Unsupervised Decomposition with Recombination-Consistent Diffusion Models

Archer Wang [1 2] **Emile Anand** [3] **Yilun Du** [4 5] **Marin Soljacic** [1 6]

Abstract

Decomposing complex data into factorized representations can reveal reusable components and enable synthesizing new samples via component recombination. We study this problem in diffusion-based models that learn factorized latent spaces without factor-level supervision. Existing compositional diffusion methods optimize reconstruction under architectural bottlenecks and rely on recombination quality emerging as a byproduct; the training objective never directly evaluates recombined samples. We introduce a *recombination-consistency objective*: a discriminator distinguishes single-source generations from generations produced by recombining latent factors across sources, providing a fully unsupervised signal that directly regularizes recombination outcomes. The objective is architecture-agnostic and composes with existing factorized decoders without modifying inference. Across CelebA-HQ, Virtual KITTI, CLEVR, and Falcor3D, our method improves recombination quality over reimplementations of prior baselines, achieving lower FID and stronger disentanglement (MIG, MCC). Applied to robotic videos on LIBERO, recombining learned action components yields diverse rollouts that significantly increase state-space coverage for exploration.

1. Introduction

Representation learning and generative modeling form two complementary pillars of machine learning. On one side, representation learning seeks to encode complex data into lower-dimensional latent spaces that capture the essential factors of variation (Bengio et al., 2014; Gidaris et al., 2018); on the other, generative modeling maps latent codes back into the data space. Their intersection has given rise to powerful frameworks such as variational auto-encoders, which simultaneously learn a latent representation z and a conditional likelihood $p(x|z)$ (Kingma & Welling, 2013). VAEs have spurred significant work on disentanglement, the goal of recovering latent factors that align with meaningful, independent attributes of the data (Huang et al., 2018; Kim & Mnih, 2018; Chen et al., 2016).

However, fully unsupervised disentanglement is provably impossible without additional assumptions (Locatello et al., 2019), and numerous efforts have explored how to incorporate suitable inductive biases. Beyond classic VAEs, recent research highlights the importance of *compositional* structure in data generation. In many domains, particularly vision, observed data arise from combinations of discrete or continuous factors (objects, attributes, positions, etc.) (Misino et al., 2022; Sehgal et al., 2024). For instance, in a dataset of boats drifting along a river with trees in the background, a compositional model can discover components that capture the boat, the water/flow, and the scenery without explicit supervision; using these components, it can recombine factors across inputs to synthesize new realistic scenes of the same type. We focus on datasets where a finite set of factors explains most variation, such as faces with different hairstyles or expressions, or simple scenes of colored blocks. Exploiting this structure can yield stronger generalization through recombination of components from different data points, and learning representations that reflect the generative process is useful for downstream tasks and for building better generative models (Schölkopf et al., 2021; Squires et al., 2023; Bansal et al., 2024). Latent recombination is both useful and revealing: coherent recombinations suggest modular factors, while artifacts indicate entanglement.

Contributions. We make recombination quality the training target for compositional diffusion models. Prior methods optimize reconstruction under architectural bottlenecks and leave recombination quality to emerge implicitly; we instead train for it directly through a discriminator that distinguishes single-source from recombined generations. The signal acts on generation outcomes rather than architecture or infer-

[1]Research Laboratory of Electronics, MIT, Cambridge, MA 02139, USA [2]NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Cambridge, MA 02139, USA [3]School of Computer Science, Georgia Institute of Technology, Atlanta, USA [4]Kempner Institute, Harvard University, Cambridge, MA, USA [5]Google DeepMind, Mountain View, CA, USA [6]Department of Physics, Massachusetts Institute of Technology, MIT, Cambridge, MA, USA. Correspondence to: Archer Wang <archerdw.@mit.edu>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

ence, allowing a simple integration with existing factorized decoders. We complement the method with a theoretical analysis: closure under coordinate-wise recombination implies a Cartesian-product structure on the latent support, and random-mask recombination contracts pairwise mutual information between latent components. Empirically, our objective improves recombination quality and disentanglement across CelebA-HQ, Virtual KITTI, CLEVR, and Falcor3D against reimplemented baselines. Extended to robotic video, it produces diverse rollouts via recombined action components and increases state-space coverage for exploration on LIBERO, evaluated through action-grounded metrics.

