Mitigating Diffusion Model Hallucinations with Dynamic Guidance

Kostas Triaridis[1], **Alexandros Graikos**[1], **Aggelina Chatziagapi**[1], **Grigorios G. Chrysos**[2], **Dimitris Samaras**[1]

[1]Stony Brook University, [2]University of Wisconsin-Madison

Abstract

Hallucinations in diffusion models are samples with structural inconsistencies that can emerge due to the excessive smoothing of the learned score function, which in turn leads to interpolations between modes of the data distribution. Since semantic interpolations are often desirable and contribute to sample diversity, we believe that a nuanced and targeted solution is required to address diffusion model hallucinations. In this work, we introduce **Dynamic Guidance**, which *mitigates* hallucinations by selectively sharpening the score function only along the pre-determined directions known to cause artifacts, while preserving valid semantic variations. This sharpening can be performed using either pre-determined classes or semantically coherent clusters that form pseudo-classes over the data distribution. The latter allows for a principled extension of Dynamic Guidance to text-to-image generation, where we select modes to correspond to fine-grained contextual differences in textual descriptions. To our knowledge, this is the first approach that addresses hallucinations at generation time rather than through post-hoc filtering. Dynamic Guidance substantially reduces hallucinations on both controlled and natural image datasets, significantly outperforming baselines.

This paper is currently under review at NeurIPS 2026, preprint available on arXiv.

A pre-print is available at this url.

Keywords: Vision and Learning.

1 Introduction

Diffusion models have emerged as the dominant paradigm for image generation [15, 34, 7] due to their ability to generate high-fidelity and diverse images. Despite their success, they still remain prone to generating hallucinations [2]; i.e., samples that could never appear in the training set and are outside the support of the theoretical data distribution. A typical example in natural images are samples with incorrect anatomy, such as human hands with the incorrect number of fingers, or cats with missing body parts (See Figure 4). Hallucinations have mostly been studied in the context of image-text misalignment, where generated images fail to sufficiently represent the information specified in a text prompt [26, 32, 25, 23, 47]. In this work, we instead study hallucinations as a fundamental issue in the diffusion model's sampling process that can arise even when the generated samples adhere to the given conditioning.

Aithal et al. [2] attributed hallucinations to mode interpolation, showing that models generate samples that lie between incompatible modes, i.e., regions of high probability density in the data distribution, producing semantically invalid content. They trace the mode interpolation issue to the fact that the learned score function of the model is excessively smooth between modes of the data distribution; the true score function is significantly sharper in the intermediate regions between modes, which means

that the size of the required denoising steps in those regions is significantly larger. They verify that mode interpolation does not occur when using the true score on a mixture of Gaussians setting.

To avoid those low-probability regions, the denoising process must take larger steps, similar to the ones that would have been taken if the true score function had been used. Conventional guidance methods, such as classifier [10] and classifier-free [14] guidance, are designed to steer samples toward high-likelihood regions of the data distribution, typically to improve sample quality in conditional generation. They essentially sharpen the score function in directions that correspond to the given condition. Hallucinations arise in low-likelihood regions, motivating the use of guidance not only for improving fidelity but also for mitigating hallucinations during sampling.

We argue that guidance with a pre-determined, fixed condition does not account for the full sampling trajectory; when the guidance condition and the initial noise are misaligned, the sample can be pushed into regions that require large corrective steps. The error in these steps scales with their magnitude (Figures 17, 18), which can cause the trajectory to overshoot or undershoot the target mode (Section 4).

We also argue that we should be selective in reducing interpolations between modes **along the specific directions** where hallucinations occur. Some of the interpolations are welcome: they correspond to valid semantic variation and are essential for maintaining diversity in the generated samples (i.e., model generalization [9]), while others lead to implausible generations, perceived as hallucinations. For instance, in the latent manifold of hand images, interpolation along directions corresponding to skin tone yields valid and diverse samples, while interpolation along directions that control finger position may generate anatomically inconsistent hands with an incorrect number of fingers (Figure 2b). We demonstrate that the score function can be sharpened *selectively* by choosing guidance modes such that interpolations between them correspond precisely to hallucinations, thereby suppressing invalid generations without reducing diversity elsewhere (Section 3).

We propose **Dynamic Guidance** (DG), which adaptively selects the target for guidance at each denoising step. Instead of committing to a fixed condition from the start, a classifier is used to identify the most likely mode given the current state, and guidance is applied toward that mode. By allowing the target to change throughout sampling, Dynamic Guidance avoids being locked into trajectories that require large, interpolation-producing steps. Dynamic Guidance is the first method specifically designed to tackle hallucinations that mitigates hallucinations during the sampling process itself, rather than relying on post-hoc detection and rejection of flawed samples or post-training of the diffusion model. Crucially, by intervening directly in the generative process, our approach prevents hallucinations from arising in the first place. Mitigating hallucinations during sampling is preferable to post-hoc detection because it avoids wasting compute on samples that will later be discarded and preserves diversity by keeping the desired interpolations. We show that we can select the modes to correspond to pre-determined classes, like ImageNet-1k classes (Section 4.2), but that is not strictly necessary: Dynamic Guidance can work with pseudo-classes created by clustering semantically coherent neighborhoods in the data distribution (Section 4.2.2). This allows for a principled extension of Dynamic Guidance to text-to-image generation, where we select modes to correspond to fine-grained contextual differences in textual descriptions (Section 4.3).

In summary, our contributions are as follows:

- We propose Dynamic Guidance, a method that mitigates diffusion model hallucinations by adaptively guiding the denoising process away from low-probability regions.
- We use controlled experiments to demonstrate that Dynamic Guidance **selectively sharpens** the score function across directions that induce hallucinations, while preserving it elsewhere.
- We present the first method to effectively mitigate hallucinations during the generation process instead of detecting them post-hoc. Our approach vastly outperforms previous detection-based methods in hallucination reduction on toy data and controlled datasets, achieving up to **70% reduction in hallucination rate**.
- We apply Dynamic Guidance to 256×256 ImageNet generation and show that it improves hallucination-related metrics, consistently outperforming static guidance methods.
- We extend DG to text-to-image generation, with a user study showing that it consistently mitigates hallucinations, reducing the perceived hallucination rate for 100% of participants.

2 Dynamic Guidance

Recent findings [2] have shown that the learned score function $s_\theta(\boldsymbol{x}_t)$ in DDPMs tends to be overly smooth in the low-density regions between modes. This lack of "sharpness" can effectively trap samples during denoising, a problem particularly pronounced by few-step sampling, where the denoiser cannot leap across these smooth regions. The resulting samples end up as interpolations between those modes, which might be "hallucinations" depending on the semantic relationship between the specific modes involved.

We propose **Dynamic Guidance**, a simple yet effective guidance algorithm that adaptively sharpens the learned score function along the sampling trajectory. The core insight is to identify potential modes c^* that a sample is naturally gravitating towards during the sampling process, and use the score function of the locally sharper conditional distribution $p(\boldsymbol{x} \mid c^*)$ rather than the overly-smooth $p(\boldsymbol{x})$ or an attenuated $p(\boldsymbol{x} \mid c)$ from a distant mode.

However, the distribution modes c are not always aligned with the conditioning y a model has been trained with. In cases where the modes are more fine-grained than the conditions, hallucinations can often emerge as interpolations between such conditions. To solve this, we resort to inference-time guidance mechanisms with a post-hoc, finer set of conditions.

For our method, having first chosen appropriate mode labels $\mathbb{C}$ whose interpolations correspond to the defined hallucinations (see Section 4.1: Single Shapes), we apply guidance dynamically, without fixing the guidance target/condition at the beginning of the sampling process. At each timestep, we identify the mode with the maximum probability given the current noisy sample $\boldsymbol{x}_t$. and perform the sampling step by applying guidance using the selected mode. We essentially calculate a sharper approximation of the score function:

$$\hat{s}_\theta(\boldsymbol{x}_t) := \sum_c \mathbb{1}_{c^*}(c) \nabla_{\boldsymbol{x}_t} \log p_\theta(\boldsymbol{x}_t \mid c), \quad c^* = \arg\max_{c \in \mathbb{C}} \log p(c|\boldsymbol{x}_t). \tag{1}$$

By recalculating the most probable class at each timestep, we ensure that the guidance signal remains aligned with the local score direction and adapts to the sample's evolving trajectory. For diffusion models with discrete conditioning, we pick class labels $\mathbb{C}$ whose interpolations correspond to the defined hallucinations and calculate the score of $p(\boldsymbol{x}|c^*)$ using classifier guidance [10]. As an additional hyperparameter, we can control the strength of the guidance λ, effectively sampling using the score of the tempered distribution $p(\boldsymbol{x}|c^*)^\lambda$. The proposed algorithm for DDIM sampling with DG using classifier guidance is summarized in Algorithm 2. In Section 4.3, we show that the idea can be extended to models that use continuous conditions c, such as text tokens, and calculate the score of $p(\boldsymbol{x}|c^*)$ using classifier-free guidance (CFG) [14].

3 Dynamic Guidance selectively sharpens the score function

We find that sampling with Dynamic Guidance mitigates hallucinations because it selectively sharpens the score function learned by the diffusion model across the directions associated with the classifier. We first validate this in a simple 2D Gaussian setup, where the theoretical score function is given analytically. We calculate the unguided score function $s_\theta(\boldsymbol{x}_t, t) = -\frac{\epsilon_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\bar{\alpha}}}$ and the guided score function $\hat{s}_\theta(\boldsymbol{x}_t, t) = s_\theta(\boldsymbol{x}_t, t) + \lambda \nabla_{\boldsymbol{x}_t} \log p(c^*|\boldsymbol{x}_t)$ and plot them together with the real score function in Figure 1. We see that Dynamic Guidance effectively sharpens the score function, which mitigates hallucinations as shown in Section 4.

For more complex datasets, we want to visualize the learned score for specific directions that correspond to meaningful semantics. We use the Single Shapes dataset and examine the score function learned by a diffusion model when unguided and when using our proposed DG. In this dataset, all training images contain 1-3 instances of the *same* shape, but diffusion models trained on it can generate images containing different shapes, which we consider hallucinations. Here, we focus on latent directions that correspond to changing the appearance of shapes, since variations there can lead to images containing different shapes, i.e hallucinations.

We first learn how to embed images from the dataset using a β-VAE [13, 6] with a disentangled 10-dimensional latent representation. Examining the VAE-learned representations, we identify those that affect specific properties of the image: Dimension 9 controls the appearance of shapes on the left side, and Dimension 5 controls the position of shapes on the right (Figures 2b, 7).

The trained β-VAE provides a transformation between latents and clean images using the learned encoder $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x}_0)$ and decoder $\boldsymbol{x} = \mathcal{D}(\boldsymbol{z})$. Our goal is to estimate the score of the latents $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$. Using the change of variables formula for this autoencoder setting ([19]), we can express the distribution of the latents as

$$p_Z(\boldsymbol{z}) = p_{X_0}(\mathcal{D}(\boldsymbol{z}))\sqrt{|\det\left(\boldsymbol{J}_{\mathcal{D}}^T(\boldsymbol{z})\boldsymbol{J}_{\mathcal{D}}(\boldsymbol{z})\right)|}, \tag{2}$$

where $\boldsymbol{J}_{\mathcal{D}}(\boldsymbol{z})$ is the Jacobian of the VAE decoder at $\boldsymbol{z}$. Taking the $\nabla_{\boldsymbol{z}} \log$ on both sides we have

$$\begin{aligned}\nabla_{\boldsymbol{z}} \log p_Z(\boldsymbol{z}) = \; & \nabla_{\boldsymbol{z}} \log p_{X_0}(\mathcal{D}(\boldsymbol{z})) \\ & + \nabla_{\boldsymbol{z}} \log \sqrt{|\det\left(\boldsymbol{J}_{\mathcal{D}}^T(\boldsymbol{z})\boldsymbol{J}_{\mathcal{D}}(\boldsymbol{z})\right)|}.\end{aligned} \tag{3}$$

To estimate the score of the latents $\nabla_{\boldsymbol{z}} \log p_Z(\boldsymbol{z})$, we compute $\nabla_{\boldsymbol{z}} \log \sqrt{|\det\left(\boldsymbol{J}_{\mathcal{D}}(\boldsymbol{z})^T\boldsymbol{J}_{\mathcal{D}}(\boldsymbol{z})\right)|}$ using automatic differentiation. To calculate the score of the clean samples $\nabla_{\boldsymbol{z}} \log p_{X_0}(\mathcal{D}(\boldsymbol{z}))$ we first rewrite it as

$$\nabla_{\boldsymbol{z}} \log p_{X_0}(\mathcal{D}(\boldsymbol{z})) = \boldsymbol{J}_{\mathcal{D}}^T(\boldsymbol{z})\nabla_{\boldsymbol{x}_0} \log p_{X_0}(\boldsymbol{x}_0), \tag{4}$$

and then we utilize the Perturb-and-Average Scoring from Wang et al. [43], which estimates the score of the clean images with an expectation over noisy image scores

$$\nabla_{\boldsymbol{x}_0} \log p_{X_0}(\boldsymbol{x}_0) \approx \mathbb{E}_{\epsilon}\left[\nabla_{\boldsymbol{x}_t} \log p_{X_t}(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon)\right]. \tag{5}$$

In practice, we first decode a latent $\boldsymbol{z}$ into a clean image $\boldsymbol{x}_0$. We then perturb $\boldsymbol{x}_0$ with multiple random noise samples at timestep t to get an approximation of the score of $\boldsymbol{z}$ by averaging out the individual predicted scores.

