
DISCO: DISCcrete nOise for Conditional Control in Text-to-Image Diffusion Models

Longquan Dai, Wu Ming, Dejiao Xue, He Wang, and Jinhui Tang*
Nanjing University of Science and Technology, Nanjing, China

{dailongquan, wuming, xuedejiao, wanghe, tangjinhui}@njjust.edu.cn

Abstract

A major challenge in using diffusion models is aligning outputs with user-defined conditions. Existing conditional generation methods fall into two major categories: classifier-based guidance, which requires differentiable target models and gradient-based correction; and classifier-free guidance, which embeds conditions directly into the diffusion model but demands expensive joint training and architectural coupling. In this work, we introduce a third paradigm: DISCcrete nOise (DISCO) guidance, which replaces the continuous conditional correction term with a finite codebook of discrete noise vectors sampled from a Gaussian prior. Conditional generation is reformulated as a code selection task, and we train prediction network to choose the optimal code given the intermediate diffusion state and the conditioning input. Our approach is differentiability-free, and training-efficient, avoiding the gradient computation and architectural redundancy of prior methods. Empirical results demonstrate that DISCO achieves competitive controllability while substantially reducing resource demands, positioning it as a scalable and effective alternative for conditional diffusion generation. Code is available at <https://github.com/dailongquan/disco>.

1 Introduction

Diffusion models [4, 8] have achieved impressive success across diverse domains [11, 25, 47, 51], scaling to billions of samples [27] and establishing themselves as foundational generative models. They are particularly effective for conditional generation tasks [1, 13, 32, 44, 46] and support a wide range of downstream applications [26, 38, 39, 30, 40, 31]. Consequently, the challenge of conditional generation—tailoring outputs to satisfy user-defined constraints such as labels, attributes, or spatiotemporal cues—is becoming increasingly critical.

Conditional generation in diffusion models typically follows one of two paradigms: classifier-based guidance[10] and classifier-free guidance[15]. Both require training a dedicated model tailored to each conditioning signal. More recently, training-free guidance, a subclass of classifier-based methods, has emerged. These approaches leverage off-the-shelf, differentiable target predictors—such as classifiers, loss functions, or energy functions—to steer generation without additional training. The predictor evaluates sample quality and provides gradients to guide the diffusion process toward desired outputs.

Classifier-based guidance[10] trains a noise-conditional classifier to estimate the class probability $p_{\theta}(c|z_t)$. Training-free guidance[1, 13, 32, 44, 46] builds on this idea from an energy-based perspective [46], using pretrained target networks—such as depth or segmentation models—to approximate $p_{\theta}(c|z_t)$ without training a dedicated classifier. While these predictors still require pretraining, their decoupling from the diffusion model reduces overall computational cost. A key limitation, however, is that these approaches require differentiable predictors, as guidance relies on computing the gradient $\nabla_{z_t} \log p(z_t|c)$ to apply conditional corrections during sampling.

*Corresponding author.

Classifier-free guidance[15] removes the need for an external classifier by interpolating between the score estimates of a conditional diffusion model and its jointly trained unconditional counterpart to form a conditional correction term. This approach relies on conditioning mechanisms [48] that adapt an unconditional model into a conditional one. A major advantage of classifier-free guidance is its ability to generate outputs more faithfully aligned with the conditioning input [18]. However, this comes at the cost of increased training complexity: the conditional branch modifies features throughout the diffusion model, requiring the full model to be loaded during training. In contrast, classifier-based methods allow the conditional network to be trained independently.

In this paper, we propose a third approach: DISCcrete nOise (DISCO) guidance, which overcomes key limitations of existing methods. Both classifier-based and classifier-free methods aim to estimate a conditional correction term $\Delta(z_t, c)$, which transforms the zero-mean Gaussian noise used in unconditional denoising into a non-zero-mean noise for conditional denoising. Our insight is that, under certain conditions, this non-zero-mean noise can be effectively approximated using a finite codebook of discrete vectors, pre-sampled from a Gaussian distribution. This reframes conditional generation as a code selection problem: given the current diffusion state z_t and conditioning input c , the model selects an optimal code from the codebook. To realize this, we introduce a discrete noise prediction network that learns to perform this selection efficiently and effectively. In summary, the contribution of our DISCO are that:

Differentiability-Free: Our discrete noise prediction network directly selects the optimal code to substitute for the non-zero-mean Gaussian noise, avoiding the need to compute gradients of the target predictor as required in classifier-based guidance.

Training Efficiency: Training the discrete noise prediction network does not require loading or fine-tuning the diffusion model, significantly reducing computational overhead. This is because both inputs— z_t and c —as well as the discrete code indices can be precomputed.

Alignment Improved: Our method explicitly trains the prediction network to select control noise, providing direct supervision for controllability. In contrast, classifier-based guidance relies on auxiliary classification loss, which does not directly align with the diffusion process.

2 Related Work

Classifier-based guidance has primarily focused on training-free approaches, where an independently trained classifier is used to guide the diffusion process during inference without modifying the diffusion model itself. In contrast, classifier-free guidance relies on learned conditioning, typically by integrating an auxiliary conditional network into the diffusion model during training. In this section, we briefly review both categories to contextualize our proposed DISCO method, which introduces a third paradigm for conditional generation that is distinct from both classifier-based and classifier-free approaches.

Classifier-based guidance was introduced by Dhariwal and Nichol [10], who trained a time-dependent classifier to approximate the posterior $\log p_t(c|z_t)$ using a pre-trained diffusion model. Later works estimate the gradient $\nabla_{z_t} \log p_t(c|z_t)$ at inference using pre-trained predictors. Chung et al. [6] applied Tweedie’s formula to linear inverse problems, which was extended to broader conditional generation by Chung et al. [7], Yu et al. [46], Bansal et al. [1], and Wang et al. [37]. Song et al. [32] reduced bias via multiple samples from a noisy Gaussian prior. Zhu et al. [52] introduced manifold projection, later refined by He et al. [13] with an autoencoder enforcing guidance in the tangent space. Yang et al. [43] further improved this by mitigating manifold drift without strong assumptions. Despite these advances, classifier-based guidance fails when the target predictor is non-differentiable.

Classifier-free guidance extends the conditional processing capabilities of diffusion models by modifying the architecture to directly incorporate conditioning information. ControlNet [48] proposes to utilize the trainable copy of the UNet encoder in the diffusion model to encode extra condition signals into latent representations and then apply zero convolution to inject into the backbone of the UNet in diffusion modal. However, ControlNet is a single-modality framework and requires a separate model for each modality. To address this, unified ControlNet-like models [17, 18, 20, 29, 22, 49, 50, 34] are proposed to handle diverse control signals by inject multi-modality into original diffusion model. Despite their flexibility, these models are typically resource-intensive to train.

