Unsupervised Concept Discovery with Dirichlet Concept Diffusion Models

Yuchong Geng **Ao Tang**
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
{yg534, atang}@cornell.edu

Abstract

We show that a generative model can discover concepts from data in an unsupervised setting while learning to generate. The Dirichlet Concept Diffusion Model (DCDM) embeds concept discovery into diffusion-based generation. DCDM learns concept centers and infers a Dirichlet distribution over them for each input. The resulting weighted center shapes the forward diffusion mean and reverse denoising process, so components are learned through the same evidence lower bound used to model the data. The method uses no class labels, attribute annotations, captions, pretrained text-to-image priors, or semantic concept supervision. Analysis shows how the formulation preserves concept-level information along diffusion paths and reduces denoising ambiguity, creating pressure for concept centers to capture stable modes. Experiments across diverse image domains show that the learned components form coherent prototypes, support interventions, and reflect recurring visual structure.

1 Introduction

Richard Feynman's maxim, "What I cannot create, I do not understand," suggests a natural test for modern generative models. If a model learns to create images, can the learning process also uncover reusable concept components that help explain the images it creates? Existing image generators can synthesize realistic samples, yet sample quality alone does not reveal whether the model has structured its representation around interpretable and reusable abstractions. What is missing is a model whose process of learning to generate exposes the recurring components driving its samples.

Researchers have long studied how observations can be organized into reusable abstractions, from early rule-based, category-based, and probabilistic accounts to modern neural and multimodal formulations [2, 3, 11, 27, 28, 31–34, 48, 49]. Concept Bottleneck Models and their variants further show that concept structure can support interpretable prediction, intervention, and probabilistic reasoning [24, 35, 53, 56]. Despite these advances, many concept-learning methods still depend on externally specified semantic information. Such information is costly to obtain and difficult to scale.

We therefore ask whether a generative model can discover concept components from data alone, with components emerging from the same evidence lower bound used to model the data distribution. No class labels, attribute annotations, captions, pretrained text-image priors, or semantic concept supervision are used to define or learn them.

The Dirichlet Concept Diffusion Model (DCDM) makes concept discovery intrinsic to the diffusion model's process of learning to generate. Standard diffusion perturbs all samples toward a single zero-centered Gaussian target, independent of their underlying content. In contrast, DCDM learns a set of concept centers and, for each input, infers a Dirichlet distribution over them. The weighted average of these centers defines the forward diffusion mean and steers the reverse denoising process.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Through these dynamics, the data-likelihood objective drives the centers to capture structure that explains and generates the data.

The proposed formulation departs from mixture and discrete latent-variable models that place component structure in a latent bottleneck and typically commit each sample to a single component [1, 6, 20, 25, 41]. Separately, prior Dirichlet VAEs place the Dirichlet on a VAE bottleneck latent [21], but do not learn explicit concept centers in data space or integrate the Dirichlet into diffusion dynamics. DCDM instead infers a Dirichlet distribution that combines multiple concept centers in different proportions, allowing recurring patterns to be shared across samples while preserving instance-level variation, and threads concept structure through the corruption path, the reverse posterior, and generation themselves. In contrast to prior generative approaches to concept discovery that obtain semantic structure through pretrained text-to-image models, post-hoc decomposition, or external concept sources [5, 8, 29, 47, 54], DCDM takes a fully unsupervised route in which concept discovery arises from a single data-likelihood objective.

We make three contributions. (1) We introduce DCDM, a fully unsupervised diffusion formulation that learns Dirichlet-weighted concept centers from the data-likelihood objective alone. (2) We analyze how the formulation preserves concept-level information through the forward diffusion path and reduces denoising ambiguity in the reverse process, creating pressure for concept centers to capture stable modes as generation proceeds from concept-level recovery to instance-level refinement. (3) We evaluate DCDM across diverse and visually complex image domains through qualitative, interventional, and quantitative diagnostics, showing that the learned components consistently reflect recurring visual structure.

2 Method

2.1 Preliminaries

DDPM. Denoising Diffusion Probabilistic Models (DDPMs) [18, 23, 46] define a Markov forward process that gradually perturbs data toward Gaussian noise and train a neural reverse process to denoise it. With variance schedule $\{\beta_t\}_{t=1}^T$, the forward transition is

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t \mid \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

The learned model approximates the corresponding reverse transitions, typically using a Gaussian denoising parameterization.

Latent Diffusion. Pixel-space diffusion is computationally expensive for high-resolution images. Latent diffusion models first encode images into a lower-dimensional VAE latent space, train the diffusion model in that space, and decode denoised latents back to images [42].

2.2 Dirichlet Concept Diffusion Model

The core idea behind the proposed Dirichlet Concept Diffusion Model (DCDM) is simple and intuitive. Instead of using a single transition kernel that perturbs all input data toward a standard Gaussian, regardless of their semantic structure, we introduce a Gaussian transition kernel whose mean is parameterized as a weighted combination of learned concept centers, which biases the diffusion process toward structured regions of the data space. Naturally, each component corresponds to a distinct "concept" within a data distribution that is often multimodal. At a high level, generating meaningful samples requires the model to capture shared structure across data points and represent them in terms of these underlying concepts. Given an input data point, the model infers a distribution over these concept centers, which composes multiple concepts into a weighted representation that parameterizes the transition dynamics. Concept discovery therefore arises from the same unsupervised objective used to learn the generative model. Figure 1 contrasts the forward processes of DDPM and DCDM.

Notation. We largely follow the standard notation used in DDPMs, with only necessary modifications. Specifically, we denote the data at timestep $t \in [0, T]$ by $\mathbf{x}_t$. We use q_ϕ for the variational

(a) DDPM
All samples drift toward a single Gaussian.

(b) DCDM
Each sample drifts toward a Dirichlet-weighted concept center.

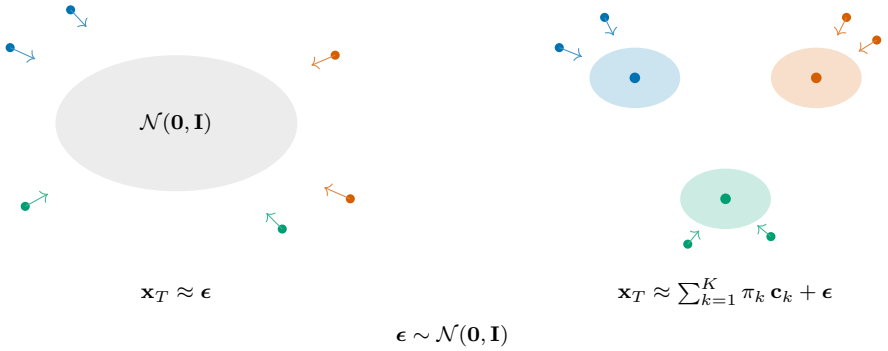

Figure 1: **Concept discovery through generative learning.** DDPM progressively removes sample-dependent structure as samples approach a common zero-centered Gaussian. DCDM instead drifts each sample toward a Dirichlet-weighted combination of learned concept centers, retaining concept-level information in the diffusion mean even at high noise. Optimizing a fully unsupervised generative objective encourages these centers to capture shared semantic structure, allowing concepts to emerge without labels or concept supervision.

forward process, p_θ for the denoising transitions in the reverse process and $p_{\theta,\phi}$ for the full generative joint. For a family of K concept centers, let $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})$ denote a latent variable on the probability simplex Δ^{K-1}, representing a distribution over the K components, where $\boldsymbol{\alpha} \in \mathbb{R}_+^K$. Let $k \in \{1, \ldots, K\}$ index the k-th component, with learnable concept center $\mathbf{c}_{\phi,k}$, which we abbreviate as $\mathbf{c}_k$ when the context is clear. We define the aggregated concept center corresponding to $\boldsymbol{\pi}$ as $\mathbf{c}(\boldsymbol{\pi}) := \sum_{k=1}^{K} \pi_k \mathbf{c}_k$. To avoid confusion, we distinguish between different mean-like quantities. The notation $\mathbf{c}_k$ denotes the learnable center of the k-th component, $\mathbf{c}(\boldsymbol{\pi})$ denotes the aggregated concept center parameterized by $\boldsymbol{\pi}$, $\widetilde{\boldsymbol{\mu}}_t(\cdot)$ denotes the posterior mean in the reverse step, and $\boldsymbol{\mu}_\theta(\cdot)$ denotes the neural-network output in the denoising process. The noise schedule is defined as the sequence $\{\beta_t\}_{t=1}^T$, where each $\beta_t \in (0, 1)$. We define $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.

2.2.1 Forward Diffusion

Similar to standard DDPM, a Gaussian transition kernel defines the forward diffusion process in our model. The diffusion gradually pulls input data toward the aggregated concept center $\mathbf{c}(\boldsymbol{\pi})$ under a shared schedule $\{\eta_t\}_{t=1}^T$. Specifically, we have:

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}) = \mathcal{N}(\mathbf{x}_t \mid \sqrt{\alpha_t}\mathbf{x}_{t-1} + \eta_t \mathbf{c}(\boldsymbol{\pi}), \beta_t \mathbf{I}). \tag{1}$$

We set $\eta_t = 1 - \sqrt{\alpha_t}$, which makes the aggregated concept center $\mathbf{c}(\boldsymbol{\pi})$ the fixed point of the noiseless dynamics. Under this choice, the marginal at timestep t admits the closed form

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1 - \sqrt{\bar{\alpha}_t})\mathbf{c}(\boldsymbol{\pi}), (1 - \bar{\alpha}_t)\mathbf{I}\right),$$

2.2.2 Reverse Diffusion

For $t \geq 2$, the one-step reverse posterior conditioned on $\mathbf{x}_0$ is the closed-form Gaussian

$$\begin{aligned} q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) &= \mathcal{N}\Big(\mathbf{x}_{t-1} \mid \widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}), \tilde{\beta}_t \mathbf{I}\Big), \\ \widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) &= A_t \mathbf{x}_t + B_t \mathbf{x}_0 + C_t \mathbf{c}(\boldsymbol{\pi}), \\ A_t = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}, &\qquad B_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1 - \bar{\alpha}_t}, \\ C_t = 1 - A_t - B_t = \frac{(1 - \sqrt{\alpha_t})(1 - \sqrt{\bar{\alpha}_{t-1}})}{1 + \sqrt{\bar{\alpha}_t}}, &\qquad \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\,\beta_t. \end{aligned} \tag{2}$$

A generator trained to approximate the posterior distribution takes the form:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t), \tilde{\beta}_t \mathbf{I}\right). \tag{3}$$

We initialize the reverse process from a concept-dependent terminal distribution $p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}) = \mathcal{N}(\mathbf{x}_T \mid \mathbf{c}(\boldsymbol{\pi}), \mathbf{I})$.

2.2.3 Fitting the Model

We fit the model by maximizing the evidence lower bound (ELBO). Given a data sample $\mathbf{x}_0$, the ELBO for its log data likelihood is given by:

$$\log p_{\theta,\phi}(\mathbf{x}_0) \geq \mathbb{E}_{q_\phi}\left[\log \frac{p_{\theta,\phi}(\mathbf{x}_0, \mathbf{x}_{1:T}, \boldsymbol{\pi})}{q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] := -\mathcal{L}(\mathbf{x}_0). \tag{4}$$

The variational encoder factorizes as

$$q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0) = q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) q_\phi(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \boldsymbol{\pi}) = q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \prod_{t=1}^{T} q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}) \tag{5}$$

where $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$ is modeled as a Dirichlet distribution, $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) = \mathrm{Dir}(\boldsymbol{\alpha})$, and is parameterized by a neural network denoted as $f^\phi_{\text{concept}} : \mathbf{x}_0 \mapsto \mathbb{R}^K_+$. It takes a clean input $\mathbf{x}_0$ and outputs concentration parameters $\boldsymbol{\alpha}$.

The Dirichlet latent variable $\boldsymbol{\pi}$ induces a compositional structure that allows multiple components to be active simultaneously and contribute to the representation in different proportions. As a result, the model can capture recurring patterns by reusing components across data instances, leading to a more structured and expressive representation of complex, multimodal data. Furthermore, this formulation naturally extends to multiple groups of components. One can introduce a collection of latent variables $\{\boldsymbol{\pi}^{(g)}\}_{g=1}^G$, each defined on its own simplex and associated with a separate set of component centers. The aggregated representation is then formed by combining contributions across groups, enabling different groups of components to represent distinct aspects of variation.

Combining the concept-dependent terminal distribution with the reverse denoising transitions, we have

$$p_{\theta,\phi}(\mathbf{x}_0, \mathbf{x}_{1:T}, \boldsymbol{\pi}) = p(\boldsymbol{\pi}) p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi}). \tag{6}$$

where $p(\boldsymbol{\pi})$ is the prior over the simplex. In this work, we fix $p(\boldsymbol{\pi})$ to be a uniform prior and leave other options, such as learnable priors, as future work.

Substituting Eqs. (5) and (6) into Eq. (4) and rearranging terms, we obtain

$$\begin{aligned}
\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)} \Bigg[\underbrace{D_{\mathbb{KL}}(q_\phi(\mathbf{x}_T \mid \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}))}_{L_T} \underbrace{-\mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0,\boldsymbol{\pi})} \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \boldsymbol{\pi})}_{L_0} \Bigg] + \\
\mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)} \left[\sum_{t=2}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\pi})} \underbrace{D_{\mathbb{KL}}(q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi}))}_{L_{t-1}} \right] + \\
\underbrace{D_{\mathbb{KL}}\left(q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \,\|\, p(\boldsymbol{\pi})\right)}_{L_{\text{reg}}}.
\end{aligned} \tag{7}$$

After dropping constants, the terminal KL L_T contributes only $\frac{1}{2}\bar{\alpha}_T\|\mathbf{x}_0 - \mathbf{c}(\boldsymbol{\pi})\|_2^2$. Since diffusion schedules choose $\bar{\alpha}_T \approx 0$, this contribution is negligible, and we omit L_T from the training objective. Following [18], we replace the reconstruction and denoising terms L_0 and L_{t-1} with the standard simplified surrogate loss $L_{\text{simple}} = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\, t\sim\text{Unif}(1,T)}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)\|_2^2\right]$, where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1-\sqrt{\bar{\alpha}_t})\mathbf{c}(\boldsymbol{\pi}) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. Consistent with DDPM, the denoising network $\boldsymbol{\epsilon}_\theta(\cdot)$ is trained to predict the noise $\boldsymbol{\epsilon}$. We then approximate the expectation over $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$ using Monte Carlo sampling during training and optimize

$$\mathcal{L}(\theta, \phi; \mathbf{x}_0) = \mathbb{E}_{\boldsymbol{\pi}\sim q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)}\big[L_{\text{simple}}\big] + D_{\mathbb{KL}}\big(q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \,\|\, p(\boldsymbol{\pi})\big). \tag{8}$$

We denote the first term in Eq. (8) by L_{diff} and the KL term by L_{reg}.

