Steering Training Trajectories in Diffusion Representation Learning

Rajat Rasal*, **Avinash Kori**, **Tian Xia**, **Ben Glocker**
Imperial College London
{rrr2417,t.xia,agk21,b.glocker}@imperial.ac.uk

Abstract

Across training runs of a diffusion autoencoder, we observe similar image reconstruction quality, but substantially different levels of disentanglement in the learned representations. In this work, we study how training dynamics contribute to this behaviour. By analysing *training trajectories* in a controlled setting - curves of reconstruction quality against disentanglement during training - we identify two recurring trajectory patterns, which we call *learning regimes*. These regimes differ in how reconstruction and disentanglement evolve, despite similar reconstruction at the end of training. In the *reconstruction regime*, models improve image fidelity before disentanglement, while in the *disentanglement regime*, models improve both reconstruction and disentanglement more gradually. We hypothesise that modulating shortcut pathways in the diffusion U-Net and controlling early noise-level exposure can shape trajectories. To that end, we introduce STEERINGDRL, combining gated residual U-Nets with a noise curriculum to steer models towards better disentanglement. Across attribute disentanglement benchmarks, STEERINGDRL has competitive disentanglement, with reduced seed sensitivity in many settings. Our method extends to spatial disentanglement with object-centric learning, improving unsupervised segmentation on synthetic and real-world datasets. Our code is publicly available at https://github.com/RajatRasal/SteeringDRL.

1 Introduction

A central hypothesis in representation learning is that decomposing observations into *disentangled* components [2, 72] enables compositional generalisation in a data-efficient manner [62, 16]. Here, disentanglement refers to the separation of invariant factors in observational data [2]. As generative models are increasingly used for foundational tasks, the ability to learn structured representations becomes critical for controllability, transfer learning, and domain adaptation [44, 7, 38, 31]. However, models can generate high-fidelity images whilst representations are poorly aligned with true causal factors [45, 24]. In our work, we find that diffusion autoencoders, with identical objectives and architectures, can produce similar image quality with substantially different latent representations.

Locatello et al. [52] prove that disentangled representation learning (DRL) requires appropriate inductive biases or appropriate distributional assumptions on both the model, such as supervision or regularisation, and the data, such as knowledge of the data-generating process. Given the challenges with obtaining ground-truth labels, research efforts have focused on designing inductive biases for unsupervised attribute disentanglement [26] and spatial disentanglement with object-centric learning (OCL) [53]. While effective, these methods often trade off between representation learning and generative fidelity [5, 3]. Here, diffusion autoencoders provide a promising alternative [63, 84, 95], potentially mitigating this trade-off [90, 24]. Our findings suggest that diffusion autoencoders admit a spectrum of representational solutions, and which one is reached is influenced by training dynamics. By analysing image reconstruction and latent disentanglement over training, we motivate inductive biases for steering diffusion autoencoders towards better disentangled latent representations.

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

Diffusion models exhibit a natural hierarchical structure: coarse semantic information is captured at high noise levels, while fine-grained details are recovered at lower noise levels [81, 50]. Cross-attention provides a mechanism for querying this hierarchy, aligning conditioning signals with spatial representations across noise levels [25, 77]. This inductive bias underpins both text-to-image and object-centric diffusion models [68], where text tokens or object-slots are aligned with regions in pixel space [86, 32]. In particular, Yang et al. [89] demonstrate that cross-attention induces disentanglement without additional regularisation. Although this simple setup is compatible with large generative models, progress in disentanglement often relied on complex regularisation [13, 83], computationally expensive compositional inductive biases [85, 35, 34], or migration to newer generative model classes. We instead revisit diffusion autoencoders with cross-attention, and ask how their existing inductive biases can be better exploited for representation learning.

We study the optimisation dynamics of unsupervised representation learning with diffusion autoencoders via *training trajectories*: curves of reconstruction against representation quality during training. In a controlled setting, trajectories exhibit two recurring patterns, which we call *regimes*. In the *disentanglement regime*, models progressively improve disentanglement, while in the *reconstruction regime*, models prioritise reconstruction early in training. We hypothesise that early noise-level exposure and architectural shortcuts in the diffusion U-Net can influence which regime a run follows. Therefore, we introduce SteeringDRL, combining gated residual U-Nets with a noise curriculum to restrict reconstruction shortcuts and control early training dynamics. For attribute disentanglement, SteeringDRL improves or remains competitive with matched baselines, while reducing seed sensitivity in many settings. For spatial disentanglement with OCL, it improves unsupervised segmentation over matched baselines on synthetic and real-world datasets. Our results show that steering training trajectories is a simple yet effective method for improving disentanglement in diffusion autoencoders.

We make the following contributions:

1. **Regimes in Training Trajectories.** We show that diffusion autoencoders have training trajectories which organise around two regimes early in training under specific conditions. We propose hypotheses for why this happens (Section 4).
2. **Steering Trajectories.** Our hypotheses motivate architectural inductive biases and training strategies to steer trajectories towards better disentanglement: **SteeringDRL** (Section 5).
3. **Scaling SteeringDRL.** We show that SteeringDRL remains effective as model capacity increases, extending from attribute disentanglement to spatial disentanglement with OCL on synthetic and real-world datasets, while improving training efficiency (Section 6).

2 Related Work

Unsupervised Representation Learning. Diffusion autoencoders [63, 84, 95] have been used to modernise unsupervised representation learning, extending earlier VAE and GAN-based approaches for attribute disentanglement [26, 39, 10, 12] and object-centric learning [6, 53, 4]. Recent diffusion methods introduce regularisations and compositional inductive biases to encourage disentanglement [88, 85, 34, 35, 13, 83]. In contrast, EncDiff [89] highlights cross-attention alone as a simple and scalable component for disentanglement in diffusion, which SlotDiffusion [86] also adopts for object-centric learning. This mechanism underpins control in foundational generative models [8, 25, 77, 86, 32]. As such, our work builds on EncDiff and SlotDiffusion, demonstrating how their simple and scalable inductive biases can be better exploited by analysing training trajectories.

Inductive Biases in Diffusion Models. A large body of work studies the design and optimisation of diffusion models, including noise schedules [11, 56], timestep sampling strategies [23, 15], and prediction parameterisations [46, 71, 37, 49]. These components interact in subtle and often opaque ways, and are typically analysed in terms of generative quality, stability, or sampling efficiency. To that end, we adopt the VDM++ formulation [42, 28, 29, 67], which unifies a wide class of generative objectives as variance-preserving processes in $\log \mathrm{SNR}$ space (λ), differing primarily by a loss-weighting function $w(\lambda)$. As such, our framework naturally extends to other diffusion-style generative models. Architectural choices, such as residual, gated or removed skip connections [60, 29, 80, 78], have been shown to improve representational alignment [91]. Similarly, some works explore timestep sampling curricula or reweighting strategies during training [41, 87, 40, 92, 82, 48], which control exposure to different noise regimes. However, their interaction with training dynamics and the emergence of representations [57] has not been systematically studied.

3 Setup: Latent Variational Diffusion Autoencoder

This section outlines the experimental setup that we build on throughout this paper. Following prior work on unsupervised diffusion representation learning [86, 89], we model images using a diffusion autoencoder [63]. This consists of a semantic encoder, which generates semantic tokens ($\mathbf{S}$) from images ($\mathbf{y}$), and the diffusion decoder, operating in the latent space ($\mathbf{x}_0$) of a pretrained VQ-VAE [19, 68] to reconstruct images from $\mathbf{S}$. We train the diffusion model in $\log \mathrm{SNR}$-space (λ) following the VDM++ framework [42, 67], which unifies a wide class of generative objectives as variance-preserving processes in $\log \mathrm{SNR}$ space (λ), differing by a loss-weighting function $w(\lambda)$.

3.1 Latent Diffusion Decoder

Let $\mathbf{x}_0$ denote the projection of an image $\mathbf{y}$ into the latent space of a pretrained VQ-VAE [68, 19]. We learn a diffusion decoder in continuous-time [43] on $\mathbf{x}_0$, keeping the VQ-VAE parameters frozen. Operating in this perceptually compressed latent space reduces computational cost and encourages the diffusion model to focus on high-level semantics rather than high-frequency pixel-level details. Here, $\mathbf{x}_0$ is mapped to a noisy intermediate $\mathbf{x}_t$ at time $t \in [0, 1]$ via the forward process,

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}), \qquad \mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $\alpha_t = \sqrt{\mathrm{sigmoid}(\lambda_t)}$ and $\sigma_t = \sqrt{\mathrm{sigmoid}(-\lambda_t)}$ satisfy the variance-preserving (VP) constraint $\alpha_t^2 + \sigma_t^2 = 1$, $\lambda_t = \log \mathrm{SNR}(t) = \log(\alpha_t^2/\sigma_t^2)$ is the log signal-to-noise ratio (logSNR). The conditional generative transitions $p_\theta(\mathbf{x}_s|\mathbf{x}_t, \mathbf{S})$ are defined with two timesteps $s < t$,

$$p_\theta(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{S}) = \mathcal{N}(\mathbf{x}_s;\, \boldsymbol{\mu}_\theta(\mathbf{x}_t; s, t, \mathbf{S}),\, \sigma^2(s,t)\mathbf{I}), \tag{2}$$

$$\text{with} \quad \sigma^2(s,t) = \frac{\sigma_{t|s}^2 \sigma_s^2}{\sigma_t^2} \quad \text{and} \quad \boldsymbol{\mu}_\theta(\mathbf{x}_t; s, t, S) = \mathbf{x}_t \frac{\alpha_{t|s}\, \sigma_s^2}{\sigma_t^2} + \hat{\mathbf{x}}_\theta(\mathbf{x}_t; t, S) \frac{\alpha_s\, \sigma_{t|s}^2}{\sigma_t^2}, \tag{3}$$

where $\mathbf{S}$ are tokens generated by the semantic encoder (Section 3.2), $p_\theta(\mathbf{x}_s|\mathbf{x}_t, \mathbf{S}) = q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0 = \hat{\mathbf{x}}_\theta(\mathbf{x}_t; t, \mathbf{S}))$, and a U-Net [69] $\hat{\boldsymbol{\epsilon}}_\theta(\cdot)$ is used to implement $\hat{\mathbf{x}}_\theta(\cdot)$ via the identity $\hat{\mathbf{x}}_\theta(\mathbf{x}_t; t, \mathbf{S}) = (\mathbf{x}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t; t, \mathbf{S}))/\alpha_t$ with $\alpha_{t|s} = \alpha_t/\alpha_s$ and $\sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2$.

3.2 Semantic Encoder

The semantic tokens $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^N$ are generated from the image $\mathbf{y}$ by a semantic encoder $\mathrm{Enc}_\phi(\cdot)$,

$$\mathbf{S} = \mathrm{Enc}_\phi(\mathbf{y}) = \big(\mathrm{IB}_\kappa \circ \mathrm{Feat}_\psi\big)(\mathbf{y}), \qquad \mathbf{z} = \mathrm{Feat}_\psi(\mathbf{y}) \tag{4}$$

where $\mathrm{Feat}_\psi(\cdot)$ is a feature extractor and $\mathrm{IB}_\kappa(\cdot)$ is an inductive bias that maps features $\mathbf{z}$ into semantic tokens $\mathbf{S}$, which condition the decoder via cross-attention. For disentanglement, [89, 33] implements $\mathrm{IB}_\kappa(\cdot)$ using a split-MLP architecture, where N parallel MLPs generate factor-wise tokens $\mathbf{S}$, while disentanglement is measured among scalars in $\mathbf{z}$. For spatial disentanglement, [86, 32] use slot attention for $\mathrm{IB}_\kappa(\cdot)$, producing a set of object representations (*slots*) $\mathbf{S}$ and object masks $\mathcal{M}$ [53].

3.3 Optimisation

Kingma and Gao [42] show that the $\boldsymbol{\epsilon}$-prediction loss in diffusion models [27] is a special case of the estimator

$$\mathcal{L}(\mathbf{x}, \mathbf{S}) = \frac{1}{2} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I}),\, \lambda \sim p(\lambda)} \left[\frac{w(\lambda)}{p(\lambda)} \big\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S}) \big\|_2^2 \right], \tag{5}$$

where $w(\lambda) = 1$ recovers the vanilla ELBO. This form follows from a change of variables from time (t) to $\log \mathrm{SNR}$ (λ) using a bijective noise schedule $f_\lambda : t \mapsto \lambda$. This induces the distribution $p(\lambda) = -\frac{d}{d\lambda} f_\lambda^{-1}(\lambda)$, truncated to $[\lambda_{\min}, \lambda_{\max}]$, when sampling $t \sim \mathcal{U}(0, 1)$. In this work, we use the cosine noise schedule (Appendix A). It follows that Equation (5) can be reformulated as

$$\mathcal{L}(\mathbf{x}, \mathbf{S}) = \frac{1}{2} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I}),\, t \sim \mathcal{U}(0,1)} \left[w(\lambda) \cdot -\frac{d\lambda}{dt} \big\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S}) \big\|_2^2 \right], \tag{6}$$

which may be more compatible with existing codebases, which commonly train in diffusion time (t). Note that monotonic loss-weightings $w(\lambda)$ maximise an ELBO under Gaussian-noise data augmentation [14]. In particular, setting $w(\lambda) = \mathrm{sigmoid}(-\lambda)$ is shown to improve image fidelity [42, 29]. For $\mathbf{x}_0$-prediction, we use the weighting $w_{\mathbf{x}_0}(\lambda) = \mathrm{sigmoid}(\lambda)$, which induces the same effective $\boldsymbol{\epsilon}$-prediction weighting $w_{\boldsymbol{\epsilon}}(\lambda) = \mathrm{sigmoid}(-\lambda)$ (Appendix A.2).

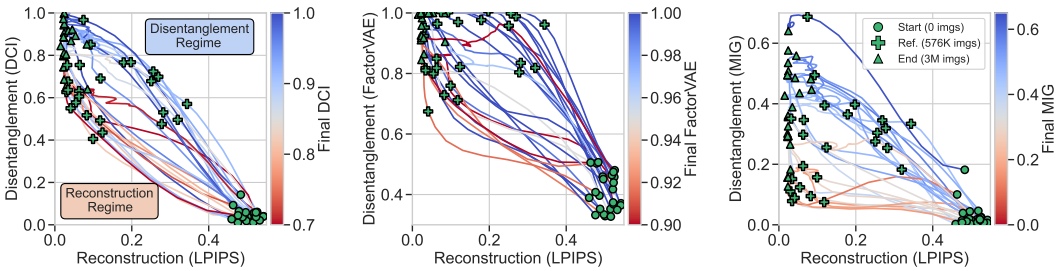


Figure 1: **Trajectories can organise around two regimes.** Training trajectories for a low-capacity diffusion autoencoder on Shapes3D (25 seeds). For interpretability, curves are coloured by final disentanglement, trajectories are smoothed and colour bars are centred at median final disentanglement.

4 Early Emergence of Regimes in Training Trajectories

We study diffusion autoencoders for attribute disentanglement in a controlled setting. Here, we observe that trajectories organise around two regimes. These observations motivate hypotheses (**H1 / H2**) about the underlying training dynamics and, in turn, our method (§5). In larger models, where the baseline regime separation may not be as clear, our method is still able to steer trajectories to achieve competitive disentanglement and/or reduce seed sensitivity (§6). Appendix E provides further analyses and outlines when this pattern is less clear, including sensitivity to VQ-VAE maturity.