To showcase how the score function is selectively sharpened, we select a generated image $\hat{\boldsymbol{x}}_0$ that is a hallucination, showing a pentagon on the left and a triangle on the right. We plot the score function given by Equation (3) along the latent dimension 9 (Figure 2b). In this instance, the ideal score should be 0 in the regions that correspond to images with a single or two triangles. In contrast, the score for an image containing two shape types should be high, pushing the sample away from the hallucination.

Indeed, this is the exact observation we make in Figure 2b; while the base diffusion-learned score does not push the sample away from the hallucination region, DG sharpens the score predictions, mitigating the hallucination. At the same time, when observing the score across a latent dimension that controls shape position, and thus is unrelated to hallucinations (latent dimension 5), we see that DG has no effect over it.

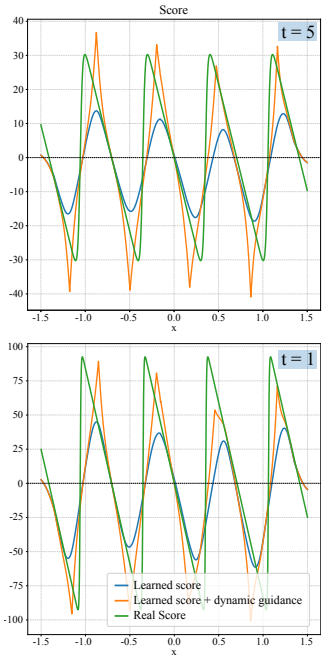


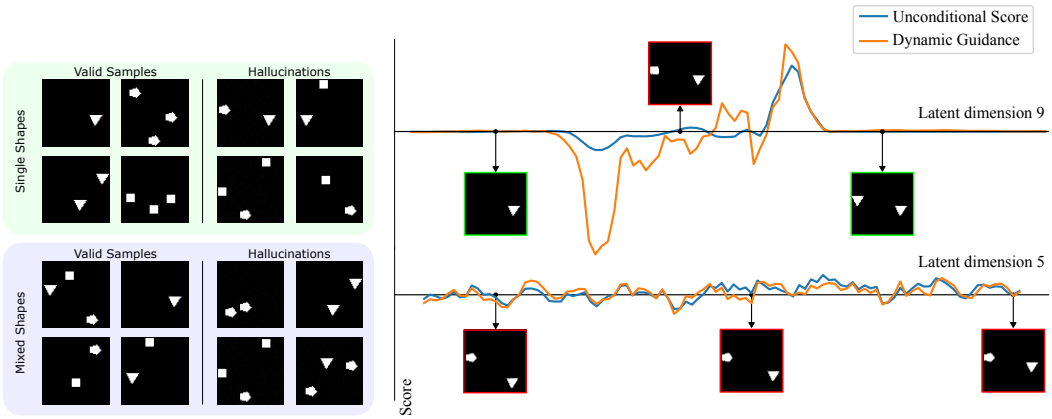
Figure 1: **Score Function Sharpening.** The learned score function of the diffusion model with and without Dynamic Guidance, compared to the true score function for a 2D mixture of Gaussians across the x dimension. The model learns a smoothed-out score function, which DG sharpens so that it more closely approximates the correct one.

4 Experiments

4.1 Controlled Settings

In this section, we describe the datasets used for our controlled study and define what is considered a hallucination for each. We also point to the labels used for the Dynamic Guidance.

Single Shapes: We construct a synthetic image dataset consisting of images containing triangles, squares, and pentagons. Each image contains one to three instances of the same shape, but never multiple shape types. In this setting, we define hallucinations as images containing different shapes simultaneously (see Figure 2a). The class labels c are given by the shape type, $c \in \{T, S, P\}$. Hallucinations, therefore, correspond to interpolations across these labels.



(a) (b)

Figure 2: **(a)** Examples of valid samples and hallucinations for the Single Shapes and Mixed Shapes datasets. **(b)** We pick an initial image that contains two different shapes (triangle + square), which is a hallucination for the Single Shapes dataset. We focus on a latent dimension that controls the appearance of the left shape **(Top)**. To resolve the hallucination, the square on the left should disappear or turn into a triangle. In the in-between region, where the left shape is square or pentagon, the unguided score function is zero, "trapping" the sample and generating a hallucination. Dynamic Guidance sharpens the score in this region, steering the sample toward valid images that only contain triangles. Dynamic Guidance does not affect the score function along dimensions that are unrelated to hallucinations, like the one controlling the position of the shape on the right **(Bottom)**.

Mixed Shapes: We adopt the synthetic image dataset from Aithal et al. [2] consisting of images of triangles, squares, and pentagons. Each image contains up to one instance of each shape, and may contain multiple shape types. In this setting, we define hallucinations as images containing 2 or more instances of the same shape (see Figure 2a). The class labels c are given by the combination of shapes, $c \in \{T, S, P, TS, TP, SP, TSP\}$. Hallucinations in this case do not necessarily correspond to interpolations across the selected class labels.

Hands: We use the Hands-11k dataset [1], which contains images of human hands. Here, we define hallucinations as images that deviate from the expected hand anatomy. For Dynamic Guidance, we use the 4 classes provided by the dataset, corresponding to the orientation of the hand: *"palmar right"*, *"palmar left"*, *"dorsal right"*, and *"dorsal left"*. Interestingly, in this setting, some hallucinations directly correspond to interpolations between the classes (images with hands facing down, with two thumbs are an interpolation between *"dorsal right"* and *"dorsal left"*), while others do not (5 fingers without a thumb). Dynamic guidance is effective in mitigating both.

4.1.1 Results

For all datasets we perform experiments for, we measure the reduction in hallucinations as follows:

$$\text{HR} = \frac{\#\text{Hallucinations}_{\text{before}} - \#\text{Hallucinations}_{\text{after}}}{\#\text{Hallucinations}_{\text{before}}} \tag{6}$$

Table 1: Single Shapes dataset.

Method	HR↑
Var. Filtering (1%)	1.05%
Var. Filtering (5%)	3.16%
Var. Filtering (10%)	11.58%
Classifier Guidance	11.05%
Dynamic Guidance	**74.21%**

Table 2: Mixed Shapes dataset.

Method	HR↑
Var. Filtering (1%)	1.35%
Var. Filtering (5%)	8.11%
Var. Filtering (10%)	17.57%
Classifier Guidance	-1.92%
Dynamic Guidance	**72.97%**

Table 3: Hands-11k dataset.

Method	HR↑
VF (1%)	1.83 ± 3.68%
VF (5%)	5.46 ± 5.16%
VF (10%)	9.19 ± 5.99%
CG	17.81 ± 17.88%
DG	**45.12 ± 5.36%**

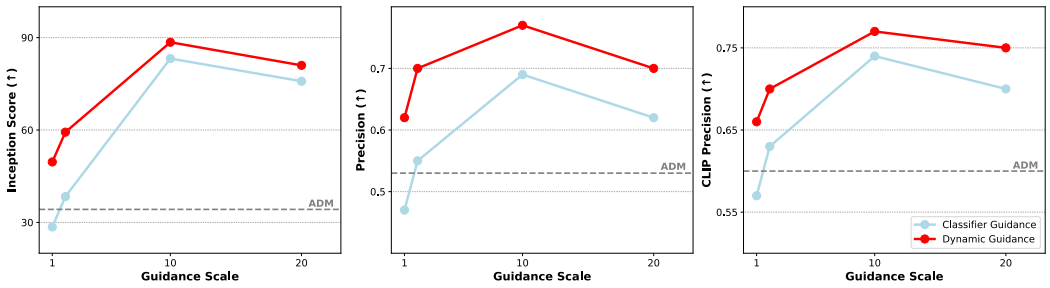


Figure 3: Performance of Classifier and Dynamic Guidance on Hallucination-related metrics on ImageNet-1k generation. Dynamic Guidance consistently improves Inception Score and precision across guidance scales, which corresponds to fewer hallucinations.

Negative values correspond to an increase in hallucinations. For all controlled image datasets, we train an ADM [10] and a noisy sample classifier using the guided-diffusion[1] codebase. We compare to variance filtering, the detection-based method proposed by [2], and Classifier Guidance. For variance filtering, we pick the best hyperparameters by performing a grid search around the values mentioned in their paper, and evaluate using different discard rates. Across these diverse benchmarks, Dynamic Guidance (DG) consistently outperforms variance filtering (VF) and Classifier Guidance (CG) in mitigating hallucinations, achieving over a 70% reduction in the Single and Mixed Shapes datasets (Tables 1, 2) and over 40% in the Hands-11k dataset with 25-step DDIM sampling.

4.2 ImageNet

Hallucinations in large-scale image benchmarks span a vast range of classes, objects, and scenes, making them nearly impossible to precisely define, let alone detect. With this in mind, we choose ImageNet1k generation as a large-scale benchmark to evaluate our approach. In this setting, hallucinations cannot be strictly defined, so we rely on proxy metrics: precision, recall [20], and Inception Score [35]. Precision measures the fraction of generated samples that fall inside the support of the estimated real data distribution. We argue that higher precision reflects fewer hallucinations, since hallucinations correspond to samples outside the *true* data distribution. Inception Score is maximized when generated images clearly belong to some class (they are low-entropy, high-confidence predictions for the classifier) and diverse (labels evenly distributed). When generating samples, conditioning and initial noise are often mismatched [23] and images fail to correspond to any valid class, visually resembling hallucinations (Figure 6). A large presence of such samples reduces the classifier's confidence and consequently hurts both the Inception Score and Precision.

In our experiments, we evaluate precision and recall using both the Inception [40] and CLIP [33] models and Inception Score. We adopt the pretrained diffusion model and classifier from Dhariwal and Nichol [10] and compare three settings: unguided generation, Classifier Guidance (CG) with varying guidance scales, and Dynamic Guidance (DG) with the same scales. The results, summarized in Table 10, show that Dynamic Guidance consistently outperforms Classifier Guidance across all settings (Figure 3). Notably, in the best setting, it achieves precision and Inception Score gains of 8 and 5 points, respectively, while maintaining diversity; recall is not significantly reduced, and the Inception Score is the highest overall. This trend holds across different metrics; we also measure FID with both Inception and CLIP and generative density and coverage [28] and report the results in Tables 11 and 12 of Appendix Section C. In Appendix Figures 17, 18, we show how 'static' CG tends to overshoot in cases where the selected label does not align with the initial noise sample.

4.2.1 Selection of guidance interval

We identify that, unlike Classifier Guidance, our proposed Dynamic Guidance (DG) can be performed just for a subset of timesteps, which we denote in our algorithm as $[T_1, T_2]$. Classifier Guidance attempts to impose a **strong constraint** on the generation process: the final sample should belong to the chosen class. On the other hand, Dynamic Guidance is more similar to Classifer-Free Guidance (CFG) in that it attempts to guide a strong signal (conditional model in the case of CFG, unconditional

[1] https://github.com/openai/guided-diffusion

model in the case of DG) with a weak guidance signal (unconditional model in the case of CFG, dynamic gradient of a classifier in the case of DG) to improve generation.

Inspired by recent work on CFG [21, 45], we show that applying DG only for some intermediate timesteps $[T_1, T_2]$ improves performance. To select T_1 and T_2 for each experiment, we chose the generation timestep at which the image begins to form (T_1), and the timestep where samples appear to have already converged to an image that cannot be modified further (T_2). We verify our choice of $[T_1, T_2]$ in Table 4, where for our ImageNet experiments the selected $T_1 = 800$ and $T_2 = 400$ achieve the best results. In Figures 9 and 10 in the Appendix, we visualize the generation process to show how our choice of T_1 and T_2 is motivated by the image formation process.

Table 4: Ablation for the timestep interval $[T_1, T_2]$ used to perform Dynamic Guidance in ImageNet.

Method	IS↑	Prec↑	Rec↑
Uncond. ADM	34.24	0.53	0.61
+ DG [1000-0]	48.20	0.73	0.33
+ DG [600-200]	61.07	0.64	**0.62**
+ DG [800-400]	**88.49**	**0.77**	0.52

Table 5: Comparison of Dynamic Guidance with a classifier trained on DINOv2 pseudo-classes to one trained on real ImageNet classes.

	Inception			CLIP	
Method	IS↑	Prec↑	Rec↑	Prec↑	Rec↑
Uncond. ADM	34.24	0.53	0.61	0.60	0.26
+ CG w/ real	83.19	0.69	0.55	0.74	0.27
+ DG w/ real	88.49	0.77	0.52	0.77	0.26
+ DG w/ clusters	75.17	0.78	0.51	0.77	0.25

4.2.2 Guidance with Pseudo-Classes

In the Mixed Shapes experiments, we discussed how DG can be used with a classifier whose labels do not directly correspond to the hallucinations we aim to avoid. The set of labels we use for the purpose of reducing hallucinations with DG does not have to be identical to the set of labels used to control a generative model. The labels used for conditioning must correspond to human-interpretable semantics, such as object categories, attributes, or text, since they determine what the model is intended to generate. In contrast, the labels used for hallucination reduction do not need to carry such semantic meaning; they may correspond to abstract latent modes or auxiliary partitions of the data that help isolate hallucination-prone directions without mapping to interpretable concepts.