Given $\mathbf{x}_0$, L_{diff} encourages the concept encoder $f^{\phi}_{\text{concept}}$, which parameterizes $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$, to produce latent distributions that enable accurate denoising of $\mathbf{x}_0$. The concept centers are shared across samples and optimized jointly with the encoder and denoiser. Gradients from L_{diff} reach the centers through both the concept-dependent forward mean and the denoiser's conditioning input. The linear aggregation $\mathbf{c}(\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathbf{c}_k$ scales each sample's gradient contribution to center $\mathbf{c}_k$ by its weight π_k, as inferred by the concept encoder through $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$. Together, these updates encourage the inferred concept distributions and shared centers to capture information useful for reconstructing the data under the diffusion process. As a result, the model naturally discovers structure in the data through the objective of improving denoising performance.

On the other hand, L_{reg} regularizes $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$ by discouraging degenerate solutions and encouraging diversity in the latent distributions. Without this regularization, the model may collapse to producing nearly identical or highly concentrated Dirichlet distributions, effectively reducing to standard DDPM behavior.

Appendix B gives the full derivations.

2.3 Latent Dirichlet Concept Diffusion

To improve computational efficiency, we extend DCDM to a latent formulation in a pretrained VAE latent space. Let $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$, where $\mathcal{E}$ denotes the encoder. All diffusion states and learnable concept centers $\{\mathbf{c}_k\}$ are then defined in latent space.

The forward process follows the same form as Eq. (1), with $\mathbf{x}$ replaced by $\mathbf{z}$.

$$q_\phi(\mathbf{z}_t \mid \mathbf{z}_{t-1}, \boldsymbol{\pi}) = \mathcal{N}(\mathbf{z}_t \mid \sqrt{\alpha_t}\mathbf{z}_{t-1} + (1-\sqrt{\alpha_t})\mathbf{c}(\boldsymbol{\pi}), \beta_t\mathbf{I}). \tag{9}$$

The reverse process and training objective follow identically in latent space, with the denoising model trained to predict noise on $\mathbf{z}_t$ under Eq. (8). During generation, the reverse process first produces $\mathbf{z}_0$, which is decoded to pixel space as $\mathbf{x}_0 = \mathcal{D}(\mathbf{z}_0)$.

3 Theoretical Perspective

We analyze why DCDM encourages concept discovery through the denoising objective. The aggregated concept center plays two roles, shaping diffusion trajectories and reducing ambiguity in denoising. Optimizing the objective therefore creates pressure for the model to discover stable latent centers that capture shared semantic structure across samples.

Diffusion-path mechanism. To analyze how the proposed forward process differs from standard diffusion, decompose the clean sample as $\mathbf{x}_0 = \mathbf{c}(\boldsymbol{\pi}) + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the instance-specific residual around the aggregated concept center. Under standard diffusion,

$$\mathbf{x}_t^{\text{std}} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} = \sqrt{\bar{\alpha}_t}\mathbf{c}(\boldsymbol{\pi}) + \sqrt{\bar{\alpha}_t}\boldsymbol{\delta} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},$$

whereas under the proposed process,

$$\mathbf{x}_t^{\text{prop}} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1-\sqrt{\bar{\alpha}_t})\mathbf{c}(\boldsymbol{\pi}) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} = \mathbf{c}(\boldsymbol{\pi}) + \sqrt{\bar{\alpha}_t}\boldsymbol{\delta} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The key structural difference is that, under standard diffusion, the concept-level signal is attenuated by $\sqrt{\bar{\alpha}_t}$, whereas under the proposed process, the aggregated center remains in

the mean at every timestep. As a result, at the same noise level, the proposed forward process preserves more data-dependent semantic information, which remains visible even at high noise levels.

The same aggregated center also enters the reverse process directly. The exact reverse posterior mean can be written in the affine form $\widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) = A_t\mathbf{x}_t + B_t\mathbf{x}_0 + C_t\mathbf{c}(\boldsymbol{\pi})$, where $C_t \geq 0$. Consequently, for fixed $\mathbf{x}_t$ and $\mathbf{x}_0$,

$$\widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}_1) - \widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}_2) = C_t\big(\mathbf{c}(\boldsymbol{\pi}_1) - \mathbf{c}(\boldsymbol{\pi}_2)\big).$$

The aggregated center induces separation between samples associated with different inferred semantics, steering both the forward and reverse processes. Consequently, samples with similar inferred semantics are encouraged to align around similar aggregated centers, whereas samples with different semantics follow separated diffusion paths. The proposed process therefore provides a mechanism that enables the model to organize samples around semantically meaningful concept centers.

Denoising mechanism. At timestep t, denoising can be viewed as a regression problem in which a noisy sample $\mathbf{x}_t$ is used to predict a target $\mathbf{u}$, which may be $\mathbf{x}_0$, $\boldsymbol{\epsilon}$, or an equivalent parameterization. A noisy sample may remain compatible with multiple plausible clean explanations, especially at large noise levels. In standard denoising, the Bayes-optimal predictor only observes $\mathbf{x}_t$, so it must average across explanations that remain consistent with the noisy sample. In the proposed model, the inferred concept distribution $\boldsymbol{\pi}$ reduces that ambiguity by steering prediction toward the concept-consistent explanation. The theorem below characterizes the Bayes-optimal denoisers with and without access to $\boldsymbol{\pi}$ and quantifies the resulting risk gap $R_{\mathrm{std}}(t) - R_{\mathrm{guided}}(t)$.

Theorem 3.1 (From concepts to details in denoising)**.** *Assume the decomposition* $\mathbf{x}_0 = \mathbf{c}(\boldsymbol{\pi}) + \boldsymbol{\delta}$*, and suppose* $\boldsymbol{\delta} \mid \boldsymbol{\pi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\delta)$*. Define* $\mathbf{G}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_\delta(\bar{\alpha}_t\boldsymbol{\Sigma}_\delta + (1-\bar{\alpha}_t)\mathbf{I})^{-1}$*. Then the Bayes-optimal denoisers for predicting* $\mathbf{x}_0$ *satisfy* $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \boldsymbol{\pi}] = \mathbf{G}_t\mathbf{x}_t + (\mathbf{I} - \mathbf{G}_t)\mathbf{c}(\boldsymbol{\pi})$ *and* $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] = \mathbf{G}_t\mathbf{x}_t + (\mathbf{I} - \mathbf{G}_t)\mathbb{E}[\mathbf{c}(\boldsymbol{\pi}) \mid \mathbf{x}_t]$*. Let* $R_{\mathrm{guided}}(t) = \mathbb{E}[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \boldsymbol{\pi}]\|^2]$ *and* $R_{\mathrm{std}}(t) = \mathbb{E}[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]\|^2]$*. Then*

$$R_{\mathrm{std}}(t) - R_{\mathrm{guided}}(t) = \mathbb{E}\Big[\mathrm{tr}((\mathbf{I} - \mathbf{G}_t)\boldsymbol{\Sigma}_c(\mathbf{x}_t)(\mathbf{I} - \mathbf{G}_t)^\top)\Big],$$

where $\boldsymbol{\Sigma}_c(\mathbf{x}_t) = \mathbb{E}[(\mathbf{c}(\boldsymbol{\pi}) - \mathbb{E}[\mathbf{c}(\boldsymbol{\pi}) \mid \mathbf{x}_t])(\mathbf{c}(\boldsymbol{\pi}) - \mathbb{E}[\mathbf{c}(\boldsymbol{\pi}) \mid \mathbf{x}_t])^\top \mid \mathbf{x}_t]$.

The proof is deferred to Appendix C.

Under the homoskedastic-Gaussian residual assumption, the theorem characterizes the gain from concept guidance in terms of the posterior uncertainty of the aggregated center $\mathbf{c}(\boldsymbol{\pi})$ that remains unresolved from $\mathbf{x}_t$, weighted by $(\mathbf{I} - \mathbf{G}_t)$. At high noise levels, $\mathbf{x}_t$ provides limited information about the residual detail $\boldsymbol{\delta}$, so the inferred $\boldsymbol{\pi}$ helps reduce ambiguity among multiple plausible clean explanations. As the noise decreases, $\mathbf{x}_t$ becomes increasingly informative about instance-specific detail, and the Bayes-optimal predictor correspondingly places more weight on the observation. Thus, the analysis implies a coarse-to-fine denoising behavior in which concept-level information helps organize early predictions, while later steps refine instance-level details. Figure 2 provides a qualitative counterpart to this interpretation, showing that the proposed trajectories resolve coherent coarse structure earlier than the baseline and preserve it through later refinement. We empirically validate the risk gap in Appendix F.1, where the measured $R_{\mathrm{std}}(t) - R_{\mathrm{guided}}(t)$ tracks the qualitative shape implied by Theorem 3.1.

Taken together, these results establish that the denoising objective creates structural and statistical pressure for stable concept centers to emerge during learning. In the diffusion process, the aggregated center remains active in both the forward trajectory and the reverse posterior, organizing generation around concept-dependent paths. On top of those paths, Theorem 3.1 shows that Bayes-optimal denoising naturally transitions from concept-level recovery at high noise toward data-point-specific refinement at low noise. These two mechanisms jointly create training pressure favoring stable, semantically organized concept centers.

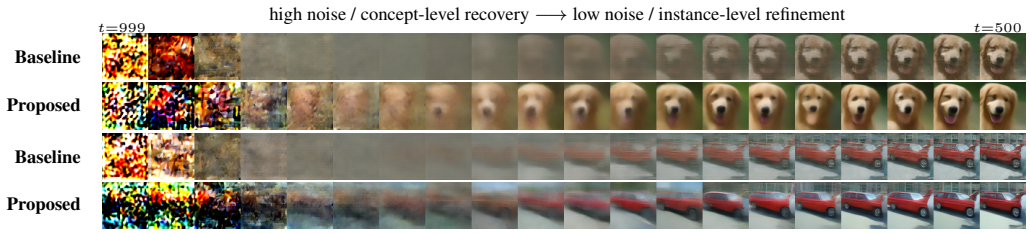

Figure 2: Matched reverse denoising trajectories for two examples. Each pair compares the baseline and the proposed model over the same reverse-time interval. Consistent with the coarse-to-fine interpretation of Theorem 3.1, the proposed trajectories resolve coherent coarse structure earlier and then refine instance-specific details on top of it.

4 Empirical Evaluation

We test whether the unsupervised components discovered by DCDM behave like meaningful visual concepts rather than arbitrary internal labels. Evaluation covers FashionMNIST [52], CIFAR-10 [26], Conceptual ImageNet[1], CelebA [30], LSUN Bedroom, and LSUN Church [55]. The study first visualizes learned components qualitatively (Sections 4.1–4.2), then quantifies concept validity through three diagnostics (Section 4.3), and finally evaluates source-generation agreement (Section 4.4). Full experimental details are in Appendix E.

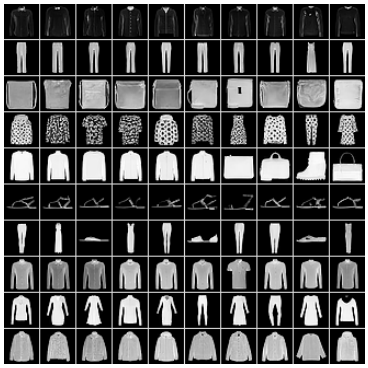

Figure 3: Uncurated generated samples from discovered DCDM concepts on FashionMNIST, organized by learned component with all components included.

4.1 Pixel-Space DCDM

We begin in pixel space, applying DCDM to FashionMNIST and CIFAR-10. Figure 3 shows uncurated FashionMNIST samples generated from each learned component. Components recover footwear, bags, outerwear, pants, and recurring texture patterns, among others. Similar component-wise coherence appears on CIFAR-10 (first row of Figure 4).

4.2 Concept Prototypes Across Datasets

For the remaining experiments, we use the latent formulation for computational efficiency. Figure 4 provides a qualitative overview of the learned concepts across datasets.

For each learned component, we generate samples from concept distributions with high corresponding posterior responsibility. The learned components consistently capture coherent visual concepts, with complete component-wise visualizations in Appendix F.2. Generation quality, measured with concept distributions sampled from the empirical aggregated posterior, favors DCDM on every dataset except CelebA, where the baseline retains a small advantage. Full results and sampling details appear in Table 7 and Appendix F.3.

4.3 Discovered Concept Validity

A discovered concept should have a targeted effect under intervention, collect samples into coherent semantic neighborhoods, and be predictable from independent visual features. Throughout these diagnostics, f_{emb} denotes a normalized frozen image encoder, and $\widehat{\boldsymbol{\pi}}_i = \mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_{0,i})}[\boldsymbol{\pi}]$ denotes the inferred posterior mean for $\mathbf{x}_{0,i}$. When its argument is a concept distribution, f_{emb} abbreviates the embedding of a generated sample using that distribution.

Concept ablation change. For a learned component to be meaningful, changing its mass should have a targeted effect on generation. *Concept ablation change* (CAC) tests this criterion by ablat-

[1] A curated ImageNet subset [44] chosen to cover a balanced range of visual semantics with construction details in Appendix E.1.