3 Preliminaries

In this section, we delve into two fundamental theorems that theoretically guarantee the practicality of our discrete noise guidance approach.

Theorem 1. Let $n \geq 2$ denote the ambient dimension, and let $\theta = \angle(\boldsymbol{\mu}, \boldsymbol{\epsilon})$ be the angle between the mean vector $\boldsymbol{\mu}$ and a random vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$. Then, $v = \cos^2 \theta = \left(\frac{\langle \boldsymbol{\mu}, \boldsymbol{\epsilon} \rangle}{\|\boldsymbol{\mu}\| \cdot \|\boldsymbol{\epsilon}\|} \right)^2$ follows the non-central beta distribution:

$$v \sim \text{Beta}(\alpha, \beta; \lambda). \quad (1)$$

where $\alpha = \frac{1}{2}$, $\beta = \frac{n-1}{2}$, and $\lambda = \|\boldsymbol{\mu}\|^2$. Let ${}_2F_1(a, b; c; z)$ be a Gaussian hypergeometric function. The expectation of $\cos^2 \theta$ is given by

$$\mathbb{E}[v] = \frac{\alpha + \lambda}{\alpha + \beta + \lambda} \cdot {}_2F_1\left(1, \alpha + 1; \alpha + \beta + 1; \frac{1}{1 + \lambda}\right) \cdot \frac{1}{\alpha + \beta}. \quad (2)$$

This theorem can be leveraged to assess whether a vector $\boldsymbol{\epsilon}$ is statistically consistent with being drawn from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$. Specifically, one can evaluate its cumulative probability under the above distribution $p = P(v \leq \cos^2 \angle(\boldsymbol{\mu}, \boldsymbol{\epsilon}))$, where $v \sim \text{Beta}(\alpha, \beta; \lambda)$. A large p -value suggests that the observed angle is likely under the assumed distribution. Thus, we have the following definition:

Definition 1. Under the setup described above, we consider that a vector $\boldsymbol{\epsilon}$ is likely to be sampled from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ if $\left(\frac{\langle \boldsymbol{\mu}, \boldsymbol{\epsilon} \rangle}{\|\boldsymbol{\mu}\| \cdot \|\boldsymbol{\epsilon}\|} \right)^2 > \mathbb{E}[v]$. Equivalently, if the angle between $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}$ satisfies $\angle(\boldsymbol{\mu}, \boldsymbol{\epsilon}) < \arccos\left(\sqrt{\mathbb{E}[v]}\right)$, we regard $\boldsymbol{\epsilon}$ as being drawn from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$.

Let $K \geq 1$ be the number of independent random samples. We consider independent random vectors $\mathcal{C}_n^K = \{\boldsymbol{\epsilon}^{(1)}, \boldsymbol{\epsilon}^{(2)}, \dots, \boldsymbol{\epsilon}^{(K)}\}$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in the n -dimensional space, and let $\boldsymbol{\mu} \in \mathbb{R}^n$ be a vector. For each $i = 1, \dots, K$, $\theta_i = \angle(\boldsymbol{\mu}, \boldsymbol{\epsilon}^{(i)})$ is defined as the angle between $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}^{(i)}$. Thus $v_i = \cos^2 \theta_i$ be independent sampled from the non-central beta distribution (1).

The maximum of $\{v_i\}$ is given by $\nu_K = \max_{1 \leq i \leq K} v_i$. As $K \rightarrow \infty$, the typical maximum of the cosine square similarity ν_K is a value b_K such that: $\mathbb{P}(\nu_K \approx b_K) \rightarrow 1$. When the sample size K becomes large, the typical maximum refers to the value that the maximum is most likely to take. It is not exactly the expected value, nor is it concerned with rare extreme fluctuations; rather, it describes where the maximum tends to be most of the time. We use it to describe how similar the random set $\mathcal{C}_n^K$ is to $\boldsymbol{\mu}$ and have the following theorem to estimate the typical maximum $\nu_{\mathcal{C}_n^K}$.

Theorem 2. Under the setup described above, let $\{v_i\}_{i=1}^K$ be K samples drawn from the non-central beta distribution $\text{Beta}(\alpha, \beta; \lambda)$. When n is moderately large and K is not too small, the typical maximum $\nu_{\mathcal{C}_n^K}$ is approximately the same:

$$\nu_{\mathcal{C}_n^K} \approx \frac{2 \log K}{n} \quad (3)$$

This theorem reveals that in high-dimensional spaces with large n , any two vectors are nearly orthogonal as the typical maximum of approaches 1. To increase the similarity between two vectors, we must decrease the dimension n and increase the number of samples K . Since each v_i measures the cosine square similarity between $\boldsymbol{\epsilon}^{(K)}$ and $\boldsymbol{\mu}$, the typical maximum $\nu_{\mathcal{C}}$ describes how the set $\mathcal{C}$ can approach $\boldsymbol{\mu}$. Thus, under the setup described above, the definition in the following is reasonable.

Definition 2. If $\nu_{\mathcal{C}_n^K} > 0.15n\mathbb{E}[v]$, we say the set $\mathcal{C}_n^K$ can provide a sample for the distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ and say $\argmin_{\boldsymbol{\epsilon}^{(i)} \in \mathcal{C}_n^K} \cos^2 \angle(\boldsymbol{\mu}, \boldsymbol{\epsilon}^{(i)})$ as the sample.

4 DISCO

In this section, we introduce DISCO, a method for conditional generation using diffusion models. We begin by showing that conditional generation can be achieved by sampling noise from a non-zero-mean Gaussian distribution. Next, we demonstrate that this sampling process can be replaced by selecting discrete codes from a predefined codebook. This insight implies that accurately predicting the correct discrete code based on the input condition enables effective conditional generation. To realize this, we describe how to train a network that predicts these discrete codes from intermediate diffusion states and the conditioning input.