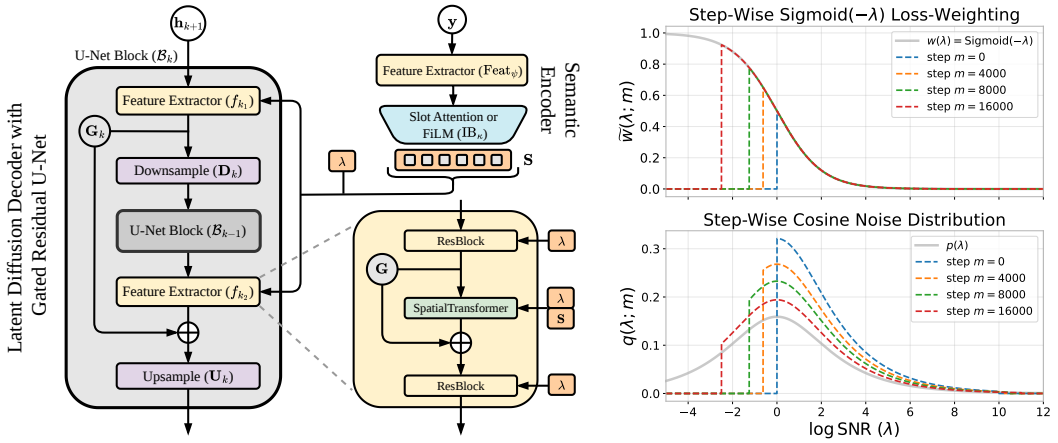
Setup. We construct a U-Net with 3M parameters, fixed up and down-sampling convolutions, using the split-MLP encoder for $\mathrm{IB}_\kappa(\cdot)$ with $N = 10$ tokens. We train using the EDM loss-weighting [37] with λ shifted by -3 via the loss in Equation (5), with the band $[\lambda_{\min}, \lambda_{\max}] = [-5, 12]$, on Shapes3D [39]. At regular intervals, we evaluate reconstruction (LPIPS [94]) against disentanglement in $\mathbf{z}$, measuring DCI disentanglement [17], FactorVAE [39] and MIG [10]. Details are in Appendix D.1.

Why do we see regimes? In Figure 1, we see that training trajectories separate into two *regimes* centred around the median final disentanglement. Similar regimes are seen with the other loss-weightings (Appendix E.1), and these trajectories cluster into two groups (Appendix E.2). Both regimes achieve similar final reconstruction, but the *disentanglement regime* yields higher final disentanglement, while the *reconstruction regime* prioritises reconstruction before disentanglement. Further, regimes correspond to different internal representations; Appendix E.5 shows that the disentanglement regime produces cross-attention maps aligned with ground-truth factors, whereas those in the reconstruction regime appear diffuse. This points to a mechanism which bypasses cross-attention; prior work shows that skip connections provide a high-bandwidth bypass around the U-Net trunk [33, 54]. We test this by zeroing out skip connections and using *SkipDropout* [33] (Appendix E.6). This improves disentanglement but substantially worsens reconstruction.

Hypothesis 1 (H1): *Since disentanglement via Equation (5) is unsupervised, the U-Net may exploit shortcut pathways via skip-connections to optimise reconstruction without organising factors in* $\mathbf{z}$.

Why do models commit early to a regime? We observe that trajectories diverge into their regimes early in training. In Figure 1, we highlight this with a reference point at 576K / 3M images seen, where reconstruction-regime models are already near-optimal in reconstruction but substantially worse in disentanglement, whereas models in the disentanglement regime improve gradually along both axes. Prior work shows that diffusion training imposes an information bottleneck over noise levels [89]; high-noise levels make reconstruction difficult, increasing reliance on clean conditioning signals, while low-noise levels recover fine-grained details as training progresses [79, 74, 81, 26] – widely observed in learning curves [37, 60, 55]. In early training, specific noise-levels may shape the relative pace of learning reconstruction vs disentanglement. We test this by retraining the current setup, modifying $[\lambda_{\min}, \lambda_{\max}]$ over 10 seeds (Appendix E.7). By 320K images seen, the original band $[-5, 12]$ yields fast reconstruction with competitive disentanglement. Removing high-noise levels with $[0, 12]$ delays reconstruction substantially, but does not collapse disentanglement. Conversely, only high-noise $[-5, 0]$ shows relatively strong early disentanglement but poor reconstruction.

Hypothesis 2 (H2): *Early exposure to specific noise levels can accelerate regime selection by changing how quickly reconstruction improves relative to disentanglement.*



(a) Diffusion Autoencoder w/ Gated Residual U-Net.

(b) log SNR curriculum.

Figure 2: **Inductive biases in STEERINGDRL. (a)** We define our U-Net recursively. We use gated residuals instead of skip connections, and gates in the SpatialTransformer. $\text{IB}_\kappa(\cdot)$ uses an amortised semantic encoder with FiLM [61] for attribute disentanglement (c.f. Appendix C.1), and SlotAttention [53] for object-centric learning. **(b)** The log SNR curriculum linearly widens the sampled log SNR band from $[0, 10]$ to $[-5, 12]$ over the first 32K steps in training.

5 STEERINGDRL: Steering Unsupervised Diffusion Representation Learning

Section 4 shows that diffusion autoencoders can follow distinct training trajectories. We hypothesise that shortcut pathways in the U-Net (**H1**) and noise-level exposure in early training (**H2**) contribute to this behaviour. We introduce two mechanisms that target these factors to improve disentanglement.

5.1 Gated Residual U-Net

U-Nets pass encoder features to the decoder via skip connections, giving the decoder access to high-frequency details and reducing reliance on the semantic tokens [46, 9]. To target **H1**, we replace concatenative skips with gated residuals [29], defining our U-Net block $\mathcal{B}(\cdot)$ recursively at level k as

$$\mathcal{B}_k(\mathbf{h}_{k+1}) = \mathbf{U}_k\Big((f_{2,k} \circ \mathcal{B}_{k-1} \circ \mathbf{D}_k)(\mathbf{h}_k) \oplus \mathbf{G}_k(\mathbf{h}_k)\Big), \qquad \mathbf{h}_k = f_{1,k}(\mathbf{h}_{k+1}), \tag{7}$$

where $\mathbf{D}(\cdot)$ and $\mathbf{U}(\cdot)$ are down and upsampling blocks, $f_1(\cdot)$ and $f_2(\cdot)$ are feature extractors conditioned on $\mathbf{S}$ via cross-attention and on λ via Fourier features, and $\mathbf{G}(\cdot)$ is a learnable gate [1, 93, 70],

$$\mathbf{G}(\mathbf{h}) = \text{Softplus}(\xi) \odot \mathbf{h}. \tag{8}$$

We initialise ξ such that $\text{Softplus}(\xi) \approx 0$ to suppress shortcuts early in training, when we observe regime commitment to occur. This introduces an information bottleneck on the skip pathways [79, 74]; skip information enters the decoder as a gated residual rather than as a separate stream via concatenation. This encourages greater reliance on the U-Net trunk and the conditioning signals in $\mathbf{S}$. Similarly, we gate residual connections inside spatial transformer blocks in $f_1(\cdot)$ and $f_2(\cdot)$ [80, 78],

$$\text{SpatialTransformer}(\mathbf{h}, \mathbf{S}; \lambda) = F(\mathbf{h}, \mathbf{S}; \lambda) \oplus \mathbf{G}(\mathbf{h}), \tag{9}$$

where $F(\cdot)$ is the transformer trunk. This conditions on $\mathbf{S}$ via cross-attention to encourage disentanglement [89, 86, 33, 32], where queries are derived from $\mathbf{h}$ and keys/values from $\mathbf{S}$. To stabilise representations, we normalise cross-attention queries using AdaLN-Zero modulated by λ [60].

5.2 log SNR Curriculum

To address **H2**, we control early noise-level exposure by training on a bounded log SNR band and gradually widening it to the full objective. At step m, we sample the active band $[\lambda_L(m), \lambda_U(m)]$ that widens linearly to $[\lambda_{\min}, \lambda_{\max}]$ while $m \le m_{\max}$:

$$\lambda_L(m) = \lambda_L(0) + \frac{m}{m_{\max}}\big(\lambda_{\min} - \lambda_L(0)\big), \qquad \lambda_U(m) = \lambda_U(0) + \frac{m}{m_{\max}}\big(\lambda_{\max} - \lambda_U(0)\big), \tag{10}$$

Table 1: **Components in STEERINGDRL interact to improve stability and retain competitive disentanglement.** Progressive ablations on Shapes3D using the EncDiff U-Net [89] ($N = 10$; 10 seeds). Macro Mean and Macro SD average the means and stds. across metrics, respectively. The curriculum we use here is shown in Figure 2b.

Model Variant	DCI ↑	FactorVAE ↑	MIG ↑	Macro Mean ↑	Macro SD ↓
U-Net + Split-MLP (EncDiff [89])	0.828 ± 0.099	0.921 ± 0.093	0.218 ± 0.087	0.656	0.093
+ Sigmoid$(-\lambda)$ Loss-Weighting	$\mathbf{0.932 \pm 0.077}$	$\mathbf{0.971 \pm 0.056}$	$\mathbf{0.399 \pm 0.102}$	**0.767**	0.078
+ log SNR Curriculum	0.853 ± 0.060	0.917 ± 0.063	0.299 ± 0.066	0.690	**0.063**
U-Net + Amortised Encoder (*Baseline*)	0.832 ± 0.087	0.904 ± 0.060	0.352 ± 0.098	0.696	0.082
+ Gated Residual U-Net	0.826 ± 0.082	0.885 ± 0.050	0.353 ± 0.093	0.688	0.075
+ Sigmoid$(-\lambda)$ Loss-Weighting	0.901 ± 0.065	0.977 ± 0.034	$\mathbf{0.386 \pm 0.092}$	0.755	0.064
+ log SNR Curriculum (**STEERINGDRL**)	$\mathbf{0.926 \pm 0.043}$	$\mathbf{0.991 \pm 0.017}$	0.353 ± 0.056	**0.757**	**0.039**

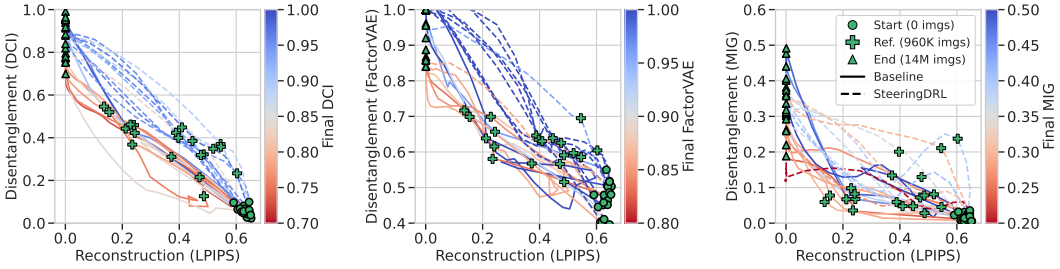


Figure 3: **STEERINGDRL steers training trajectories.** Trajectories for models in Table 1 show that STEERINGDRL (dashed lines) can shift the *baseline* (solid lines) towards the disentanglement regime; reconstruction is slower (STEERINGDRL ref. points lag behind *baseline* for reconstruction), trajectories shift upwards, final disentanglement is higher (dashed lines are bluer than solid lines).

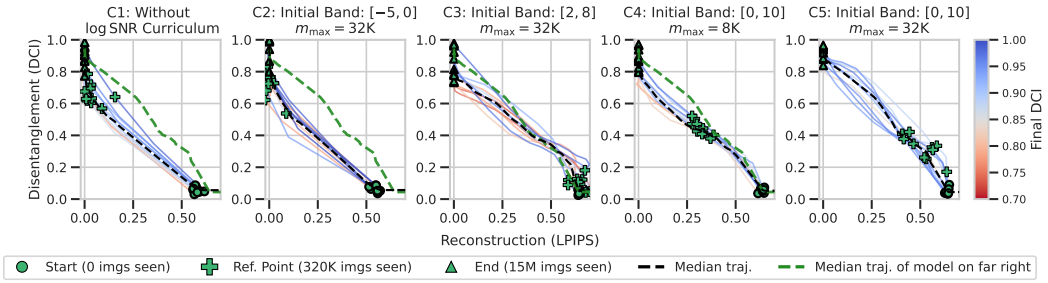


Figure 4: log SNR **curriculum selects the regime.** With architecture fixed (C1: Gated Residual U-Net + sigmoid$(-\lambda)$ loss-weighting, Table 1), we vary the curriculum (C2 to C5). Starting from high-noise (C2) biases trajectories toward the reconstruction regime, whereas starting from low-noise (C5) steers them into the disentanglement regime. This shows that regime selection occurs early in training, and that curricula can be designed to follow different regimes whilst attaining similar disentanglement. C5 beats C2 in disentanglement and reduces variance across runs (Appendix E.9).

with initial band $[\lambda_L(0), \lambda_U(0)]$ at $m = 0$. This defines a step-wise sampling density

$$q(\lambda; m) = \frac{p(\lambda)}{Z(m)} \cdot \mathbf{1}\{\lambda \in [\lambda_L(m), \lambda_U(m)]\}, \qquad Z(m) = t_L(m) - t_U(m) \tag{11}$$

for noise schedules induced by uniform sampling $t \sim \mathcal{U}(0, 1)$, where $t_U(m) = f_\lambda^{-1}(\lambda_U(m))$ and $t_L(m) = f_\lambda^{-1}(\lambda_L(m))$. Sampling λ from $q(\lambda; m)$ rather than $p(\lambda)$, requires the importance correction $w(\lambda)/q(\lambda; m) = Z(m)\, w(\lambda)/p(\lambda)$ to our objective in Equation (5):

$$\mathcal{L}(\mathbf{x}, \mathbf{S}; m) = \frac{1}{2}\, \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}),\, \lambda \sim q(\lambda; m)} \left[Z(m)\, \frac{w(\lambda)}{p(\lambda)}\, \left\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda, \mathbf{S}; \lambda) \right\|_2^2 \right], \tag{12}$$

where the effective step-wise loss-weighting at step m is,

$$\widetilde{w}(\lambda; m) = w(\lambda) \cdot \mathbf{1}\{\lambda \in [\lambda_L(m), \lambda_U(m)]\}. \tag{13}$$

This directly extends to the objectives in Equation (6) and $\mathbf{x}_0$-prediction (Appendix C.2).

Table 2: **STEERINGDRL improves attribute disentanglement over matched EncDiff baselines on Cars3D and Shapes3D and remains competitive on MPI3D-toy.** Methods marked in blue are run under matched conditions: EncDiff and STEERINGDRL use matched U-Net capacity, images seen in training and evaluation protocols, over 10 seeds. Baselines taken from literature are grey.