For our ImageNet experiments, we create pseudo-classes by clustering the training images using DINOv2 [30]. We create 5000 clusters using the hierarchical clustering method described by Vo et al. [42], assign pseudo-labels to the clusters, and train a classifier to predict those pseudo-labels. In Table 5, Dynamic Guidance with a classifier trained on DINOv2 pseudo-classes performs as well as DG with ImageNet classes in terms of precision and recall. With regard to InceptionScore DG with pseudo-classes still improves performance compared to the unguided model, but underperforms DG with ImageNet classes. We attribute this to the bias of IS towards exact ImageNet classes specifically.

In the Appendix Figure 25 we also show that it is possible to use distinct sets of labels for conditioning (ImageNet classes) and guiding a model with DG (DINO clusters). The two can act orthogonally and DG can improve generations even in the class-conditional case. This means that it is possible to pick labels that correspond to required semantics for training a conditional model and a distinct set of labels that help reduce hallucinations when used with DG.

4.3 Text-to-Image generation

We extend Dynamic Guidance to text-conditioned latent diffusion models [34]. In text-to-image models, we identify that an ambiguous prompt, such as *"a horse in a field"* leaves the horse's pose and action unspecified, which regularly causes hallucinations. During denoising, the model may hedge among multiple valid interpretations, resulting in anatomical errors (e.g., malformed limbs or improbable poses).

To address this, we propose an approximation to the sharpening of the score function by modifying the prompt during sampling. Analogous to the $\arg\max$ operation on a classifier, we employ a Vision-Language Model (VLM) to observe intermediate denoised predictions and dynamically adjust the text prompt during generation. We utilize a set of classes designed to resolve ambiguities that

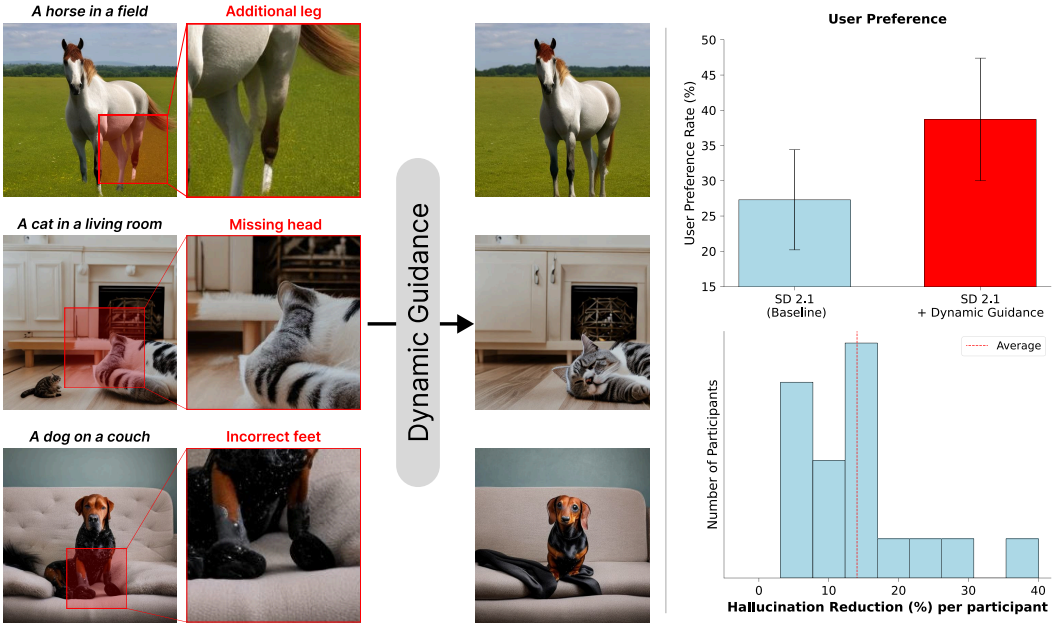


Figure 4: (**Left**) Text-to-image models can generate hallucinated images that contain anatomical errors, such as extra legs, or a missing head. Dynamic Guidance effectively mitigates those hallucinations, producing higher-quality samples. (**Right**) We evaluate Dynamic Guidance on text-to-image generation with a user study. Remarkably, Dynamic Guidance reduces the hallucination rate for **all** participants, with an average reduction in hallucinations of 14.1%. Participants also showed strong preference for images generated using Dynamic Guidance over the baseline.

cause hallucinations; given a specific input prompt P, we ask a Large Language Model (LLM) to identify the primary axes of ambiguities contained in the prompt. Then the LLM is tasked with creating a diverse set of branches, designed to resolve this ambiguity: for example, for the *"a horse in a field"* prompt, the LLM proposes branches to resolve the pose and action ambiguity such as *"a horse galloping with legs extended in a field"* and *"a horse lying down in a field"*.

During denoising, we allow the text conditioning to adapt within a guidance window $[T_1, T_2]$. For each step, we calculate the current clean latent prediction $\boldsymbol{z}_0(\boldsymbol{z}_t)$, decode it into an image using the VAE, and ask a VLM to identify the closest match given multiple options corresponding to the created branches. If the VLM identifies a match, we update the text prompt and continue sampling; otherwise, the VLM may also respond "STAY" if the image is too noisy to classify, in which case we retain the current prompt. We show that Guidance with a dynamically changing prompt can effectively mitigate hallucinations, especially the ones corresponding to incorrect anatomy (Figure 4).

We implement Dynamic Guidance on top of Stable Diffusion 2.1 [34], and use a classifier-free guidance scale of 7.5 [14] for sampling. We use Claude [4] to create the branches that are used as "classes" and Qwen2-VL-7B [44] as the classifier. We evaluate Dynamic Guidance against the Stable Diffusion 2.1 baseline with a user study of 18 participants. To highlight the simplicity of direct application of Dynamic Guidance on a given text-to-image model we choose some reasonable default hyper-parameters, avoiding the need for extensive tuning (More details in Appendix Section B.6). We recognize that the definition of hallucinations in complex image generation settings is quite ambiguous, and identify that, even when given strict guidelines, different people have different "thresholds" for identifying a sample as a hallucination. Even with this variance, **Dynamic Guidance remarkably reduced the perceived hallucination rate for all participants**. In total, Dynamic Guidance reduced the perceived hallucination rate for 70% of prompts, resulting in an average reduction of 14.1%, from 52.7% to 45.3%. Additionally, participants preferred images generated using Dynamic Guidance with a rate of 38.7% versus 27.3% for the baseline. We also show that using the generated, more complex, branches as base prompts can lead to worse generations and new hallucinations (similarly to the observations by Park et al. [31]). We show examples in which those are resolved by guiding adaptively, using our proposed Dynamic Guidance in Figure 13.

5 Related Work

Diffusion Models Introduced by Sohl-Dickstein et al. [36], diffusion models [15, 39] are characterized by the forward process, where Gaussian noise is gradually added to a sample, and a learned reverse process, where a network learns to denoise samples. The original denoising diffusion formulation was shown to be equivalent to score-matching [17, 41, 39], linking the denoiser predictions to the score function of the noisy data. We adopt the score-function view of diffusion models in this paper to study the generation and mitigation of *hallucinations*.

Guiding Diffusion Models The reverse diffusion process can be controlled to draw samples with constraints [10, 14, 11]. Dhariwal and Nichol [10] introduced an external classifier into the diffusion model's sampling, using its gradients to influence the sampling trajectory towards a target class, while Ho and Salimans [14] jointly trained a conditional and an unconditional diffusion model and sampled from a combination of their scores. A more recent work [18] suggests that it is possible to improve the fidelity of the generations by guiding the model with a weaker one. While those guidance methods have aimed to improve the fidelity of generations by strategically sampling from well-learned high probability regions, another line of work has focused on incorporating arbitrary training-free guidance by imposing specific linear [8] or non-linear [46] constraints, either through backpropagation [8, 46] or approximate Newton iterations [12].

Hallucinations in Diffusion Models Prior work has examined hallucinations in diffusion models [2, 32, 25], but these are most often defined in terms of text–image misalignment [23, 26, 5], where the diffusion model fails to generate an image that matches the conditioning prompt. Such hallucinations are more linked to issues of text-image alignment and text representations, and most approaches address them by improving prompt adherence through post-training [27, 16, 22]. In contrast, we take a view of hallucinations that is more directed to the diffusion model: rather than treating them as failures of the text condition and the model's adherence to it, we study them as a fundamental property of the diffusion model itself. Our focus is on the model's score function and its tendency to interpolate across modes in ways that yield unrealistic samples, independent of the text condition.

Mode Interpolation Aithal et al. [2] analyze the problem of hallucinations in diffusion models through mode interpolation. This, however, is not the first time the phenomenon of mode interpolation has been observed in generative models, and not all instances of mode interpolation result in hallucinations. Deschenaux et al. [9] show that you can guide a diffusion model to produce interpolations of *desired* attributes not present in the training data, while there are other recent works that detect "mode mixture" in GANs [3] and diffusion models [24].

6 Limitations

While Dynamic Guidance effectively mitigates hallucinations, it can also impact the diversity of generated samples, as it also introduces certain biases. The impact of Dynamic Guidance on the diversity-hallucination trade-off depends on how the selected classes relate to the hallucination direction. When the chosen classes align with hallucination-relevant directions (e.g., Single Shapes), Dynamic Guidance sharpens the score only where necessary, preserving diversity. When this alignment is not possible, it still mitigates hallucinations but may slightly reduce diversity, as suggested by the modest recall drop and increased coverage on ImageNet (Section C). Since the method relies on a classifier to determine the most likely mode at each timestep, any bias in the classifier's predictions can affect the sampling trajectory, leading to a preference for conditions that are easier to identify. Consequently, certain semantic modes can receive disproportionately more probability mass, leading to biased final distributions. In addition, the initial noise introduces bias in the composition of the generated image, further skewing the distribution of generated samples. In practice, we find that certain classes are over-represented (Figures 23 and 24) when using DG.

7 Conclusion

In this work, we addressed the problem of hallucinations in diffusion models by introducing Dynamic Guidance (DG), a method that mitigates hallucinations during the generative process itself. Unlike

prior detection-based approaches, DG prevents hallucinations from arising by selectively sharpening the score function along hallucination-inducing directions while preserving benign interpolations that support diversity. Our experiments across toy data, controlled and real-world image datasets demonstrate consistent improvements, with DG achieving substantial hallucination reduction even under realistic low-step DDIM sampling. On the large-scale benchmark ImageNet, we further show improvements in proxy metrics such as precision and Inception Score, validating that our method generates samples that remain closer to the true data distribution. Finally, we showed that DG can also be applied to text-to-image models, effectively mitigating hallucinations. We believe Dynamic Guidance provides a principled step toward more reliable diffusion models and opens opportunities for future work in understanding and controlling hallucinations in large-scale generative models.

Acknowledgements

This research was supported by NSF grants IIS-2123920, IIS-2212046.

References

[1] M. Afifi. 11k hands: gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*, 2019. doi: 10.1007/s11042-019-7424-8. URL `https://doi.org/10.1007/s11042-019-7424-8`.

[2] S. K. Aithal, P. Maini, Z. C. Lipton, and J. Z. Kolter. Understanding hallucinations in diffusion models through mode interpolation. In *NeurIPS*, 2024.

[3] D. An, Y. Guo, N. Lei, Z. Luo, S.-T. Yau, and X. Gu. Ae-ot: A new generative model based on extended semi-discrete optimal transport. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkldyTNYwH`.

[4] Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic, 2024. URL `https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf`.

[5] A. Borji. Qualitative failures of image generation models and their application in detecting deepfakes. *Image and Vision Computing*, 137:104771, 2023.

[6] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

[7] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.

[8] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=OnD9zGAGT0k`.

[9] J. Deschenaux, I. Krawczuk, G. Chrysos, and V. Cevher. Going beyond compositions, ddpms can produce zero-shot interpolations. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[10] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] A. Graikos, N. Malkin, N. Jojic, and D. Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

[12] A. Graikos, N. Jojic, and D. Samaras. Fast constrained sampling in pre-trained diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Sy2fzU9gl`.

[14] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

[15] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, 2020.

[16] Z. Hu, F. Zhang, L. Chen, K. Kuang, J. Li, K. Gao, J. Xiao, X. Wang, and W. Zhu. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23604–23614, 2025.

[17] A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[18] T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, and S. Laine. Guiding a diffusion model with a bad version of itself. In *NeurIPS*, 2024.

[19] U. Köthe. A review of change of variable formulas for generative modeling. *arXiv preprint arXiv:2308.02652*, 2023.

[20] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

[21] T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

[22] M. Le, G. Mittal, T. Meng, A. S. M. Iftekhar, V. Suryanarayanan, B. Patra, D. Samaras, and M. Chen. Hummingbird: High fidelity image generation via multimodal context alignment. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6kPBThI6ZJ.

[23] S. Li, H. Le, J. Xu, and M. Salzmann. Enhancing compositional text-to-image generation with reliable random seeds. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5BSlakturs.

[24] Z. Li, S. Li, Z. Wang, N. Lei, Z. Luo, and D. X. Gu. Dpm-ot: a new diffusion probabilistic model based on optimal transport. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22624–22633, 2023.

[25] Y. Lim, H. Choi, and H. Shim. Evaluating image hallucination in text-to-image generation with question-answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26290–26298, 2025.

[26] Q. Liu, A. Kortylewski, Y. Bai, S. Bai, and A. Yuille. Discovering failure modes of text-guided diffusion models via adversarial search. *arXiv preprint arXiv:2306.00974*, 2023.

[27] K. Mrini, H. Lu, L. Yang, W. Huang, and H. Wang. Fast prompt alignment for text-to-image generation. *arXiv preprint arXiv:2412.08639*, 2024.

[28] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pages 7176–7185. PMLR, 2020.