4.1 Non-zero-mean Gaussian Distribution for Conditional Generation

To derive conditional diffusion models, Dhariwal and Nichol [10] introduced *classifier guidance*, which modifies the score $\epsilon_\theta(\mathbf{z}_t, \mathbf{c}) \approx -\sigma_t \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t | \mathbf{c})$ by adding the gradient of the log-likelihood from a classifier model $p_\theta(\mathbf{c} | \mathbf{z}_t)$, which is proportional to $\exp(\mathcal{E}(\mathbf{z}_t, \mathbf{c}))$, as shown below:

$$\tilde{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) = \epsilon_\theta(\mathbf{z}_t, \mathbf{c}) - w \sigma_t \nabla_{\mathbf{z}_t} \log p_\theta(\mathbf{c} | \mathbf{z}_t) = \epsilon_\theta(\mathbf{z}_t, \mathbf{c}) + \underbrace{-w \sigma_t \nabla_{\mathbf{z}_t} \log \mathcal{E}(\mathbf{z}_t, \mathbf{c})}_{\text{Conditional Correction } \Delta(\mathbf{z}_t, \mathbf{c})}, \quad (4)$$

where w is a parameter controlling the strength of the classifier guidance. The modified score $\tilde{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c})$ is then used in place of $\epsilon_\theta(\mathbf{z}_t, \mathbf{c})$ when sampling from the diffusion model, resulting in approximate samples from the distribution: $\tilde{p}_\theta(\mathbf{z}_t | \mathbf{c}) \propto p_\theta(\mathbf{z}_t | \mathbf{c}) p_\theta(\mathbf{c} | \mathbf{z}_t)^w$. This approach effectively increases the probability of data where the classifier $p_\theta(\mathbf{c} | \mathbf{z}_t)$ assigns high likelihood to the correct conditions.

Since classifier guidance relies on gradients from an image classifier, Ho and Salimans [15] aim to remove the need for the classifier and introduce *classifier-free guidance*. This approach modifies $\epsilon_\theta(\mathbf{z}_t, \mathbf{c})$ in a way that produces the same effect as classifier guidance but without using a classifier. They train an unconditional denoising diffusion model $p_\theta(\mathbf{z})$, parameterized by $\epsilon_\theta(\mathbf{z}_t, \mathbf{c} = \emptyset)$, along with the conditional model $p_\theta(\mathbf{z} | \mathbf{c})$, parameterized by $\epsilon_\theta(\mathbf{z}_t, \mathbf{c})$, where $\emptyset$ denotes the absence of the class identifier $\mathbf{c}$. Sampling is then carried out using the following linear combination of the conditional and unconditional score estimates:

$$\tilde{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) = \epsilon_\theta(\mathbf{z}_t, \mathbf{c}) + \underbrace{w(\epsilon_\theta(\mathbf{z}_t, \mathbf{c}) - \epsilon_\theta(\mathbf{z}_t, \emptyset))}_{\text{Conditional Correction } \Delta(\mathbf{z}_t, \mathbf{c})} \quad (5)$$

The DDPM denoising step [16] for conditional generation thus becomes

$$\begin{aligned} \mathbf{z}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \tilde{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) \right) + \rho_t \epsilon', \quad \epsilon' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{z}_t, \mathbf{c}) \right) + \rho_t \epsilon, \quad \epsilon \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}), \mathbf{I}) \end{aligned} \quad (6)$$

where $\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}) = -\frac{1 - \alpha_t}{\sqrt{\alpha_t} \sqrt{1 - \bar{\alpha}_t}} \Delta(\mathbf{z}_t, \mathbf{c})$ and the last term ϵ is sampled from a non-zero-mean Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}), \mathbf{I})$.

4.2 Discrete Noise for Conditional Generation

Equation (6) shows that in conditional DDPM sampling, noise is drawn from a non-zero-mean Gaussian distribution. We propose replacing this noise with a code selected from a fixed codebook $\mathcal{C}_n^K = \{\epsilon^{(1)}, \dots, \epsilon^{(K)}\}$, where each code $\epsilon^{(k)}$ is independently sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and remains unchanged during training and inference. The core idea is that an appropriate code from this set can serve as an effective discrete approximation of the non-zero-mean Gaussian noise, enabling a simple codebook lookup in place of continuous sampling.

Theorem 1 provides a criterion for selecting a good substitute for Gaussian noise. It shows that the squared cosine similarity between $\boldsymbol{\mu}$ and $\epsilon \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$ follows a non-central beta distribution. If there exists a code $\epsilon^{(i)} \in \mathcal{C}_n^K$ whose squared cosine similarity with $\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c})$ exceeds the expected value $\mathbb{E}[v]$ of this distribution, then $\epsilon^{(i)}$ can be considered a valid substitute under Definition 1.

Theorem 2 evaluates whether the codebook $\mathcal{C}_n^K$ can provide such a substitute. It defines the typical maximum similarity $\nu_{\mathcal{C}_n^K}$ between $\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c})$ and the codes in $\mathcal{C}_n^K$ as in Equation (3). If $\nu_{\mathcal{C}_n^K} > \mathbb{E}[v]$, the codebook is deemed sufficient to approximate samples from the target non-zero-mean Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}), \mathbf{I})$.

Since the latent space of SD 1.5 has size $4096 = 4 \times 64 \times 64$, the dimensionality of the noise ϵ in Equation (6) is also 4096. According to Theorem 2, in such high-dimensional spaces, a predefined codebook $\mathcal{C}_n^K$ cannot provide a good approximation to samples from a non-zero-mean Gaussian distribution, as most vectors tend to be nearly orthogonal. To address this, we divide ϵ into 256 blocks $\{\epsilon_i\}_{i=1}^{256}$, each of size $4 \times 4 \times 4$, so that $\epsilon = \text{cat}(\{\epsilon_i\})$, and each block has dimensionality 64. In this configuration, the expected value of $\text{Beta}(0.5, 31.5; \lambda)$ is 0.0278, with $\lambda = \|\boldsymbol{\mu}_i\|^2 \approx 256$ based on empirical statistics. For a predefined codebook $\mathcal{C}_{64}^{4096}$ with 4096

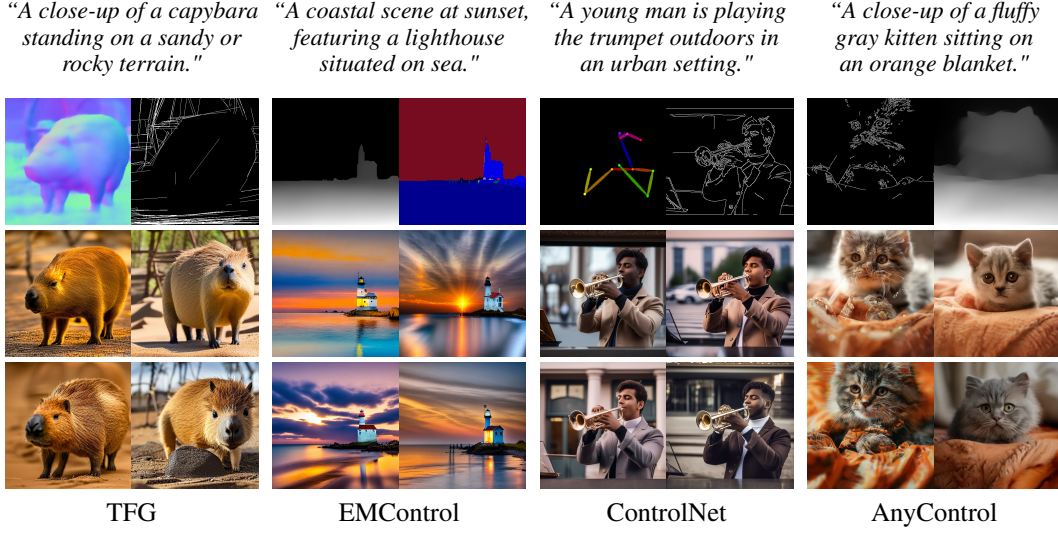


Figure 1: Conditional Generation with Discrete Noise. The first two rows present the text prompt and the corresponding conditioning input. The third row shows the outputs generated using continuous noise. The final row displays the results generated using discrete noise. The results demonstrate that our method maintains generation quality without degradation.