		Cars3D		Shapes3D		MPI3D-toy	
N	Method	FactorVAE ↑	DCI ↑	FactorVAE ↑	DCI ↑	FactorVAE ↑	DCI ↑
-	DisDiff [88]	0.976 ± 0.018	0.232 ± 0.019	0.902 ± 0.043	0.723 ± 0.013	0.617 ± 0.070	0.337 ± 0.057
-	FDAE [85]	0.918 ± 0.027	0.232 ± 0.418	0.987 ± 0.023	0.917 ± 0.038	–	–
-	DyGA [33]	0.941 ± 0.002	0.414 ± 0.013	1.000 ± 0.000	0.938 ± 0.001	0.930 ± 0.004	0.627 ± 0.002
-	Comp. IB [35]	0.877 ± 0.089	0.365 ± 0.073	0.975 ± 0.059	0.837 ± 0.105	0.708 ± 0.060	0.458 ± 0.052
20	EncDiff [89]	0.765 ± 0.037	0.258 ± 0.024	0.995 ± 0.010	0.916 ± 0.041	$\mathbf{0.899 \pm 0.024}$	0.670 ± 0.034
20	**STEERINGDRL**	$\mathbf{1.000 \pm 0.001}$	$\mathbf{0.624 \pm 0.071}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.954 \pm 0.030}$	0.870 ± 0.021	$\mathbf{0.690 \pm 0.031}$
10	EncDiff [89]	0.714 ± 0.035	0.208 ± 0.035	0.921 ± 0.093	0.828 ± 0.099	$\mathbf{0.805 \pm 0.061}$	$\mathbf{0.627 \pm 0.050}$
10	**STEERINGDRL**	$\mathbf{0.987 \pm 0.031}$	$\mathbf{0.527 \pm 0.112}$	$\mathbf{0.991 \pm 0.017}$	$\mathbf{0.926 \pm 0.043}$	0.800 ± 0.043	0.602 ± 0.039

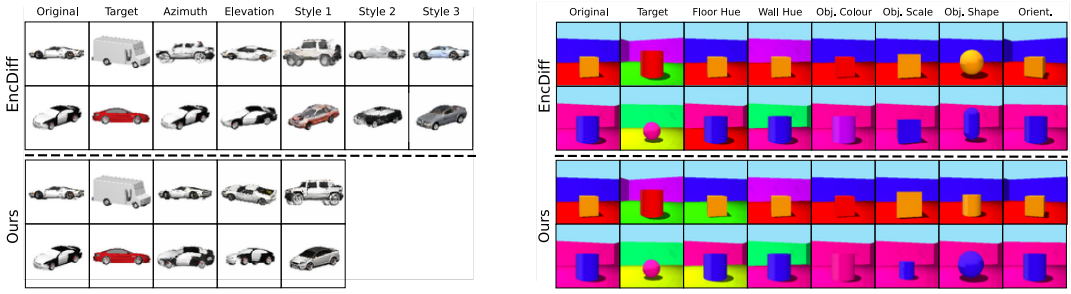


Figure 5: **Latent feature discovery and controlled image editing.** We manually select the latents for true attributes in the target image and swap them with the source. Compared to EncDiff, STEERINGDRL better isolates Cars3D attributes and produces more consistent edits on Shapes3D.

6 Experiments

We evaluate STEERINGDRL for disentanglement in two settings. In Section 6.1, we study attribute disentanglement, seeing how the trends in Section 4 apply to larger U-Nets. In Section 6.2, we apply our approach to spatial disentanglement on synthetic and real-world datasets, analysing generalisation when $\mathrm{IB}_\kappa(\cdot)$ uses Slot Attention [53] and $\mathrm{Feat}_\psi(\cdot)$ is pretrained. Setup details are in Appendix D.

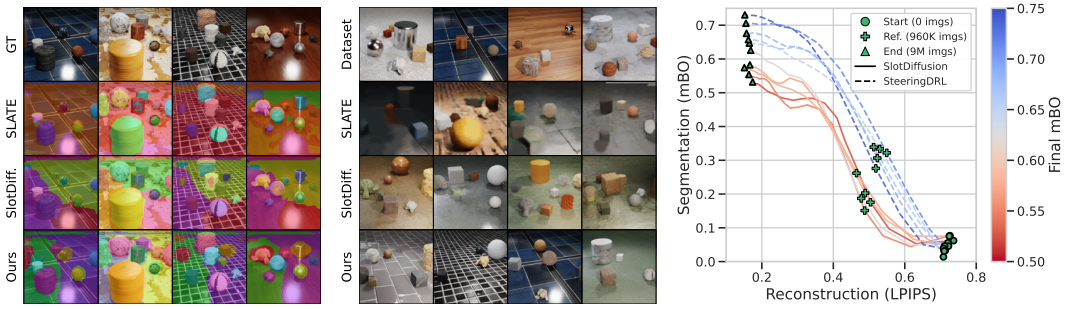
6.1 Attribute Disentanglement

We study unsupervised attribute disentanglement on *Shapes3D* [39], *Cars3D* [65], and *MPI3D-Toy* [21], where scalars in $\mathbf{z}$ should capture the true generative factors of the dataset. We follow the evaluation protocol from Section 4, now building on top of a better parameterised U-Net and a more mature VQ-VAE checkpoint, with both matched in capacity and training to the EncDiff baseline [89]. We use the loss in Equation (5) and the $\log \mathrm{SNR}$ curriculum as defined in Equation (12).

Inductive Biases in STEERINGDRL. In Table 1, we progressively apply our inductive biases to EncDiff [89] on Shapes3D. We replace the split-MLP with a parameter-efficient amortised encoder (Appendix C.1), matching EncDiff performance. With EncDiff, the sigmoid loss-weighting has the highest Macro Mean; adding our curriculum does not consistently improve individual metrics. With our gated residual U-Net, STEERINGDRL has comparable Macro Means and substantially lower Macro SD, although MIG does not improve. Figure 3 shows the corresponding training trajectories. When using a larger U-Net and a more mature VQ-VAE checkpoint than Section 4, the marked baseline here no longer exhibits regime separation and instead follows the reconstruction regime. Despite this harder setting, adding STEERINGDRL steers trajectories toward the disentanglement regime, improving disentanglement without sacrificing reconstruction. In Figure 4, we show that curricula can select different regimes, with the disentanglement-oriented trajectory showing better performance under the settings tested. Varying the initial band changes the training trajectory, while $m_{\max}$ determines the convergence speed within a regime. Overall, this shows that the curriculum influences regime selection, while our architectural bottlenecks and loss-weighting improve stability within it, as further supported by the skip-connection ablations in Appendix E.6.

Table 3: **STEERINGDRL improves segmentation and image fidelity with object-centric diffusion models.** Segmentation, reconstruction, and composition metrics on ClevrTex [36]. Methods marked in blue are run under the same conditions: Resnet34 for $\text{Feat}_\psi(\cdot)$, SlotAttention for $\text{IB}_\kappa(\cdot)$, the same data splits and evaluation protocol over 5 seeds. Baselines taken from literature are marked in grey.

	Segmentation			Recon.	Comp.
Method	**FG-ARI ↑**	**mIoU ↑**	**mBO ↑**	**LPIPS ↓**	**FID ↓**
SLATE$^+$ [32]	70.71 ± 3.35	52.96 ± 1.40	54.90 ± 1.15	–	69.23
StableLSD (200 steps, DDPM) [32]	64.41 ± 8.53	62.52 ± 1.88	63.93 ± 1.93	–	29.53
Modular IB [35]	87.68	58.88	59.12	–	–
MetaSlot [51]	77.60 ± 1.10	64.20 ± 0.60	62.80 ± 0.70	–	–
SLATE [75]	61.39 ± 9.07	55.00 ± 5.47	56.40 ± 7.19	0.397 ± 0.009	90.26 ± 3.72
SlotDiffusion (50 steps, DDIM) [86]	63.90 ± 5.11	55.92 ± 1.17	58.83 ± 1.70	0.167 ± 0.011	42.95 ± 3.07
STEERINGDRL (50 steps, DDIM)	$\mathbf{74.39 \pm 4.12}$	$\mathbf{68.31 \pm 2.39}$	$\mathbf{69.69 \pm 2.07}$	0.157 ± 0.005	26.67 ± 2.17
STEERINGDRL (200 steps, DDPM)	"	"	"	$\mathbf{0.147 \pm 0.005}$	$\mathbf{23.36 \pm 2.01}$



(a) Segmentations. (b) Random Compositions (c) Training Trajectories.

Figure 6: **Training trajectories support visual improvements.** (a) STEERINGDRL discovers more objects with more complete segmentations than SlotDiffusion. (b) Compositions preserve object structure and background textures. (c) Training trajectories show trends analogous to Figure 3.

Benchmark Datasets. In Table 2, we compare STEERINGDRL from Table 1 to a capacity-matched EncDiff. On Cars3D, STEERINGDRL improves disentanglement over EncDiff at $N \in \{10, 20\}$, while for Shapes3D we additionally reduce variance across seeds. We remain competitive on both metrics for MPI3D-toy while consistently reducing variance. Figure 5 illustrates the differences in disentanglement at $N = 10$. On Cars3D, STEERINGDRL better isolates the challenging `Style/Object Type` factor, which EncDiff spreads over multiple latents. On Shapes3D, STEERINGDRL recovers all ground-truth factors, while EncDiff often confuses `Floor Hue`, `Wall Hue`, and `Object Shape`.

6.2 Spatial Disentanglement with Object-Centric Learning

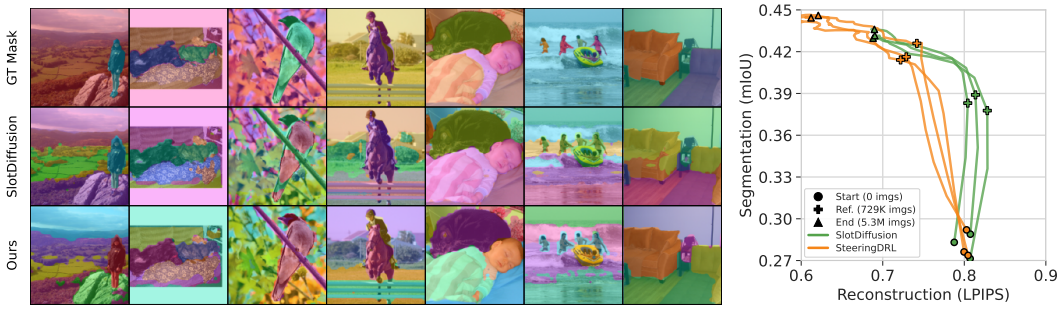
We study spatial disentanglement with OCL on *ClevrTex* [36] and *PascalVOC* [20], using segmentation metrics on $\mathcal{M}$ as a proxy for representation quality. We build on SlotDiffusion [86], follow its evaluation protocol, and use the objective in Equation (6) with curriculum in Appendix C.2. We use an effective $\text{sigmoid}(-\lambda)$ weighting, and a neutral initial band $[-2, 2]$ with $m_{\text{max}} = 200\text{k}$ for the curriculum; ClevrTex uses $\mathbf{x}_0$-prediction (Appendix A.2) and PascalVOC uses $\boldsymbol{\epsilon}$-prediction.

Synthetic Dataset (ClevrTex). In Table 3, we compare STEERINGDRL to matched baselines and see improvements on all metrics. For segmentation, improvements are driven by better identification of occluded objects and regions with low foreground-background contrast (Figure 6a), where SlotDiffusion in particular often under-segments or misses objects. Improved composition and reconstruction are linked to better preservation of background textures (Figure 6b). Notably, Figure 6c shows that our model exhibits training dynamics similar to the attribute disentanglement setting (Figure 3), while learning semantics faster and converging to higher segmentation scores than SlotDiffusion. This is consistent with **H1** and **H2**; controlling early noise-level exposure and introducing a skip bottleneck can also steer larger models toward better spatial disentanglement.

Training Efficiency with Pretrained Features (PascalVOC). We scale STEERINGDRL to a much larger U-Net ($\sim$ 138M params), using a pretrained DINO ViT for $\text{Feat}_\psi(\cdot)$. In Table 4, we improve

Table 4: **STEERINGDRL improves spatial disentanglement via object-centric learning on Pascal VOC.** Instance and class-wise segmentation metrics on PascalVOC [20]. Methods marked blue are trained under the same conditions: DINO ViT [58] for $\text{Feat}_\psi(\cdot)$, SlotAttention for $\text{IB}_\kappa(\cdot)$, the same data splits and evaluation protocol over 3 seeds. Other baselines from literature are marked in grey.

Model	FG-ARI	mIoU_i	mIoU_c	mBO_i	mBO_c
DINOSAUR MLP [73]	24.6	39.1	41.0	39.7	41.2
DINOSAUR Trans. [73]	23.1	42.0	47.5	43.2	47.8
StableLSD [32]	8.7	31.5	35.4	32.1	35.4
SA + DINO ViT [86]	12.3	–	–	24.6	24.9
SLATE + DINO ViT [75, 86]	15.6	–	–	35.9	41.5
SlotDiffusion [86]	15.9	43.2	47.7	47.2	52.2
STEERINGDRL	**17.6**	**44.5**	**48.6**	**49.1**	**53.6**



(a) Segmentations. (b) Training Trajectories.

Figure 7: **STEERINGDRL accelerates object-centric learning with pretrained features. (a)** STEERINGDRL better segments dominant objects while SlotDiffusion often undersegments. **(b)** STEERINGDRL reaches higher segmentation and reconstruction earlier in training than SlotDiffusion.

over matched baselines on all metrics. Our improvements stem from better segmentation of dominant objects that SlotDiffusion often merges (Figure 7a). We also find meaningful unannotated regions. Figure 7b shows different trajectories to ClevrTex despite using a similar curriculum. On ClevrTex, where $\text{Feat}_\psi(\cdot)$ is learned from scratch, representations and reconstructions are learned together progressively. With pretrained features, semantics are already available, so learning shifts from feature discovery to slot binding, allowing STEERINGDRL to converge faster than SlotDiffusion with better segmentations and reconstructions (see Appendix E.11.2 for more results). We see a similar pattern on the COCO dataset in the early stages of training (c.f. Appendix E.11.3).

7 Conclusion

We studied a key challenge in diffusion autoencoders for unsupervised disentanglement: models with similar reconstruction quality can learn representations with different levels of disentanglement. In a controlled setting, we observe training trajectories organising around two regimes that differ in how reconstruction and disentanglement develop. Our hypotheses about these dynamics motivate STEERINGDRL, which steers models even when the baseline trajectories show no clear regime separation. Against matched baselines, STEERINGDRL is highly competitive on attribute disentanglement, with reduced seed sensitivity in many settings, and improves spatial disentanglement with OCL. These results support **H1** and **H2** as useful design principles for disentanglement in diffusion autoencoders: restrict reconstruction shortcuts and control when models see specific noise levels.

Limitations and Future Work. We use the standard VQ-VAE architecture from EncDiff and SlotDiffusion; although we control for its capacity in matched comparisons and analyse its sensitivity in Appendix E, testing alternative autoencoders [96], pixel-space generative models [47] and different conditioning mechanisms [60] remains future work. We fix the curriculum's starting band and widening duration across datasets in each disentanglement setting, but new settings may require tuning; a robust unsupervised proxy may help with this [40, 41]. Our work supports **H1** and **H2** as factors affecting disentanglement, but there may be confounders that our analysis did not cover.

Acknowledgments and Disclosure of Funding

We thank Pavithra Manoj, Fabio De Sousa Ribeiro, Xiaodan Xing, Vishal Jain, and Raghav Mehta for their detailed and insightful feedback on early versions of this manuscript. We also thank Nairouz Shehata for her support with the tooling used to produce the figures. Finally, we thank Bernhard Kainz for his support and encouragement during this project.

R.R. is supported by EPSRC through a Doctoral Training Partnerships PhD Scholarship. A.K. is supported by the EPSRC Doctoral Prize. T.X. is supported through the Imperial College London UKRI Impact Acceleration Account EP/X52556X/1. B.G. was supported by the Royal Academy of Engineering as part of the Kheiron/RAEng Research Chair. The research was supported by the NIHR Imperial Biomedical Research Centre.

References

[1] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In Uncertainty in artificial intelligence, pages 1352–1361. PMLR, 2021.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8): 1798–1828, 2013.

[3] Jack Brady, Roland S. Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius Von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pages 3038–3062. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/brady23a.html.

[4] Jack Brady, Julius von Kügelgen, Sébastien Lachapelle, Simon Buchholz, Thomas Kipf, and Wieland Brendel. Interaction asymmetry: A general principle for learning composable abstractions. arXiv preprint arXiv:2411.07784, 2024.

[5] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. arXiv preprint arXiv:1804.03599, 2018.

[6] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390, 2019.

[7] Pierre Chambon, Christian Bluethgen, Jean-Benoit Delbrouck, Rogier Van der Sluijs, Małgorzata Połacin, Juan Manuel Zambrano Chaves, Tanishq Mathew Abraham, Shivanshu Purohit, Curtis P Langlotz, and Akshay Chaudhari. Roentgen: vision-language foundation model for chest x-ray generation. arXiv preprint arXiv:2211.12737, 2022.

[8] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023.

