arXiv preprint arXiv:2510.01378*, 2025. 4

[30] Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025. 6

[31] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 4, 5

[32] 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. In *Proc. ICCV*, 2023. 4, 5

[33] Fu Xiaomeng and Li Jia. Tcfg: Truncated classifier-free guidance for efficient and scalable text-to-image acceleration. In *Proc. ICCV*, 2025. 4

[34] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *Proc. ICLR*, 2025. 6