vectors sampled from a 64-dimensional standard Gaussian, the typical maximum squared cosine similarity is 0.263. Since this exceeds the expected similarity from the non-zero-mean distribution, we conclude that $\mathcal{C}_{64}^{4096}$ is sufficient to approximate samples from $\mathcal{N}(\boldsymbol{\mu}_i(\mathbf{z}_t, \mathbf{c}), \mathbf{I})$, where $\|\boldsymbol{\mu}_i\| \approx 16$, and $\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}) = \text{cat}(\{\boldsymbol{\mu}_i(\mathbf{z}_t, \mathbf{c})\})$. This enables us to reformulate Equation (6) as follows:

$$\begin{aligned} \mathbf{z}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) \right) + \rho_t \text{cat}(\{\boldsymbol{\epsilon}_i\}), \\ \boldsymbol{\epsilon}_i &= \boldsymbol{\epsilon}^{(k_i)}, \quad k_i = \underset{1 \leq j \leq 4096}{\text{argmax}} \cos^2 \angle(\boldsymbol{\mu}_i(\mathbf{z}_t, \mathbf{c}), \boldsymbol{\epsilon}^{(j)}) \end{aligned} \quad (7)$$

To validate our approach, we replace the non-zero-mean Gaussian noise used in TFG [44], EMControl [37], ControlNet [48], and AnyControl [34] with our proposed discrete noise. The evaluated algorithms are categorized into two types: classifier-based guidance methods (e.g., TFG and EMControl) and classifier-free guidance methods (e.g., ControlNet and AnyControl). We apply Equations (4) and (5) to compute the conditional correction, followed by Equation (7) to perform conditional generation with discrete noise. The results are presented in Figure 1, where the third row shows outputs generated using the original methods, and the fourth row displays the results with our approach. The comparable visual quality confirms the effectiveness of our substitution strategy.

4.3 Conditional Control by Discrete Noise Guidance

Both classifier guidance and classifier-free guidance estimate the condition term $\Delta(\mathbf{z}_t, \mathbf{c})$ based on the intermediate state $\mathbf{z}_t$ and the condition $\mathbf{c}$. In the previous section, we discretized the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}_t, \mathbf{c}), \mathbf{I})$ and validated its effectiveness. Building on this, we propose training a network that directly predicts the discrete noise code from $\mathbf{c}$ and $\mathbf{x}_t$, thereby enabling explicit control over the generation process. Notably, this approach differs fundamentally from prior methods, as it predicts the discrete control noise rather than the condition term $\Delta(\mathbf{z}_t, \mathbf{c})$ itself.

4.3.1 Discrete Noise Predication

In this subsection, we describe how the discrete noise indices $\{k_i\}_{i=1}^{256}$ is predicted from the intermediate features $\mathbf{z}_t = \text{cat}(\{\mathbf{z}_{t,i}\}_{i=1}^{256})$ and the conditioning input $\mathbf{c}$, where each $\mathbf{z}_{t,i}$ is a $4 \times 4 \times 4$ block of the intermediate state $\mathbf{z}_t$. We adopt the Diffusion Transformer (DiT) [21] as our backbone, which integrates time-step conditioning into a standard encoder-only transformer [36] and employs rotary positional embeddings [33] for spatial encoding.

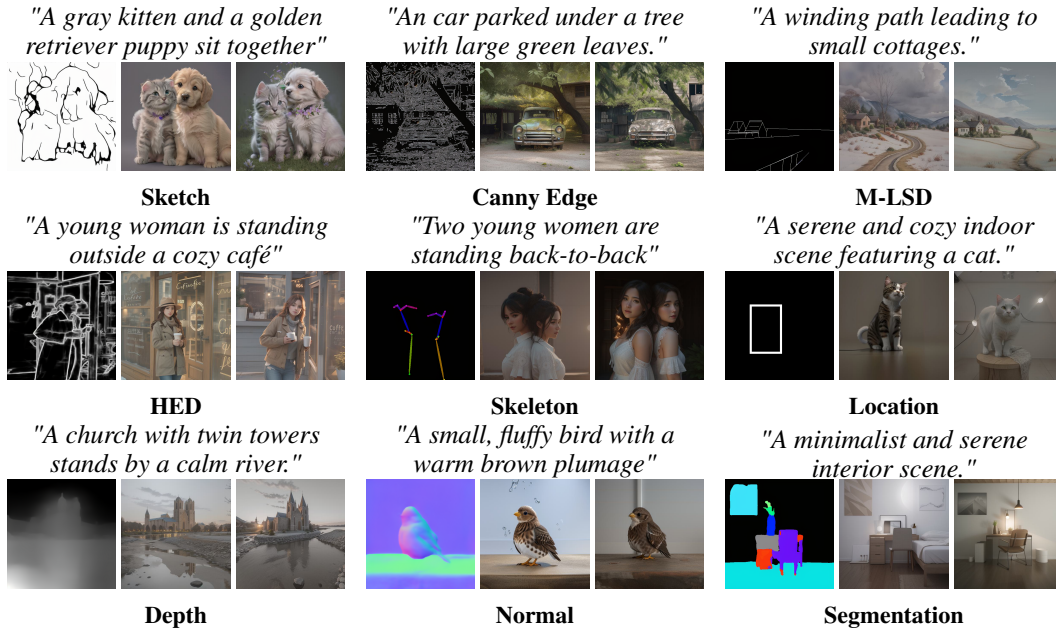


Figure 2: Conditional Generation with Diverse Conditions via Discrete Noise. Discrete noise enables seamless integration of various conditions into diffusion models. To demonstrate its versatility, we showcase ten diverse applications: Sketch, Canny Edge, MLSD, HED, Skeleton, Location, Depth, Normal, and Segmentation.