[9] Guanjie Chen, Xinyu Zhao, Yucheng Zhou, Xiaoye Qu, Tianlong Chen, and Yu Cheng. Towards stabilized and efficient diffusion transformers through long-skip-connections with spectral constraints. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 17708–17718, 2025.

[10] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. Advances in neural information processing systems, 31, 2018.

[11] Ting Chen. On the importance of noise scheduling for diffusion models. arXiv preprint arXiv:2301.10972, 2023.

[12] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems, 29, 2016.

[13] Jinjin Chi, Taoping Liu, Mengtao Yin, Ximing Li, Yongcheng Jing, and Dacheng Tao. Disentangled representation learning via flow matching. arXiv preprint arXiv:2602.05214, 2026.

[14] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509, 2019.

[15] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11472–11481, 2022.

[16] Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. Building machines that learn and think with people. Nature human behaviour, 8(10):1851–1863, 2024.

[17] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

[18] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. arXiv preprint arXiv:1907.13052, 2019.

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12873–12883, 2021.

[20] Mark Everingham, Luc Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88 (2):303–338, 2010.

[21] Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d97d404b6119214e4a7018391195240a-Paper.pdf.

[22] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In International conference on machine learning, pages 2424–2433. PMLR, 2019.

[23] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7441–7451, 2023.

[24] Jonathan Heek, Emiel Hoogeboom, Thomas Mensink, and Tim Salimans. Unified latents (ul): How to train your latents. arXiv preprint arXiv:2602.17270, 2026.

[25] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626, 2022.

[26] 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.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[28] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In International Conference on Machine Learning, pages 13213–13232. PMLR, 2023.

[29] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. arXiv preprint arXiv:2410.19324, 2024.

[30] Lawrence Hubert and Phipps Arabie. Comparing partitions. Journal of Classification, 2(1): 193–218, 1985. doi: 10.1007/BF01908075.

[31] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 23115–23127, 2024.

[32] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. arXiv preprint arXiv:2303.10834, 2023.

[33] Youngjun Jun, Jiwoo Park, Kyobin Choo, Tae Eun Choi, and Seong Jae Hwang. Disentangling disentangled representations: Towards improved latent units via diffusion models. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 3559–3569. IEEE, 2025.

[34] Whie Jung, Jaehoon Yoo, Sungjin Ahn, and Seunghoon Hong. Learning to compose: Improving object centric learning by injecting compositionality. arXiv preprint arXiv:2405.00646, 2024.

[35] Whie Jung, Dong Hoon Lee, and Seunghoon Hong. Disentangled representation learning via modular compositional bias. arXiv preprint arXiv:2510.21402, 2025.

[36] Laurynas Karazija, Iro Laina, and Christian Rupprecht. ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021.

[37] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems, 35: 26565–26577, 2022.

[38] Samar Khanna, Patrick Liu, Linqi Zhou, Chenlin Meng, Robin Rombach, Marshall Burke, David Lobell, and Stefano Ermon. Diffusionsat: A generative foundation model for satellite imagery. arXiv preprint arXiv:2312.03606, 2023.

[39] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In International conference on machine learning, pages 2649–2658. PMLR, 2018.

[40] Jin-Young Kim, Hyojun Go, Soonwoo Kwon, and Hyun-Gyoon Kim. Denoising task difficulty-based curriculum for training diffusion models. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=96GMFXsbJE.

[41] Myunsoo Kim, Donghyeon Ki, Seong-Woong Shim, and Byung-Jun Lee. Adaptive non-uniform timestep sampling for diffusion model training. arXiv preprint arXiv:2411.09998, 2024.

[42] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. Advances in Neural Information Processing Systems, 36:65484–65516, 2023.

[43] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. Advances in neural information processing systems, 34:21696–21707, 2021.

[44] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

[45] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 18262–18272, 2025.

[46] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720, 2025.

[47] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 36115–36125, 2026.

[48] Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, and Qing Qu. Understanding representation dynamics of diffusion models via low-dimensional modeling. arXiv preprint arXiv:2502.05743, 2025.

[49] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.

[50] Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. arXiv preprint arXiv:2404.02747, 2024.

[51] Hongjia Liu, Rongzhen Zhao, Haohan Chen, and Joni Pajarinen. Metaslot: Break through the fixed number of slots in object-centric learning. arXiv preprint arXiv:2505.20772, 2025.

[52] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In international conference on machine learning, pages 4114–4124. PMLR, 2019.

[53] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. Advances in neural information processing systems, 33:11525–11538, 2020.

[54] Jiajun Ma, Shuchen Xue, Tianyang Hu, Wenjia Wang, Zhaoqiang Liu, Zhenguo Li, Zhi-Ming Ma, and Kenji Kawaguchi. The surprising effectiveness of skip-tuning in diffusion sampling. arXiv preprint arXiv:2402.15170, 2024.

[55] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In European Conference on Computer Vision, pages 23–40. Springer, 2024.

[56] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International conference on machine learning, pages 8162–8171. PMLR, 2021.

[57] Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. Advances in Neural Information Processing Systems, 36:50173–50195, 2023.

[58] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023.

[59] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

[60] William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4195–4205, 2023.

[61] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[62] Steven T Piantadosi, Joshua B Tenenbaum, and Noah D Goodman. The logical primitives of thought: Empirical foundations for compositional cognitive models. Psychological review, 123 (4):392, 2016.

[63] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10619–10629, 2022.

[64] W.M. Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66(336):846–850, 1971.

[65] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/e07413354875be01a996dc560274708e-Paper.pdf.

[66] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In ICLR, 2022.

[67] Fabio De Sousa Ribeiro, Ben Glocker, et al. Demystifying variational diffusion models. Foundations and Trends® in Computer Graphics and Vision, 17(2):76–170, 2025.

[68] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022.

[69] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

[70] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. IEEE transactions on pattern analysis and machine intelligence, 45(4):4713–4726, 2022.

[71] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512, 2022.

[72] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. Neural computation, 4(6):863–879, 1992.

[73] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=b9tUk-f_aG.

[74] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

[75] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=h0OYV0We3oh.

[76] Krishnakant Singh, Simone Schaub-Meyer, and Stefan Roth. Glass: Guided latent slot diffusion for object-centric learning. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 28673–28683, 2025.

[77] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Türe. What the daam: Interpreting stable diffusion using cross attention. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5644–5659, 2023.

[78] Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention residuals, 2026.

[79] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. arXiv preprint physics/0004057, 2000.

[80] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 32–42, 2021.

[81] Binxu Wang and John J Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later. arXiv preprint arXiv:2303.02490, 2023.

[82] Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 12934–12944, 2025.

[83] Liming Wang, Muhammad Jehanzeb Mirza, Yishu Gong, Yuan Gong, Jiaqi Zhang, Brian H. Tracey, Katerina Placek, Marco Vilela, and James R. Glass. Can diffusion models disentangle? a theoretical perspective. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. URL https://openreview.net/forum?id=u1HclHIsLQ.

[84] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. In International conference on machine learning, pages 36336–36354. PMLR, 2023.

[85] Ancong Wu and Wei-Shi Zheng. Factorized diffusion autoencoder for unsupervised disentangled representation learning. Proceedings of the AAAI Conference on Artificial Intelligence, 38(6): 5930–5939, Mar. 2024. doi: 10.1609/aaai.v38i6.28407. URL https://ojs.aaai.org/index.php/AAAI/article/view/28407.

[86] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. Advances in Neural Information Processing Systems, 36:50932–50958, 2023.

[87] Tianshuo Xu, Peng Mi, Ruilin Wang, and Yingcong Chen. Towards faster training of diffusion models: An inspiration of a consistency phenomenon. arXiv preprint arXiv:2404.07946, 2024.

[88] Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. arXiv preprint arXiv:2301.13721, 2023.

[89] Tao Yang, Cuiling Lan, Yan Lu, and Nanning Zheng. Diffusion model with cross attention as an inductive bias for disentanglement. Advances in Neural Information Processing Systems, 37: 82465–82492, 2024.

[90] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940, 2024.

[91] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In International Conference on Learning Representations, 2025.

[92] Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I-Chao Chang, and Hanwang Zhang. Exploring diffusion time-steps for unsupervised representation learning. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=bWzxhtl1HP.

[93] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. arXiv preprint arXiv:1901.09321, 2019.

[94] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595, 2018.

[95] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. Advances in neural information processing systems, 35: 22117–22130, 2022.

[96] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In International Conference on Learning Representations, volume 2026, pages 35791–35820, 2026.

A Background

A.1 Cosine Noise Schedule

We use the cosine noise schedule in all of our experiments. It is defined as

$$\lambda = f_\lambda(t) = -2\log\left(\tan\left(\frac{\pi t}{2}\right)\right), \qquad t = f_\lambda^{-1}(\lambda) = \frac{2}{\pi}\arctan(e^{-\lambda/2}), \tag{14}$$

with corresponding density

$$p(\lambda) = -\frac{d}{d\lambda} f_\lambda^{-1}(\lambda) = \frac{\operatorname{sech}(\lambda/2)}{2\pi}. \tag{15}$$

A.2 Objective for $\mathbf{x}_0$-Prediction

Our main objective is written in $\boldsymbol{\epsilon}$-prediction form, where the loss is weighted by $w_{\boldsymbol{\epsilon}}(\lambda)$. For some experiments, however, we use $\mathbf{x}_0$-prediction. Here, we derive the equivalent weighting required for $\mathbf{x}_0$-prediction to induce the same effective $\boldsymbol{\epsilon}$-prediction objective. For variance-preserving diffusion, the forward process is

$$\mathbf{x}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}, \qquad \alpha_\lambda^2 = \operatorname{sigmoid}(\lambda), \qquad \sigma_\lambda^2 = \operatorname{sigmoid}(-\lambda), \tag{16}$$

so that

$$\frac{\alpha_\lambda^2}{\sigma_\lambda^2} = \exp(\lambda). \tag{17}$$

Given an $\mathbf{x}_0$-prediction model $\hat{\mathbf{x}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S})$, the corresponding noise prediction is

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S}) = \frac{\mathbf{x}_\lambda - \alpha_\lambda \hat{\mathbf{x}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S})}{\sigma_\lambda}. \tag{18}$$

Substituting $\mathbf{x}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$ gives

$$\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta = \frac{\alpha_\lambda}{\sigma_\lambda}\left(\mathbf{x} - \hat{\mathbf{x}}_\theta\right). \tag{19}$$

Therefore,

$$\left\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta\right\|_2^2 = \frac{\alpha_\lambda^2}{\sigma_\lambda^2}\left\|\mathbf{x} - \hat{\mathbf{x}}_\theta\right\|_2^2 = \exp(\lambda)\left\|\mathbf{x} - \hat{\mathbf{x}}_\theta\right\|_2^2. \tag{20}$$

It follows that an $\boldsymbol{\epsilon}$-prediction objective with weighting $w_{\boldsymbol{\epsilon}}(\lambda)$,

$$w_{\boldsymbol{\epsilon}}(\lambda)\left\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta\right\|_2^2, \tag{21}$$

is equivalent to an $\mathbf{x}_0$-prediction objective with weighting

$$w_{\mathbf{x}}(\lambda) = \exp(\lambda) w_{\boldsymbol{\epsilon}}(\lambda). \tag{22}$$

In our main experiments for attribute disentanglement, we use

$$w_{\boldsymbol{\epsilon}}(\lambda) = \operatorname{sigmoid}(-\lambda). \tag{23}$$

Using Equation (22), the corresponding $\mathbf{x}_0$-prediction weighting is

$$w_{\mathbf{x}}(\lambda) = \exp(\lambda)\operatorname{sigmoid}(-\lambda) = \operatorname{sigmoid}(\lambda). \tag{24}$$

Thus, when training $\mathbf{x}_0$-prediction models for spatial disentanglement with ClevrTex, we use $w_{\mathbf{x}}(\lambda) = \operatorname{sigmoid}(\lambda)$ to preserve the same effective $\boldsymbol{\epsilon}$-space weighting as $w_{\boldsymbol{\epsilon}}(\lambda) = \operatorname{sigmoid}(-\lambda)$.

B Metrics

B.1 Attribute Disentanglement

For attribute disentanglement metrics (DCI, FactorVAE and MIG), we use code from Ren et al. [66].

DCI Disentanglement [17]. We report the disentanglement component of the DCI framework [17]. This measures whether the predictive information in each latent dimension is concentrated on a single ground-truth factor. We train a gradient-boosted tree classifier from Scikit-Learn [59] to predict each of the M ground-truth factors from the learned representation vector $\mathbf{z} \in \mathbb{R}^D$ produced by $\text{Feat}_\psi(\cdot)$, using 10,000 training and 5,000 test images. We use the absolute values of the weights to form a feature importance matrix $\mathbf{R} \in \mathbb{R}^{D \times M}$, where the rows correspond to latents and the columns to ground truth factors. For each latent dimension i, we form a distribution over true factors,

$$p_{ij} = \frac{R_{ij}}{\sum_{j'=1}^{M} R_{ij'}}, \qquad \mathbf{p}_i = (p_{i1}, ..., p_{iM}). \tag{25}$$

The per-latent disentanglement score is then defined as

$$D_i = 1 - H_M(\mathbf{p}_i), \qquad H_M(\mathbf{p}_i) = -\sum_{j=1}^{M} p_{ij} \log_M p_{ij}. \tag{26}$$

If latent dimension i is important for only one ground-truth factor, then $H_M(\mathbf{p}_i) = 0$ and $D_i = 1$. If it is equally important for all M factors, then $H_M(\mathbf{p}_i) = 1$ and $D_i = 0$. The per-latent relative importance is defined by normalising each row w.r.t every cell in $\mathbf{R}$,

$$\rho_i = \frac{\sum_{j=1}^{M} R_{ij}}{\sum_{i'=1}^{D} \sum_{j=1}^{M} R_{i'j}}. \tag{27}$$

The final score is the importance-weighted average over latent dimensions,

$$\text{DCI-D} = \sum_{i=1}^{D} \rho_i D_i. \tag{28}$$

High values indicate that individual latent dimensions are predictive of single ground-truth factors. We report this value on the test samples.

FactorVAE [39]. This tests whether a fixed generative factor can be identified by finding which latent dimension changes the least. We first estimate the global variance of each latent coordinate in $\mathbf{z} \in \mathbb{R}^D$ using 10,000 training images, and prune inactive dimensions whose standard deviation is below 0.05. Then, we repeatedly sample mini-batches of size 64, with the value of one ground-truth factor v_j fixed across the batch, and sampling all other factors randomly. For each mini-batch, we compute the variance of each active latent coordinate and normalise it by its global variance. The active latent dimension with the smallest normalised variance casts a vote for the fixed factor v_j. Aggregating these votes over 10,000 images gives a majority-vote classifier from latent dimensions to factors. We report the accuracy of this classifier on 5,000 test samples. High values indicate that fixing a ground-truth factor reliably makes one latent dimension remain stable while the others vary.

Mutual Information Gap (MIG) [10]. This measures whether each ground-truth factor is captured by one clearly dominant latent dimension rather than being spread across several latents. We sample 10,000 images and compute their learned representations $\mathbf{z} \in \mathbb{R}^D$. Each latent dimension z_i is discretised into 20 bins, after which we estimate the discrete mutual information between every discretised latent dimension $\widetilde{z}_i \in \{1, ..., 20\}$ and every ground-truth factor v_j. This gives a mutual information matrix $\mathbf{M} \in \mathbb{R}^{D \times M}$, where $M_{ij} = I(\widetilde{z}_i, v_j)$. For each factor v_j, we sort the mutual information values across latent dimensions and compute the gap between the largest and second-largest values, normalised by the entropy $H(v_j)$:

$$\text{MIG} = \frac{1}{M} \sum_{j=1}^{M} \frac{M_{(1)j} - M_{(2)j}}{H(v_j)}, \tag{29}$$

where $M_{(1)j}$ and $M_{(2)j}$ denote the largest and second-largest entries in column j of $\mathbf{M}$. High values indicate that each ground-truth factor has a single dominant latent dimension that contains substantially more information about it than any other latent dimension.