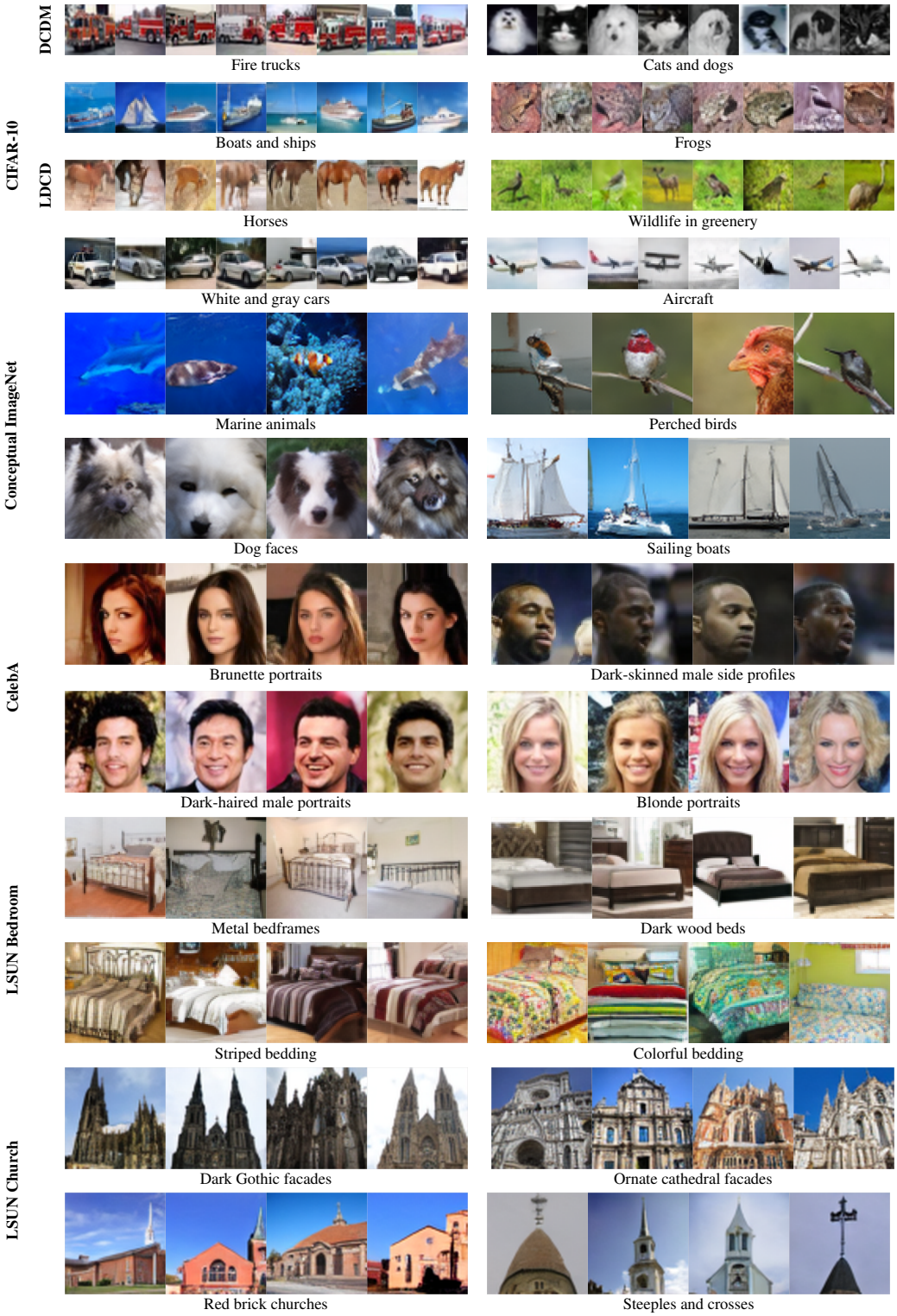

Figure 4: Generated prototypes associated with learned concept components across datasets. Text below each strip gives a post hoc semantic interpretation. The CIFAR-10 top row uses pixel-space DCDM, while the remaining rows use the latent-space formulation. Components are learned without labels, attribute annotations, text captions, or pretrained text-image priors.

ing one component from an inferred concept distribution $\boldsymbol{\pi}$. A content-bearing component should produce a larger semantic shift than a control edit that removes a low-mass non-target component.

For $\boldsymbol{\pi} \in \Delta^{K-1}$, let $\boldsymbol{\pi}^{\setminus k}$ denote $\boldsymbol{\pi}$ with π_k replaced by zero and the remaining mass renormalized onto the simplex. For a target concept k and a low-mass non-target concept k' chosen as a control, we measure $\Delta_{\text{tgt}} = \|f_{\text{emb}}(\boldsymbol{\pi}) - f_{\text{emb}}(\boldsymbol{\pi}^{\setminus k})\|_2$ and define Δ_{ctrl} analogously by replacing k with k'. The CAC lift, $\Delta_{\text{tgt}} - \Delta_{\text{ctrl}}$, isolates concept-specific semantic change from generic sensitivity to perturbing $\boldsymbol{\pi}$.

Figure 5 shows that targeted ablation smoothly removes a specific visual attribute, while the control panel leaves the image largely unchanged, indicating the effect is semantic rather than generic. The CAC columns of Table 2 confirm the effect quantitatively. CAC lift is positive on every dataset, ranging from 0.72 to 1.21, and remains positive under a stricter control that ablates the second-highest-mass non-target component (Appendix F.4). The learned components are therefore interventionally meaningful and correspond to identifiable visual regularities in generated images.

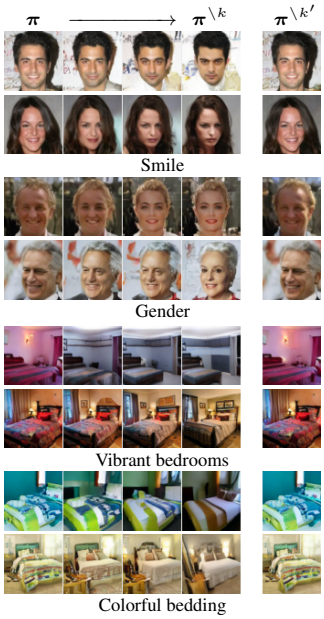

Figure 5: CAC sweeps with controls at right.

Concept coherence and predictability. We use two complementary tests to evaluate whether the discovered concepts correspond to visual structure. Neighborhood Coherence (NC) is a chance-normalized metric measuring whether samples generated under the same concept are close in an independent embedding space, with NC = 0 corresponding to chance and NC = 1 to perfect alignment. Linear probe predictability measures whether independent image features can predict $\widehat{\boldsymbol{\pi}}_i$. Across encoder choices and datasets, NC is positive on every dataset, ranging from 0.186 ± 0.009 to 0.387 ± 0.012, and probe R^2 ranges from 0.437 ± 0.003 to 0.741 ± 0.001 (Appendix F.5). Together, these results indicate that DCDM concepts capture semantically related visual structure and track visual content rather than internal labels.

We further evaluate whether the inferred concept distributions capture CelebA's annotated visual attributes, following InfoDiffusion's evaluation protocol [51]. Attribute annotations are used only for evaluation. As shown in Table 1, DCDM scores higher than both published InfoDiffusion configurations on all three reported metrics, providing complementary evidence that the inferred concept distributions capture annotated visual attributes on CelebA.

Table 1: CelebA representation evaluation following InfoDiffusion's released protocol. InfoDiffusion results are taken from Wang et al. [51]. DCDM values are means ± standard deviations across five evaluation folds using one checkpoint.

Method	Latent dim.	TAD ↑	Captured attrs. ↑	Macro AUROC ↑
InfoDiffusion ($\lambda = 0.01$)	32	0.299 ± 0.006	3.0 ± 0.0	0.836 ± 0.002
InfoDiffusion ($\lambda = 0.1$)	32	0.192 ± 0.004	2.8 ± 0.4	0.848 ± 0.001
DCDM	24	$\mathbf{0.595 \pm 0.033}$	$\mathbf{8.6 \pm 0.5}$	$\mathbf{0.858 \pm 0.001}$

4.4 Source-Generation Agreement

Source-generation agreement (SGA) measures whether a sample generated from the inferred concept distribution remains semantically close to the source image. Given a real image $\mathbf{x}_{0,i}$, we use $\widehat{\boldsymbol{\pi}}_i$ to generate a sample $\mathbf{x}'_{0,i}$ and measure $S_{\text{match},i} = \cos(f_{\text{emb}}(\mathbf{x}_{0,i}), f_{\text{emb}}(\mathbf{x}'_{0,i}))$. We define $S_{\text{mis},i}$ analogously by replacing the generated sample with $\mathbf{x}'_{0,j}$ from a mismatched source $j \neq i$. The SGA lift, $S_{\text{match},i} - S_{\text{mis},i}$, is computed per source image and averaged over the dataset.

The SGA columns of Table 2 show positive lifts on every dataset, ranging from 0.17 to 0.67, well above the reported 2σ variation. Appendix F.6 reports qualitative visualizations and fuller metrics across encoder choices that exhibit the same pattern. The positive lift shows that the inferred concept distribution carries source-image semantics that shape generation.

Table 2: CAC and SGA metrics.

	CAC			SGA		
Dataset	$\Delta_{\text{tgt}} \uparrow$	$\Delta_{\text{ctrl}} \downarrow$	lift $\uparrow$	$S_{\text{match}} \uparrow$	$S_{\text{mis}} \downarrow$	lift $\uparrow$
CIFAR-10	1.283 ± 0.004	0.069 ± 0.004	1.214 ± 0.007	0.759 ± 0.001	0.094 ± 0.000	0.665 ± 0.001
C-ImageNet	1.250 ± 0.003	0.356 ± 0.009	0.893 ± 0.010	0.242 ± 0.000	0.031 ± 0.000	0.211 ± 0.000
CelebA	0.990 ± 0.010	0.162 ± 0.001	0.828 ± 0.010	0.648 ± 0.000	0.335 ± 0.000	0.313 ± 0.000
LSUN Bedroom	1.017 ± 0.018	0.297 ± 0.004	0.720 ± 0.022	0.481 ± 0.001	0.314 ± 0.002	0.167 ± 0.002
LSUN Church	1.189 ± 0.008	0.264 ± 0.004	0.925 ± 0.007	0.360 ± 0.001	0.098 ± 0.000	0.262 ± 0.001

Additional ablations and auxiliary diagnostics are reported in Appendix F. Ablating the number of learned components K shows that DCDM remains robust in centering generation around coherent discovered concepts (Table 12). As concept capacity increases, DCDM captures more source-image-specific semantic detail, while excessive capacity can lead to overlapping semantic roles across different components. K controls the capacity of the shared concept dictionary and need not match the number of semantic classes. Our single-group CIFAR-10 sweep suggests a practical heuristic of choosing the smallest K at which CAC and SGA have largely plateaued while neighborhood coherence remains strong. We also compare against a foundation-model-distilled variant of DCDM in Appendix F.8, which is closer in spirit to prior concept-discovery methods that rely on pretrained priors. Unsupervised DCDM matches the distilled variant on the interventional metrics (CAC, SGA), indicating that external semantic supervision is not required for the learned components to control generation. Distillation primarily improves geometric alignment between the concept space and external semantic representations (NC, probe R^2).

5 Discussion and Limitations

DCDM suggests that a plain data log-likelihood objective can make learned visual structure explicit. The learned components not only enable coherent sample generation but also produce, for each sample, an inferred concept distribution whose components can be perturbed to test what each contributes. Because our goal is fully unsupervised concept discovery, we evaluate the discovered components through complementary desiderata including targeted changes when component mass is perturbed, coherence and predictability, and source-generation agreement. These diagnostics test whether components act as reusable visual concepts during generation, rather than only whether post hoc labels can be assigned to samples. By design, DCDM adds little computational overhead over standard diffusion. The concept posterior is estimated by Monte Carlo during training, and the reverse denoiser stays architecturally close to a plain diffusion network, so the only meaningful added cost is the concept encoder itself.

A current limitation is that the number of components fixes the granularity of the discovered concept space. Across moderate capacity changes, the learned components remain coherent and interpretable, while excessive capacity can fragment related structure or assign overlapping semantic roles to multiple components. A natural direction is adaptive concept growth and merging, in the spirit of Chinese restaurant processes and rational models of categorization [2]. The inferred concept distributions in DCDM may provide indirect signals for such a method. Diverse samples assigned nearly identical concept distributions suggest that the current components are too coarse and warrant adding new ones, while components with similar activation patterns across data are candidates for merging. A complementary direction is hierarchical concepts that vary along the diffusion path. As Theorem 3.1 shows denoising progresses from concept-level recovery at high noise to instance-specific refinement at low noise, coarser concepts could organize early timesteps while finer ones shape later refinement, extending DCDM from a flat concept space to a hierarchical one.

If creation is to serve as evidence of understanding, DCDM shows that generative models are capable of not only creating realistic samples but also exposing the reusable structure behind creation in forms humans can inspect and test.

Acknowledgments

This work was supported in part by the Office of Naval Research (ONR) under award N000142612461.

References

[1] Ehsan Abbasnejad, Anthony Dick, and Anton van den Hengel. Infinite variational autoencoder for semi-supervised learning, 2016. URL https://arxiv.org/abs/1611.07800.

[2] John R Anderson. The adaptive nature of human categorization. *Psychological review*, 98(3): 409, 1991.

[3] Dana Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988.

[4] Denis Blessing, Xiaogang Jia, and Gerhard Neumann. End-to-end learning of gaussian mixture priors for diffusion sampler. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iXbUquaWbl.

[5] Yusuf Dalva and Pinar Yanardag. Noiseclr: A contrastive learning approach for unsupervised discovery of interpretable directions in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24209–24218, June 2024.

[6] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[7] Yilun Du and Leslie Kaelbling. Compositional generative modeling: A single model is not all you need, 2024. URL https://arxiv.org/abs/2402.01103.

[8] Yilun Du, Shuang Li, Yash Sharma, Joshua B. Tenenbaum, and Igor Mordatch. Unsupervised learning of compositional energy concepts, 2021. URL https://arxiv.org/abs/2111.03042.

[9] Khashayar Gatmiry, Jonathan Kelner, and Holden Lee. Learning mixtures of gaussians using diffusion models, 2025. URL https://arxiv.org/abs/2404.18869.

[10] Yuchong Geng and YUCHEN LU. Atelier: A human-owned protocol for auditable AI-assisted research. In *ICML 2026 Workshop: AI as a Tool for Mathematics, Computer Science, and Machine Learning*, 2026. URL https://openreview.net/forum?id=9h3ZudPP3G.

[11] Yuchong Geng and Ao Tang. A concept-centric approach to multi-modality learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=8WAAPP32c7.

[12] Yuchong Geng, Dongyue Zhang, Po-han Li, Oguzhan Akcin, Ao Tang, and Sandeep P. Chinchali. Decentralized sharing and valuation of fleet robotic data. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1795–1800. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/geng22a.html.

[13] Sang gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior, 2022. URL https://arxiv.org/abs/2106.06406.

[14] Hanzhong Guo, Cheng Lu, Fan Bao, Tianyu Pang, Shuicheng Yan, Chao Du, and Chongxuan Li. Gaussian mixture solvers for diffusion models, 2023. URL https://arxiv.org/abs/2311.00941.

[15] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18100–18115. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/72dad95a24fae750f8ab1cb3dab5e58d-Paper-Conference.pdf.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[17] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[19] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi: 10.1214/aoms/1177703732. URL https://doi.org/10.1214/aoms/1177703732.

[20] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.01144.

[21] Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder, 2019. URL https://arxiv.org/abs/1901.02739.

[22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:3380833.

[23] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *CoRR*, abs/2107.00630, 2021. URL https://arxiv.org/abs/2107.00630.

[24] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020.

[25] Lingjing Kong, Guangyi Chen, Biwei Huang, Eric P. Xing, Yuejie Chi, and Kun Zhang. Learning discrete concepts in latent hierarchical models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=bO5bUxvH6m.

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[27] John Kruschke. Alcove: A connectionist model of human category learning. In R.P. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3. Morgan-Kaufmann, 1990. URL https://proceedings.neurips.cc/paper_files/paper/1990/file/8fe0093bb30d6f8c31474bd0764e6ac0-Paper.pdf.

[28] Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. A Competence-Aware Curriculum For Visual Concepts Learning via Question Answering. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, page 141–157, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58535-8. doi: 10.1007/978-3-030-58536-5_9. URL https://doi.org/10.1007/978-3-030-58536-5_9.

[29] Nan Liu, Yilun Du, Shuang Li, Joshua B. Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models, 2023. URL https://arxiv.org/abs/2306.05357.

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[31] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *CoRR*, abs/1904.12584, 2019. URL http://arxiv.org/abs/1904.12584.

[32] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretabile, leakproof concept-based models, 2022.