2. Preliminaries

Compositional modeling can be learned through $p_\theta(x) \propto \prod_i p_\theta^i(x)$. Given a narrow slice of training, this compositional formulation allows us to generalize to unseen combinations of variables (Du & Kaelbling, 2024), and this factorization can be traced back to the field of probabilistic graphical modeling (Koller & Friedman, 2009). Given an input image or video of a scene, we model it by encoding it into several low-dimensional latent components, and we encourage each component to capture a different concept so that, together, they can be used to generate or score the input. Du & Mordatch (2019) showed how sampling through the joint distribution on all latents is represented by generation on an energy-based model (EBM) that is the sum of each conditional EBM, but this required supervision, as labels were used to represent concepts.

Learning compositional visual representations for images with diffusion models. Diffusion-based models have emerged as the state-of-the-art generative models for images (Ho et al., 2020; Song et al., 2020), and recent works have investigated using this class of generative models to learn representations (Song et al., 2021; Liu et al., 2022; Netanyahu et al., 2024). Previous works have also explored their use in compositional generation, such as combining visual concepts (Liu et al., 2021; Du et al., 2023a; Janner et al., 2022; Park et al., 2020) and robotic skills (Ajay et al., 2023; Zhou et al., 2024). The most relevant work is **Decomp Diffusion** (Su et al., 2024), an unsupervised method that, given an image, infers a set of latent components that condition a shared diffusion denoiser. We build on Decomp Diffusion as a strong baseline for learning compositional components with diffusion models, and study how additional feedback during training can improve recombination quality and representation structure.

Recent work has shown that using video generation to guide robotic control policy achieves great success (Du et al., 2023b; Ko et al., 2024; Shen et al., 2026; Fu et al., 2026; Ma et al., 2026). We conduct an additional experiment which focuses on videos of task demonstrations by robotics, and we study how good representations lead to generated recombined videos with better physical realism and also lead to the agent exploring its environment much more effectively with a pre-trained policy (Gaujac et al., 2021; Luo & Du, 2024; Chi et al., 2023; Zhang et al., 2024).

Relation to latent editing and concept manipulation. StyleGAN latent arithmetic and recent diffusion-based editing methods address a distinct problem from ours. StyleGAN-based approaches manipulate pretrained GANs through interpolation or semantic directions discovered post-hoc (Karras et al., 2019; Härkönen et al., 2020), while Textual Inversion, Concept Sliders, and instruction-guided diffusion editing rely on concepts specified through text, exemplars, or learned embeddings (Gal et al., 2023; Gandikota et al., 2024). These methods perform single-image edits along a known direction and assume the target concept is provided; recombination or editing quality is expected to emerge from latent smoothness or the supervision channel. By contrast, we discover factors without labels, prompts, or predefined directions, and train directly on the outcome of cross-source recombination via a discriminator. The central question in our setting is not whether a known concept can be edited in a pretrained model, but whether a model can discover factors that recombine into coherent samples.

Disentanglement and feedback signals. Disentanglement aims to represent data in terms of latent factors that can be independently manipulated, so that recombining components yields novel yet coherent samples (Denton & Fergus, 2018; Brooks et al., 2023). This property is closely tied to compositional generalization: if each latent encodes a distinct generative mechanism, then swapping one component should behave like an intervention on that mechanism while keeping the rest fixed. Traditional unsupervised disentanglement methods encourage factorial latents via regularization (e.g., β-VAE; FactorVAE); however, fully unsupervised disentanglement is ill-posed without additional assumptions: many different latent parameterizations can explain the same observational distribution (Higgins et al., 2017; Kim & Mnih, 2018; Locatello et al., 2019; Black et al., 2024). In practice, identifying the "right" compositional structure often requires external feedback that reveals whether a recombination corresponds to a plausible intervention. Complementary lines of work study identifiability through structure and interventions, showing that interventional data can recover meaningful factors under suitable conditions (Squires et al., 2023; Schölkopf et al., 2021). Feedback can be incorporated by fine-tuning diffusion models with reward gradients (Wang et al., 2025; Zekri & Boulle, 2025). ReFL (Xu et al., 2023) evaluates a reward on the one-step clean prediction $\hat{x}_0$ at a randomly chosen denoising step t (rather than only on the final sample), while DRaFT (Clark et al., 2024) backpropagates reward gradients through the

sampling procedure. Our feedback mechanism is inspired by the ReFL-style use of intermediate $\hat{x}_0$ predictions.