B.2 Spatial Disentanglement

For the segmentation metrics for spatial disentanglement (FG-ARI, mBO, mIoU), we use the code from Wu et al. [86]. All of these metrics are converted to percentages for our tables.

Foreground Adjusted Rand Index (FG-ARI) [64, 30, 22]. This measures whether foreground pixels that belong to the same ground-truth object are assigned to the same predicted segment. The adjusted Rand Index (ARI) measures clustering similarity between pixels in a ground truth and predicted mask, where a score of 1 corresponds to perfect agreement, while a score near 0 corresponds to chance-level agreement. Following common practice in object-centric learning [53], we exclude pixels labelled as background in the ground-truth mask before computing ARI. Higher values indicate that foreground object instances are better recovered. Since FG-ARI ignores ground-truth background pixels, it does not penalise errors that only affect background segmentation [18, 36].

Mean Intersection-over-Union (mIoU). This measures the spatial overlap between predicted masks and ground-truth masks. For a predicted mask P and ground-truth mask G, the IoU is

$$\mathrm{IoU}(P, G) = \frac{|P \cap G|}{|P \cup G|}. \tag{30}$$

Since predicted slots are unordered, we first compute all pairwise IoUs between predicted masks and the relevant ground-truth masks, and then use Hungarian matching to find an assignment that maximises total IoU. The matched IoUs are averaged to obtain mIoU. We report both instance-wise and class-wise variants. For mIoU_i, the ground-truth masks correspond to object instances. For mIoU_c, the ground-truth masks correspond to semantic classes. Thus, mIoU_i measures recovery of individual object instances, while mIoU_c measures semantic class-level overlap. Higher values indicate better spatial agreement between predicted and ground-truth masks.

Mean Best Overlap (mBO). Mean best overlap also measures mask overlap, but unlike mIoU it does not enforce a one-to-one matching between predicted and ground-truth masks. For each ground-truth mask G_k, we compute its best overlap with any predicted mask,

$$\mathrm{BO}(G_k) = \max_l \frac{|G_k \cap P_l|}{|G_k \cup P_l|}, \tag{31}$$

where $\{P_l\}$ are the predicted masks. The mean best overlap is then

$$\mathrm{mBO} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{BO}(G_k). \tag{32}$$

We report both instance-wise and class-wise variants. For mBO_i, the ground-truth masks $\{G_k\}_{k=1}^K$ correspond to object instances. For mBO_c, they correspond to semantic classes. Thus, mBO_i measures whether each object instance is covered by at least one predicted mask, while mBO_c measures whether each semantic class region is covered by at least one predicted mask. Higher values indicate better best-case overlap between ground-truth masks and predicted masks.

C Method

C.1 Amortised Semantic Encoder with FiLM Modulation

To improve the parameter efficiency of the semantic encoder (Section 3.2), we design $\mathrm{IB}_\kappa(\cdot)$ attribute disentanglement using a shared backbone with FiLM modulation [61] in place of the split-MLP [89],

$$\textbf{Split-MLP:} \quad \mathbf{s}_i = \mathrm{IB}_{\xi_i}(z_i), \qquad \textbf{Ours:} \quad \mathbf{s}_i = \mathrm{IB}_{\xi}\big(z_i; \{\gamma_i^{(l)}, \beta_i^{(l)}\}_{l=1}^{L}\big), \tag{33}$$

where the split-MLP uses separate parameters ξ_i for each latent scalar z_i, while our approach shares weights ξ across all latents. The FiLM parameters $\{\gamma_i^{(l)}, \beta_i^{(l)}\}_{l=1}^{L}$ provide latent-specific feature-wise modulation across L layers of the shared network via scaling and shifting of intermediate activations. This design preserves factor-wise independence (each token $\mathbf{s}_i$ depends only on the scalar z_i) while reducing parameters by approximately $\mathcal{O}(N)$. In this work, we set $L = 2$. We found this to provide similar disentanglement as the split-MLP, seen in Table 1, when paired with a deeper feature extractor, which we refer to as Residual ConvNet in Table 5.

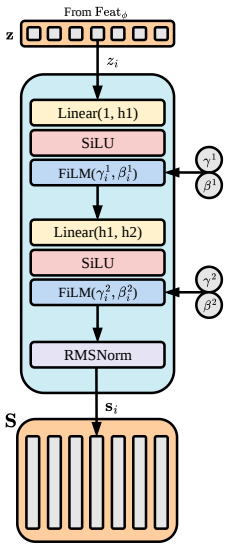


Figure 8: Architecture for Amortised Semantic Encoder

C.2 Deriving the $\log \mathrm{SNR}$ Curriculum Loss

Our curriculum restricts training at step m to an active $\log \mathrm{SNR}$ band $[\lambda_L(m), \lambda_U(m)]$. This defines the truncated density

$$q(\lambda; m) = \frac{p(\lambda)}{Z(m)} \mathbf{1}\{\lambda \in [\lambda_L(m), \lambda_U(m)]\}, \qquad Z(m) = \int_{\lambda_L(m)}^{\lambda_U(m)} p(\lambda)\, d\lambda. \tag{34}$$

Sampling from $q(\lambda; m)$ therefore requires the importance correction

$$\frac{w(\lambda)}{q(\lambda; m)} = Z(m) \frac{w(\lambda)}{p(\lambda)}. \tag{35}$$

Thus, the curriculum objective in λ-space is

$$\mathcal{L}(\mathbf{x}, \mathbf{S}; m) = \frac{1}{2} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}),\, \lambda \sim q(\lambda; m)} \left[Z(m) \frac{w(\lambda)}{p(\lambda)} \left\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S}) \right\|_2^2 \right]. \tag{36}$$

Equivalently, when implementing the objective by sampling time, we sample t uniformly over the corresponding active time interval $[t_U(m), t_L(m)]$, where $t_U(m) = f_\lambda^{-1}(\lambda_U(m))$ and $t_L(m) = f_\lambda^{-1}(\lambda_L(m))$. For decreasing noise schedules, this interval has mass

$$Z(m) = t_L(m) - t_U(m). \tag{37}$$

Using $\lambda = f_\lambda(t)$, the derivative-form curriculum objective is then

$$\mathcal{L}(\mathbf{x}, \mathbf{S}; m) = \frac{1}{2} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}),\, t \sim \mathcal{U}(t_U(m), t_L(m))} \left[Z(m)\, w(\lambda) \left(-\frac{d\lambda}{dt} \right) \left\| \boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_\lambda; \lambda, \mathbf{S}) \right\|_2^2 \right]. \tag{38}$$

Table 5: Variations in model architectures and training settings across datasets.

Dataset	Shapes3D	Cars3D	MPI3D	ClevrTex	PascalVOC
Feature Extractor ($\mathrm{Feat}_\psi(\cdot)$)	Residual ConvNet	Residual ConvNet	Residual ConvNet	ResNet34	DINO ViT-S/8
Disentanglement Inductive Bias ($\mathrm{IB}_\kappa(\cdot)$)	Amortised FiLM	Amortised FiLM	Amortised FiLM	Slot Attention	Slot Attention
Number of Tokens / Slots (N)	10/20	10/20	10/20	11	6
Token / Slot Dimension	16	16	16	192	192
Slot Attention Iterations	–	–	–	3	3
Prediction Target	$\boldsymbol{\epsilon}$	$\boldsymbol{\epsilon}$	$\boldsymbol{\epsilon}$	$\mathbf{x}_0$	$\boldsymbol{\epsilon}$
Loss Weighting	$\mathrm{Sigmoid}(-\lambda)$	$\mathrm{Sigmoid}(-\lambda)$	$\mathrm{Sigmoid}(-\lambda)$	$\mathrm{Sigmoid}(\lambda)$	$\mathrm{Sigmoid}(-\lambda)$
Initial Curriculum Band ($[\lambda_L(0), \lambda_U(0)]$)	$[0, 10]$	$[0, 10]$	$[0, 10]$	$[-2, 2]$	$[-2, 2]$
Full $\log \mathrm{SNR}$ Band ($[\lambda_{\min}, \lambda_{\max}]$)	$[-5, 12]$	$[-5, 12]$	$[-5, 12]$	$[-8.5, 6.5]$	$[-15, 15]$
Widening Steps ($m_{\max}$)	32K	32K	32K	200K	200K
Learning Rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	10^{-4}	10^{-4}
LR Schedule	Constant	Constant	Constant	Cosine	Cosine
EMA	U-Net + encoder EMA; inverse warmup, max 0.9999			–	–
Optimiser	AdamW	AdamW	AdamW	Adam	Adam
Batch Size	128	128	128	64	64
Base U-Net Architecture	EncDiff [89]			SlotDiffusion [86]	
Base VQ-VAE Architecture	Esser et al. [19] with LPIPS Discriminator			Ckpts from SlotDiffusion Repo [86]	
Latent Resolution $\mathbf{x}_0$	16×16	16×16	16×16	32×32	56×56

D Experimental Setup

D.1 Analysis

For the trajectory analysis in Section 4, we modify the lightweight `imagev3` U-Net implementation from `k-diffusion`[1] by adding EncDiff SpatialTransformer blocks for conditioning on the output of $\mathrm{Enc}_\phi(\cdot)$, together with a middle bottleneck layer. This U-Net uses fixed convolutions for up- and down-sampling. We use the `Encoder4` encoder for $\mathrm{Feat}_\psi(\cdot)$ and the split-MLP for $\mathrm{IB}_\kappa(\cdot)$ from EncDiff [89] to construct $\mathrm{Enc}_\phi(\cdot)$. Models are trained until they have seen 3M images, which is sufficient for most runs to reach stable reconstruction and disentanglement scores. For this analysis, we use the EncDiff VQ-VAE checkpoint trained for 30K steps. We use 5 model initialisation seeds and 5 λ-sampling seeds, giving 25 runs in total. This lets us test whether early disentanglement correlates with model initialisation or noise-level sampling; we did not find strong evidence for either. As discussed in Appendix E.8, this checkpoint makes regime separation more visible, whereas longer VQ-VAE training improves reconstruction and reduces the separation between regimes.

D.2 Main Experiments

Architecture and training details for all datasets are summarised in Table 5.

For the main attribute disentanglement experiments, we follow EncDiff [89] unless otherwise specified. We use the same datasets, evaluation protocol, and VQ-VAE architecture (now trained to 150K steps), while replacing the split-MLP encoder with our Scalar-to-Token FiLM encoder and the standard U-Net with our gated residual U-Net. All attribute models use $\boldsymbol{\epsilon}$-prediction with $\mathrm{sigmoid}(-\lambda)$ weighting and the $\log \mathrm{SNR}$ curriculum described in Section 5.2. We train until the model has seen 15M images, which is sufficient for convergence. This takes approximately 16 hours on an NVIDIA A40 or 10 hours on an NVIDIA RTX 4090.

For object-centric learning, we build on SlotDiffusion [86], instantiate $\mathrm{IB}_\kappa(\cdot)$ with Slot Attention, and follow its evaluation protocol on ClevrTex and PascalVOC. ClevrTex uses $\mathbf{x}_0$-prediction with the equivalent weighting in Appendix A.2, while PascalVOC uses $\boldsymbol{\epsilon}$-prediction. We use the VQ-VAE checkpoints provided by the SlotDiffusion codebase. For fair comparison to SlotDiffusion, we train ClevrTex for 400 epochs and PascalVOC for 500 epochs. Trajectory plots are computed on the validation set for ClevrTex. For PascalVOC, the val and test set are the same in the SlotDiffusion evaluation protocol. On H100 GPUs, both object-centric models can be trained within one day.

We obtain the full $\log \mathrm{SNR}$ band by mapping the noise schedules in the matched baselines, indexed by time, into $\log \mathrm{SNR}$-space, with minor adjustments for our formulation.

[1] https://github.com/crowsonkb/k-diffusion

D.3 OCL Curricula

For spatial disentanglement on ClevrTex and PascalVOC, we use the same initial log SNR band $[-2, 2]$ and widening duration $m_{\max} = 200\text{K}$. The final bands follow the corresponding baseline schedules: $[-8.5, 6.5]$ for ClevrTex and $[-15, 15]$ for PascalVOC (Table 5). Figure 9 illustrates the PascalVOC curriculum.

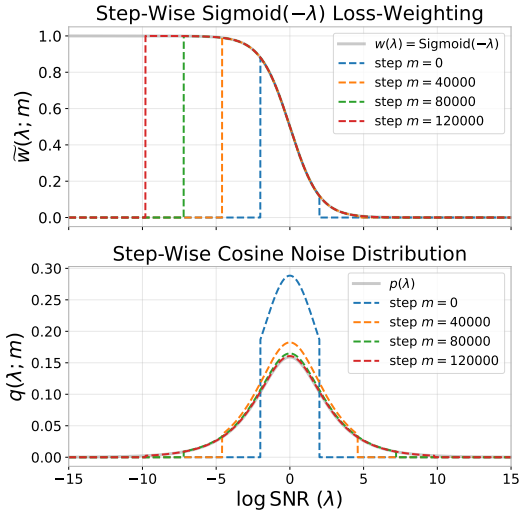


Figure 9: $\log \mathrm{SNR}$ curriculum for object-centric learning on the Pascal VOC dataset.

E Additional Results

E.1 Training Trajectories

We repeat the trajectory analysis from Figure 1 under alternative loss-weighting functions. In Figure 10, we use the non-monotonic weighting $w(\lambda) = \mathrm{sech}(\lambda)$, which places most weight near intermediate $\log \mathrm{SNR}$ values, akin to the EDM with shift -3 we used in Section 4. The same qualitative behaviour appears; runs separate into trajectories that prioritise reconstruction early and trajectories that maintain stronger disentanglement throughout training. This suggests that the regime structure is not specific to a single loss-weighting choice, but reflects a broader optimisation property of diffusion autoencoders which becomes clear in this low-capacity setting.

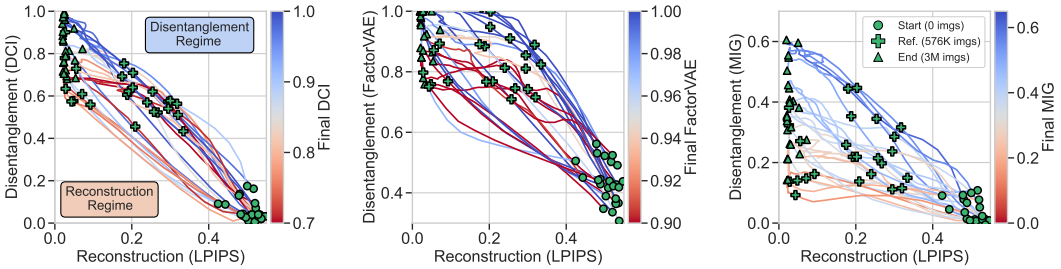


Figure 10: **Training trajectories organise around two distinct regimes.** This follows the same setup as Figure 3, now using a different non-monotonic loss-weighting function, $w(\lambda) = \mathrm{sech}(\lambda)$.