[33] Lingjie Mei, Jiayuan Mao, Ziqi Wang, Chuang Gan, and Joshua B. Tenenbaum. FALCON: Fast Visual Concept Learning by Integrating Images, Linguistic descriptions, and Conceptual Relations. *ArXiv*, abs/2203.16639, 2022.

[34] Tom M. Mitchell. *Machine Learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997. ISBN 978-0-07-042807-2. URL https://www.worldcat.org/oclc/61321007.

[35] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models, 2023. URL https://arxiv.org/abs/2304.06129.

[36] Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *CoRR*, abs/2201.00308, 2022. URL https://arxiv.org/abs/2201.00308.

[37] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019.

[38] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. *CoRR*, abs/1709.07871, 2017. URL http://arxiv.org/abs/1709.07871.

[39] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. *CoRR*, abs/2111.15640, 2021. URL https://arxiv.org/abs/2111.15640.

[40] 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*, pages 8748–8763. PmLR, 2021.

[41] Dushyant Rao, Francesco Visin, Andrei A. Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:202772066.

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[45] Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the ddpm objective, 2023. URL https://arxiv.org/abs/2307.01178.

[46] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

[47] Jocelin Su, Nan Liu, Yanbo Wang, Joshua B. Tenenbaum, and Yilun Du. Compositional image decomposition with diffusion models, 2024. URL https://arxiv.org/abs/2406.19298.

[48] Ryan Teehan, Brenden Lake, and Mengye Ren. College: Concept embedding generation for large language models, 2024. URL https://arxiv.org/abs/2403.15362.

[49] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1025. URL https://aclanthology.org/P18-1025.

[50] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering, 2024. URL https://arxiv.org/abs/2409.02426.

[51] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models, 2023. URL https://arxiv.org/abs/2306.08757.

[52] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

[53] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations, 2024. URL https://arxiv.org/abs/2401.14142.

[54] Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models, 2023. URL https://arxiv.org/abs/2301.13721.

[55] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[56] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models, 2023. URL https://arxiv.org/abs/2205.15480.

[57] Martin Zach, Thomas Pock, Erich Kobler, and Antonin Chambolle. *Explicit Diffusion of Gaussian Mixture Model Based Image Priors*, page 3–15. Springer International Publishing, 2023. ISBN 9783031319754. doi: 10.1007/978-3-031-31975-4_1. URL http://dx.doi.org/10.1007/978-3-031-31975-4_1.

Appendix Contents

A Related Work **16**

B Derivations for the Dirichlet Concept Diffusion Model **16**

B.1 Closed-form marginal of the forward process 16

B.2 Derivation of the reverse posterior 17

B.3 ELBO derivation 18

B.4 Gradients for individual concept centers 19

B.5 Terminal KL term 20

B.6 Connection to the simplified noise-prediction objective 20

B.7 Closed form of the Dirichlet regularizer 21

C Proofs for Theoretical Results **21**

D Model Architecture **22**

E Experimental Details **23**

E.1 Conceptual ImageNet 23

E.2 Training and Model Details 23

F Additional Experiments **24**

F.1 Empirical Risk Gap from Concept Guidance 24

F.2 Global Concept Visualizations 24

F.3 Generation Quality Metrics with 100-Step DDIM 34

F.4 Extended CAC Diagnostics 34

F.5 Extended Neighborhood and Probe Diagnostics 34

F.6 Extended SGA Diagnostics 35

F.7 Concept Capacity Ablation 35

F.8 Optional Concept Distillation 36

A Related Work

Concept Learning. Pioneering work on concept learning defines concepts as Boolean rules [3, 34], often based on associations with attributes or other concepts. Kruschke [27] introduce ALCOVE, a connectionist model that uses attention over stored exemplars to perform categorization, while Anderson [2] model concept learning as Bayesian inference over a mixture of Gaussians, where each component represents a concept defined over observed features.

More recent approaches aim to learn continuous concept representations that support generalization and multimodal reasoning [11, 28, 31–33, 48, 49]. Such representations may also support selective data sharing by identifying observations that complement a learner's existing experience [12]. In parallel, Concept Bottleneck Models (CBMs) incorporate concepts as intermediate representations within neural networks [24], with subsequent work extending CBMs to weaker supervision and pretrained settings [35, 56]. More recent formulations such as Energy-Based Concept Bottleneck Models further enable probabilistic reasoning and intervention over concepts through a unified energy-based framework [53].

While these approaches demonstrate the benefits of structuring models around interpretable concepts, they typically assume concepts are provided, weakly supervised, or derived from pretrained representations. Such assumptions raise a fundamental question of how concepts can be obtained directly from data in a scalable and unsupervised manner, which is the focus of our work.

Concept Discovery in Generative Models. Recent work extracts or composes concepts within generative models through additive energy-based decomposition [8], product-of-experts compositional modeling [7], score aggregation across separate denoisers [47], and discovery via pretrained text-to-image models [5, 29]. A complementary line learns structured latents directly from diffusion models, including semantic encoders that condition the reverse process [39], information-theoretic extensions with flexible priors [51], and post-hoc score decomposition into per-factor sub-fields [54].

Mixture-Based Generative Models. Mixture-based latent variable models compose multiple components to capture multimodal data, from Gaussian mixture VAEs [1, 6, 20, 41] to disentanglement-oriented variants [17, 22]. In diffusion models, related work modifies reverse transitions for component recovery [9, 14, 45], introduces structured Gaussian priors [57], analyzes implicit clustering [50], jointly learns a mixture prior [4], or shifts the forward kernel toward a structured non-zero target such as a per-sample VAE reconstruction [36], a data-dependent prior from observed conditioning [13], or the output of a pre-trained mean estimator for supervised conditional density modeling [15].

Our formulation makes concept structure intrinsic to the generative dynamics. Like the kernel-shifting models above, we center the forward kernel on a structured non-zero target. Unlike those models, the target is a Dirichlet-weighted aggregation of *latent* concept centers learned jointly with the diffusion model, rather than a target derived from observed conditioning or a pretrained estimator. The per-sample mixture is inferred via amortized variational inference, and the entire model is trained end-to-end under a single denoising ELBO. Each concept is realized as a learned anchor in data space, with weights on the Dirichlet simplex. Concept discovery emerges from learning to generate, rather than from an auxiliary objective, post-hoc decomposition, reverse-process conditioning, or external supervision. Methods that discover concepts on top of pretrained text-to-image diffusion models [5, 29, 54] inherit the text-image supervision used to train those models and operate in an easier problem setting.

B Derivations for the Dirichlet Concept Diffusion Model

Throughout, we fix a data point $\mathbf{x}_0$ and a concept distribution $\boldsymbol{\pi}$. To simplify notation, we write $\mathbf{m} := \mathbf{c}(\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathbf{c}_k$. Recall that $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, with $\bar{\alpha}_0 = 1$.

B.1 Closed-form marginal of the forward process

The forward transition in DCDM is

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\alpha_t}\mathbf{x}_{t-1} + \eta_t \mathbf{m}, \beta_t \mathbf{I}\right).$$

Equivalently, we may write $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \eta_t\mathbf{m} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For general coefficients $\eta_1, \ldots, \eta_T$, unrolling the recursion gives $\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{\pi}] = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + c_t\mathbf{m}$, where

$$c_t = \sum_{s=1}^{t} \eta_s \prod_{r=s+1}^{t} \sqrt{\alpha_r} = \sum_{s=1}^{t} \eta_s \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_s}}$$

Thus, for arbitrary η_t, the coefficient of $\mathbf{m}$ is an accumulated weighted sum. We choose $\eta_t = 1 - \sqrt{\alpha_t}$. Under this choice, the expression for c_t becomes a telescoping sum:

$$\begin{aligned} c_t &= \sum_{s=1}^{t} (1 - \sqrt{\alpha_s}) \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_s}} \\ &= \sum_{s=1}^{t} \left(\sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_s}} - \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{s-1}}} \right) \\ &= 1 - \sqrt{\bar{\alpha}_t}. \end{aligned}$$

Therefore $\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{\pi}] = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1 - \sqrt{\bar{\alpha}_t})\,\mathbf{m}$.

It remains to compute the variance. Because the additional term $\eta_t\mathbf{m}$ is deterministic given $\boldsymbol{\pi}$, it affects only the mean, so the conditional variance follows the standard DDPM recursion. Writing $\mathrm{Var}(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{\pi}) = v_t\mathbf{I}$, we have $v_t = \alpha_t v_{t-1} + \beta_t$ with $v_0 = 0$, and therefore $v_t = 1 - \bar{\alpha}_t$. Hence,

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \left(1 - \sqrt{\bar{\alpha}_t}\right)\mathbf{m}, (1 - \bar{\alpha}_t)\,\mathbf{I}\right). \tag{10}$$

B.2 Derivation of the reverse posterior

We now derive the closed-form posterior $q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi})$. From Eq. (10),

$$q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_{t-1}^{\mathrm{fwd}}, (1 - \bar{\alpha}_{t-1})\mathbf{I}\right),$$

where $\boldsymbol{\mu}_{t-1}^{\mathrm{fwd}} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + (1 - \sqrt{\bar{\alpha}_{t-1}})\,\mathbf{m}$.

The one-step forward transition is

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\alpha_t}\mathbf{x}_{t-1} + (1 - \sqrt{\alpha_t})\,\mathbf{m}, \beta_t\mathbf{I}\right).$$

By Bayes' rule, $q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \propto q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}) q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \boldsymbol{\pi})$. Taking logs and keeping only terms involving $\mathbf{x}_{t-1}$ gives

$$\begin{aligned} \log q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) = &-\frac{1}{2\beta_t} \left\| \mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1} - (1 - \sqrt{\alpha_t})\,\mathbf{m} \right\|_2^2 \\ &- \frac{1}{2(1 - \bar{\alpha}_{t-1})} \left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}^{\mathrm{fwd}} \right\|_2^2 + \mathrm{const.} \end{aligned}$$

which is quadratic in $\mathbf{x}_{t-1}$, so the posterior is Gaussian. Its precision is

$$\begin{aligned} \tilde{\beta}_t^{-1} &= \frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \\ &= \frac{\alpha_t(1 - \bar{\alpha}_{t-1}) + \beta_t}{\beta_t(1 - \bar{\alpha}_{t-1})} \\ &= \frac{1 - \alpha_t\bar{\alpha}_{t-1}}{\beta_t(1 - \bar{\alpha}_{t-1})} = \frac{1 - \bar{\alpha}_t}{\beta_t(1 - \bar{\alpha}_{t-1})}. \end{aligned}$$

Therefore $\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t$.

The posterior mean is obtained by multiplying the posterior covariance by the natural parameter:

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) = \tilde{\beta}_t \left[\frac{\sqrt{\alpha_t}}{\beta_t} \left(\mathbf{x}_t - (1 - \sqrt{\alpha_t})\,\mathbf{m}\right) + \frac{1}{1 - \bar{\alpha}_{t-1}} \boldsymbol{\mu}_{t-1}^{\mathrm{fwd}} \right].$$

Substituting the expression for $\boldsymbol{\mu}_{t-1}^{\text{fwd}}$ and grouping terms yields

$$\widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) = A_t \mathbf{x}_t + B_t \mathbf{x}_0 + C_t \mathbf{m},$$

where $A_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$ and $B_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}$. The coefficient of $\mathbf{m}$ is

$$C_t = \tilde{\beta}_t \left[-\frac{\sqrt{\alpha_t}(1-\sqrt{\alpha_t})}{\beta_t} + \frac{1-\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} \right].$$

Let $r := \sqrt{\alpha_t}$ and $s := \sqrt{\bar{\alpha}_{t-1}}$. Then $\beta_t = 1 - r^2$, $\bar{\alpha}_t = r^2 s^2$, and

$$\begin{aligned}
C_t &= \frac{(1-r^2)(1-s^2)}{1-r^2 s^2} \left[-\frac{r(1-r)}{1-r^2} + \frac{1-s}{1-s^2} \right] \\
&= \frac{(1-r^2)(1-s^2)}{1-r^2 s^2} \left[-\frac{r}{1+r} + \frac{1}{1+s} \right] \\
&= \frac{(1-r^2)(1-s^2)}{1-r^2 s^2} \left[\frac{1-rs}{(1+r)(1+s)} \right] \\
&= \frac{(1-r)(1-s)}{1+rs}.
\end{aligned}$$

Returning to the original notation, $C_t = \frac{(1-\sqrt{\alpha_t})(1-\sqrt{\bar{\alpha}_{t-1}})}{1+\sqrt{\bar{\alpha}_t}}$.

The identity $C_t = 1 - A_t - B_t$ can also be seen by shifting coordinates around the concept center. Let $\mathbf{y}_t = \mathbf{x}_t - \mathbf{m}$. In these coordinates, the forward process reduces to the standard DDPM form, so the posterior mean is $A_t \mathbf{y}_t + B_t \mathbf{y}_0$. Transforming back to $\mathbf{x}$-coordinates gives

$$\widetilde{\boldsymbol{\mu}}_t = \mathbf{m} + A_t(\mathbf{x}_t - \mathbf{m}) + B_t(\mathbf{x}_0 - \mathbf{m}) = A_t \mathbf{x}_t + B_t \mathbf{x}_0 + (1 - A_t - B_t)\mathbf{m}.$$

Hence $C_t = 1 - A_t - B_t$.

Combining the results, for $t \geq 2$,

$$q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_{t-1} \mid A_t \mathbf{x}_t + B_t \mathbf{x}_0 + C_t \mathbf{m}, \tilde{\beta}_t \mathbf{I}\right). \tag{11}$$

B.3 ELBO derivation

We factorize the variational distribution as

$$\begin{aligned}
q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0) &= q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) q_\phi(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \boldsymbol{\pi}), \\
q_\phi(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \boldsymbol{\pi}) &= \prod_{t=1}^{T} q_\phi(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\pi}).
\end{aligned}$$

The generative model can be factorized as

$$p_{\theta,\phi}(\mathbf{x}_{0:T}, \boldsymbol{\pi}) = p(\boldsymbol{\pi}) p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi}).$$

Starting from the marginal likelihood, Jensen's inequality gives

$$\begin{aligned}
\log p_{\theta,\phi}(\mathbf{x}_0) &= \log \mathbb{E}_{q_\phi} \left[\frac{p_{\theta,\phi}(\mathbf{x}_{0:T}, \boldsymbol{\pi})}{q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right] \\
&\geq \mathbb{E}_{q_\phi} \left[\log \frac{p_{\theta,\phi}(\mathbf{x}_{0:T}, \boldsymbol{\pi})}{q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right].
\end{aligned}$$

Define the negative ELBO by

$$\mathcal{L}(\mathbf{x}_0) := -\mathbb{E}_{q_\phi} \left[\log \frac{p_{\theta,\phi}(\mathbf{x}_{0:T}, \boldsymbol{\pi})}{q_\phi(\boldsymbol{\pi}, \mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right].$$

Using the factorized forms above,

$$\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)}{p(\boldsymbol{\pi})}\right] + \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \boldsymbol{\pi})}{p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi})}\right].$$

The first term is simply $L_{\text{reg}} = D_{\mathbb{KL}}\left(q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \,\|\, p(\boldsymbol{\pi})\right)$.