[29] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[30] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

[31] D. Park, S. Kim, T. Moon, M. Kim, K. Lee, and J. Cho. Rare-to-frequent: Unlocking compositional generation power of diffusion models on rare concepts with LLM guidance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BgxsmpVoOX.

[32] Z. Qin, D. Cheng, H. Wang, H. Yi, Y. Shao, Z. Fan, K. Li, and Q. Lao. Evaluating hallucination in text-to-image diffusion models with scene-graph based question-answering agent. *arXiv preprint arXiv:2412.05722*, 2024.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[36] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[37] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[38] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[39] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[41] P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[42] H. V. Vo, V. Khalidov, T. Darcet, T. Moutakanni, N. Smetanin, M. Szafraniec, H. Touvron, camille couprie, M. Oquab, A. Joulin, H. Jegou, P. Labatut, and P. Bojanowski. Automatic data curation for self-supervised learning: A clustering-based approach. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=G7p8djzWOl.

[43] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12619–12629, 2023.

[44] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[45] X. Wang, N. Dufour, N. Andreou, M.-P. Cani, V. F. Abrevaya, D. Picard, and V. Kalogeiton. Analysis of classifier-free guidance weight schedulers. *arXiv preprint arXiv:2404.13040*, 2024.

[46] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023.

[47] L. Zhao, Y. Deng, W. Zhang, and Q. Gu. Mitigating object hallucination in large vision-language models via image-grounded guidance. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=w0xYx9CJhY.

Appendix

Contents

A Background on Diffusion Models **14**

A.1 Denoising Diffusion Probabilistic Models 14

A.2 Denoising Diffusion Implicit Models 14

A.3 Connections to Score Based Generative Models 14

A.4 Classifier Guidance 14

B Implementation Details **14**

B.1 Dynamic Guidance Algorithm with DDPM, DDIM and Classifier Guidance 14

B.2 2D Mixture of Gaussians 15

B.3 Single Shapes - Mixed Shapes 15

B.4 Hands 15

B.5 ImageNet 16

B.6 Text to Image 16

B.7 Compute details 17

C Additional Comparisons / Metrics **21**

C.1 Additional Analysis of Dynamic Guidance 22

C.2 Hallucination Mitigation during Sampling 22

C.3 Importance of Adaptive Selection 26

C.4 Sensitivity to Classifier Quality 27

C.5 Potential Class Bias in Generation 27

C.6 Additional Qualitative Results 27

D Potential Societal Impact **38**

A Background on Diffusion Models

A.1 Denoising Diffusion Probabilistic Models

DDPMs [15] learn to draw samples from a given data distribution $q(\boldsymbol{x})$. They consist of a forward process, in which Gaussian noise of increasing variance, controlled by a pre-defined schedule $\bar{\alpha}_t$, is iteratively added to a sample $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$ to produce a noisy sample $\boldsymbol{x}_t$, and a reverse process that learns how to denoise samples by predicting the added noise with a neural network $\epsilon_\theta(\boldsymbol{x}_t, t)$.

Once the denoising network is trained, DDPMs can draw new samples, starting from random Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, by following the inverse process transitions

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)), \tag{7}$$

where the mean is given from the predicted noise $\epsilon_\theta(\boldsymbol{x}_t, t)$ and the variance $\Sigma_\theta(\boldsymbol{x}_t, t))$ can either be fixed or learned [29, 10].

A.2 Denoising Diffusion Implicit Models

Song et al. [37] introduced DDIM, which allows deterministic sampling from a trained diffusion model with fewer steps. We adopt this sampling approach when using fewer than 100 sampling steps, following Nichol and Dhariwal [29]. In DDIM, the reverse sampling steps are defined as:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\, \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{“predicted } \boldsymbol{x}_0\text{”}} + \sqrt{1-\bar{\alpha}_{t-1}} \underbrace{\epsilon_\theta(x_t, t)}_{\text{“direction pointing to } x_t\text{”}}, \tag{8}$$

where the model uses a *prediction* of the clean image to perform the denoising.

A.3 Connections to Score Based Generative Models

The score function $s(x)$ of a probability distribution $p(x)$ is defined as the gradient of the log probability density function, $s(x) = \nabla_x \log p(x)$. Score-based generative modeling [38] aims to learn this score function of the data distribution from samples drawn from the same distribution. In the context of diffusion models, denoising diffusion has been shown to also approximate the score function [39]

$$s_\theta(x_t, t) = -\frac{\epsilon_\theta(\boldsymbol{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}. \tag{9}$$

A.4 Classifier Guidance

Data generated by diffusion models often fail to reproduce the clarity of the training data. A widely-used technique to increase fidelity of samples is classifier guidance [10], which uses the gradient of a classifier $p(y|\boldsymbol{x}_t)$, trained on noisy samples $\boldsymbol{x}_t$, to guide the denoiser network towards synthesizing more realistic samples. Classifier guidance modifies the predicted noise from a network with a term that maximizes the classifier likelihood

$$\epsilon'_\theta(\boldsymbol{x}_t, t) = \epsilon_\theta(\boldsymbol{x}_t, t) - \lambda\sqrt{1-\bar{\alpha}_t}\nabla_{\boldsymbol{x}_t} \log p(y \mid \boldsymbol{x}_t), \tag{10}$$

where λ is a hyperparameter, controlling the strength of the guidance.

B Implementation Details

B.1 Dynamic Guidance Algorithm with DDPM, DDIM and Classifier Guidance

We describe the algorithm using DDPM in Algorithm 1. In the main paper, we employed DDIM (Algorithm 2) as it is the more practical sampling algorithm for models larger than the toy 2D Gaussian setting.

Algorithm 1 Dynamic Guidance with DDPM

Input: timesteps T, dynamic guidance steps T_1 T_2, denoiser $\mu\theta$, classifier p_ϕ
$\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T \dots 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
  if $T_1 \geq t \geq T_2$ **then**
    $y^* = \arg\max p_\phi(y|\boldsymbol{x}_t)$
    $\boldsymbol{x}_{t-1} = \mu_\theta(\boldsymbol{x}_t, t) + \sigma_t^2 \lambda \nabla_{\boldsymbol{x}_t} \log p_\phi(y^*|\boldsymbol{x}_t) + \sigma_t \mathbf{z}$
  else
    $\boldsymbol{x}_{t-1} = \mu_\theta(\boldsymbol{x}_t, t) + \sigma_t \mathbf{z}$
  end if
end for
Return $\boldsymbol{x}_0$

Algorithm 2 Dynamic Guidance with DDIM

Input: timesteps T, dynamic guidance steps T_1 T_2, denoiser ϵ_θ, classifier p_ϕ
$\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$
for $t = T \dots 1$ **do**
  if $T_1 \geq t \geq T_2$ **then**
    $y^* = \arg\max_y \log p_\phi(y|\boldsymbol{x}_t)$
    $\hat{\epsilon} = \epsilon_\theta(\boldsymbol{x}_t) - \lambda\sqrt{1-\bar{\alpha}}\nabla_{\boldsymbol{x}_t} \log p_\phi(y^*|\boldsymbol{x}_t)$
  else
    $\hat{\epsilon} = \epsilon_\theta(\boldsymbol{x}_t)$
  end if
  $\tilde{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}_\theta(\boldsymbol{x}_t, t)\right)$
  $\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\tilde{\boldsymbol{x}}_0 + \sqrt{1-\alpha_{t-1}}\,\hat{\epsilon}_\theta(\boldsymbol{x}_t, t)$
end for
Return $\boldsymbol{x}_0$

B.2 2D Mixture of Gaussians

We create a synthetic toy dataset with a mixture of 25 2D Gaussians arranged in a square grid, similar to Aithal et al. [2]. Since the true distribution $p(\boldsymbol{x}_0)$ is known in this setup, we define hallucinations as samples $\boldsymbol{x}$ for which $p(\boldsymbol{x}) < p(\boldsymbol{x}_{4\sigma})$, where $\boldsymbol{x}_{4\sigma}$ is a threshold of 4 standard deviations away from the nearest Gaussian component. We cluster the generated dataset into 25 clusters and use the cluster assignments as labels for DG.

For the 2D Mixture of Gaussians dataset, as both Aithal et al. [2] and we observe, the number of training data and iterations greatly affect the number of hallucinations, so we train the DDPM [15] for 20k, 50k, and 100k iterations and generate 100k samples with DDPM for evaluation (Table 6). For variance filtering, we evaluate the performance when discarding 1%, 2.5%, and 5% of the generated samples. We observe that variance filtering requires discarding a very large percentage of samples to match Dynamic Guidance, while our method outperforms it in most cases.

Table 6: 2D Mixture of Gaussians dataset.

Method	# Training iterations		
	20k	50k	100k
Variance Filtering (1%)	6.1%	12.71%	17.8%
Variance Filtering (2.5%)	15.4%	34.5%	45.7%
Variance Filtering (5%)	34.5%	60.5%	71.3%
Dynamic Guidance	**69.5%**	**76.1%**	**72.1%**

B.3 Single Shapes - Mixed Shapes

For both Shape datasets, we generate 50k images to train the diffusion and classifiers, and sample 10k images using 50-step DDIM for evaluation. Hallucinations are quantified by automatically detecting shapes in the generated images using OpenCV. To evaluate variance filtering, we test thresholds that discard 1%, 5%, and 10% of generated samples. Even at the highest threshold (10%), variance filtering fails to reliably identify and remove hallucinations. In contrast, Dynamic Guidance consistently mitigates more than 50% of hallucinations across a broad range of guidance scales, and achieves over a 70% reduction at the optimal scale (Tables 1,2).

B.4 Hands

For the hands dataset [1], we train both the diffusion models and the classifiers on the 11k images provided. For evaluation, we sample 100 images with each method using 25-step DDIM and manually label each image as hallucinated or not. We perform this experiment 10 times and manually label all 3000 generated images (1000 unguided, 1000 for DG, and 1000 for CG). We report the mean and standard deviation for the experiments. To evaluate variance filtering, we test thresholds that discard 1%, 5%, and up to 10% of generated samples. Even at the highest threshold (10%), variance filtering

fails to reliably identify and remove hallucinations. DG vastly outperforms both CG and variance filtering, mitigating more than 45% of hallucinations on average (Table 3).

B.5 ImageNet

We build on the guided-diffusion codebase[2] and implement Dynamic Guidance following Algorithm 2. We use the pretrained 256x256 unconditional and conditional ADM models and the 256x256 noisy image classifier. For the experiment in Section 4.2.2 we train the noisy classifier using the training script provided in the guided-diffusion codebase and use the hyperparameters described in [10].

B.6 Text to Image

We use Stable Diffusion 2.1 with a DDIM scheduler for deterministic sampling. Generation uses 20 denoising steps, guidance scale 7.5, and 512×512 resolution. For all our experiments we use a default guidance window of [850, 300] on a 1000-step schedule. Within this window, we query the VLM every 5 denoising steps.

Classifier VLM Configuration We use Qwen2-VL-7B in bfloat16 precision. The VLM receives a multiple-choice prompt with branch options plus a "STAY" option for uncertain cases:

VLM Branch Selection Prompt

```
Look at this partially-generated image.  I need you to determine which
description best matches what is ALREADY FORMING in the image.

The image is ambiguous about:  {ambiguity_axis}

Options:
A. {branch_1}
B. {branch_2}
...
N. STAY - I cannot tell yet which option best matches the image

If none of them seem reasonable or there is no way to tell then pick N to
STAY. It's okay to pick a letter even if not completely sure.

Respond with ONLY the letter (A, B, C, etc.)  and nothing else.
```

The `{ambiguity_axis}` placeholder is replaced with the specific ambiguity type (e.g., "body pose", "breed", "species"). The branch options are populated from the corresponding branch prompts in Tables 7 and 8. The STAY option allows the VLM to defer judgment when the intermediate image is too noisy to classify reliably.

Branch Prompt Design For each base prompt, we define a set of branch prompts that resolve the specified ambiguity. We identify a primary axis of semantic ambiguity and generate a set of branch prompts that resolve it. The following prompt was used to generate candidate branches:

[2] https://github.com/openai/guided-diffusion

Instructions

You will be shown pairs of AI-generated images for the same prompt. For each pair, please:

1. **Choose your preferred image** from the pair by clicking on it, or click "Tie" if they are equally good.
2. **Mark hallucinations** using the checkbox under each image if you notice any.

What counts as a hallucination?

- **Incorrect anatomy:** Missing or extra body parts (e.g., hands, legs, ears, eyes) or body parts in wrong places
- **Wrong count:** Incorrect number of body parts (e.g., a 5-legged dog, 3 arms)
- **Missing objects:** Objects described in the prompt that don't appear in the image (e.g., prompt says "a vase and a plate on a table" but image only shows a plate)

Figure 5: Instructions given to the participants in the user study.

Branch Generation Prompt

```
Given a text-to-image prompt, identify the primary axis of ambiguity (e.g.,
body pose, breed, species, object type) and generate branch prompts that
resolve it.

Requirements:
      • Each branch should be mutually exclusive
      • Branches should preserve the base structure while resolving the
        ambiguity
      • Keep branches simple and within the model's capability
      • Avoid low-quality or rare interpretations

Base prompt:  "{base_prompt}"

Output format:
Ambiguity axis:  <axis>
Branches:
- <branch_1>
- <branch_2>
...
```

Tables 7 and 8 list all prompts used in the user study. Figure 5 shows the instructions given to the participants.