We adopt a BERT-style training scheme [9] for our DiT model. Specifically, we formulate the training as a discrete noise index completion task: a subset of the input discrete noise indices is randomly masked, and the model is trained to recover the missing indices $\{k_i\}_{i \in M}$ using the condition c , the intermediate state z_t , and the unmasked indices $\{k_i\}_{i \in \overline{M}}$, where M denotes the set of masked positions. Unlike traditional Transformers that operate solely on discrete token sequences, our model uses a hybrid input that combines discrete indices with continuous visual features.

The hybrid inputs to DiT consist of condition c , the intermediate state z_t , and unmasked discrete noise indices $\{k_i\}_{i \in \overline{M}}$. For each masked index, we replace its original codebook index with a special index $[MASK]$. Conditioned on the unmasked indices, intermediate features $\{z_{t,i}\}_{i=1}^{256}$, and condition c , the Transformer is trained to estimate the likelihood of the indices at the masked positions:

$$P(\{k_i\}_{i \in M} | \{k_i\}_{i \in \overline{M}}, \{z_{t,i}\}_{i=1}^{256}, c) = \prod_{i \in M} P(k_i | \{k_i\}_{i \in \overline{M}}, \{z_{t,i}\}_{i=1}^{256}, c) \quad (8)$$

The model is optimized by minimizing the softmax cross-entropy loss between the predicted probabilities and the ground-truth indices. During the inference time, the model starts with the intermediate features $z_t = \text{cat}(\{z_{t,i}\}_{i=1}^{256})$ and the conditioning input c and the discrete noise indices all filled with $[MASK]$. The predicted discrete noise indices are generated by the hybrid-transformer.

4.3.2 Training Dataset Preparation

Training the discrete noise prediction network requires data pairs $(z_{t,i}, k_i)$ for a given condition c . A straightforward approach is to use intermediate states $z_t = \text{cat}(\{z_{t,i}\}_{i=1}^{256})$ and corresponding discrete noise indices $\{k_i\}_{i=1}^{256}$ generated by an existing conditional method such as FreeDoM [46]. However, this method inherits the limitations of the underlying generation model, potentially compromising data quality. Instead, we adopt a reconstruction-based strategy. Given c , we define the energy function in Equation (4) as $\mathcal{E}(z_t, c) = \|z_{0|t} - c\|^2$, where $z_{0|t} = (z_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(z_t, c)) / \sqrt{\bar{\alpha}_t}$. This formulation frames conditional denoising as reconstructing z_0 from noisy input z_t . By applying Equation (7) during reconstruction, we obtain training pairs $(z_{t,i}, k_i)$. Structural cues (e.g., depth or edge maps) extracted from the reconstructed image z_0 are then used as conditioning inputs.

Table 1: Quantitative Comparison for Controllable Generation Under Different Conditions. The best results are highlighted in bold. We use FID, CLIP, and CLIP-Ac to evaluate generation quality, and MSE, SSIM, mIoU, and mAP to assess controllability. A dash “-” indicates that the method does not provide a publicly available model for evaluation.

	Method	Depth	Canny	HED	M-LSD	Segmentation	Normal	Skeleton	Location	Sketch
FID ↓	UGD	-	-	-	-	25.1856	-	-	-	-
	FreeControl	24.9604	17.9107	17.4471	-	30.8189	28.9317	34.7962	-	-
	ControlNet	19.8064	18.9211	16.1678	21.5897	20.3947	29.5817	26.1679	29.6951	24.0166
	T2I-Adapter	20.2817	24.8351	-	-	-	-	24.9437	-	-
	AnyControl	18.6452	19.3719	-	-	19.3647	-	25.7867	-	-
	DISCO	18.0431	21.3836	15.7542	22.0951	21.1360	27.4390	25.5083	28.8875	23.6784
CLIP ↑	UGD	-	-	-	-	0.2916	-	-	-	-
	FreeControl	0.3017	0.3214	0.3039	-	0.3045	0.3043	0.3016	-	-
	ControlNet	0.3061	0.3085	0.3008	0.2915	0.2997	0.2987	0.3011	0.2917	0.3023
	T2I-Adapter	0.2990	0.3045	-	-	-	-	0.3111	-	-
	AnyControl	0.3063	0.3032	-	-	0.3069	-	0.3089	-	-
	DISCO	0.2952	0.3329	0.3035	0.2951	0.3189	0.2943	0.2920	0.2968	0.3106
CLIP-ac ↑	UGD	-	-	-	-	5.3704	-	-	-	-
	FreeControl	5.0129	5.0010	5.0048	-	5.0520	5.0704	5.4050	-	-
	ControlNet	5.0973	5.1213	5.1683	5.2763	5.3920	5.2437	5.2956	5.2737	5.0048
	T2I-Adapter	5.3498	5.1650	-	-	-	-	5.4802	-	-
	AnyControl	5.1342	5.0485	-	-	4.9362	-	4.8970	-	-
	DISCO	5.4225	5.4264	5.4575	5.4341	5.3905	5.4284	5.4794	5.3977	5.4730
Controllability		MSE↓	SSIM↑	SSIM↑	SSIM↑	mIoU↑	MSE↓	mAP↑	mAP↑	SSIM↑
	UGD	-	-	-	-	0.6843	-	-	-	-
	FreeControl	99.3874	0.4679	0.6159	-	0.7005	0.3569	0.5262	-	-
	ControlNet	87.3678	0.4754	0.4836	0.7447	0.4428	0.3572	0.4332	0.3092	0.5262
	T2I-Adapter	89.8904	0.4786	-	-	-	-	0.5283	-	-
	AnyControl	88.9634	0.5083	-	-	0.3382	-	0.3764	-	-
	DISCO	86.6756	0.4412	0.4752	0.7793	0.3916	0.3443	0.4043	0.2563	0.6118

5 Experiment

This section presents a comprehensive quantitative and qualitative evaluation of our method, emphasizing its training efficiency and enhanced alignment performance. Additionally, we include an analysis of how dimensionality and codebook size influence the final results.

5.1 Training Details & Efficiency

Unlike ControlNet-style methods [34, 45, 48], our discrete noise prediction network does not require loading a frozen Stable Diffusion backbone into GPU memory, which significantly improves training efficiency. Our training process completes in 3 days on a single NVIDIA RTX 3090 (24GB). In contrast, ControlNet [48], IP-Adapter [45], and AnyControl [34] require an A6000 GPU (48GB) and 3+ days for training, according to our experiments.

We train the discrete noise prediction network for 35 epochs with a batch size of 32, using the Adam optimizer with a learning rate of 1×10^{-5} . During training, both input images and condition maps are resized to 512×512 . We utilize approximately 118,000 images from the COCO2017 dataset [19] across all conditioning types. For human pose estimation, we use a subset of about 6,500 images specifically selected from the “person” category.