For the second term, we use the reverse-time factorization of the forward process conditioned on $\mathbf{x}_0$:

$$q_\phi(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \boldsymbol{\pi}) = q_\phi(\mathbf{x}_T \mid \mathbf{x}_0, \boldsymbol{\pi}) \prod_{t=2}^{T} q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}).$$

Substituting this expression gives

$$\begin{aligned}\mathcal{L}(\mathbf{x}_0) = {} & D_{\mathbb{KL}}\left(q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \,\|\, p(\boldsymbol{\pi})\right) \\ & + \mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)}\Bigg[D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_T \mid \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi})\right) \\ & \quad + \sum_{t=2}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\pi})} D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi})\right) \\ & \quad - \mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0,\boldsymbol{\pi})} \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \boldsymbol{\pi})\Bigg].\end{aligned}$$

Thus, defining

$$\begin{aligned} L_T &:= D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_T \mid \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi})\right), \\ L_{t-1} &:= \mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\pi})} D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi})\right), \\ L_0 &:= -\mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0,\boldsymbol{\pi})} \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \boldsymbol{\pi}), \end{aligned}$$

the negative ELBO can be written compactly as

$$\mathcal{L}(\mathbf{x}_0) = \mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)}\left[L_T + \sum_{t=2}^{T} L_{t-1} + L_0\right] + L_{\text{reg}}. \tag{12}$$

B.4 Gradients for individual concept centers

We make explicit how the simplified denoising objective in Eq. (8) updates each learnable center. For one training example, sample $\boldsymbol{\pi} \sim q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$, $t \sim \text{Unif}\{1, \ldots, T\}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and write

$$\begin{aligned} \mathbf{m} &= \sum_{j=1}^{K} \pi_j \mathbf{c}_j, \\ \mathbf{x}_t &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \left(1 - \sqrt{\bar{\alpha}_t}\right)\mathbf{m} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \\ \ell &= \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{m}, t)\right\|_2^2. \end{aligned}$$

The concept encoder predicts the Dirichlet parameters from $\mathbf{x}_0$ independently of the center bank, so $\boldsymbol{\pi}$ is fixed when taking the partial derivative with respect to $\mathbf{c}_k$. Consequently,

$$\frac{\partial \mathbf{m}}{\partial \mathbf{c}_k} = \pi_k \mathbf{I}.$$

Treating $\mathbf{x}_t$ and $\mathbf{m}$ as separate inputs to the denoiser, the chain rule gives

$$\nabla_{\mathbf{c}_k} \ell = \pi_k \left[\underbrace{\left(1 - \sqrt{\bar{\alpha}_t}\right) \nabla_{\mathbf{x}_t} \ell}_{\text{through the forward mean}} + \underbrace{\nabla_{\mathbf{m}} \ell|_{\mathbf{x}_t \text{ fixed}}}_{\text{through denoiser conditioning}}\right]. \tag{13}$$

The first term propagates gradients through the concept-dependent forward mean, while the second captures the denoiser's explicit dependence on the aggregated center with $\mathbf{x}_t$ held fixed. Both contributions are scaled by the inferred weight π_k. Averaging the per-example gradients over a minibatch gives a Monte Carlo estimate of the gradient of the batch-averaged L_{diff}. Each shared center therefore accumulates denoising signals from multiple samples in proportion to their inferred weights. The Dirichlet regularizer L_{reg} does not directly depend on the centers under the stated encoder parameterization and contributes no additional center-gradient term.

B.5 Terminal KL term

The terminal distribution under the forward process is

$$q_\phi(\mathbf{x}_T \mid \mathbf{x}_0, \boldsymbol{\pi}) = \mathcal{N}\left(\sqrt{\bar{\alpha}_T}\mathbf{x}_0 + \left(1 - \sqrt{\bar{\alpha}_T}\right)\mathbf{c}(\boldsymbol{\pi}), (1-\bar{\alpha}_T)\,\mathbf{I}\right),$$

whereas the generative terminal distribution is

$$p_\phi(\mathbf{x}_T \mid \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{c}(\boldsymbol{\pi}), \mathbf{I}\right).$$

Therefore,

$$\begin{aligned} L_T &= \frac{1}{2}\left[d(1-\bar{\alpha}_T) + \left\|\sqrt{\bar{\alpha}_T}\mathbf{x}_0 + \left(1-\sqrt{\bar{\alpha}_T}\right)\mathbf{c}(\boldsymbol{\pi}) - \mathbf{c}(\boldsymbol{\pi})\right\|_2^2 - d - d\log(1-\bar{\alpha}_T)\right] \\ &= \frac{1}{2}\left[d(1-\bar{\alpha}_T) - d - d\log(1-\bar{\alpha}_T)\right] + \frac{1}{2}\bar{\alpha}_T\left\|\mathbf{x}_0 - \mathbf{c}(\boldsymbol{\pi})\right\|_2^2. \end{aligned}$$

The first bracket depends only on the noise schedule and dimension. The only parameter-dependent part is proportional to

$$\bar{\alpha}_T\left\|\mathbf{x}_0 - \mathbf{c}(\boldsymbol{\pi})\right\|_2^2.$$

Since typical diffusion schedules choose T so that $\bar{\alpha}_T$ is close to zero, this term is negligible and is omitted in the training objective.

B.6 Connection to the simplified noise-prediction objective

For $t \geq 2$, suppose the reverse model uses the same variance as the true reverse posterior, i.e.,

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi}) = \mathcal{N}\left(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t), \tilde{\beta}_t\mathbf{I}\right).$$

Then

$$D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi})\right) = \frac{1}{2\tilde{\beta}_t}\left\|\widetilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)\right\|_2^2 + \text{const.}$$

Using the forward marginal, we can sample

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \left(1-\sqrt{\bar{\alpha}_t}\right)\mathbf{c}(\boldsymbol{\pi}) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Given a noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)$, define the induced prediction of $\mathbf{x}_0$ as

$$\widehat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \left(1-\sqrt{\bar{\alpha}_t}\right)\mathbf{c}(\boldsymbol{\pi}) - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)}{\sqrt{\bar{\alpha}_t}}.$$

We parameterize the reverse mean by substituting $\widehat{\mathbf{x}}_0$ into the exact posterior mean, namely

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t) = A_t\mathbf{x}_t + B_t\widehat{\mathbf{x}}_0 + C_t\mathbf{c}(\boldsymbol{\pi}).$$

Since the exact posterior mean is

$$\widetilde{\boldsymbol{\mu}}_t = A_t\mathbf{x}_t + B_t\mathbf{x}_0 + C_t\mathbf{c}(\boldsymbol{\pi}),$$

we have

$$\begin{aligned} \widetilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta &= B_t\left(\mathbf{x}_0 - \widehat{\mathbf{x}}_0\right) \\ &= \frac{B_t\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t) - \boldsymbol{\epsilon}\right). \end{aligned}$$

Therefore, the exact denoising KL is equal, up to a constant, to

$$D_{\mathbb{KL}}\left(q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\pi}) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \boldsymbol{\pi})\right) = w_t \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)\right\|_2^2 + \text{const},$$

where

$$w_t = \frac{B_t^2(1-\bar{\alpha}_t)}{2\tilde{\beta}_t\bar{\alpha}_t} = \frac{\beta_t}{2\alpha_t(1-\bar{\alpha}_{t-1})}.$$

Following the standard DDPM simplification, we drop the time-dependent weighting and optimize the unweighted surrogate

$$L_{\text{simple}} = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi}), t)\right\|_2^2\right], \tag{14}$$

where $t \sim \text{Unif}\{1, \ldots, T\}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \left(1 - \sqrt{\bar{\alpha}_t}\right)\mathbf{c}(\boldsymbol{\pi}) + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}.$$

The final training objective is therefore

$$\mathcal{L}(\theta, \phi; \mathbf{x}_0) = \mathbb{E}_{\boldsymbol{\pi}\sim q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)}\left[L_{\text{simple}}\right] + D_{\mathbb{KL}}\left(q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) \,\|\, p(\boldsymbol{\pi})\right). \tag{15}$$

B.7 Closed form of the Dirichlet regularizer

If $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) = \text{Dir}(\boldsymbol{\alpha})$ and $p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\gamma})$, then

$$\begin{aligned} D_{\mathbb{KL}}\left(\text{Dir}(\boldsymbol{\alpha}) \,\|\, \text{Dir}(\boldsymbol{\gamma})\right) = \log\Gamma(\alpha_0) - \sum_{k=1}^K \log\Gamma(\alpha_k) - \log\Gamma(\gamma_0) + \sum_{k=1}^K \log\Gamma(\gamma_k) \\ + \sum_{k=1}^K (\alpha_k - \gamma_k)\left[\psi(\alpha_k) - \psi(\alpha_0)\right], \end{aligned}$$

where $\alpha_0 = \sum_{k=1}^K \alpha_k$, $\gamma_0 = \sum_{k=1}^K \gamma_k$, and $\psi(\cdot)$ is the digamma function.

C Proofs for Theoretical Results

Proof of Theorem 3.1. Using the notation $\mathbf{m} = \mathbf{c}(\boldsymbol{\pi})$ introduced above, write $\mathbf{x}_0 = \mathbf{m} + \boldsymbol{\delta}$. Then the proposed forward process satisfies

$$\mathbf{x}_t = \mathbf{m} + \sqrt{\bar{\alpha}_t}\boldsymbol{\delta} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}.$$

Equivalently,

$$\mathbf{x}_t - \mathbf{m} = \sqrt{\bar{\alpha}_t}\boldsymbol{\delta} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}.$$

Conditioned on $\mathbf{m}$, the pair $(\boldsymbol{\delta}, \mathbf{x}_t - \mathbf{m})$ is jointly Gaussian with mean zero. Its covariance blocks are

$$\begin{aligned} \text{Cov}(\boldsymbol{\delta} \mid \mathbf{m}) &= \boldsymbol{\Sigma}_\delta, \\ \text{Cov}(\mathbf{x}_t - \mathbf{m} \mid \mathbf{m}) &= \bar{\alpha}_t\boldsymbol{\Sigma}_\delta + (1-\bar{\alpha}_t)\mathbf{I}, \end{aligned}$$

and

$$\text{Cov}(\boldsymbol{\delta}, \mathbf{x}_t - \mathbf{m} \mid \mathbf{m}) = \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_\delta.$$

Therefore, by the Gaussian conditioning formula,

$$\mathbb{E}[\boldsymbol{\delta} \mid \mathbf{x}_t, \mathbf{m}] = \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_\delta(\bar{\alpha}_t\boldsymbol{\Sigma}_\delta + (1-\bar{\alpha}_t)\mathbf{I})^{-1}(\mathbf{x}_t - \mathbf{m}).$$

Define

$$\mathbf{G}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_\delta(\bar{\alpha}_t\boldsymbol{\Sigma}_\delta + (1-\bar{\alpha}_t)\mathbf{I})^{-1}.$$

Then

$$\mathbb{E}[\boldsymbol{\delta} \mid \mathbf{x}_t, \mathbf{m}] = \mathbf{G}_t(\mathbf{x}_t - \mathbf{m}),$$

and hence

$$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{m}] = \mathbf{m} + \mathbb{E}[\boldsymbol{\delta} \mid \mathbf{x}_t, \mathbf{m}] = \mathbf{m} + \mathbf{G}_t(\mathbf{x}_t - \mathbf{m}) = \mathbf{G}_t\mathbf{x}_t + (\mathbf{I} - \mathbf{G}_t)\mathbf{m}.$$

Taking conditional expectation with respect to $\mathbf{m} \mid \mathbf{x}_t$ yields

$$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] = \mathbf{G}_t \mathbf{x}_t + (\mathbf{I} - \mathbf{G}_t)\mathbb{E}[\mathbf{m} \mid \mathbf{x}_t].$$

Subtracting the two predictors gives

$$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{m}] - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] = (\mathbf{I} - \mathbf{G}_t)(\mathbf{m} - \mathbb{E}[\mathbf{m} \mid \mathbf{x}_t]).$$

Now let

$$R_{\text{guided}}(t) = \mathbb{E}[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{m}]\|^2], \qquad R_{\text{std}}(t) = \mathbb{E}[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]\|^2].$$

Since conditional expectation is the Bayes-optimal predictor under square loss, the standard orthogonality identity implies

$$R_{\text{std}}(t) - R_{\text{guided}}(t) = \mathbb{E}[\|\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{m}] - \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]\|^2].$$

Substituting the expression above,

$$R_{\text{std}}(t) - R_{\text{guided}}(t) = \mathbb{E}[\|(\mathbf{I} - \mathbf{G}_t)(\mathbf{m} - \mathbb{E}[\mathbf{m} \mid \mathbf{x}_t])\|^2].$$

Condition on $\mathbf{x}_t$. By the definition of conditional covariance,

$$\mathbf{\Sigma}_c(\mathbf{x}_t) = \mathbb{E}[(\mathbf{m} - \mathbb{E}[\mathbf{m} \mid \mathbf{x}_t])(\mathbf{m} - \mathbb{E}[\mathbf{m} \mid \mathbf{x}_t])^\top \mid \mathbf{x}_t].$$

Therefore

$$\mathbb{E}[\|(\mathbf{I} - \mathbf{G}_t)(\mathbf{m} - \mathbb{E}[\mathbf{m} \mid \mathbf{x}_t])\|^2 \mid \mathbf{x}_t] = \operatorname{tr}((\mathbf{I} - \mathbf{G}_t)\mathbf{\Sigma}_c(\mathbf{x}_t)(\mathbf{I} - \mathbf{G}_t)^\top).$$

Taking expectation over $\mathbf{x}_t$ gives

$$R_{\text{std}}(t) - R_{\text{guided}}(t) = \mathbb{E}[\operatorname{tr}((\mathbf{I} - \mathbf{G}_t)\mathbf{\Sigma}_c(\mathbf{x}_t)(\mathbf{I} - \mathbf{G}_t)^\top)].$$

The claimed formula follows. □

D Model Architecture

Our DCDM consists of two trainable parts, the variational encoder q_ϕ and the decoder p_θ. The variational encoder contains a concept encoder f^ϕ_{concept}, which parameterizes $q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0)$, together with the learnable concept centers $\{\mathbf{c}_k\}_{k=1}^K$. The decoder is implemented as a reverse denoising network and is written under the noise-prediction parameterization as $\boldsymbol{\epsilon}_\theta(\cdot)$. Given a clean input $\mathbf{x}_0$, the concept encoder infers a distribution over concepts. This distribution is used to aggregate the learnable concept centers into $\mathbf{c}(\boldsymbol{\pi})$. The denoising network then predicts the forward-process noise using the noisy latent $\mathbf{x}_t$, the aggregated concept center $\mathbf{c}(\boldsymbol{\pi})$, and the timestep t.