B.7 Compute details

For all experiments, we use a cluster of NVIDIA RTX 6000 Ada GPUs. For the discrete conditioning models (Shapes, ImageNet), Dynamic Guidance requires training a noisy image classifier. This incurs some computational overhead during training, which, however, is significantly lower than that of training the Diffusion Model itself. We trained the classifier on pseudo-classes for the experiment of Section 4.2.2 on 4 NVIDIA RTX 6000 Ada GPUs for 50 hours. For the rest of the experiments we used pre-trained publicly availanble checkpoints. For a more extensive analysis of the compute required to train ADM models and classifiers, we refer to Appendix A of [10].

During generation, when using discrete condition models, DG requires a forward and a backward pass through the classifier, exactly like CG, for every step that it is active. This means that it requires some computational overhead compared to the unguided model. Crucially, compared to CG, DG can be applied to a subset of generation steps only, so it incurs a lower computational cost than CG. To compare the three settings, we sample 160 256x256 images with the unguided model, DG and CG in a single NVIDIA RTX 6000 Ada and compare the generation time in Table 9.

Table 9: Comparison of generation times for different guidance methods.

Method	**Generation time (s)**
ADM	351
Classifier Guidance (applied to all timesteps)	417
Dynamic Guidance (applied to timesteps 800–400)	380

Table 7: Base prompts and their corresponding branch prompts used in the user study (Part 1/2).

Base Prompt	**Ambiguity**	**Branch Prompts**
a cat in a living room	body pose	• a cat sitting upright in a living room • a cat lying down curled up in a living room • a cat standing on all fours in a living room • a cat stretching with arched back in a living room • a cat walking across a living room • a cat grooming itself in a living room • a cat jumping in a living room • a cat scratching furniture in a living room
a dog playing fetch	body pose	• a dog running with a ball in its mouth • a dog jumping to catch a ball • a dog waiting for a ball to be thrown • a dog chasing a ball • a dog returning with a ball • a dog dropping a ball at feet
a cat sleeping	breed	• a persian cat sleeping • a siamese cat sleeping • a maine coon cat sleeping • a tabby cat sleeping • an orange cat sleeping • a black cat sleeping • a white cat sleeping • a gray cat sleeping • a calico cat sleeping
a dog on a couch	breed	• a poodle on a couch • a husky on a couch • a dachshund on a couch • a bulldog on a couch • a corgi on a couch • a chihuahua on a couch • a golden retriever on a couch • a shiba inu on a couch • a pit bull on a couch
a bird on a branch	species	• a robin on a branch • a blue jay on a branch • a cardinal on a branch • a sparrow on a branch • a crow on a branch • a parrot on a branch • an owl on a branch • a woodpecker on a branch

Table 8: Base prompts and their corresponding branch prompts used in the user study (Part 2/2).

Base Prompt	**Ambiguity**	**Branch Prompts**
a boat on water	vessel type	• a sailboat on water • a speedboat on water • a rowboat on water • a fishing boat on water • a yacht on water • a canoe on water • a kayak on water
a cat on a windowsill	body pose	• a cat sitting upright on a windowsill • a cat lying stretched out on a windowsill • a cat curled up on a windowsill • a cat looking out the window on a windowsill • a cat sleeping on a windowsill
a dog by a fireplace	body pose	• a dog sleeping by a fireplace • a dog sitting by a fireplace • a dog lying on its side by a fireplace • a dog with head on paws by a fireplace • a dog stretching by a fireplace
a frog	pose	• a frog sitting on a lily pad • a frog jumping • a frog in water • a frog on a rock
a rabbit	pose	• a rabbit sitting upright • a rabbit hopping • a rabbit lying down • a rabbit eating • a rabbit in grass

C Additional Comparisons / Metrics

Figure 6: Images generated with Classifier and Dynamic Guidance using the same initial noises. **(Top)** Given a specific label ("llama") that is misaligned with the initial noise, the model can generate low-quality samples that visually resemble hallucinations. **(Bottom)** Even for randomly selected labels, when those are fixed, misalignment can produce generations that would be considered hallucinations.

Table 10: Performance of Classifier (CG) and Dynamic Guidance (DG) on Hallucination-related metrics on ImageNet-1k generation.

	Inception			CLIP	
Method	IS↑	Prec↑	Rec↑	Prec↑	Rec↑
Uncond. ADM	34.24	0.53	0.61	0.60	0.26
+ CG $\lambda = 1$	28.52	0.47	0.62	0.57	0.25
+ DG $\lambda = 1$	49.63	0.62	0.59	0.66	**0.27**
+ CG $\lambda = 2$	38.39	0.55	**0.64**	0.63	**0.27**
+ DG $\lambda = 2$	59.31	0.70	0.56	0.70	0.26
+ CG $\lambda = 10$	83.19	0.69	0.55	0.74	**0.27**
+ DG $\lambda = 10$	**88.49**	**0.77**	0.52	**0.77**	0.26
+ CG $\lambda = 20$	75.84	0.62	0.55	0.70	0.23
+ DG $\lambda = 20$	80.99	0.70	0.52	0.75	0.23

We also compute Inception and CLIP FID and observe that in most settings Dynamic Guidance outperforms Classifier Guidance. For high guidance scales ($\lambda = 10$), Dynamic Guidance tends

to generate classes that are better matched with a wider range of initial noises, and so it ends up creating a distribution of generated samples that is non-uniform with regard to the ImageNet classes. This greatly affects Inception FID since the Inception model is heavily biased towards balanced ImageNet generation (trained on ImageNet-1k), whereas performance on CLIP FID is not affected. To fairly compare Dynamic Guidance to Classifier Guidance that strictly enforces a balanced generated distribution for evaluation, we generate a larger amount of samples and perform stratified sampling on the generated set to approximate a balanced distribution. We report the results in Table 11 in the row titled + *DG* $\lambda = 10$ *(balanced)*.

Table 11: Inception and CLIP FID on ImageNet.

	FID	
Method	Inception↓	CLIP↓
Uncond. ADM	37.48	42.93
+ CG $\lambda = 1$	43.72	48.02
+ DG $\lambda = 1$	27.46	38.21
+ CG $\lambda = 2$	32.96	41.15
+ DG $\lambda = 2$	25.36	36.66
+ CG $\lambda = 10$	17.05	30.01
+ DG $\lambda = 10$	25.57	32.68
+ DG $\lambda = 10$ (balanced)	**15.52**	**27.93**
+ CG $\lambda = 20$	21.45	40.50
+ DG $\lambda = 20$	25.58	37.07

Table 12: Density and Coverage on ImageNet-1k generations, based on CLIP.

Method	**Density** ↑	**Coverage** ↑
Uncond.	0.7003	0.6662
+CG ($\lambda = 1$)	0.5366	0.4473
+DG ($\lambda = 1$)	0.7023	0.5921
+CG ($\lambda = 10$)	0.9058	0.7516
+DG ($\lambda = 10$)	0.9724	0.6761
+CG ($\lambda = 20$)	0.7999	0.6943
+DG ($\lambda = 20$)	0.9285	0.6385

We also report additional metrics aiming to better illustrate the trade-off between reducing hallucinations and loss of diversity in generation. We compute diversity and coverage, as described by Naeem et al. [28] using CLIP, and report the results in Table 12. We see that DG improves the density of the generations from 0.70 to 0.97 while not sacrificing coverage (0.66 vs 0.67).

C.1 Additional Analysis of Dynamic Guidance

Score Sharpening and Guidance Gradients. We identify that our trained β-VAE has learned latent dimensions that change the image in distinct, independent ways. We provide visual examples in Figure 7(a): latent dimensions 0 and 9 control the appearence of shapes placed in different positions in the image, while other dimensions, like 5, control the position of shapes. We show that Dynamic Guidance isolates latent dimensions corresponding to hallucinations (dimension 9), while not affecting unrelated ones (dimension 5), as the gradients along hallucination-relevant directions are strong and informative, whereas gradients along irrelevant directions are noisy and close to zero (Figure 7(b)). This is more pronounced when classes are selected so that interpolation between them directly aligns with hallucinations, like in the setting of the Single Shapes dataset.

C.2 Hallucination Mitigation during Sampling

To understand how hallucinations form during sampling and how Dynamic Guidance changes the sampling process to mitigate them, we visualize the sampling process for a sample that would otherwise be a hallucination and observe how Dynamic Guidance corrects it (Figure 8). We also show an example where classifier guidance fails to fix a hallucinated sample and how it is fixed using Dynamic Guidance (Figure 9).

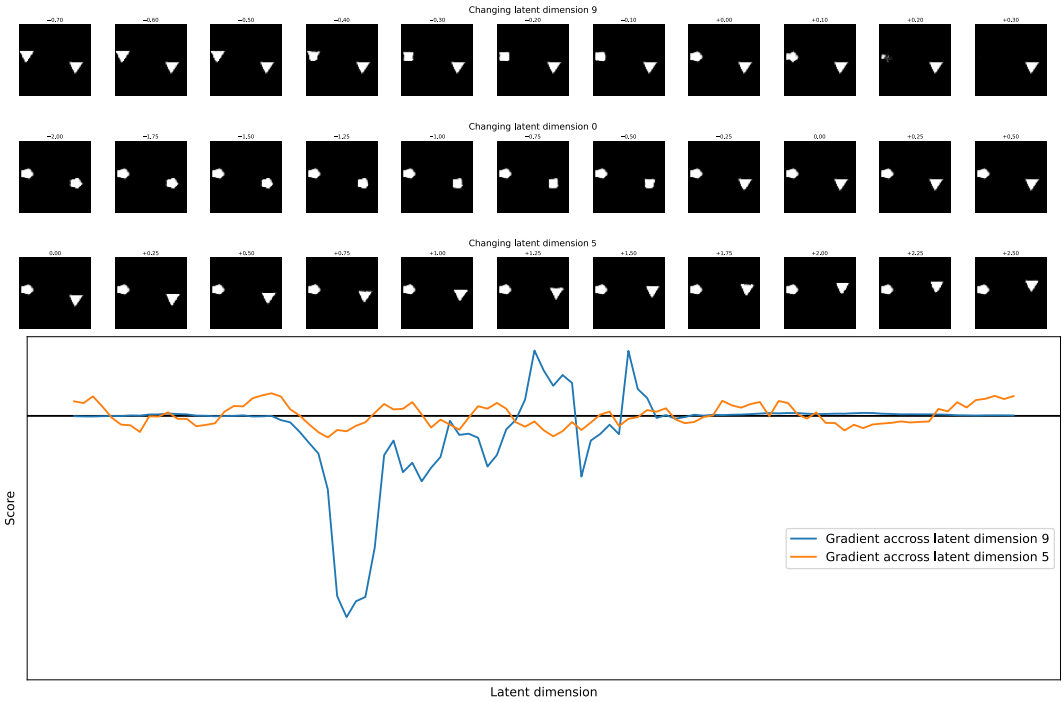


Figure 7: (a) Change in different latent dimensions learned by the β-VAE alters the image in distinct ways; latent dimension 9 controls the appearance of the shape on the left, latent dimension 0 controls the appearance of the shape on the right, and latent dimension 5 controls the position of the shape on the right. (b) Dynamic Guidance isolates latent dimensions corresponding to hallucinations (dimension 9), while not affecting unrelated ones (dimension 5), as the gradients along hallucination-relevant directions are strong and informative, whereas gradients along irrelevant directions are noisy and close to zero. This is more pronounced when classes are selected so that interpolation between them directly aligns with hallucinations.

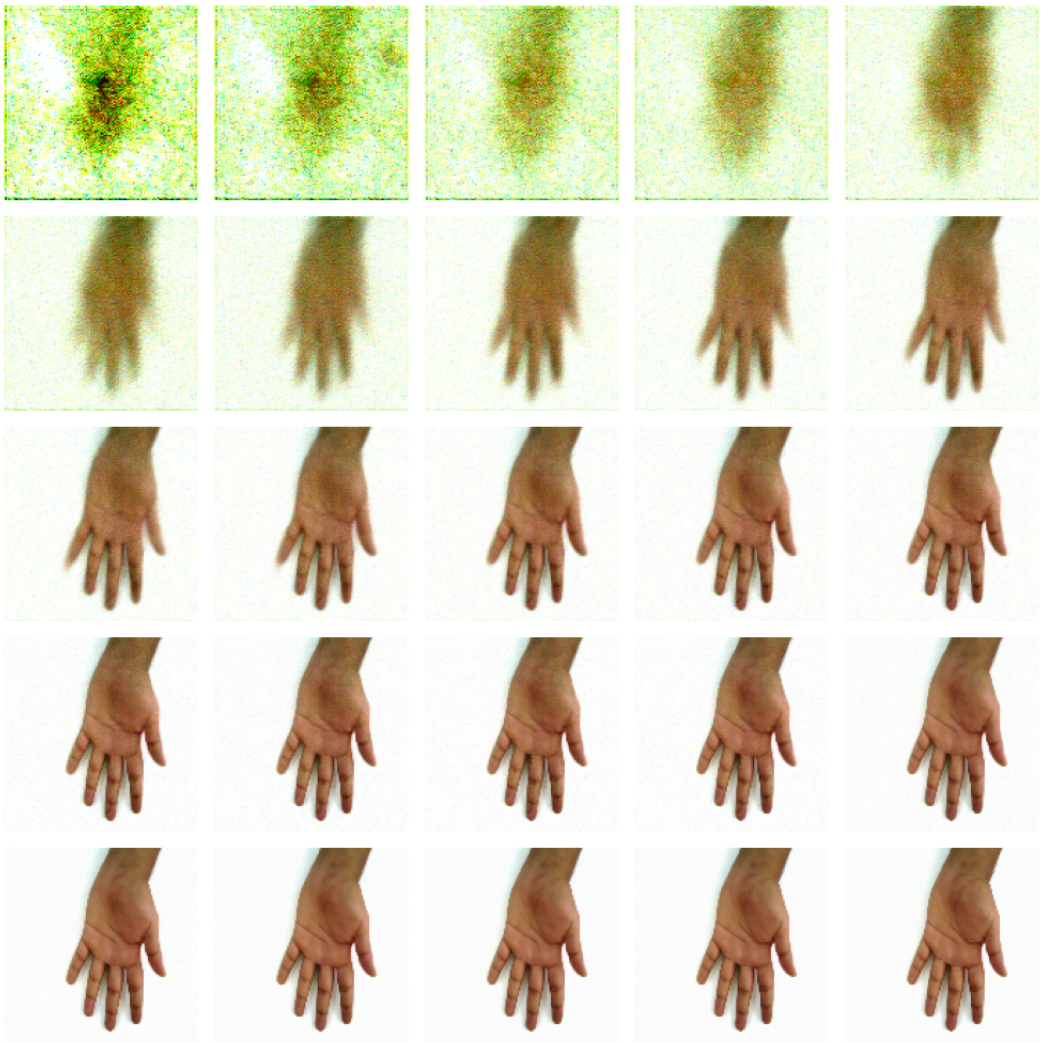


Figure 8: $\hat{\boldsymbol{x}}_0$ during sampling with Dynamic Guidance using initial noise $\boldsymbol{x}_T$ that would result in a hallucination. We see that Dynamic Guidance guides the model to generate the thumb that would otherwise be missing, resulting in a sample with correct anatomy.