5.2 Conditional Generation

DISCO effectively integrates diverse conditioning inputs directly into the image generation process of diffusion models, enabling precise and flexible control over outputs. To demonstrate this capability, we present ten single-condition examples in Figure 2, covering a wide range of modalities, including Canny edge [2], MLSD [12], HED [41], Skeleton [3], Location [24], Depth [42], Normal [35], and Segmentation [5]. These examples highlight the versatility and robustness of DISCO in accommodating a broad spectrum of visual conditions.

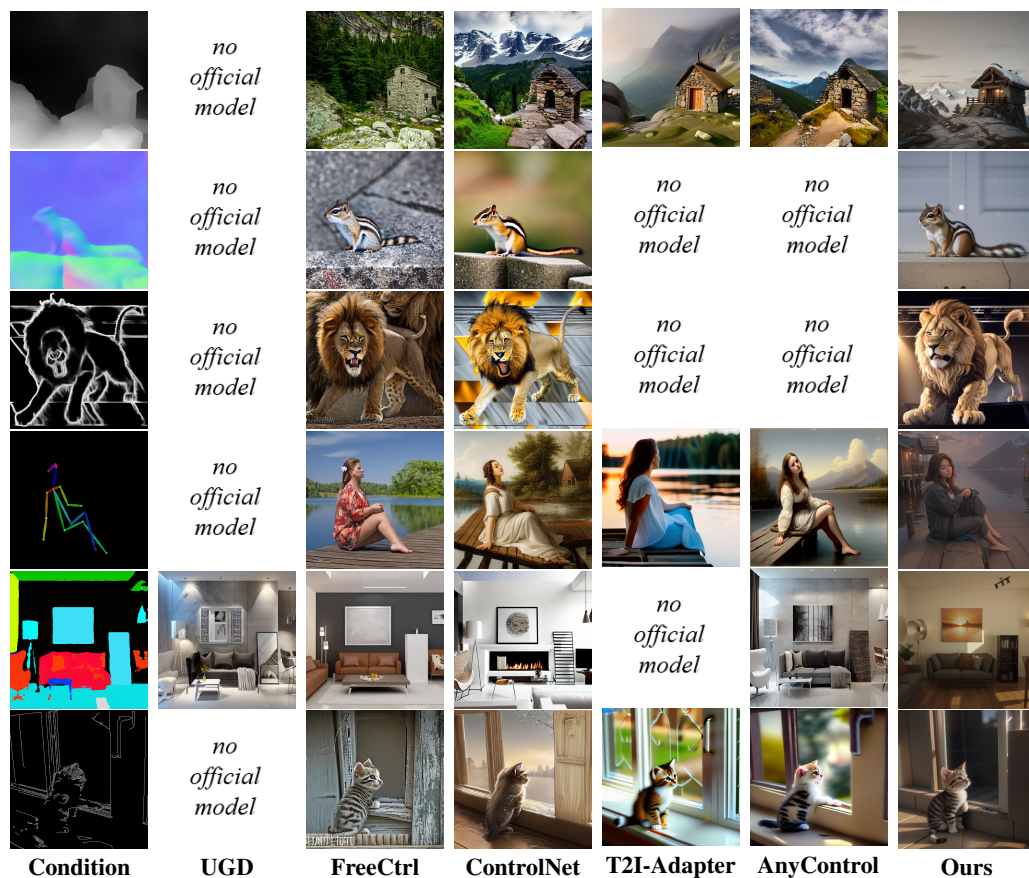


Figure 3: Qualitative Comparison for Controllable Generation under Different Conditions: We compare our method with five other approaches. Among them, UGD and FreeCtrl belong to the classifier-based guidance category, while ControlNet, T2I-Adapter, and AnyControl are part of the classifier-free guidance category. The results demonstrate that our method is competitive with state-of-the-art techniques.

5.3 Qualitative and Quantitative Comparison

In this section, we present both qualitative and quantitative comparisons to demonstrate the controllability of DISCO. It is important to note that not all baseline methods provide publicly available implementations for handling all types of conditions, such as Depth, Canny, HED, M-LSD, Segmentation, Normal, Skeleton, Location, and Sketch. For conditions that are not supported by a given method, we leave the corresponding entries blank.

For a thorough quantitative evaluation, we adopt several metrics, including FID [14], CLIP Score [23], and CLIP Aesthetic Score [28] to assess generation quality. To measure condition controllability, we use SSIM, mAP, MSE, mIoU. As shown in Table 1, our method consistently outperforms existing approaches across various conditions. These results demonstrate that DISCO effectively handles diverse conditioning inputs while producing high-quality, coherent outputs aligned with both textual prompts and constraints. Overall, DISCO achieves superior performance on most metrics, validating its advantage over prior methods.

Figure 3 shows examples of DISCO handling various conditions, demonstrating its ability to generate high-quality results that accurately follow the spatial constraints imposed by the conditioned inputs. The prompts for the six rows are: “a stone hut perched on a rugged, rocky outcrop”, “a chipmunk perched on a weathered concrete surface”, “a majestic lion with full mane is captured in raw”, “a young woman sitting on a wooden dock by the water”, and “a small, curious kitten sitting on a wooden windowsill”. As illustrated, the generated outputs effectively fulfill the requirements of both the textual prompts and visual conditions.

Table 2: Quantitative comparison with state-of-the-art methods on multiple conditioning modalities. Higher values indicate better performance. DISCO achieves the best results across all conditions.

	Depth	Canny	HED	MLSD	Seg	Normal	Skeleton	Location	Sketch
UGD	-	-	-	-	0.12	-	-	-	-
FreeControl	0.10	0.13	0.30	-	0.11	0.34	0.11	-	-
ControlNet	0.21	0.28	0.33	0.42	0.23	0.27	0.14	0.37	0.41
T2I-Adapter	0.22	0.11	-	-	-	-	0.25	-	-
AnyControl	0.18	0.17	-	-	0.24	-	0.19	-	-
DISCO (Ours)	0.29	0.31	0.37	0.58	0.30	0.39	0.31	0.63	0.59