E.2 Analysing Regimes in Training Trajectories

Here we present a post-hoc analysis of the same 25 trajectories shown in Figures 1 and 10. We assess the two-regime structure by constructing a summary statistic of each training trajectory. For each run, at a step s, we define the normalised reconstruction progress

$$R(s) = \frac{\mathrm{LPIPS}(s_1) - \mathrm{LPIPS}(s)}{\mathrm{LPIPS}(s_1)}, \tag{39}$$

and normalised disentanglement progress for the metrics $m \in \{\mathrm{DCI}, \mathrm{FactorVAE}, \mathrm{MIG}\}$ as

$$D_m(s) = \frac{m(s) - m(s_1)}{1 - m(s_1)}, \tag{40}$$

where s is the number of training images seen and s_1 is the first checkpoint included in this analysis. The two quantities measure the relative progress of the reconstruction and disentanglement, and are therefore both equal zero at s_1. We set $s_1 = 64K$, which is the first evaluated checkpoint after $s = 0$. In Figures 1 and 10, the starting point at $s = 0$ is marked by a circle and the reference point at $s_{\mathrm{ref}} = 576\mathrm{K}$ is marked by a plus. The relative progress between s_1 and s_{ref} is

$$J_m = \frac{1}{s_{\mathrm{ref}} - s_1} \int_{s_1}^{s_{\mathrm{ref}}} [D_m(s) - R(s)] \, \mathrm{d}s = \frac{1}{512\mathrm{K}} \int_{64\mathrm{K}}^{576\mathrm{K}} [D_m(s) - R(s)] \, \mathrm{d}s. \tag{41}$$

Positive values indicate that disentanglement progresses more quickly than reconstruction over this interval, while negative values indicate the converse. The integral is approximated using the trapezium rule over the nine measurements from 64K to 576K. We compute J_m for each of the 25 runs denoted J_m^r, $r \in \{1, \ldots, 25\}$. For each metric m, we calculate the mean and standard deviation across runs:

$$\mu_m = \frac{1}{25} \sum_{r=1}^{25} J_m^r, \qquad \sigma_m = \sqrt{\frac{1}{25} \sum_{r=1}^{25} (J_m^r - \mu_m)^2}. \tag{42}$$

The Z-score $Z_m^r = (J_m^r - \mu_m)/\sigma_m$ is averaged over the three metrics to obtain a per-run statistic

$$J_c^r = \frac{1}{3} \left(Z_{\mathrm{DCI}}^r + Z_{\mathrm{FactorVAE}}^r + Z_{\mathrm{MIG}}^r \right), \tag{43}$$

where larger values of J_c^r indicate greater early disentanglement progress relative to reconstruction. All normalisation and standardisation operations are performed separately for each loss weighting.

For each loss weighting, we fit one-dimensional Gaussian mixture models (GMMs) to the J_c^r values using `scikit-learn`. We compare $K \in \{1, 2, 3, 4, 5\}$ mixture components and compare models using the Bayesian information criterion (BIC), for which lower values are preferred. Relative to the two component model, we also define

$$\Delta\mathrm{BIC}_{K-2} = \mathrm{BIC}_K - \mathrm{BIC}_2, \tag{44}$$

Positive values of $\Delta\mathrm{BIC}_{k-2}$ indicate that the two-component model is preferred over the corresponding k-component alternative. For the selected $K = 2$ model, we identify the component with the larger mean J_c^r as the disentanglement regime and the other as the reconstruction regime.

Evidence for Two Regimes. The two-regime structure visible in Figures 1 and 10 is supported quantitatively. For each loss weighting, we fit GMMs with $K \in \{1, 2, 3\}$ using `n_init=100` and select the model with the lowest BIC. For both EDM-3 and sech, BIC favours $K = 2$ over $K = 1$ and $K = 3$. In Table 6, both resulting regimes contain a substantial fraction of the 25 trajectories.

Table 6: BIC differences and regime assignments for the two loss weightings. Positive $\Delta\mathrm{BIC}_{k-2}$ values indicate that the two-component model is preferred over the $K \in \{1, 3\}$ component alternative.

Weighting	$\Delta\mathrm{BIC}_{1-2}$	$\Delta\mathrm{BIC}_{3-2}$	Recon. / Dis. Regime
EDM-3	5.548	5.452	13/12
Sech	3.086	2.992	9/16

Stability of Model Selection. To assess sensitivity to GMM initialisation, we repeat model selection 100 times. In each trial, we fit one GMM for every $K \in \{1, 2, 3, 4, 5\}$ using `n_init=1`, with `random_state` set to each value from 0 to 99. We then select the K with the lowest BIC within each trial. For both loss weightings, we see that $K = 2$ has the lowest BIC in Table 7. Here we found that all 100 repeats select $K = 2$ via the metric in Equation (44).

Table 7: **Clustering favours two regimes.** We report the mean ± std BIC over 100 repeats of the J_c^r clustering procedure over $K \in \{1, 2, 3, 4, 5\}$, and find that lowest values for $K = 2$.

Weighting	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
EDM-3	76.05 ± 0.00	70.50 ± 0.00	76.61 ± 1.63	83.25 ± 1.49	87.21 ± 2.95
Sech	75.51 ± 0.00	72.42 ± 0.00	79.31 ± 1.66	85.59 ± 2.19	88.01 ± 1.70

Uncertainty in Individual Assignments. We next assess the ambiguity of the assignments made by the selected $K = 2$ model. For each loss weighting, we use the GMM fitted with `n_init=100` to the 25 values of J_c^r. For each trajectory, the GMM returns a posterior probability for each component, and we assign a run to the component with the larger probability. Table 8 reports summary statistics of the maximum posterior probability across runs. All sech weighted trajectories and 24 of the 25 EDM-3-weighted trajectories have maximum posterior probabilities above 0.90, indicating that the fitted models make largely unambiguous assignments.

Table 8: **Regime assignments for each run have high confidence.** Mean, median, and minimum for the maximum posterior probability across the 25 trajectories. Values closer to one indicate less ambiguous assignments.

Weighting	**Mean**	**Median**	**Minimum**	**Runs below** 0.90
EDM-3	0.995	1.000	0.898	1/25
Sech	0.992	0.999	0.951	0/25

Stability of the Partitions. Finally, we test whether all 100 separately fitted GMMs with `n_init = 1` and $K = 2$ recover the same partition of the runs. Since GMM component labels may be permuted across fits, we measure agreement using the adjusted Rand index (ARI), which is invariant to this ordering. We compute the ARI between the assignments produced by every pair of initialisations, where an ARI of one indicates identical partitions up to label permutation. In Table 9, every pair has an ARI of one, showing that all 100 fits recover the same partition of the 25 runs.

Table 9: **The two-regime partition is stable across initialisations.** Mean and minimum pairwise ARI across the 100 independently initialised $K = 2$ GMMs. An ARI of one denotes identical partitions up to a label permutation.

Weighting	Mean Pairwise ARI	Minimum Pairwise ARI
EDM-3	1.000	1.000
Sech	1.000	1.000

E.3 Analysing Early Regime Selection

For each loss weighting and disentanglement metric, we compute Pearson's r across the 25 runs between the disentanglement at the reference point and its final value. This tests whether a run's early disentanglement is informative of its final disentanglement. As shown in Table 10, all correlations are positive and range from 0.520 to 0.827. These scores suggest that early disentanglement is positively correlated with the final outcome under both weightings and all three metrics in Figures 1 and 10.

Table 10: Pearson's r correlation between disentanglement at the reference point of 576K images seen, marked with a + in Figures 1 and 10, and the final disentanglement for each metric.

Weighting	DCI	FactorVAE	MIG
EDM-3	0.662	0.520	0.827
Sech	0.622	0.781	0.767

E.4 Sensitivity to VQ-VAE Training

The analysis in Figures 1 and 10 uses a VQ-VAE trained for 30K steps, for which the separation between regimes is particularly clear. This setting motivates our hypotheses **H1** and **H2**, which we test through further experiments and use to design our method for steering training trajectories. To assess how this pattern changes with VQ-VAE, we repeat the GMM and correlation analyses with the EDM-3 loss weighting and using the 150K checkpoint employed in our main attribute disentanglement experiments (Section 6.1).

With this VQ-VAE, BIC now favours $K = 1$, aligned with a more continuous spectrum of trajectories rather than two clearly separated regimes. However, final disentanglement continues to vary substantially across runs, and early disentanglement remains positively correlated with its final value (Table 11). The correlations range from 0.485 to 0.776, showing that early disentanglement remains informative of final disentanglement across all three metrics. These results indicate that VQ-VAE training affects how distinctly the trajectory patterns separate, while the broader relationship between early and final representation quality persists. Since our main attribute experiments use the 150K checkpoint and still exhibit trajectory steering, STEERINGDRL does not require the clean regime separation in its base model to be effective. This supports **H1** and **H2** as design hypotheses that remain useful beyond the controlled, motivating diagnostic setting.

Table 11: **Early disentanglement remains informative with a mature VQ-VAE.** Pearson correlations between disentanglement at 576K images seen and its final value across 25 runs using the 150K VQ-VAE checkpoint. We also report the distribution of final scores.

Metric	Pearson's r	Final median [IQR]	Final range
DCI	0.560	0.823 [0.789, 0.891]	0.634–0.951
FactorVAE	0.485	0.929 [0.885, 0.992]	0.680–1.000
MIG	0.776	0.363 [0.274, 0.396]	0.133–0.569

E.5 Cross-Attention Maps in each Regime

We visualise cross-attention maps from the low-capacity U-Net used in Section 4. We use the second downsampling layer in the U-Net encoder, which operates at spatial resolution 8^2, and visualise maps at the same layer and logSNR level for both models. For readability, attention maps are upsampled and smoothed. Figure 11 compares models from the reconstruction and disentanglement regimes. Although both models achieve strong reconstructions, their internal attention patterns differ substantially. In the disentanglement regime, several attention maps are spatially localised and align with all ground-truth factors. In contrast, attention maps in the reconstruction regime are more diffuse and less consistently aligned with individual factors. This supports the interpretation that the two regimes correspond not only to different metric values, but also to different internal representations.

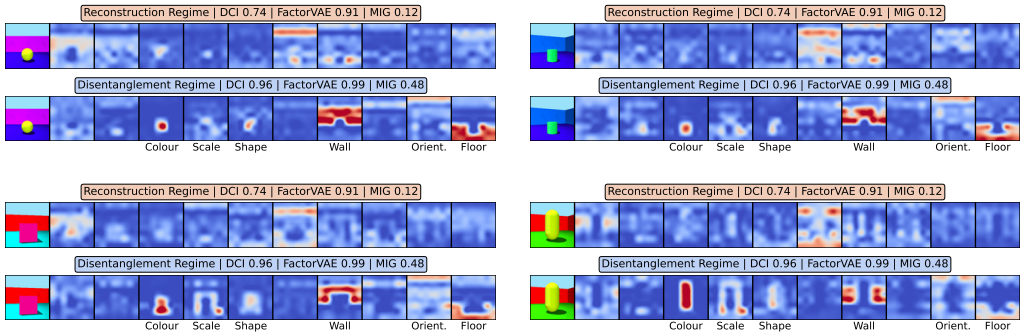


Figure 11: **Cross-attention maps reveal semantic structure in the disentanglement regime.** These are taken from the models used in Figure 1. At the same U-Net layer and logSNR level, attention maps in the disentanglement regime are localised and align with ground-truth factors, but those in the reconstruction regime are diffuse.

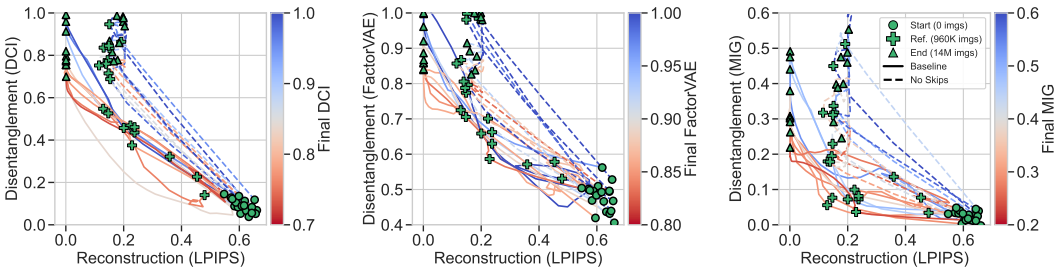
E.6 Skip Connections

Table 12: **Skip pathways trade reconstruction for disentanglement.** We ablate skip connections in the EncDiff U-Net on Shapes3D over 10 seeds. Zeroing skip features removes the encoder-decoder shortcut entirely, substantially improving disentanglement metrics but sharply degrading reconstruction. Skip Dropout provides a softer intervention, preserving reconstruction while giving smaller disentanglement gains. Combining zeroed skips with the $\log \mathrm{SNR}$ curriculum further improves disentanglement, suggesting that restricting shortcut pathways can make the curriculum's trajectory-steering effect more pronounced (akin to Table 1).

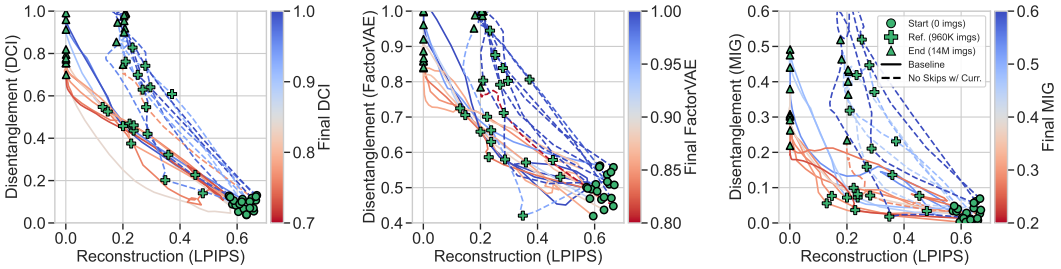
Model	LPIPS ↓	DCI ↑	FactorVAE ↑	MIG ↑
EncDiff U-Net + Split-MLP [89]	0.0025 ± 0.0003	0.828 ± 0.099	0.921 ± 0.093	0.218 ± 0.087
+ $\mathrm{sigmoid}(-\lambda)$*	$\mathbf{0.0020 \pm 0.0001}$	0.932 ± 0.077	0.971 ± 0.056	0.399 ± 0.102
+ $\log \mathrm{SNR}$ Curriculum	0.0020 ± 0.0003	0.853 ± 0.060	0.917 ± 0.063	0.299 ± 0.066
Zeroed Skips + $\mathrm{sigmoid}(-\lambda)$*	0.1573 ± 0.0221	0.905 ± 0.079	0.940 ± 0.062	0.439 ± 0.109
+ $\log \mathrm{SNR}$ Curriculum	0.2029 ± 0.0049	$\mathbf{0.942 \pm 0.064}$	$\mathbf{0.975 \pm 0.059}$	$\mathbf{0.548 \pm 0.161}$
Skip Dropout [33] + $\mathrm{sigmoid}(-\lambda)$	0.0030 ± 0.0003	0.892 ± 0.075	0.954 ± 0.052	0.408 ± 0.088
+ $\log \mathrm{SNR}$ Curriculum	0.0030 ± 0.0002	0.876 ± 0.101	0.935 ± 0.061	0.397 ± 0.123

We test the role of skip connections in unsupervised diffusion representation learning by modifying the skip pathways of the EncDiff U-Net used in our Shapes3D analysis. In the *zeroed skips* setting, we replace the encoder skip features passed to the decoder with zeros, removing the shortcut pathway and forcing the decoder to rely more heavily on the U-Net trunk and conditioning representation. We also compare against Skip Dropout [33], which stochastically drops skip-connection features.

Table 12 shows that removing skip connections substantially worsens reconstruction and has mixed effects on disentanglement. Under the same sigmoid weighting, zeroing skips increases MIG but lowers mean DCI and FactorVAE. Adding the log SNR curriculum to the zeroed-skip model improves all three disentanglement metrics. However, reconstruction degrades sharply, indicating that the skip pathway is useful for image fidelity even when it is harmful for representation learning. Skip



(a) Zeroed skips.



(b) Zeroed skips + $\log \mathrm{SNR}$ curriculum from $[0, 10]$ to $[-5, 12]$ over 32K steps.

Figure 12: **Removing skip connections steers trajectories toward disentanglement.** We follow the setup in Figure 1 and compare the EncDiff baseline against zeroed-skip variants. Zeroing skip connections shifts training trajectories upward toward the disentanglement regime, but also moves them leftward more slowly, reflecting worse reconstruction. Adding the $\log \mathrm{SNR}$ curriculum on top of zeroed skips steers trajectories to the high-disentanglement region, especially for MIG. Together with Table 12, these results support H1: skip connections act as reconstruction-oriented shortcuts, and regulating them can make optimisation rely more on the representation pathway.