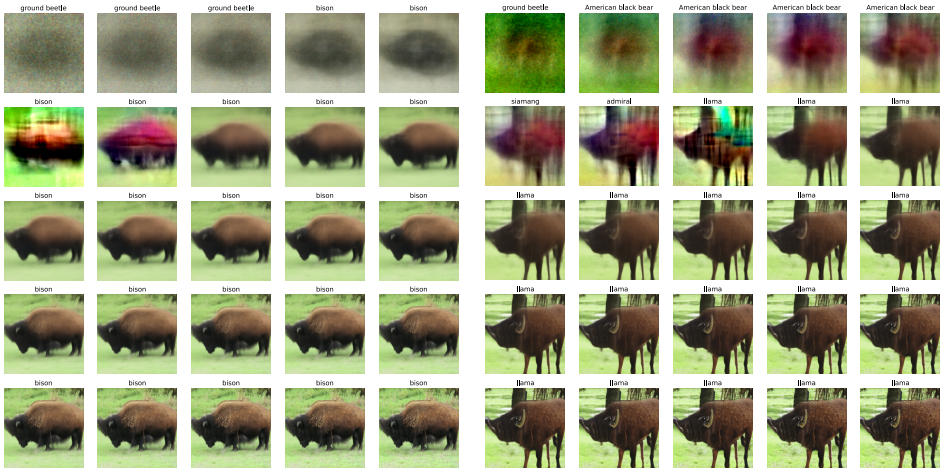


Figure 9: (a) DG steps: When using dynamic guidance, the generated image has the label “bison". The initial noise adds a blob of black pixels in the middle on a green background, for which the local mode is a bison, as predicted by the classifier. By applying the gradients from dynamic guidance, we end up at a realistic-looking bison image, where both the model and classifier agree. (b) CG steps: For the same initial noise, we choose a class that is unlikely to contain a large blob of black pixels in the middle (e.g., “llama"). We see that the classifier guidance gradient updates attempt to correct the image by changing the shape of the generated object and adding thin legs.

C.3 Importance of Adaptive Selection

To highlight the importance of dynamically adjusting the guidance signal we investigate how often the dominant class changes during sampling in Figure 10. We show that the dominant class can change multiple times during the intermediate steps where we perform the guidance. The class changes more in the early timesteps, where the image has not fully formed yet, and multiple candidate classes could potentially be feasible generations. To quantify this, we generate 100 images with an unconditional ADM trained on ImageNet and Dynamic Guidance and observe how often the dominant class changes for 25-step generation on average (Figure 11). We observe that for most samples the label changes at least 2 times, which is expected, as for early timesteps the image has not formed fully yet and the sample could feasibly belong to multiple similar classes.

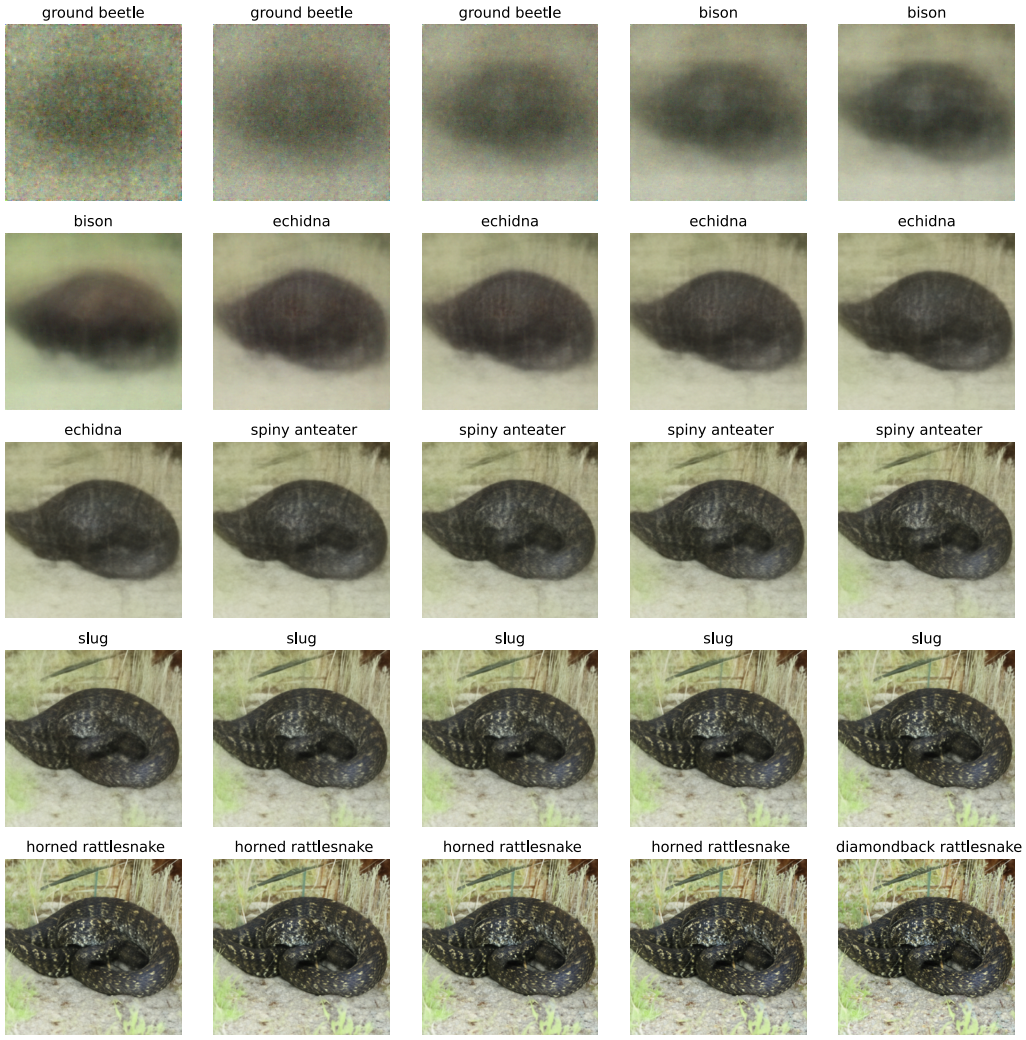


Figure 10: The intermediate image can be close to multiple modes during generation. The selected mode can change multiple times during sampling, and Dynamic Guidance adaptively chooses the closest one.

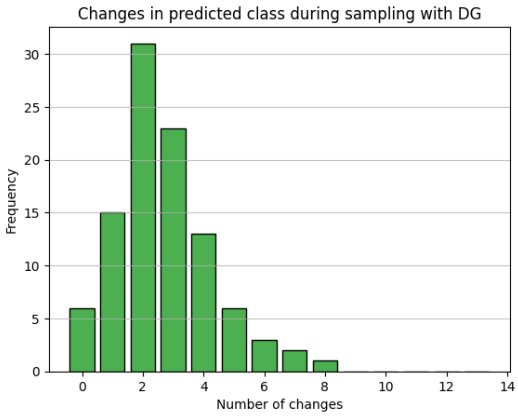


Figure 11: Histogram for the number of selected class changes during generations.

C.4 Sensitivity to Classifier Quality

To assess the dependence of Dynamic Guidance on specific strong classifiers, we evaluate using weaker classifiers corresponding to earlier training checkpoints. In Figure 12 we provide hallucination reduction metrics on ImageNet using DG with checkpoints of the DiNO pseudo-class classifier during different stages of its training. We observe that DG performs well even with a weaker classifier and is effective in mitigating hallucinations (high precision). The gradients provided by the classifier are informative enough to guide the generation away from hallucinations, even when the classifier is slightly worse at differentiating between pseudo-classes or its predictions are less certain.

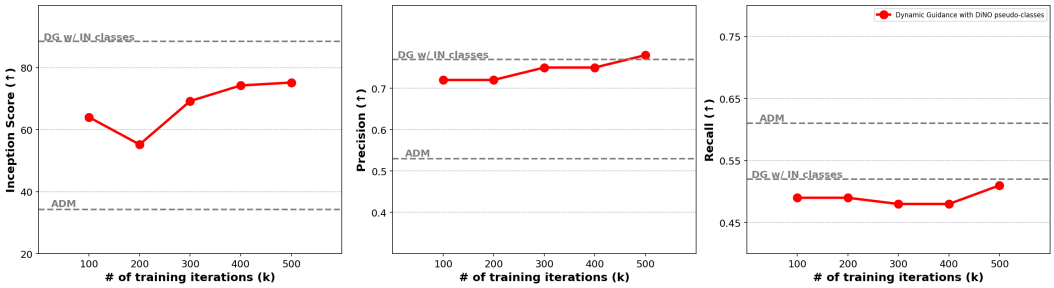


Figure 12: Hallucination related metrics for different training iterations.

C.5 Potential Class Bias in Generation

To better understand the potential bias introduced by the noisy image classifier, we plot the final classification for the generated images using Classifier Guidance and Dynamic Guidance for different guidance scales (Figures 23 and 24).

C.6 Additional Qualitative Results

Hands-11k. Figures 15 and 16 show 100 images of hands generated with DDIM and Dynamic Guidance, respectively.

ImageNet. Figures 19, 20, 21, and 22 show examples of random images generated with Classifier or Dynamic Guidance with different guidance scales. Figures 17 and 18, we show the difference in the norms of the denoising steps when using Classifier or Dynamic Guidance.

Class-Conditional ImageNet with Pseudo-Class Guidance. In Figure 25, we show that Dynamic Guidance can be applied to a conditional model and improve generations even though the diffusion and guidance signals are not conditioned on the same classes.

Figure 13: Using the generated, more complex, branches as base prompts can lead to worse generations and new hallucinations.

Text-to-Image. In Figures 13 and 14 we show examples of hallucinations in text-to-image generation that are fixed with Dynamic Guidance.

Figure 14: Examples of hallucinations in text-to-image generation that are fixed with Dynamic Guidance. In the three images generated by the baseline, the animals have an extra, hallucinated limb that is removed with Dynamic Guidance.

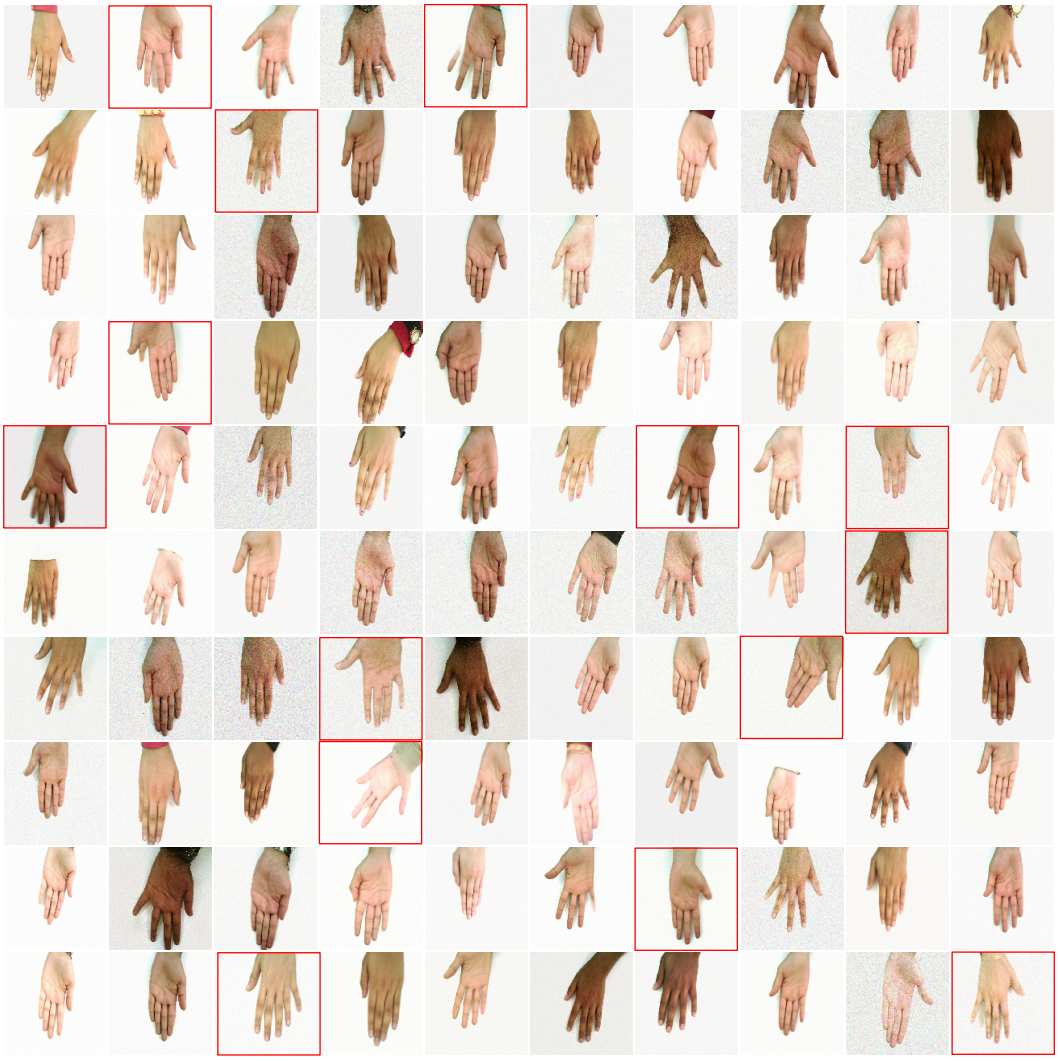


Figure 15: Generated samples from the hands dataset using DDIM. Hallucinations in red.