Variational encoder. The concept encoder f^ϕ_{concept} is implemented as a ResNet-based network [16]. Given a clean input $\mathbf{x}_0$, it outputs concept logits $\mathbf{h} = f^\phi_{\text{concept}}(\mathbf{x}_0) \in \mathbb{R}^K$. We apply a softplus transformation to obtain positive Dirichlet concentration parameters, $\boldsymbol{\alpha}(\mathbf{x}_0) = \operatorname{softplus}(\mathbf{h})$, and define

$$q_\phi(\boldsymbol{\pi} \mid \mathbf{x}_0) = \operatorname{Dir}(\boldsymbol{\alpha}(\mathbf{x}_0)).$$

In addition to the ResNet encoder, the variational parameter set ϕ includes the learnable spatial concept centers $\{\mathbf{c}_k\}_{k=1}^K$, where $\mathbf{c}_k \in \mathbb{R}^{C \times H \times W}$. These centers are registered as model parameters and optimized jointly by gradient descent. Given a sampled or inferred concept distribution $\boldsymbol{\pi} \in \Delta^{K-1}$, the aggregated concept center is computed as $\mathbf{c}(\boldsymbol{\pi}) = \sum_{k=1}^K \pi_k \mathbf{c}_k$. For batched concept distributions paired with noisy latents $\mathbf{x}_t \in \mathbb{R}^{B \times C \times H \times W}$, this setup gives $\mathbf{c}(\boldsymbol{\pi}) \in \mathbb{R}^{B \times C \times H \times W}$.

Reverse denoising network. The reverse model p_θ is implemented using a UNet backbone [43]. The timestep embedding, ResBlocks, attention blocks, encoder–decoder structure, skip connections, and noise-prediction target are unchanged. Relative to the base UNet, the only architectural additions are two lightweight concept-input paths that allow the denoising network to use the inferred concept distribution.

The first concept-input path is spatial. Before passing the noisy latent to the UNet, we concatenate the aggregated concept center with the noisy latent along the channel dimension:

$$\mathbf{u}_t = \mathrm{Concat}_{\mathrm{ch}}\left[\mathbf{x}_t, \mathbf{c}(\boldsymbol{\pi})\right] \in \mathbb{R}^{B\times 2C\times H\times W}.$$

Thus, when the latent has $C = 4$ channels, the UNet receives an 8-channel tensor as input and still predicts a 4-channel noise tensor. This channel concatenation implements the dependence of the denoising network on both $\mathbf{x}_t$ and $\mathbf{c}(\boldsymbol{\pi})$ in our formulation.

The second concept-input path is global. The FiLM branch maintains a learned vector embedding $\mathbf{e}_k$ for each concept k. From the same concept distribution $\boldsymbol{\pi}$, we compute a probability-weighted concept embedding $\mathbf{v}(\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \mathbf{e}_k$. The vector $\mathbf{v}(\boldsymbol{\pi})$ is passed to enabled ResBlocks through a FiLM-style modulation branch [38]. For a ResBlock ℓ, a small linear layer maps $\mathbf{v}(\boldsymbol{\pi})$ to per-channel scale and shift vectors,

$$(\mathbf{s}_\ell, \mathbf{r}_\ell) = W_\ell \mathbf{v}(\boldsymbol{\pi}) + \mathbf{a}_\ell.$$

These vectors are broadcast over spatial dimensions and applied to the normalized hidden activation $\mathbf{h}_\ell$ as

$$\mathbf{h}_\ell \leftarrow \mathbf{h}_\ell \odot (1 + \mathbf{s}_\ell) + \mathbf{r}_\ell.$$

This FiLM-style path gives the ResBlocks a lightweight global concept signal in parallel with the spatial concept-center input. We use this design rather than a transformer cross-attention interface to keep the additional GPU memory cost close to that of the base UNet.

E Experimental Details

Unless otherwise stated, quantitative metrics are reported as mean $\pm 2\sigma$ over three runs with different random seeds.

E.1 Conceptual ImageNet

Conceptual ImageNet is a semantically controlled subset of ImageNet constructed using WordNet hierarchy information. We select ImageNet leaf classes whose synsets fall under a small set of broad WordNet hypernym roots covering animals, natural scenes, and vehicles, as listed in Table 3. We cap the number of selected leaf classes per coarse group to avoid domination by dense WordNet branches such as dog breeds, while preserving ImageNet's near-balanced sampling at the leaf-class level. The resulting subset contains 103 ImageNet leaf classes, 132,214 training images, and 5,150 validation images.

Table 3: Conceptual ImageNet coarse groups and WordNet roots.

Coarse group	WordNet root synsets	Leaf classes
Bird	`bird.n.01`	20
Geological place	`geological_formation.n.01`	10
Marine animals	`fish.n.01`, `aquatic_mammal.n.01`, `crab.n.01`	20
Wheeled vehicle	`motor_vehicle.n.01`	18
Dog	`dog.n.01`	20
Nonland vehicle	`aircraft.n.01`, `vessel.n.02`	15
Total	–	103

E.2 Training and Model Details

We summarize training hyperparameters for the LDCD models in Table 4. The same trained models are used throughout the experiments. All diffusion models are trained with AdamW. For latent experiments, the autoencoder is trained following the latent-diffusion setup of Rombach et al. [42]. Concept centers use orthogonal initialization. Each matched generative baseline uses the comparable U-Net backbone and training schedule, with the concept encoder, concept centers, spatial concept concatenation, and FiLM concept branch removed. Each run used a single NVIDIA A10 GPU. For computational efficiency, CelebA, Conceptual ImageNet, LSUN Bedroom, and LSUN Church

Table 4: Training hyperparameters for LDCD.

Dataset	LR	Train steps	Batch size	Diffusion schedule
CIFAR-10	5×10^{-5}	500k	128	cosine
CelebA	6×10^{-5}	400k	128	cosine
C-ImageNet	6×10^{-5}	1.5M	128	cosine
LSUN Bedroom	3×10^{-5}	1M	128	cosine
LSUN Church	6×10^{-5}	800k	128	cosine

images are resized to 64×64 before latent encoding, while CIFAR-10 is kept at its native 32×32 resolution.

Table 5 reports the latent resolution and U-Net architecture used by these models.

Table 5: Model hyperparameters for DCDM.

Dataset	Latent size	Model channels	Channel mult.	Res. blocks	Attn. res.	Head channels
CIFAR-10	$4\times8\times8$	256	$(1, 2, 4)$	2	$(8, 4, 2)$	32
CelebA	$4\times16\times16$	256	$(1, 2, 4, 4)$	2	$(8, 4, 2)$	32
C-ImageNet	$4\times16\times16$	256	$(1, 2, 4, 4)$	2	$(8, 4, 2)$	32
LSUN Bedroom	$4\times16\times16$	256	$(1, 2, 4, 4)$	2	$(8, 4, 2)$	32
LSUN Church	$4\times16\times16$	256	$(1, 2, 4, 4)$	2	$(8, 4, 2)$	32

Table 6 summarizes the concept-space configuration used per dataset.

Table 6: Concept-space configuration per dataset.

Dataset	Total concepts	Groups	Concepts per group
CIFAR-10	16	1	16
CelebA	24	4	6
C-ImageNet	24	1	24
LSUN Bedroom	24	1	24
LSUN Church	24	1	24

F Additional Experiments

F.1 Empirical Risk Gap from Concept Guidance

We provide direct empirical evidence for Theorem 3.1 by measuring the per-timestep risk gap $R_{\mathrm{std}}(t) - R_{\mathrm{guided}}(t)$ on real data. For each held-out LSUN Church sample $\mathbf{x}_0$ and each timestep t, we form a noisy latent under each model's forward process with shared $\boldsymbol{\epsilon}$, run the corresponding noise predictor, recover $\widehat{\mathbf{x}}_0$ from the noise prediction, and compute the per-pixel $\mathbf{x}_0$-MSE. Averaging over the full LSUN Church test set with 4 noise seeds per timestep yields the curves used to estimate $R_{\mathrm{std}}(t)$ (baseline) and $R_{\mathrm{guided}}(t)$ (DCDM).

The measured gap concentrates at high noise and decays toward zero as $t \to 0$, matching the qualitative $(\mathbf{I} - \mathbf{G}_t)\boldsymbol{\Sigma}_c(\mathbf{x}_t)(\mathbf{I} - \mathbf{G}_t)^\top$ shape implied by the theorem. At high noise $\mathbf{G}_t \approx \mathbf{0}$ and the concept-conditional posterior of $\mathbf{c}(\boldsymbol{\pi})$ is broad, while at low noise $\mathbf{G}_t \to \mathbf{I}$ and the residual $\boldsymbol{\delta}$ is largely determined by $\mathbf{x}_t$.

F.2 Global Concept Visualizations

The following visualizations provide a broader view of what the learned components capture across datasets. Each row corresponds to one learned component k and shows uncurated samples generated from the posteriors $\widehat{\boldsymbol{\pi}}_i$ of the top-10 in-distribution samples ranked by $\widehat{\pi}_{i,k}$. The figures include every learned component rather than only selected examples, which gives additional insight into the structure learned by the model, including recurring visual modes, component diversity, and failure cases. Presenting all components keeps the evidence behind our interpretations available for inspection, consistent with broader work on auditable AI-assisted research [10].

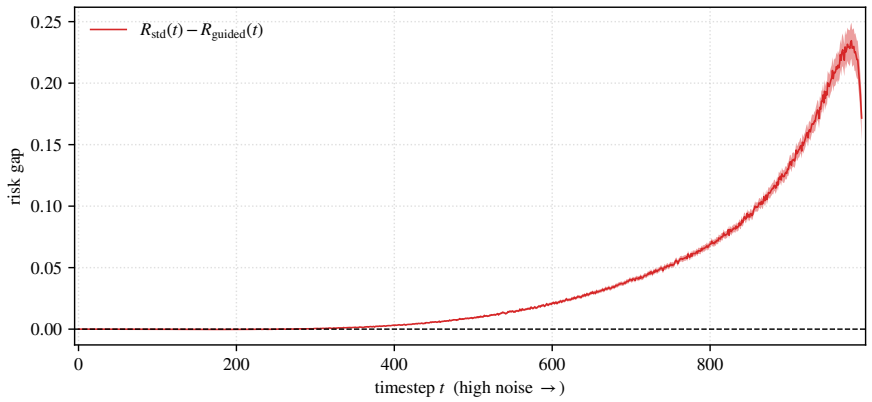

Figure 6: Empirical risk gap $R_{\text{std}}(t) - R_{\text{guided}}(t)$ on LSUN Church, computed from $\mathbf{x}_0$-prediction MSE. Shaded band is 95% CI. Consistent with Theorem 3.1, the gap concentrates at high noise and decays toward zero as $t \to 0$.



Figure 7: Global concept visualization for CelebA. The model uses 24 components across 4 groups. Rows cover all learned components, with uncurated prototype samples shown for each component.

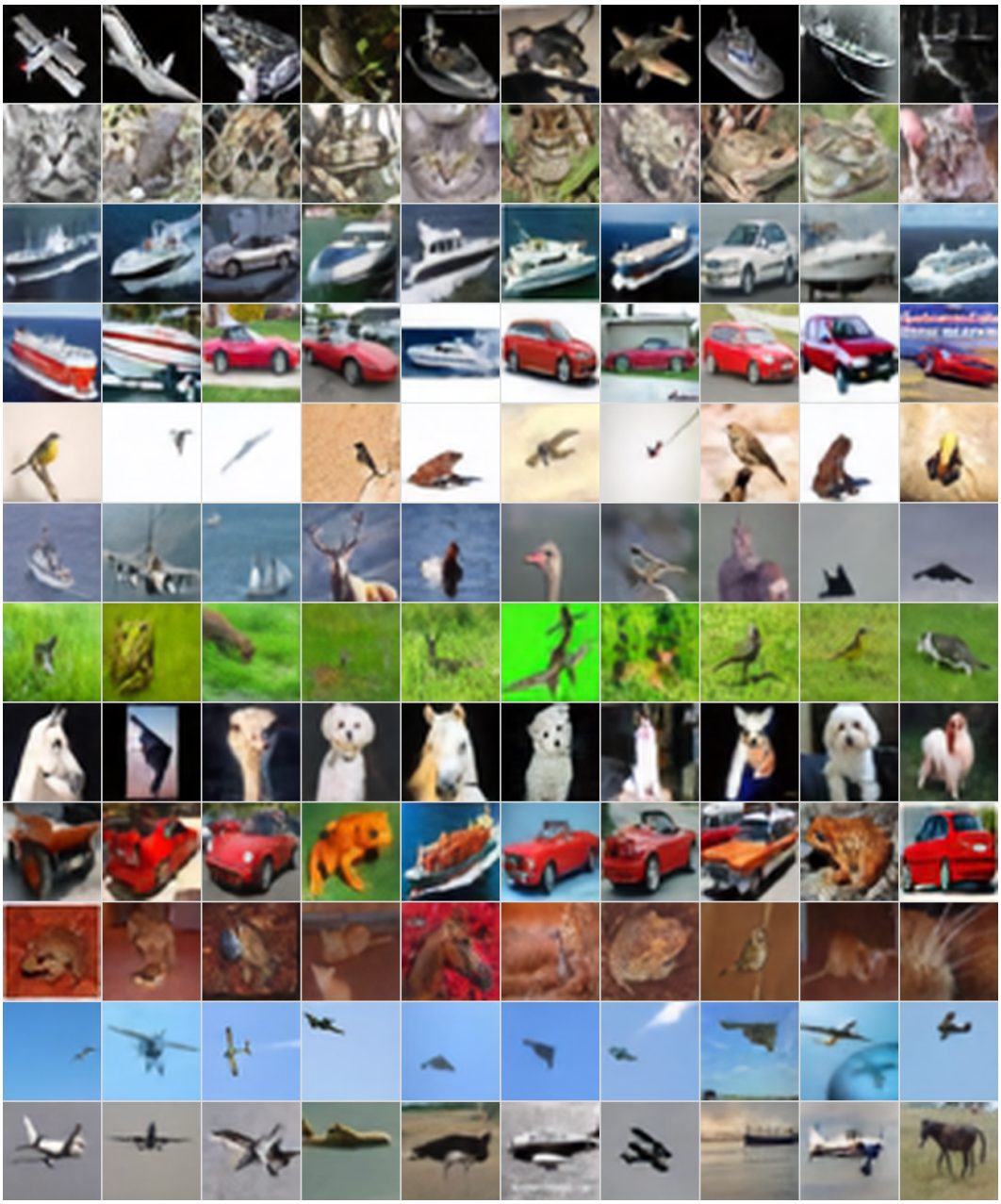

Figure 8: Global concept visualization for CIFAR-10. The model uses 12 components in one group. Rows cover all learned components, with uncurated prototype samples shown for each component.



Figure 9: Global concept visualization for CIFAR-10. The model uses 16 components in one group. Rows cover all learned components, with uncurated prototype samples shown for each component.