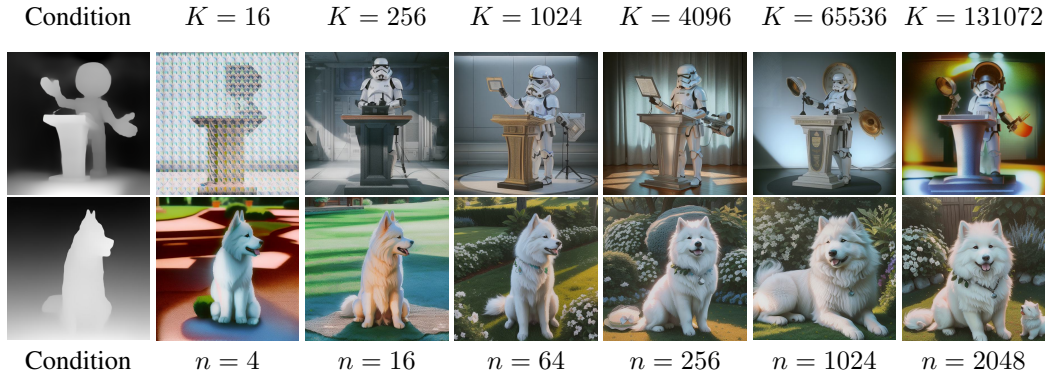


Figure 4: Conditional Generation Quality with Different Dimensionalities n and Codebook Sizes K . In the first row, we fix $n = 64$ and vary K to examine its effect on conditional generation, using the prompt “stormtrooper behind a lectern.” In the second row, we fix $K = 4096$ and vary n , with the prompt “a fluffy white Samoyed dog sitting gracefully on a lush, green lawn.”

5.4 User Study

In this section, we conducted a user study to further validate the effectiveness of our method, the results of which are summarized in Table 2. For each conditioning type, we randomly selected 50 images. The outputs from different methods were presented to the participants in a random order to avoid positional bias. Participants were then asked to select the best result based on three criteria: *realism*, *consistency with the text prompt*, and *consistency with the input condition*.

Each participant answered three formal questions per condition, plus one validation question, resulting in a total of 28 questions. Responses that failed the validation question were excluded from the subsequent analysis. To compute the final aggregate score, we employed a weighted scoring scheme across the three criteria: realism (30%), text alignment (30%), and condition alignment (40%).

We invited volunteers from our research group, which comprises nearly 100 students, to participate in the study. In total, we collected 76 questionnaires. After filtering out responses that failed the validation check, 72 were considered valid for the final evaluation. As shown in Table 2, our method DISCO achieves the best performance across all nine control conditions, significantly outperforming existing approaches.

5.5 Study for Dimensionality and Codebook Size

DISCO is built on the assumption that a predefined codebook $\mathcal{C}_n^K$, constructed from a zero-mean Gaussian distribution, can effectively approximate samples from a non-zero-mean Gaussian distribution. This assumption relies on the existence of at least one code vector in the codebook that is closely aligned in direction with the target mean—i.e., the angle between the selected code vector and the mean of the non-zero-mean Gaussian distribution is small to act as a reliable proxy for conditional generation. Theorem 2 guarantees that this assumption holds when the dimension n is moderately large and the codebook size K is not too small (i.e., $n = 64$, $K = 4096$). Here, we analyze how varying the values of n and K impacts the performance of conditional generation.

In the first row of Figure 4, we fix n and vary K . When K is small, the generated outputs fail to align with the conditional input, as the typical maximum of cosine square similarity $\nu_{C_n^K}$ is close to 0, meaning the codebook cannot approximate samples from the non-zero-mean Gaussian. When K is very large, $\nu_{C_n^K}$ approaches 1, causing the selected code vector to converge with the mean, reducing randomness in sampling and thereby decreasing output quality. In the second row of Figure 4, we fix K and vary n . With small n , $\nu_{C_n^K}$ approaches 1, leading to degraded quality. With large n , $\nu_{C_n^K}$ approaches 0, causing the codebook to fail in aligning with the target mean and making conditional control ineffective. We thus conclude that our method is robust for conditional generation only when n and K are within appropriate ranges, and not at their extreme values.

The configuration ($n = 64 = 4 \times 4 \times 4$, $K = 4096$) is not the only viable option for discrete noise control. Alternatives such as ($n = 16 = 2 \times 2 \times 4$, $K = 32$) and ($n = 256 = 8 \times 8 \times 4$, $K = 65536$) are also feasible, as defined in Definition 2. However, we exclude these due to training efficiency concerns. For example, $n = 16$ requires the discrete noise prediction network to predict 1024 indices, increasing DiT's computational cost by $16\times$ compared to the $n = 64$ setting, which requires only 256 predictions. Conversely, a larger codebook size of $K = 65536$ significantly raises memory and computational demands relative to $K = 4096$. Thus, we adopt ($n = 64$, $K = 4096$) as a balanced and efficient choice.

6 Conclusion

In this paper, we present two key findings: (1) conditional generation in diffusion models can be reformulated as sampling noise from a non-zero-mean Gaussian distribution; and (2) this distribution can be approximated by selecting noise vectors from a predefined codebook, constructed from standard Gaussian samples, which serve as discrete proxies for conditional guidance. Based on this discrete conditional generation perspective, we propose a novel approach—discrete noise prediction—which differs fundamentally from both classifier-based and classifier-free guidance. Specifically, we employ DiT to transform the input $(z_{t,i}, c)$ into a noise index k_i , indicating the selected entry in the codebook. Experimental results demonstrate the superiority and adaptability of DISCO.

7 Acknowledgment

This work was supported by the National Natural Science Foundation of China (62372237, 62332010) and the Major Science and Technology Projects in Jiangsu Province under Grant BG2024042.

References

- [1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal Guidance for Diffusion Models. In *International Conference on Learning Representations*, 2024.
- [2] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986.
- [3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [4] Stanley Chan. Tutorial on Diffusion Models for Imaging and Vision. *Foundations and Trends® in Computer Graphics and Vision*, 2024.
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2022.
- [6] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving Diffusion Models for Inverse Problems using Manifold Constraints. In *Advances in Neural Information Processing Systems*, 2022.
- [7] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *International Conference on Learning Representations*, 2023.
- [8] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion Models in Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, 2021.
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *International Conference on Machine Learning*, 2024.
- [12] Geonmo Gu, Byungsoo Ko, SeoungHyun Go, Sung-Hyun Lee, Jingeun Lee, and Minchul Shin. Towards Light-Weight and Real-Time Line Segment Detection. In *AAAI Conference on Artificial Intelligence*, 2022.
- [13] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold Preserving Guided Diffusion. In *International Conference on Learning Representations*, 2024.
- [14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems*, 2017.
- [15] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020.