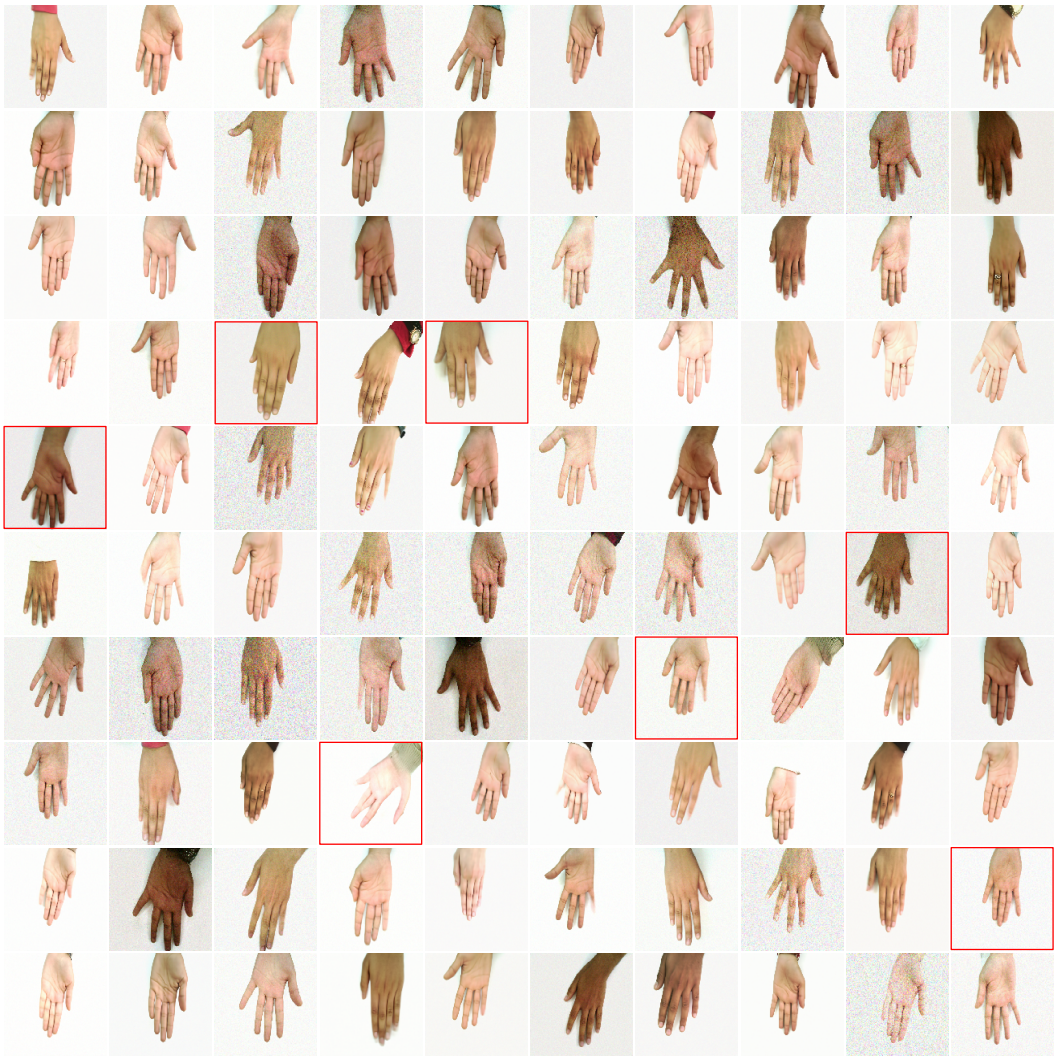


Figure 16: Generated samples from the hands dataset using Dynamic Guidance. Hallucinations in red.

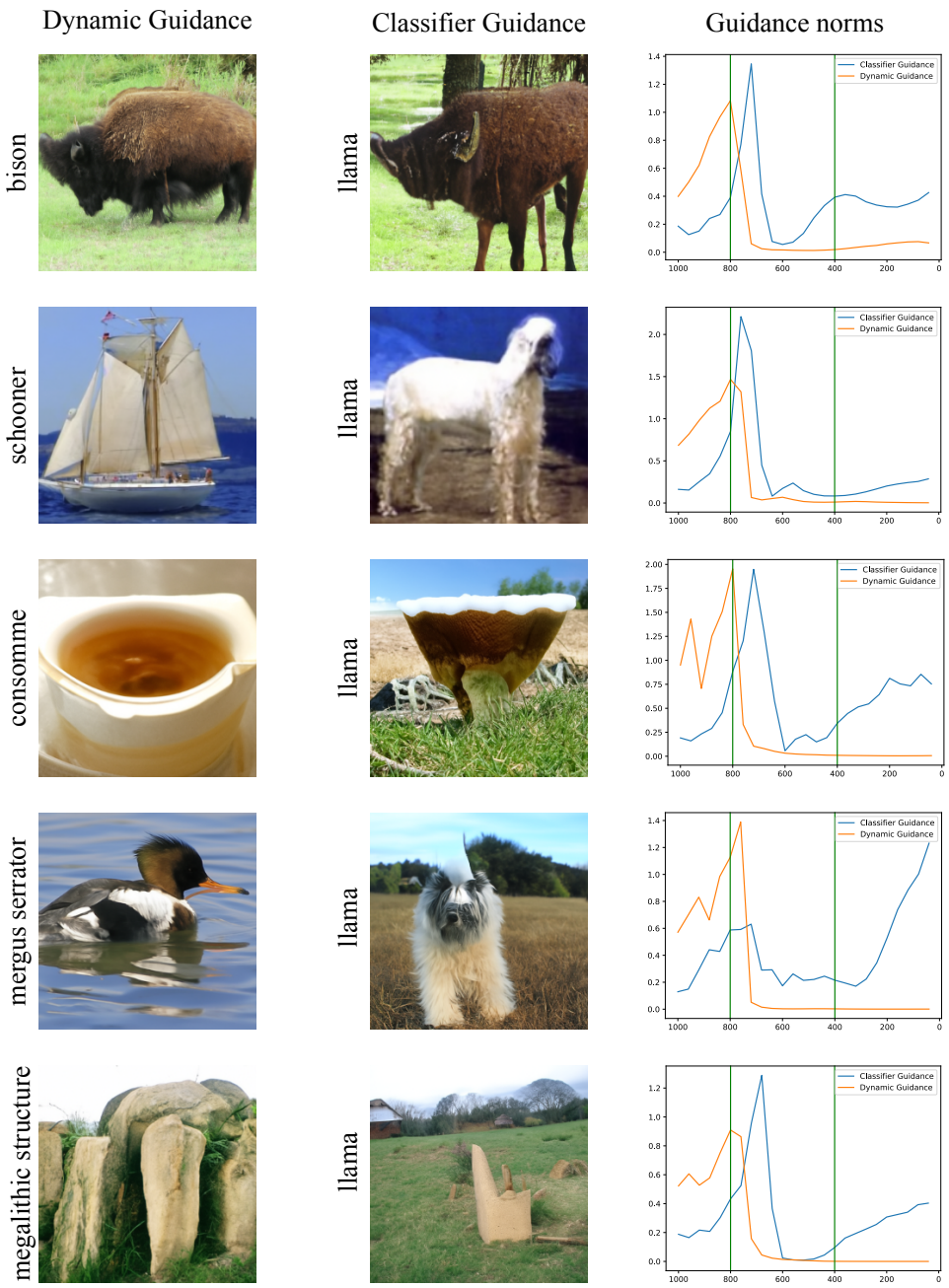


Figure 17: Generated samples from ImageNet using Classifier Guidance with a fixed label and Dynamic Guidance. In classifier-guided samples, the norm of the denoising step gets bigger towards the end of sampling, meaning that the required denoising steps become significantly larger.

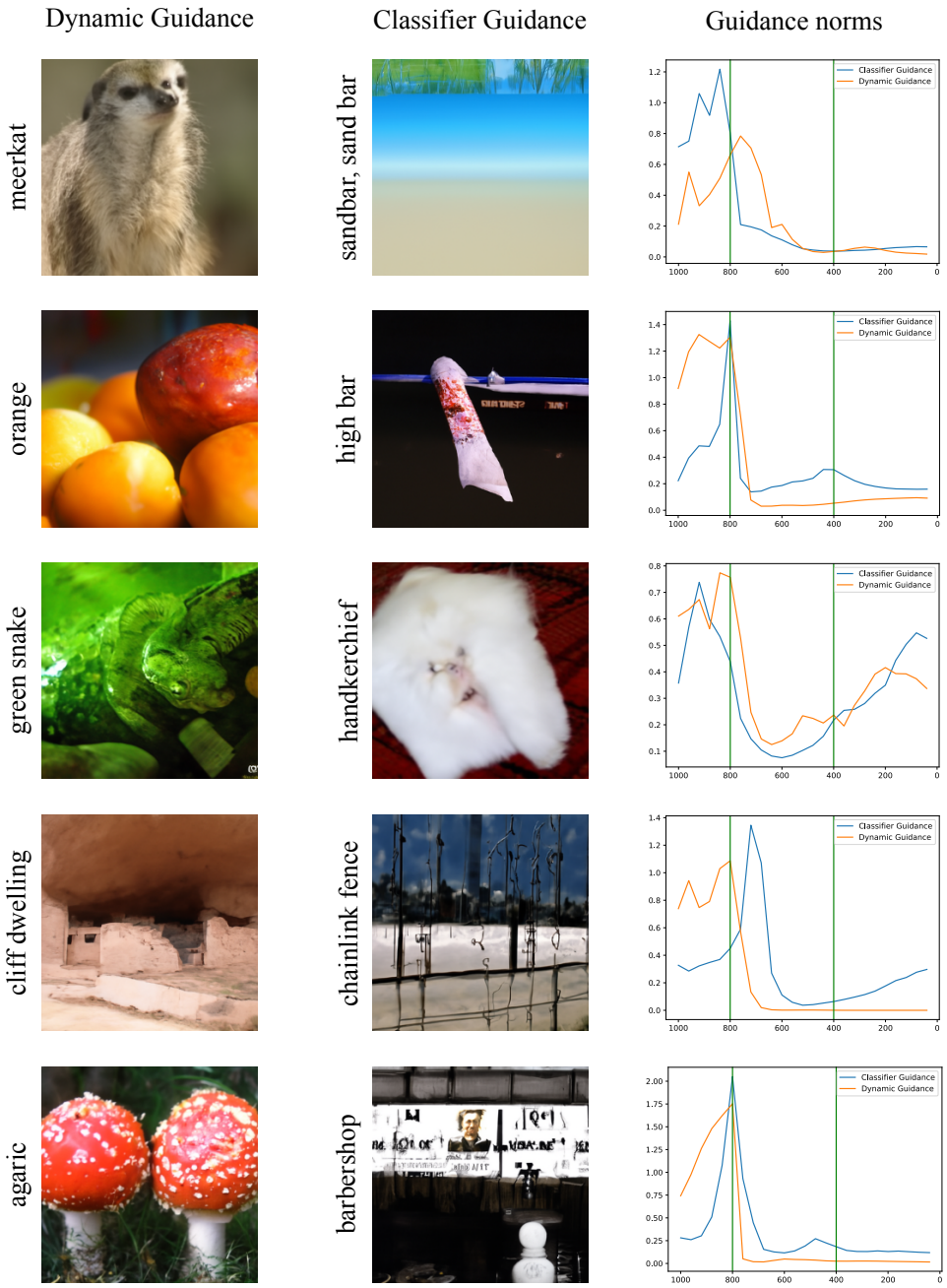


Figure 18: Generated samples from ImageNet using Classifier Guidance with a random label and Dynamic Guidance. In classifier-guided samples, the norm of the denoising step gets bigger towards the end of sampling, meaning that the required denoising steps become significantly larger.