Figure 10: Global concept visualization for CIFAR-10 with concept distillation. The model uses 16 components in one group. Rows cover all learned components, with uncurated prototype samples shown for each component.

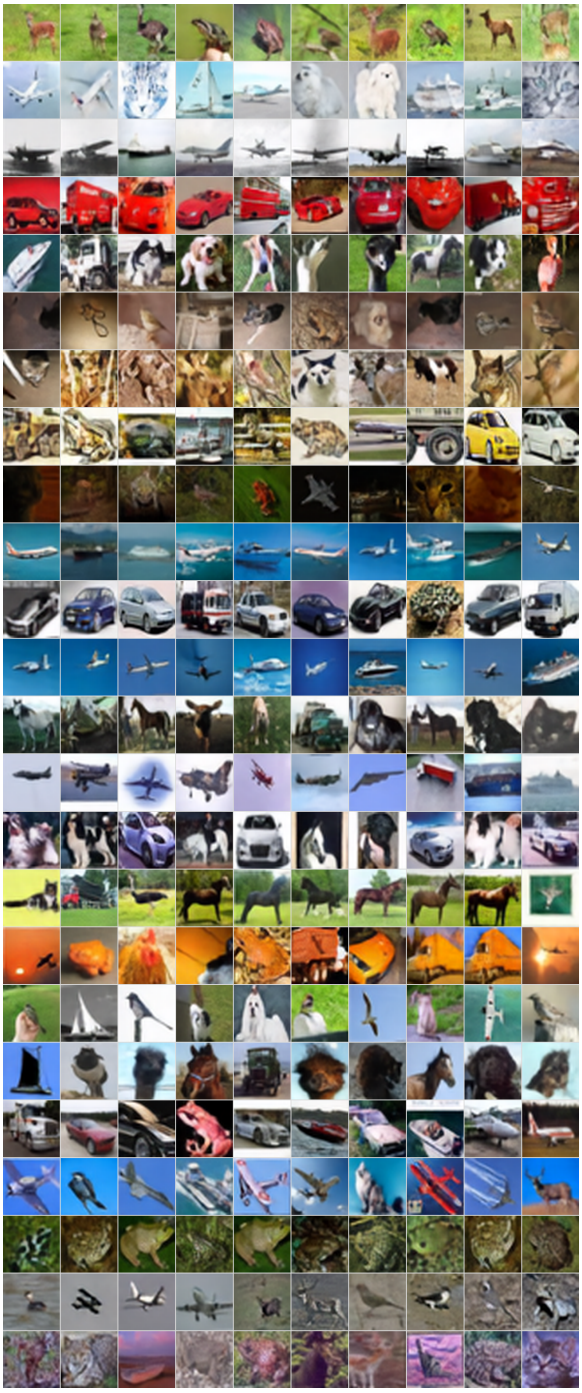

Figure 11: Global concept visualization for CIFAR-10. The model uses 24 components in one group. Rows cover all learned components, with uncurated prototype samples shown for each component.

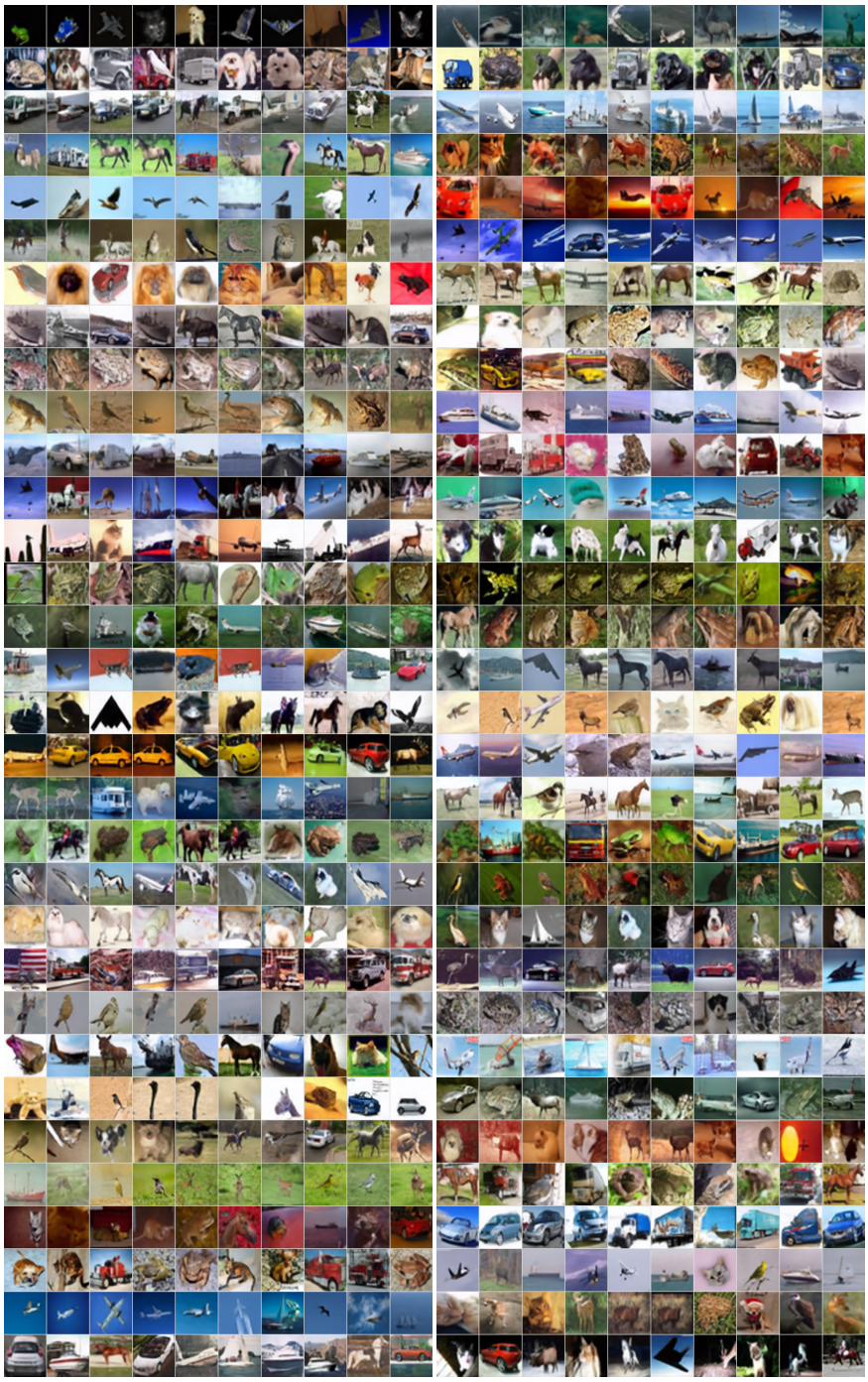

Figure 12: Global concept visualization for CIFAR-10. The model uses 64 components across 8 groups. Rows cover all learned components, with uncurated prototype samples shown for each component.

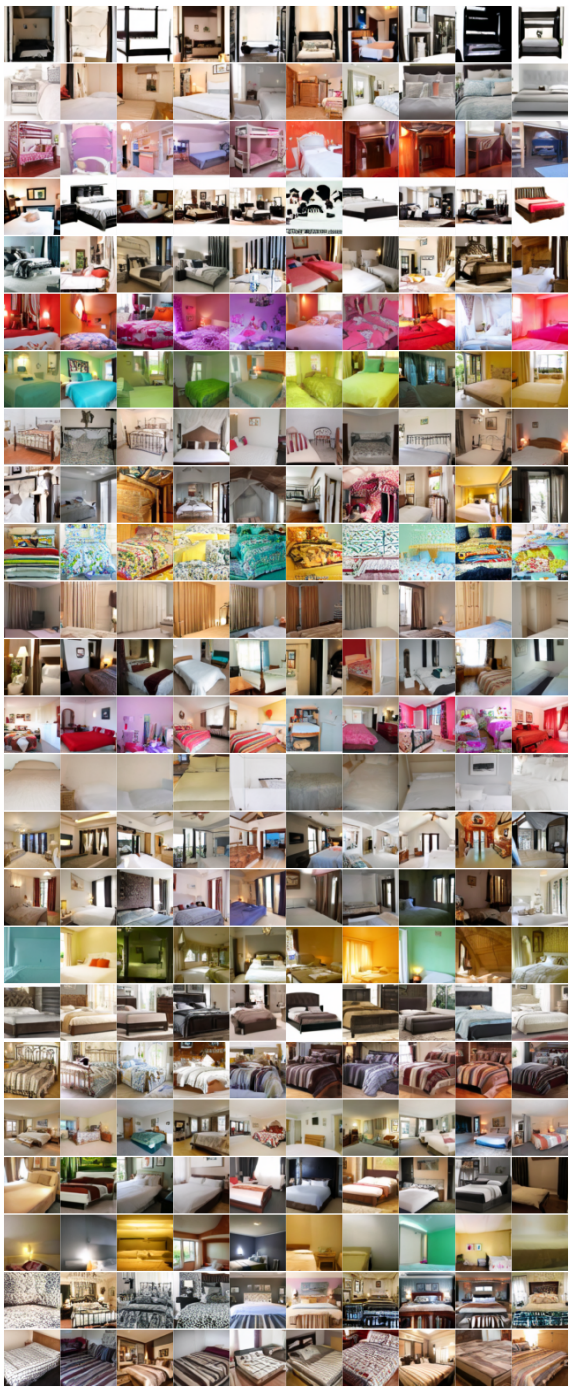

Figure 13: Global concept visualization for LSUN Bedroom. The visualization is shown without text annotations. Rows cover all learned components, with uncurated prototype samples shown for each component.



Figure 14: Global concept visualization for LSUN Church. The model uses 24 components in one group. Rows cover all learned components, with uncurated prototype samples shown for each component.

F.3 Generation Quality Metrics with 100-Step DDIM

We report generation quality metrics from 100-step DDIM sampling as auxiliary diagnostics. Table 7 compares DCDM with the corresponding generative baseline for each dataset. Bold entries indicate the better method within each dataset for each metric.

Table 7: Generation quality metrics using 100-step DDIM sampling.

Dataset	Method	FID ↓	KID ↓	IS ↑	Precision ↑	Recall ↑	LPIPS ↑	MS-SSIM ↓
CIFAR-10	Baseline	4.815±0.080	0.02702±0.00038	6.308±0.034	0.689±0.029	0.378±0.023	**0.257±0.006**	0.387±0.019
	Ours	**4.611±0.030**	**0.00379±0.00006**	**8.874±0.034**	**0.710±0.025**	**0.687±0.011**	0.221±0.002	**0.284±0.038**
CelebA	Baseline	**4.091±0.025**	**0.00283±0.00002**	2.782±0.010	0.724±0.012	**0.582±0.014**	**0.280±0.013**	**0.326±0.023**
	Ours	4.593±0.014	0.00331±0.00012	**2.817±0.001**	**0.743±0.006**	0.566±0.012	0.270±0.001	0.327±0.011
C-ImageNet	Baseline	21.366±0.183	0.01450±0.00017	**15.022±0.141**	**0.599±0.034**	0.627±0.032	0.550±0.010	**0.142±0.000**
	Ours	**19.295±0.146**	**0.01231±0.00025**	10.700±0.021	0.591±0.004	**0.639±0.006**	**0.561±0.020**	0.151±0.025
LSUN Bedroom	Baseline	6.173±0.020	0.00412±0.00020	2.567±0.017	0.626±0.006	0.505±0.031	**0.489±0.008**	0.145±0.006
	Ours	**5.028±0.062**	**0.00328±0.00012**	**2.575±0.012**	**0.627±0.007**	**0.520±0.005**	0.484±0.012	**0.143±0.013**
LSUN Church	Baseline	6.941±0.012	0.00513±0.00010	2.828±0.012	0.693±0.010	0.499±0.020	**0.459±0.003**	0.169±0.014
	Ours	**5.142±0.039**	**0.00345±0.00010**	**2.876±0.006**	**0.701±0.010**	**0.530±0.019**	0.458±0.013	**0.167±0.019**

F.4 Extended CAC Diagnostics

Tables 8 and 9 report the full CAC diagnostics under DINOv2 and CLIP. In addition to Δ_{tgt}, Δ_{ctrl}, and the lift reported in the main text, we include a high-mass control Δ_{hi} that ablates the second-highest-mass non-target component (matching the target in $\boldsymbol{\pi}$-perturbation magnitude), the half-ablation magnitude Δ_{half} obtained by replacing $\boldsymbol{\pi}^{\setminus k}$ with the partial-ablation distribution at $\lambda = 0.5$, and direction consistencies measured as the mean pairwise cosine similarity of ablation-induced embedding shifts across samples. The corresponding $\text{lift}_{\text{hi}} = \Delta_{\text{tgt}} - \Delta_{\text{hi}}$ remains positive on every dataset and encoder. The direction-consistency gap (target $\sim$ 0.1–0.2, control $\sim$ 0), together with the smooth ablation sweep in Figure 5, shows that targeted ablations move samples along coherent semantic directions while control ablations produce essentially random shifts.

Table 8: CAC diagnostics under DINOv2.

	Magnitude						Direction consistency	
Dataset	Δ_{tgt} ↑	Δ_{ctrl} ↓	Δ_{hi} ↓	Δ_{half}	lift ↑	lift_{hi} ↑	target ↑	control ↓
CIFAR-10	1.283 ± 0.004	0.069 ± 0.004	1.169 ± 0.007	0.971 ± 0.033	1.214 ± 0.007	0.114 ± 0.011	0.105 ± 0.002	0.001 ± 0.001
C-ImageNet	1.250 ± 0.003	0.356 ± 0.009	1.125 ± 0.008	1.039 ± 0.007	0.893 ± 0.010	0.125 ± 0.011	0.082 ± 0.001	0.002 ± 0.001
CelebA	0.990 ± 0.010	0.162 ± 0.001	0.547 ± 0.004	0.508 ± 0.014	0.828 ± 0.010	0.443 ± 0.011	0.221 ± 0.006	0.002 ± 0.002
LSUN Bedroom	1.017 ± 0.018	0.297 ± 0.004	0.916 ± 0.011	0.806 ± 0.016	0.720 ± 0.022	0.101 ± 0.027	0.087 ± 0.006	0.001 ± 0.001
LSUN Church	1.189 ± 0.008	0.264 ± 0.004	1.029 ± 0.009	0.967 ± 0.011	0.925 ± 0.007	0.160 ± 0.014	0.108 ± 0.002	0.001 ± 0.001

Table 9: CAC diagnostics under CLIP.