Dropout gives a softer version of this effect, preserving reconstruction quality while improving disentanglement, but the gains are smaller and the $\log \mathrm{SNR}$ curriculum does not further improve performance. This mirrors the behaviour of the standard U-Net with curriculum in Table 1, suggesting that when skip connections are present, their reconstruction bias can dominate the curriculum signal.

The trajectory plots in Figure 12 make this behaviour visible. With zeroed skips, trajectories shift toward the disentanglement regime, achieving higher final representation scores but worse LPIPS. Adding the $\log \mathrm{SNR}$ curriculum preserves this shift and further concentrates trajectories in the high-disentanglement region, particularly for MIG. Together, these results support **H1**: skip connections are not merely an implementation detail, but a reconstruction-oriented shortcut that can determine whether optimisation uses the representation pathway or bypasses it.

Zeroed skips + $\log \mathrm{SNR}$ curriculum. The zeroed-skips + $\log \mathrm{SNR}$ curriculum ablation directly probes the interaction between H1 and H2. Removing skip pathways shifts trajectories toward the disentanglement regime, while adding the $\log \mathrm{SNR}$ curriculum shifts them further into the disentanglement region, especially for MIG. This suggests that noise-level exposure can steer optimisation more effectively once reconstruction shortcuts are restricted. Together, this supports our main conclusion: the curriculum selects the regime, while architectural bottlenecks and loss-weighting improve stability within it. Since zeroing skips sharply degrades reconstruction, we instead regulate skip pathways with gated residuals, preserving a weaker reconstruction pathway.

E.7 Early Noise-Level Exposure

Table 13 shows an early-time slice at 320K images, using the 150K VQ-VAE checkpoint. The three bands are chosen to distinguish full noise-level support, removal of high-noise levels, and high-noise only. The full band $[-5, 12]$ gives the strongest early reconstruction among the three settings, reaching LPIPS 0.0348 ± 0.0138, while also achieving competitive disentanglement. Removing high-noise levels with $[0, 12]$ slows reconstruction, increasing LPIPS to 0.1721 ± 0.0413, but does not collapse the representation: DCI and FactorVAE decrease moderately, while MIG remains comparable to

the full band. Conversely, the high-noise band $[-5, 0]$ gives the worst reconstruction, with LPIPS 0.2929 ± 0.0021, but the strongest early disentanglement scores. These results show that early noise-level exposure changes the relative pace of reconstruction and semantic organisation: full support accelerates reconstruction, whereas restricted bands preserve or improve early disentanglement despite worse image fidelity. This supports **H2**: noise-level exposure changes the path through reconstruction-disentanglement space, and may therefore influence regime selection.

Table 13: **Representative early fixed-band dynamics used to support H2.** Metrics are measured at 320K images over 10 seeds using the 150K VQ-VAE checkpoint. The selected bands compare full support, removing high-noise levels, and using only high-noise levels.

Description	$\lambda_{\min}$	$\lambda_{\max}$	LPIPS↓	DCI↑	FactorVAE↑	MIG↑
Full band	-5	12	0.0348 ± 0.0138	0.614 ± 0.053	0.716 ± 0.122	0.100 ± 0.049
No high-noise	0	12	0.1721 ± 0.0413	0.508 ± 0.073	0.673 ± 0.078	0.118 ± 0.073
Only high-noise	-5	0	0.2929 ± 0.0021	0.635 ± 0.070	0.786 ± 0.065	0.184 ± 0.063

E.8 Noise-Level Exposure and VQ-VAE Effects

To isolate the role of noise-level support, we train the toy diffusion autoencoder with fixed $\log \mathrm{SNR}$ bands under the $\mathrm{sigmoid}(-\lambda)$ loss-weighting used in our main experiments. We report results with two VQ-VAE checkpoints: an earlier 30K-step checkpoint used for the trajectory analysis in Section 4, and a later 150K-step checkpoint used to more closely follow EncDiff and obtain stronger reconstruction. The 30K VQ-VAE produces the clearest regime separation, while the 150K VQ-VAE improves reconstruction and makes the regime separation less pronounced.

The fixed-band ablations show that reconstruction is highly sensitive to the high-noise part of the objective, whereas disentanglement is less tied to full noise-level support. For the 30K VQ-VAE, removing high-noise levels worsens LPIPS from 0.0177 to 0.4044, but disentanglement does not collapse: FactorVAE and MIG increase, while DCI decreases. The strongest disentanglement for this checkpoint comes from restricted bands, with $[-5, 5]$ giving the best DCI and FactorVAE, and $[0, 12]$ giving the best MIG. However, using only high-noise levels $[-5, 0]$ performs poorly across all disentanglement metrics, suggesting that high-noise exposure alone is insufficient to organise factors in $\mathbf{z}$. For the 150K VQ-VAE, removing high-noise levels again worsens reconstruction, increasing LPIPS from 0.0017 to 0.0671, and using only high-noise levels gives very poor LPIPS (0.2975). Disentanglement varies less sharply for this later checkpoint, but restricted bands remain competitive with the full band, with several bands improving FactorVAE or MIG despite worse reconstruction. Thus, removing high-noise levels substantially worsens reconstruction, while some restricted bands achieve competitive or improved disentanglement.

Table 14: Fixed logSNR band ablations (VQ-VAE trained for 150K steps, Diffusion Autoencoder sees 3M images, averaged over 10 seeds).

Description	$\lambda_{\min}$	$\lambda_{\max}$	LPIPS ↓	DCI ↑	FactorVAE ↑	MIG ↑
Full band	-5	12	0.0017 ± 0.0007	0.744 ± 0.093	0.814 ± 0.085	0.186 ± 0.085
No high-noise	0	12	0.0671 ± 0.0167	0.703 ± 0.117	0.793 ± 0.120	0.231 ± 0.123
No low-noise	-5	10	0.0015 ± 0.0002	0.684 ± 0.041	0.789 ± 0.045	0.176 ± 0.045
No extremes	0	10	0.0614 ± 0.0182	0.716 ± 0.071	0.825 ± 0.082	0.231 ± 0.109
Middle band	-5	5	0.0021 ± 0.0001	0.722 ± 0.064	0.829 ± 0.107	0.187 ± 0.069
Only high-noise	-5	0	0.2975 ± 0.0006	0.739 ± 0.097	0.859 ± 0.096	0.231 ± 0.071

Table 15: Fixed logSNR band ablations (VQ-VAE trained for 30K steps, Diffusion Autoencoder sees 3M images, averaged over 10 seeds).

Description	$\lambda_{\min}$	$\lambda_{\max}$	LPIPS ↓	DCI ↑	FactorVAE ↑	MIG ↑
Full band*	-5	12	0.0177 ± 0.0399	0.750 ± 0.102	0.868 ± 0.088	0.268 ± 0.121
No high-noise*	0	12	0.4044 ± 0.0725	0.665 ± 0.202	0.892 ± 0.153	0.381 ± 0.157
No low-noise	-5	10	0.0054 ± 0.0012	0.790 ± 0.105	0.872 ± 0.122	0.283 ± 0.139
No extremes	0	10	0.3938 ± 0.0892	0.694 ± 0.158	0.917 ± 0.112	0.358 ± 0.147
Middle band	-5	5	0.0063 ± 0.0011	0.834 ± 0.118	0.926 ± 0.085	0.313 ± 0.134
Only high-noise	-5	0	0.2021 ± 0.0006	0.641 ± 0.069	0.778 ± 0.083	0.166 ± 0.059

These results motivate our curriculum. Rather than removing reconstruction-favouring noise levels entirely, we delay exposure to them. This gives the representation time to organise under a bounded noise range before the model sees the full reconstruction objective. The VQ-VAE checkpoint also matters: as the VQ-VAE is trained longer, reconstruction improves, but disentanglement becomes less separated across bands. One possible explanation is that a more mature VQ-VAE encodes finer visual detail directly in the latent space, making reconstruction easier and reducing the pressure to organise information through $\mathbf{z}$. We therefore use the 30K VQ-VAE to study regime emergence, where the effect is most visible, and the 150K VQ-VAE for experiments that more closely follow EncDiff.

Table 16: Relative change when training the VQ-VAE longer (150K vs 30K). Values are $(150\text{K} - 30\text{K})/30\text{K}$.

Description	$\lambda_{\min}$	$\lambda_{\max}$	Δ LPIPS $\downarrow$	Δ DCI $\uparrow$	Δ FactorVAE $\uparrow$	Δ MIG $\uparrow$
Full band	-5	12	-0.904	-0.008	-0.062	-0.306
No high-noise	0	12	-0.834	$+0.057$	-0.111	-0.394
No low-noise	-5	10	-0.722	-0.134	-0.095	-0.378
No extremes	0	10	-0.844	$+0.032$	-0.100	-0.355
Middle band	-5	5	-0.667	-0.134	-0.105	-0.403
Only high-noise	-5	0	$+0.472$	$+0.153$	$+0.104$	$+0.392$

E.9 $\log \mathrm{SNR}$-curriculum

We ablate the initial $\log \mathrm{SNR}$ band and widening duration used by the curricula in Figure 4. All settings use the same gated residual U-Net and $\mathrm{sigmoid}(-\lambda)$ loss-weighting, and widen to the same final band $[-5, 12]$. Thus, the ablation isolates how the early noise-level support and the speed of widening affect the training trajectory.

Table 17 shows that all curricula reach similar final reconstruction, with LPIPS saturated at approximately 0.0005. The main differences are therefore in representation quality. With a fixed widening duration of 32K, changing the initial band changes the final disentanglement scores: the high-noise initial band $[-5, 0]$ gives strong DCI and FactorVAE, while the moderate band $[0, 10]$ gives the best overall combination once the widening duration is tuned. For the $[0, 10]$ initial band, widening over 32K steps provides the best balance across disentanglement metrics, as seen by the Macro Mean and Macro SD, whereas faster widening (8K) and slower widening (64K) both reduce at least one disentanglement metric. This suggests that curriculum design matters beyond simply delaying access to high or low-noise levels; the rate at which the full objective is introduced also affects the representation learned.

Figure 13 shows the corresponding training trajectories for DCI, FactorVAE, and MIG. The trajectories reveal that curricula change the path through reconstruction-disentanglement space before final convergence. In particular, the best-performing curriculum follows a different trajectory, delaying early commitment to a reconstruction-dominated path and reaching high final disentanglement with lower variance across seeds. These results support **H2**: early noise-level exposure can steer regime selection by changing the relative pace of reconstruction and semantic organisation.

Table 17: **Curriculum ablations on Shapes3D.** We vary the initial $\log \mathrm{SNR}$ band $[\lambda_L(0), \lambda_U(0)]$ and widening duration $m_{\max}$, keeping the final band fixed to $[\lambda_{\min}, \lambda_{\max}] = [-5, 12]$. Metrics are reported after 15M images over 10 seeds. Macro Mean and Macro SD average the means and standard deviations across DCI, FactorVAE, and MIG, respectively.

$\lambda_L(0)$	$\lambda_U(0)$	$m_{\max}$	LPIPS ↓	DCI ↑	FactorVAE ↑	MIG ↑	Macro Mean ↑	Macro SD ↓
0	12	32K	0.0005 ± 0.0000	0.834 ± 0.065	0.935 ± 0.045	0.329 ± 0.097	0.699	0.069
2	8	32K	0.0005 ± 0.0000	0.860 ± 0.100	0.941 ± 0.068	0.290 ± 0.097	0.697	0.088
-5	0	32K	0.0005 ± 0.0000	0.914 ± 0.096	0.976 ± 0.039	0.348 ± 0.090	0.746	0.075
0	10	8K	0.0005 ± 0.0000	0.883 ± 0.063	0.968 ± 0.050	$\mathbf{0.375 \pm 0.146}$	0.742	0.086
0	10	32K	0.0005 ± 0.0000	$\mathbf{0.926 \pm 0.043}$	$\mathbf{0.991 \pm 0.017}$	0.353 ± 0.056	**0.757**	**0.039**
0	10	64K	0.0005 ± 0.0000	0.864 ± 0.038	0.954 ± 0.046	0.266 ± 0.099	0.695	0.061

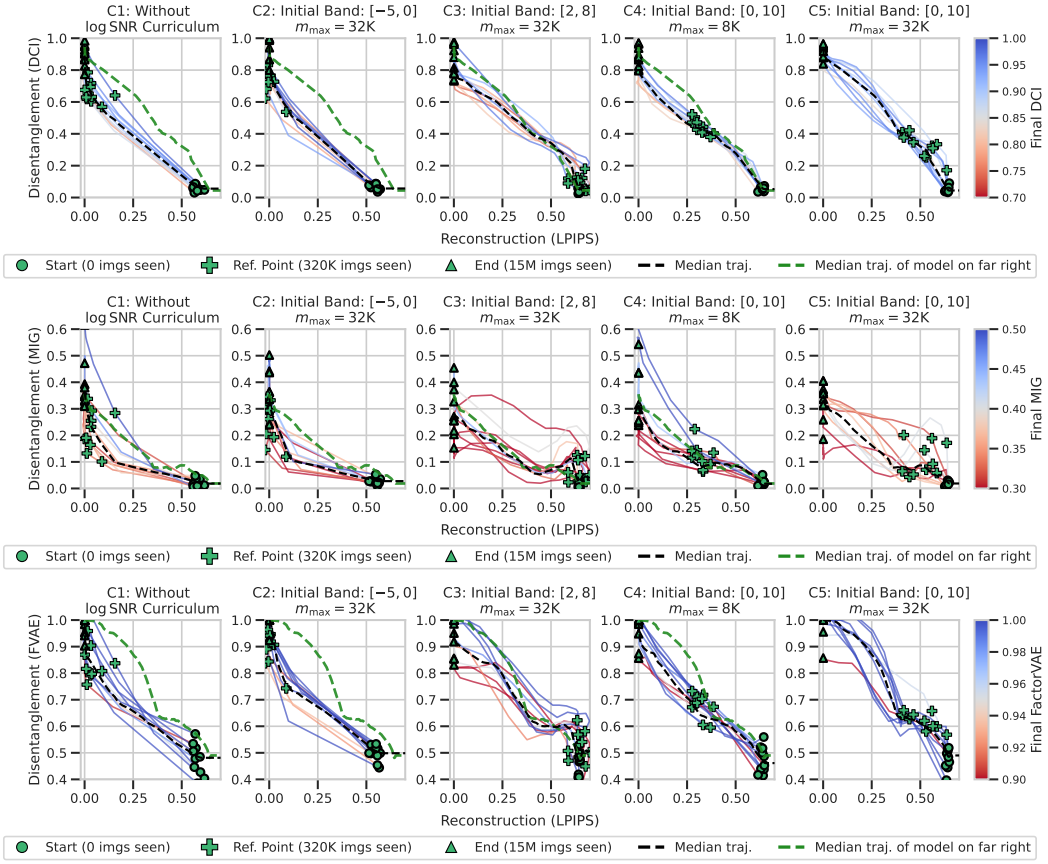
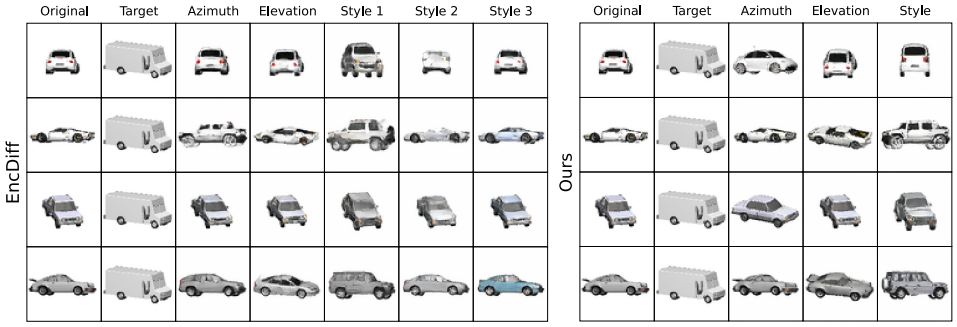
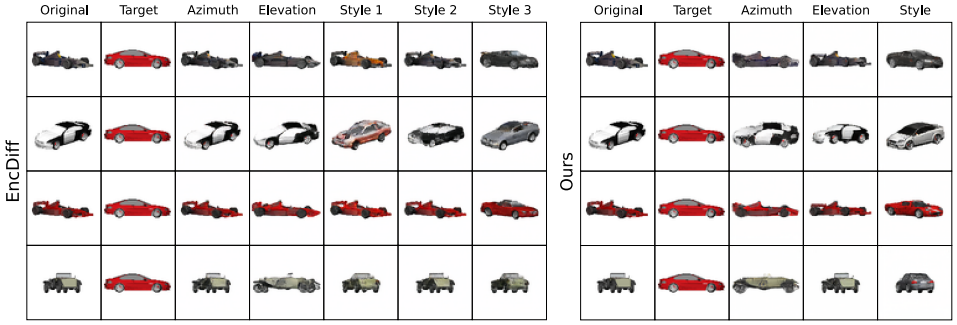


Figure 13: $\log \mathrm{SNR}$ **curricula steer training trajectories across disentanglement metrics.** Training trajectories for curriculum ablations on Shapes3D, using DCI, FactorVAE, and MIG as the disentanglement axis. Panels correspond to different curricula; colours indicate final disentanglement, and dashed curves show median trajectories. Changing the initial band and widening duration changes the path through reconstruction–disentanglement space, not just the final score. The moderate curriculum $[0, 10]$ with $m_{\max} = 32\mathrm{K}$ gives the strongest overall disentanglement, supporting curriculum design as a mechanism for regime selection.