Figure 19: Random ImageNet samples generated with Classifier Guidance using $\lambda = 1$.

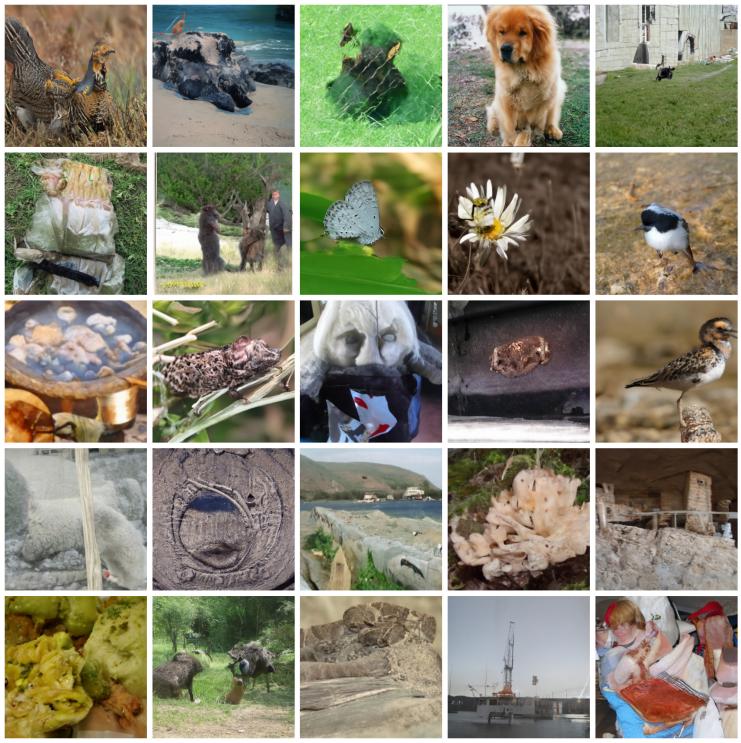


Figure 20: Random ImageNet samples generated with Dynamic Guidance using $\lambda = 1$.

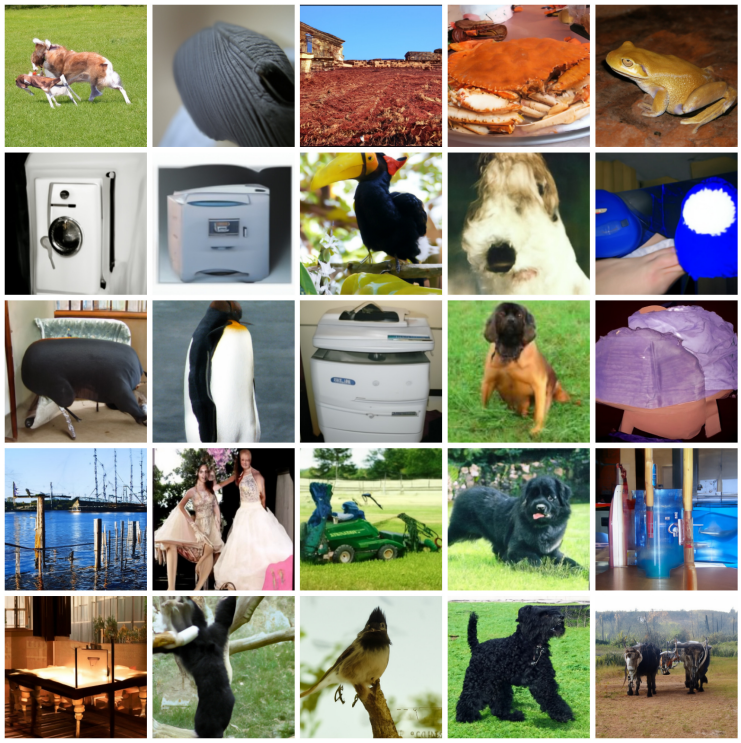


Figure 21: Random ImageNet samples generated with Classifier Guidance using $\lambda = 10$.

Figure 22: Random ImageNet samples generated with Dynamic Guidance using $\lambda = 10$.

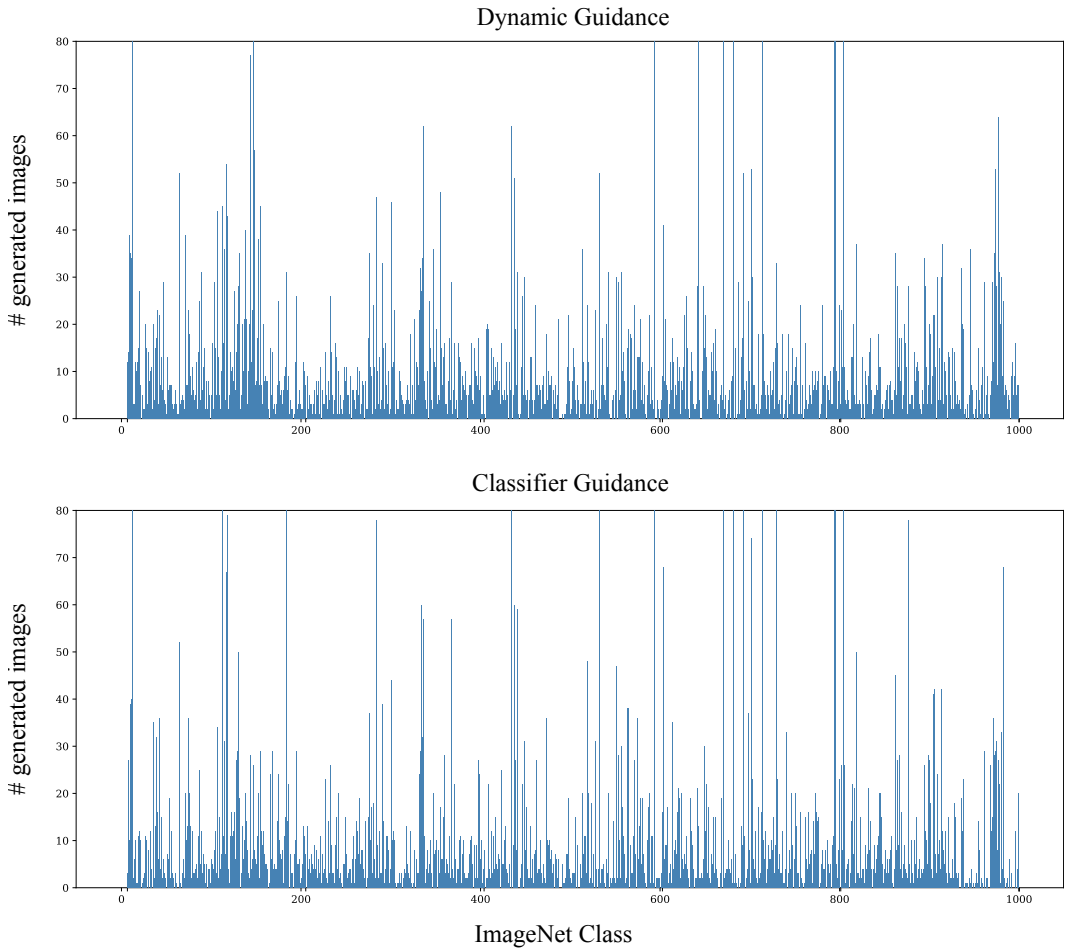


Figure 23: Distribution of final predicted ImageNet classes for samples generated with Classifier and Dynamic Guidance using $\lambda = 1$.

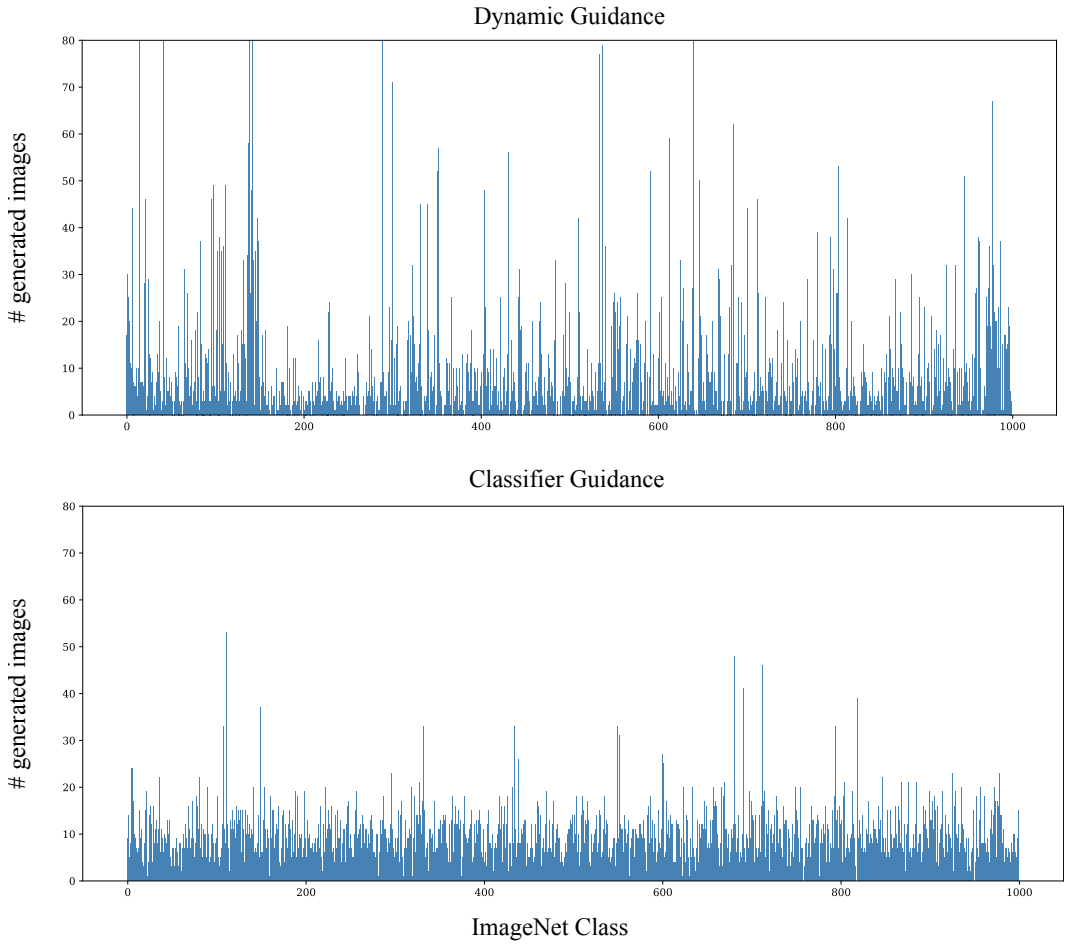


Figure 24: Distribution of final predicted ImageNet classes for samples generated with Classifier and Dynamic Guidance using $\lambda = 10$.

Figure 25: Samples generated from an ImageNet class-conditional ADM model, without guidance, and with Dynamic Guidance using the classifier trained on pseudoclasses created by clustering with DINOv2 embeddings (Section 4.2.2). We show that the conditioning and guidance labels do not need to be the same; Dynamic Guidance can improve generations even though the diffusion and guidance signals are not conditioned on the same classes. We practically see that the two can act orthogonally, with class-conditioning guiding the sample towards specific class-related attributes and and dynamic guidance helping avoid bad generations within the class. This is especially apparent in the second and last examples where Dynamic Guidance prevents hallucinations (cat with no face, dog with 5 visible legs) by guiding the samples towards clusters that represent multiple animals in an image.

D Potential Societal Impact

This work studies how to controllably mitigate hallucinations in diffusion models. Currently, people have been using common forms of hallucinations such as incorrect anatomy in generated pictures of humans or animals to detect and identify synthetically generated content. This means that developing methods that improve diffusion models by mitigating such hallucinations can potentially make identifying synthetically generated content such as deepfakes significantly more challenging for humans, even if automated detection is unaffected.

However, we contend that, as models naturally improve through increased scale and data diversity there is going to be a decrease in generated artifacts that are common across different models and very easily identifiable such as poorly generated hands, even though hallucinations will still be present. By understanding how to controllably mitigate hallucinations, our work facilitates the responsible and effective use of diffusion models in practical scenarios. Additionally, this evolution shows the requirement for additional future work on watermarking synthetically generated content so that it can be identified beyond the reliance on common visible failure modes of image generators.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that the proposed dynamic guidance can mitigate hallucinations in diffusion models. Our experiments show confidently that dynamic guidance is an effective way to reduce hallucinations, as defined for each specific dataset/model examined.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a paragraph in the main text describing limitations of our method, specifically regarding the bias a classifier can introduce to the sampling process and the diversity-hallucination trade-off.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: There are no theoretical results to prove.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide pseudocode to reproduce the proposed method. All datasets and models are adequately described or taken from public sources.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not provided at this time. We will provide an implementation upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details are provided in the experiments section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars where suitable, specifically in the experiments requiring human annotation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the compute resources used in Section B.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS code of conduct.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Societal impacts are discussed in Section D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We experiment with existing models that already have such safeguards in place or models that produce 'toy' images.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Any public datasets or models used adhere to the licensing provided.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The instructions given to participants are included in Figure 5 in the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The user study conducted collected no data or identifiable information **about** the participants, so no IRB approval was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our proposed Dynamic Guidance uses an LLM to analyze and synthesize the text prompts used for generating images in one experiment. Usage is described in detail in Section B.6.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.