	Magnitude						Direction consistency	
Dataset	Δ_{tgt} ↑	Δ_{ctrl} ↓	Δ_{hi} ↓	Δ_{half}	lift ↑	lift_{hi} ↑	target ↑	control ↓
CIFAR-10	0.612 ± 0.007	0.039 ± 0.002	0.528 ± 0.007	0.417 ± 0.012	0.573 ± 0.007	0.085 ± 0.010	0.166 ± 0.008	0.002 ± 0.004
C-ImageNet	0.665 ± 0.003	0.150 ± 0.005	0.571 ± 0.012	0.508 ± 0.001	0.515 ± 0.003	0.094 ± 0.010	0.132 ± 0.004	0.004 ± 0.001
CelebA	0.660 ± 0.005	0.093 ± 0.005	0.357 ± 0.005	0.324 ± 0.011	0.567 ± 0.010	0.303 ± 0.010	0.198 ± 0.003	0.003 ± 0.004
LSUN Bedroom	0.456 ± 0.005	0.114 ± 0.002	0.400 ± 0.007	0.341 ± 0.003	0.342 ± 0.005	0.056 ± 0.011	0.148 ± 0.008	0.003 ± 0.004
LSUN Church	0.608 ± 0.003	0.117 ± 0.001	0.511 ± 0.002	0.469 ± 0.001	0.491 ± 0.005	0.097 ± 0.005	0.149 ± 0.003	0.001 ± 0.001

F.5 Extended Neighborhood and Probe Diagnostics

Table 10 reports the full NC and linear-probe diagnostics. NC uses the leave-source-out nearest-neighbor protocol at $m = 10$, and probe rows use $\widehat{\boldsymbol{\pi}} = \mathbb{E}_{q_\phi(\boldsymbol{\pi}|\mathbf{x}_0)}[\boldsymbol{\pi}]$ as the target. The final column reports the saturation ceiling of A^D_{adj} under hard DINOv2 k-means and serves as a metric calibration. A_{adj} scores hard 1-of-K assignments, while DCDM is compositional by design and produces soft Dirichlet distributions over multiple components.

Table 10: NC and linear-probe diagnostics.

Dataset	NC@10 LSO		DINO probe		CLIP probe		K-means A_{adj}^{D}
	DINO	CLIP	R^2	A_{adj}	R^2	A_{adj}	
CIFAR-10	0.325 ± 0.006	0.371 ± 0.006	0.510 ± 0.003	0.420 ± 0.006	0.584 ± 0.003	0.453 ± 0.008	0.972 ± 0.003
C-ImageNet	0.364 ± 0.008	0.358 ± 0.004	0.451 ± 0.001	0.398 ± 0.004	0.471 ± 0.002	0.407 ± 0.003	0.963 ± 0.002
CelebA	0.387 ± 0.012	0.314 ± 0.010	0.741 ± 0.001	0.634 ± 0.003	0.646 ± 0.002	0.548 ± 0.002	0.986 ± 0.001
LSUN Bedroom	0.186 ± 0.009	0.200 ± 0.006	0.437 ± 0.003	0.303 ± 0.013	0.418 ± 0.003	0.283 ± 0.008	0.931 ± 0.003
LSUN Church	0.344 ± 0.010	0.337 ± 0.001	0.541 ± 0.001	0.374 ± 0.005	0.555 ± 0.001	0.375 ± 0.001	0.955 ± 0.002

F.6 Extended SGA Diagnostics

Table 11 reports source-generation agreement. The S_{mis} column is the unrelated generated-image baseline.

Table 11: SGA diagnostics.

	DINOv2					CLIP				
Dataset	S_{match}	S_{mis}	S_{random}	S_{NN}	lift	S_{match}	S_{mis}	S_{random}	S_{NN}	lift
CIFAR-10	0.759 ± 0.001	0.094 ± 0.000	0.084 ± 0.000	0.709 ± 0.000	0.665 ± 0.001	0.949 ± 0.000	0.762 ± 0.000	0.757 ± 0.000	0.941 ± 0.000	0.187 ± 0.000
C-ImageNet	0.242 ± 0.000	0.031 ± 0.000	0.026 ± 0.000	0.767 ± 0.000	0.211 ± 0.000	0.789 ± 0.000	0.663 ± 0.000	0.636 ± 0.000	0.920 ± 0.000	0.126 ± 0.000
CelebA	0.648 ± 0.000	0.335 ± 0.000	0.317 ± 0.000	0.817 ± 0.000	0.313 ± 0.000	0.837 ± 0.000	0.692 ± 0.000	0.670 ± 0.000	0.924 ± 0.000	0.146 ± 0.000
LSUN Bedroom	0.481 ± 0.001	0.314 ± 0.002	0.304 ± 0.000	0.794 ± 0.000	0.167 ± 0.002	0.881 ± 0.001	0.837 ± 0.000	0.823 ± 0.000	0.948 ± 0.000	0.044 ± 0.000
LSUN Church	0.360 ± 0.001	0.098 ± 0.000	0.088 ± 0.000	0.751 ± 0.000	0.262 ± 0.001	0.837 ± 0.000	0.731 ± 0.000	0.710 ± 0.000	0.928 ± 0.000	0.105 ± 0.000

Figure 15 provides qualitative examples corresponding to the SGA metric.

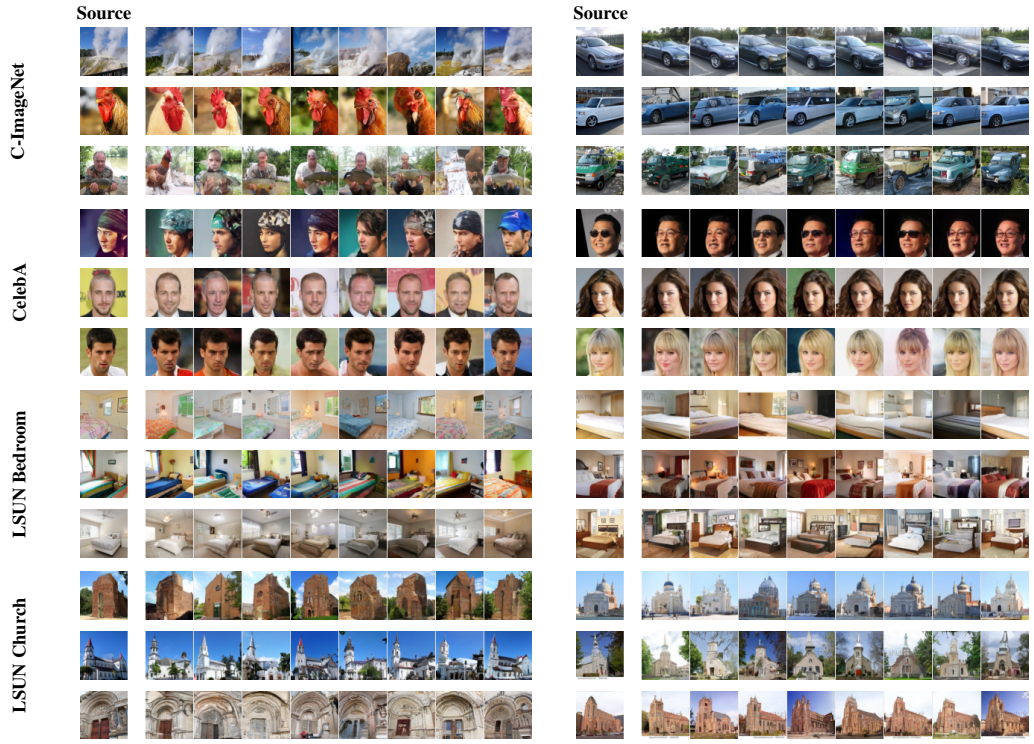

Figure 15: Source-generation agreement visualization. Each strip begins with a source image and then shows generations from the corresponding inferred concept distribution.

F.7 Concept Capacity Ablation

We evaluate how the number of learned components affects the quality of discovered concepts on CIFAR-10. Table 12 reports DINO-primary metrics. Here, K denotes the total number of concept components across all groups, and G denotes the number of groups.

Across the single-group variants, CAC remains strong as K increases from 12 to 24, indicating that the learned components continue to interventionally shape generation under moderate changes

Table 12: Concept capacity ablation on CIFAR-10.

Dataset	K	G	CAC lift ↑	NC@10 LSO ↑	Probe R^2 ↑	SGA lift ↑
CIFAR-10	12	1	1.134 ± 0.010	0.447 ± 0.010	0.563 ± 0.001	0.599 ± 0.001
	16	1	1.214 ± 0.007	0.325 ± 0.006	0.510 ± 0.003	0.665 ± 0.001
	24	1	1.216 ± 0.002	0.246 ± 0.002	0.481 ± 0.003	0.670 ± 0.000
	64	8	0.731 ± 0.009	0.107 ± 0.002	0.457 ± 0.002	0.674 ± 0.000

in concept capacity. At the same time, NC and probe R^2 decrease with larger K, suggesting that additional components split CIFAR-10 into finer or partially overlapping factors that are less aligned with DINO-local neighborhoods. In contrast, SGA improves with K, indicating that higher concept capacity can preserve more source-image-specific information. The high-capacity grouped setting with $K = 64$ and $G = 8$ follows the same tradeoff more strongly. SGA remains high, but CAC and NC drop substantially. The result suggests that high-capacity grouped concept spaces can over-fragment the data distribution, preserving source-image information while weakening component-level coherence.

F.8 Optional Concept Distillation

Complementing DCDM, we consider an optional learning scheme referred to as *concept distillation*. While the diffusion objective in Eq. (8) alone enables the model to discover meaningful structure in a fully unsupervised manner, the learned structure typically reflects visual semantics present in the input distribution. To encourage the model to capture higher-level semantic relationships beyond purely visual similarity, we introduce an additional training signal from a foundation model. In practice, the proposed model does not rely on this auxiliary objective, and the majority of experiments are conducted without concept distillation.

The insight behind concept distillation is that a good concept learner should understand data in a way aligned with an oracle's knowledge. Prior work building on the Concept Bottleneck Model [24] enforces a similar principle by explicitly using concept annotations from domain experts or crowdsourcing as a learning signal for supervised concept learning on intermediate neural network layers. However, such supervision is costly and often unavailable. In contrast, our approach replaces human labeling with knowledge distillation from a foundation model. Concretely, given an input $\mathbf{x}_0$, the concept encoder $f^{\phi}_{\text{concept}}$ produces logits $\mathbf{h}$, which parameterize Dirichlet parameters $\boldsymbol{\alpha}$ and induce a concept distribution $\boldsymbol{\pi} \in \Delta^{K-1}$. Concept distillation is applied directly to $\mathbf{h}$, encouraging the relational structure of $\mathbf{h}$ to align with semantic representations obtained from a foundation model.

We employ a foundation model f_{oracle} to generate a natural language description from $\mathbf{x}_0$ and a prompt p. The description is then embedded using a CLIP text encoder f_{clip} [40], producing a semantic embedding $\mathbf{e} \in \mathbb{R}^d$, where $d \gg K$. Given a batch of unperturbed inputs $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$, their corresponding embeddings $\{\mathbf{e}^i\}_{i=1}^m$, and logits $\{\mathbf{h}^i\}_{i=1}^m$, we apply the relational distillation loss of [37]:

$$L_{\text{distill}} = \sum_{(\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k) \in \mathbf{X}^3} l_\delta\big(\psi(\mathbf{h}^i, \mathbf{h}^j, \mathbf{h}^k), \psi(\mathbf{e}^i, \mathbf{e}^j, \mathbf{e}^k)\big)$$

$$\psi(\mathbf{v}^i, \mathbf{v}^j, \mathbf{v}^k) = \langle \hat{\mathbf{v}}^{i,j}, \hat{\mathbf{v}}^{k,j} \rangle, \quad \hat{\mathbf{v}}^{i,j} = \frac{\mathbf{v}^i - \mathbf{v}^j}{\|\mathbf{v}^i - \mathbf{v}^j\|_2}$$

where l_δ denotes the Huber loss [19]. This objective encourages the concept encoder to preserve the relative semantic relationships between inputs as captured by the foundation model. We evaluate whether the optional weak semantic supervision changes the learned concepts. Table 13 compares the vision-only model against concept-distilled DCDM on CIFAR-10.

Table 13: Optional concept-distillation ablation on CIFAR-10.

Dataset	Concept-distilled	CAC lift ↑	NC@10 LSO ↑	Probe R^2 ↑	SGA lift ↑
CIFAR-10	×	1.214 ± 0.007	0.325 ± 0.006	0.510 ± 0.003	0.665 ± 0.001
CIFAR-10	✓	1.202 ± 0.005	0.524 ± 0.006	0.638 ± 0.005	0.663 ± 0.001

Concept distillation primarily affects semantic alignment rather than generative control. The vision-only model matches the concept-distilled variant on CAC and SGA, showing that external semantic supervision is not needed for learned components to interventionally shape generation or preserve source-image information. Distillation improves NC and probe R^2, suggesting that weak semantic supervision from a foundation model aligns the learned concept space more closely with external feature geometry.

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 state the paper's scope as unsupervised concept discovery within diffusion-based generation. The Method, Theoretical Perspective, Empirical Evaluation, and Discussion and Limitations sections support these claims and describe their scope.

   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: Section 5 discusses limitations, including the fixed granularity of the concept space and the potential for excessive capacity to fragment related visual structure.

   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: [Yes]

Justification: Theorem 3.1 states the assumptions used for the denoising-risk analysis. Appendix C provides the full proof, with additional derivations in Appendix B.

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: The method, model architecture, dataset construction, training settings, and evaluation protocols are described in the main text and appendices. Code and reproduction instructions will be released upon publication of the paper.

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 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: The experiments use public datasets and publicly documented pretrained components that are cited in the paper. Code and instructions to reproduce the reported experiments will be released upon publication of the paper.

   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: The main text describes datasets and evaluation protocols. Appendix E, Appendix D, and Appendix E.2 provide dataset construction details, model architecture, and training hyperparameters.

   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: Appendix E states that quantitative metrics are reported as mean $\pm 2\sigma$ over three runs with different random seeds unless otherwise stated. The empirical risk-gap plot in Appendix F.1 reports a 95% confidence interval.

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: Appendix E.2 reports that each run used a single NVIDIA A10 GPU and provides per-dataset training steps and batch sizes. The reported setup is modest relative to large-scale generative-model training.

   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 uses public benchmark datasets and standard generative-model evaluation protocols. The authors have reviewed the NeurIPS Code of Ethics and are not aware of any deviation from it.

   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: The paper is primarily methodological, so its direct societal impact is limited. It can support more interpretable generative modeling, while it also inherits standard misuse risks associated with image generation.

    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: The paper does not release a high-risk pretrained model, scraped dataset, or deployment system. It studies a methodological contribution on public research datasets.

    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: The paper cites the public datasets and pretrained components used in the experiments, including FashionMNIST, CIFAR-10, ImageNet, CelebA, LSUN, CLIP, DINOv2, and the latent-diffusion autoencoder. These open research assets are used under their published terms.

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 the corresponding URL when possible.
- 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: The paper does not release a new dataset, benchmark, or model asset. The Conceptual ImageNet subset is documented as part of the experimental setup in Appendix E.1.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset, code, or 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: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

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: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

    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: [N/A]

    Justification: The core method does not use LLMs as an important, original, or non-standard component. The optional foundation-model distillation experiment is an auxiliary ablation and is described in Appendix F.8.

    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.