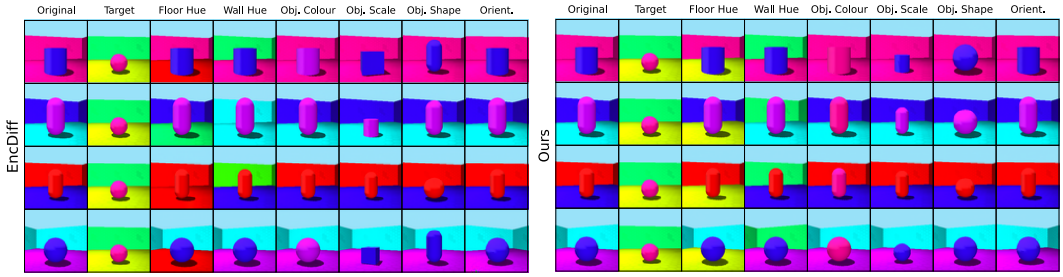
E.10 Attribute Disentanglement



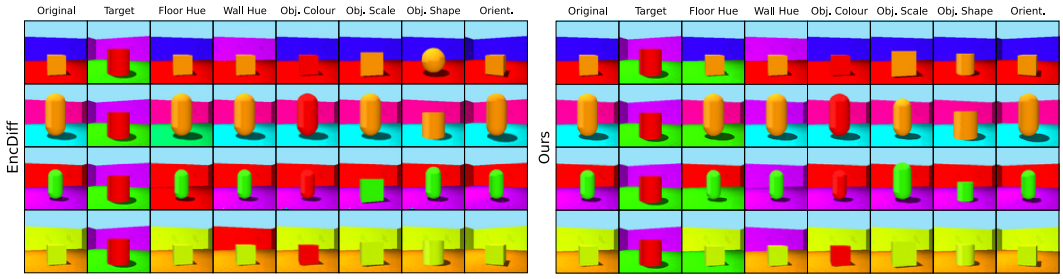
(a) Cars3D example 1



(b) Cars3D example 2



(c) Shapes3D example 1

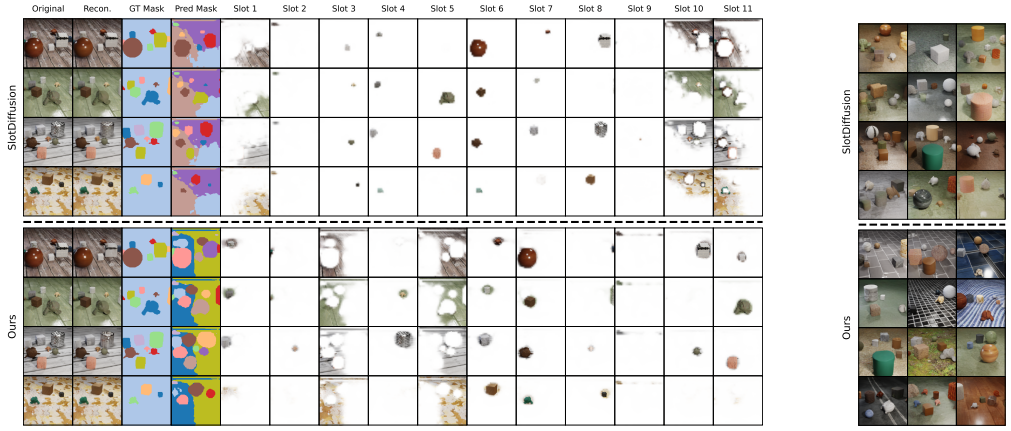


(d) Shapes3D example 2

Figure 14: **Latent edits reveal interpretable factors across datasets.** Each panel compares EncDiff and STEERINGDRL by replacing a single source slot with the corresponding target slot. On Cars3D, edits capture viewpoint (`Azimuth`, `Elevation`) and object identity (`Object Type / Style`). On Shapes3D with $N = 10$, we can more consistently perform all edits, whereas EncDiff often struggles with `Floor Hue` and `Object Shape`.

E.11 Spatial Disentanglement with Object-Centric Learning

E.11.1 ClevrTex



(a) Segmentation and slot decomposition (b) Random compositions

Figure 15: **STEERINGDRL produces more spatially coherent and compositional slot representations on ClevrTex than SlotDiffusion.** (a) Reading down each column, a slot captures objects in similar spatial locations across scenes, with fewer background slots compared to SlotDiffusion. (b) Fewer background slots reduce fragmentation when slots are shuffled for random composition.

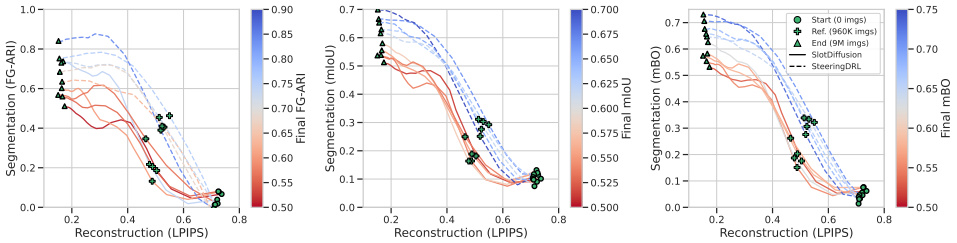


Figure 16: **ClevrTex training trajectories across segmentation metrics.** We plot reconstruction–segmentation trajectories for FG-ARI, mIoU and mBO using ResNet34 features. STEERINGDRL learns semantic slot structure earlier than SlotDiffusion and converges to higher segmentation scores.

Table 18: **ClevrTex results with ResNet18 features.** Here we use ResNet18 for $\text{Feat}_\psi(\cdot)$. Methods are run under the same data split and evaluation protocol over 5 seeds.

	Segmentation			Recon.	Comp.
Method	**FG-ARI ↑**	**mIoU ↑**	**mBO ↑**	**LPIPS ↓**	**FID ↓**
SLATE	62.75 ± 5.89	56.41 ± 4.02	60.23 ± 4.01	0.402 ± 0.005	88.73 ± 5.37
SlotDiffusion	66.85 ± 3.22	56.01 ± 1.98	60.20 ± 2.34	0.173 ± 0.015	47.84 ± 1.36
STEERINGDRL	$\mathbf{67.82 \pm 1.40}$	$\mathbf{59.20 \pm 1.91}$	$\mathbf{63.29 \pm 1.87}$	$\mathbf{0.164 \pm 0.004}$	$\mathbf{44.01 \pm 1.85}$

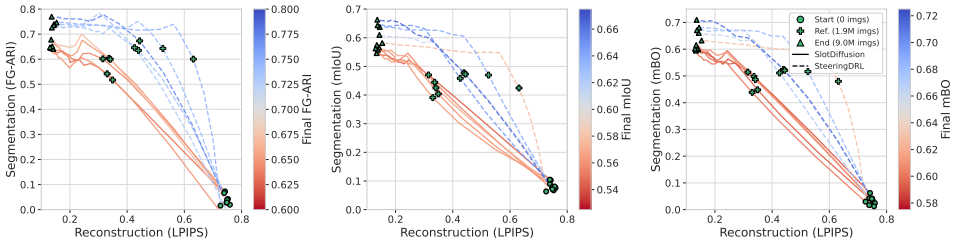


Figure 17: **ClevrTex training trajectories with ResNet18 features.** STEERINGDRL still improves segmentation faster and converges to higher scores than SlotDiffusion.

E.11.2 PascalVOC

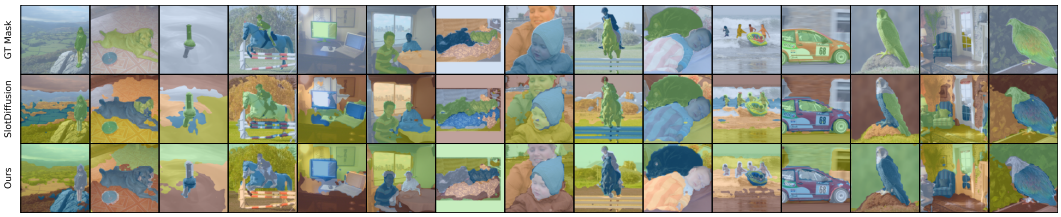


Figure 18: **PascalVOC qualitative masks with pretrained DINO features.** STEERINGDRL better exploits the semantic features in DINO to segment dominant objects, whereas SlotDiffusion often merges objects with the background or nearby regions. However, both methods still exhibit object-part segmentation, such as separating the head and body of birds, suggesting that further gains may require explicit semantic alignment as in GLASS [76].

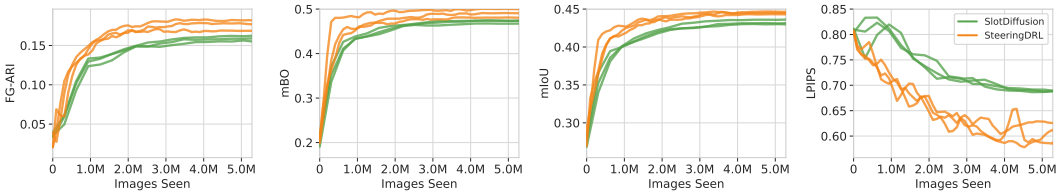


Figure 19: **Individual PascalVOC trajectories show faster and stronger convergence.** We plot each metric against training progress for SlotDiffusion and STEERINGDRL. With the same pretrained $\mathrm{Feat}_\psi(\cdot)$, STEERINGDRL improves FG-ARI, mBO and mIoU earlier in training, while also reducing LPIPS faster. This supports the view that, with pretrained semantic features, our inductive biases improve the efficiency of object-centric binding. While we do improve reconstruction metrics, the visualisations remain poor, and not comparable to larger model such as StableLSD [32] or GLASS [76], which use higher quality VQ-VAEs.

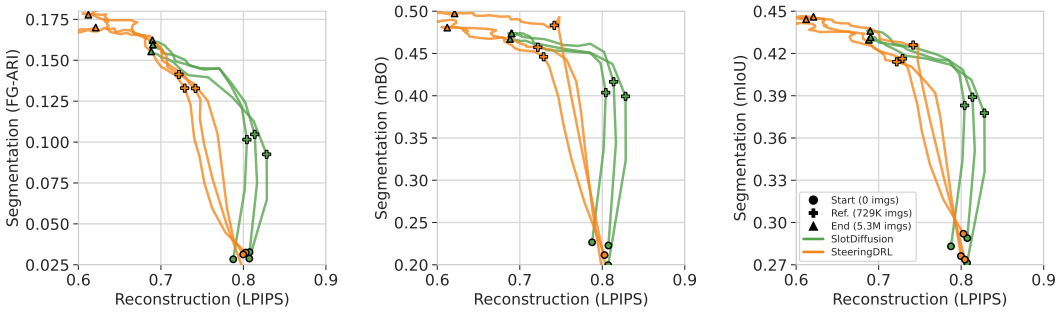


Figure 20: **Training trajectories on PascalVOC differ from from-scratch OCL.** We plot reconstruction-segmentation trajectories for FG-ARI, mBO and mIoU on PascalVOC. Unlike Clevr-Tex, where representations must emerge from scratch, PascalVOC uses pretrained DINO features for $\mathrm{Feat}_\psi(\cdot)$. STEERINGDRL exploits these pretrained semantics more quickly than SlotDiffusion, reaching higher segmentation scores and better reconstruction earlier in training.

E.11.3 COCO

Table 19: **Preliminary COCO results follow the trends observed on Pascal VOC.** For spatial disentanglement with object-centric learning, we retain the matched SlotDiffusion baseline's training and evaluation setup and swap in the architectural and curriculum components from Section 5. STEERINGDRL accelerates improvements in segmentation (FG-ARI) and reconstruction (LPIPS), while mIoU and mBO show early gains before remaining closely matched to SlotDiffusion. Values are reported as mean ± standard deviation over 3 seeds; the best result at each checkpoint is bold.

Step	Method	FG-ARI ↑	mIoU ↑	mBO ↑	LPIPS ↓
20K	SlotDiffusion	0.282 ± 0.013	0.287 ± 0.003	0.293 ± 0.005	0.863 ± 0.010
20K	STEERINGDRL	$\mathbf{0.321 \pm 0.012}$	$\mathbf{0.305 \pm 0.001}$	$\mathbf{0.304 \pm 0.002}$	$\mathbf{0.751 \pm 0.010}$
40K	SlotDiffusion	0.323 ± 0.009	0.306 ± 0.001	0.306 ± 0.003	0.824 ± 0.024
40K	STEERINGDRL	$\mathbf{0.352 \pm 0.006}$	$\mathbf{0.313 \pm 0.003}$	$\mathbf{0.313 \pm 0.003}$	$\mathbf{0.730 \pm 0.010}$
60K	SlotDiffusion	0.347 ± 0.006	$\mathbf{0.314 \pm 0.002}$	$\mathbf{0.315 \pm 0.002}$	0.786 ± 0.010
60K	STEERINGDRL	$\mathbf{0.360 \pm 0.012}$	$\mathbf{0.314 \pm 0.002}$	0.313 ± 0.003	$\mathbf{0.706 \pm 0.001}$
80K	SlotDiffusion	0.349 ± 0.001	0.313 ± 0.001	0.315 ± 0.004	0.790 ± 0.019
80K	STEERINGDRL	$\mathbf{0.374 \pm 0.008}$	$\mathbf{0.317 \pm 0.001}$	$\mathbf{0.317 \pm 0.002}$	$\mathbf{0.688 \pm 0.005}$
100K	SlotDiffusion	0.357 ± 0.004	$\mathbf{0.319 \pm 0.002}$	$\mathbf{0.320 \pm 0.002}$	0.772 ± 0.010
100K	STEERINGDRL	$\mathbf{0.381 \pm 0.007}$	0.318 ± 0.001	0.318 ± 0.001	$\mathbf{0.681 \pm 0.008}$