- [17] Minghui Hu, Jianbin Zheng, Daqing Liu, Chuanxia Zheng, Chaoyue Wang, Dacheng Tao, and Tat-Jen Cham. Cocktail: Mixing Multi-Modality Control for Text-Conditional Image Generation. In *Advances in Neural Information Processing Systems*, 2023.
- [18] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. ControlNet++: Improving Conditional Controls with Efficient Consistency Feedback. In *European Conference on Computer Vision*, 2024.
- [19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, 2014.
- [20] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning Adapters to Dig Out More Controllable Ability for Text-to-Image Diffusion Models. *AAAI Conference on Artificial Intelligence*, 2024.
- [21] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *International Conference on Computer Vision*, 2023.
- [22] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, Stefano Ermon, Yun Fu, and Ran Xu. UniControl: A Unified Diffusion Model for Controllable Visual Generation In the Wild. In *Advances in Neural Information Processing Systems*, 2023.
- [23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, 2021.
- [24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2016.
- [25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Conference on Computer Vision and Pattern Recognition*, 2022.
- [26] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving Linear Inverse Problems Provably via Posterior Sampling with Latent Diffusion Models. In *Advances in Neural Information Processing Systems*, 2023.
- [27] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An Open Large-Scale Dataset for Training Next Generation image-Text Models. In *Advances in Neural Information Processing Systems*, 2022.
- [28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. *Advances in Neural Information Processing Systems*, 2022.
- [29] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. In *Conference on Computer Vision and Pattern Recognition*, 2024.
- [30] Zhenning Shi, Haoshuai Zheng, Chen Xu, Changsheng Dong, Bin Pan, Xueshuo Xie, Along He, Tao Li, and Huazhu Fu. Resfusion: Denoising Diffusion Probabilistic Models for Image Restoration Based on Prior Residual Noise. In *Advances in Neural Information Processing Systems*, 2024.
- [31] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency. In *International Conference on Learning Representations*, 2024.

- [32] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation. In *International Conference on Machine Learning*, 2023.
- [33] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 2024.
- [34] Yanan Sun, Yanchen Liu, Yinhao Tang, Wenjie Pei, and Kai Chen. AnyControl: Create Your Artwork with Versatile Control on Text-to-Image Generation. In *European Conference on Computer Vision*, 2024.
- [35] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *arXiv*, 2019.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017.
- [37] He Wang, Longquan Dai, and Jinhui Tang. EMControl: Adding Conditional Control to Text-to-Image Diffusion Models via Expectation-Maximization. In *AAAI Conference on Artificial Intelligence*, 2025.
- [38] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C. Kot, and Bihan Wen. SinSR: Diffusion-Based Image Super-Resolution in a Single Step. In *Conference on Computer Vision and Pattern Recognition*, 2024.
- [39] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. SeeSR: Towards Semantics-Aware Real-World Image Super-Resolution. In *Conference on Computer Vision and Pattern Recognition*, 2024.
- [40] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. DiffIR: Efficient Diffusion Model for Image Restoration. In *International Conference on Computer Vision*, 2023.
- [41] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. In *International Conference on Computer Vision*, 2015.
- [42] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Conference on Computer Vision and Pattern Recognition*, 2024.
- [43] Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with Spherical Gaussian Constraint for Conditional Diffusion. In *International Conference on Machine Learning*, 2024.
- [44] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y Zou, and Stefano Ermon. TFG: Unified Training-Free Guidance for Diffusion Models. In *Advances in Neural Information Processing Systems*, 2024.
- [45] Hu Ye, Jun Zhang, Sibio Liu, Xiao Han, and Wei Yang. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv*, 2023.
- [46] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-Free Energy-Guided Conditional Diffusion Model. In *International Conference on Computer Vision*, 2023.
- [47] Lvmin Zhang and Maneesh Agrawala. Packing Input Frame Context in Next-Frame Prediction Models for Video Generation. *arXiv*, 2025.
- [48] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *International Conference on Computer Vision*, 2023.

- [49] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models. In *Advances in Neural Information Processing Systems*, 2023.
- [50] Jianbin Zheng, Daqing Liu, Chaoyue Wang, Minghui Hu, Zuopeng Yang, Changxing Ding, and Dacheng Tao. MMoT: Mixture-of-Modality-Tokens Transformer for Composed Multimodal Conditional Image Synthesis. *International Journal of Computer Vision*, 2024.
- [51] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. StoryDiffusion: Consistent Self-Attention for Long-Range Image and Video Generation. In *Advances in Neural Information Processing Systems*, 2024.
- [52] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhong Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising Diffusion Models for Plug-and-Play Image Restoration. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

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: The abstract and introduction clearly state DISCO's contributions and scope. Claims align with theoretical (Section 3) and experimental results (Section 5).

Guidelines:

- The answer NA 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 NA 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: [No]

Justification: The paper does not include the limitations.

Guidelines:

- The answer NA 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: [Yes]

Justification: We provide the proof in the supplementary material.

Guidelines:

- The answer NA 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 offer an exhaustive discussion of our DISCO approach in Section 4, where the theoretical underpinnings and procedural details are laid out. Additionally, experimental setups and their configurations are elaborated upon in Section 5, providing clarity on how the method's effectiveness is empirically validated, enabling others to replicate the results with the described methodology and datasets.

Guidelines:

- The answer NA 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: [\[Yes\]](#)

Justification: Our intention is to make the code publicly available as open source once our paper is accepted. However, preparing the code for open source involves extensive work to facilitate the reproduction of our results by the readers. Currently, our focus is entirely on the preparation of the paper, which leaves us with limited time to concurrently undertake the necessary steps to ready the code for open source distribution.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: In Section 5.1, we detail our experimental setup, including data splits, hyperparameters, and their selection process. Upon acceptance of the paper, we will release the code, which will include all the detailed settings necessary for replication of our results.

Guidelines:

- The answer NA 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: [\[No\]](#)

Justification: We consider that error bars are not essential for our comparative analysis, as the focus is on qualitative and quantitative metrics that sufficiently validate the method's performance.

Guidelines:

- The answer NA 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: Our training and inference processes were carried out on a 3090 GPU, as specified in Section 5.1, including details on the type of compute resources used.

Guidelines:

Guidelines:

- The answer NA 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: We have meticulously adhered to the NeurIPS Code of Ethics in all aspects of our work.

Guidelines:

- The answer NA 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: [NA]

Justification: We introduce a conditional generation method for Stable Diffusion. Therefore, both potential positive societal impacts and negative societal impacts of the work performed are inherently addressed by the Stable Diffusion framework,

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We propose a conditional generation approach for Stable Diffusion, with all required safeguards naturally incorporated within the Stable Diffusion framework.

Guidelines:

- The answer NA 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: We utilize publicly available datasets, such as COCO2017, and cite the original sources in the paper, respecting their licenses and terms of use.

Guidelines:

- The answer NA 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: [NA]

Justification: Our paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core methodology of this research does not involve large language models (LLMs) as important, original, or non-standard elements. The DISCO method relies on diffusion models without LLM usage in its methodology.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.