Recent diffusion-based approaches can be viewed as replacing explicit energy minimization with denoising dynamics that implicitly follow additive energy gradients, yielding a more stable and scalable mechanism for compositional generation (Wu et al., 2023; Kirilenko et al., 2024; Zhu et al., 2024). Decomp Diffusion links EBM composition to diffusion by decomposing the denoiser's prediction into an average of factor-specific contributions. Concretely, it mirrors the EBM decomposition $E_\theta(x) = \sum_i E_\theta(x; z_i)$ with $\epsilon_\theta(x_t, t) = \frac{1}{K}\sum_{k=1}^{K} \epsilon_\theta(x_t, t, z_k)$, where z_k are learned latent factors, K is the number of factors, and ϵ_θ denotes the denoising network. In their best performing model, the authors demonstrate a noticeable improvement in disentanglement compared to β-VAE (Khemakhem et al., 2020; Higgins et al., 2017; Burgess et al., 2018), InfoGAN (Chen et al., 2016), GENESIS-V2 (Engelcke et al., 2021), and COMET (Du et al., 2021). We provide a more detailed exposition of COMET and Decomp Diffusion in Appendix A.

3. Compositional Generation with Discriminator

Without explicit interventions during training, unsupervised generative models provide no control over the structure of their learned representations or over the behavior of latent recombination. Lachapelle et al. (2023) argue that for additive models, recombined samples may not always lie on the manifold of reasonable observations, and a theoretical characterization of these models is also studied in Wiedemer et al. (2023) for the supervised setting. We adopt this perspective, but replace expensive environment feedback with a scalable, fully unsupervised surrogate. We use a convolutional discriminator to distinguish single-source from recombined predictions, providing a learned signal that penalizes incoherent composition in recombined samples.

We train the discriminator to distinguish single-source generations from recombined generations, and use its gradients to refine the encoder/generator so that recombinations become indistinguishable from valid samples. In other words, recombination provides a probe of whether factors are independently manipulable, and discriminator feedback supplies a practical signal for encouraging compositional structure. A key challenge in compositional generative modeling is balancing compositional generalization with the plausibility of newly composed samples (Bagautdinov et al., 2018). Naively recombining latent factors often yields out-of-distribution generations, whereas overly strong constraints can collapse diversity. Our method introduces a tunable discriminator-driven feedback signal that selectively penalizes implausible recompositions while preserving compositional diversity.

Factorized Generative Model. Let $\mathcal{X}$ denote the data space (e.g. images or videos) and $\mathcal{M} \subset \mathcal{X}$ denote the data manifold corresponding to the support of the true distribution p_{data}. We assume each observation $x \in \mathcal{X}$ is generated by a factorized latent representation $z = (z_1, \dots, z_K)$, where each $z_i \in \mathcal{Z}_i$ corresponds to a distinct generative factor (e.g. object shape, position, color). The generative model is then described by: $x \sim p_\theta(\cdot \mid z), z \sim p(z) = \prod_{i=1}^{K} p(z_i)$, and induces a model distribution $p_\theta(x)$ over $\mathcal{X}$. During training, we learn both an encoder $\text{Enc}_\phi : \mathcal{X} \to \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ and a conditional distribution $p_\theta(\cdot|z) : \mathcal{Z} \to \Delta(\mathcal{X})$ by minimizing a reconstruction-based diffusion loss $\mathcal{L}_{\text{MSE}}(\theta, \phi)$.

Compositional Recombination. At inference time, we construct *compositional* latent codes by recombining factors from multiple inputs $\tilde{z} = S \odot z^{(A)} + (1 - S) \odot z^{(B)}$, for $S \in \{0, 1\}^K$ where $\odot$ denotes element-wise multiplication and $z^{(A)}, z^{(B)}$ are latent representations of two source samples and S is a binary mask indicating which factors to swap. However, the resulting generation $\tilde{x} \sim p_\theta(\cdot|\tilde{z})$ may correspond to a combination of factors that was never observed jointly in p_{data}, and hence $\tilde{x}$ may lie off the manifold $\mathcal{M}$ which might lead to unrealistic recombinations (Montero et al., 2022).

Discriminator Feedback To discourage unrealistic recombinations, we introduce a recombination discriminator $D_\psi : \mathcal{X} \to [0, 1]$ trained to distinguish single-source generations from recombined generations. Ideally, single-source samples are obtained by decoding latents inferred from a single data point and recombined samples are constructed by mixing latent factors across multiple sources and decoding them.

Formally, given two data points x^A and x^B and timestep t, let $\hat{x}^{\text{single}} = G_\theta(x_t^A, t, z_{1:K})$ and $\hat{x}^{\text{recomb}} = G_\theta(x_t^A, t, \tilde{z}_{1:K})$, where denoising predictions come from contributions associated with each latent, we have $G_\theta(x, t, z_{1:K}) = \frac{1}{K}\sum_{j=1}^{K} G_\theta(x, t, z_j)$. We introduce a discriminator D_ψ that distinguishes single-source generations from recombined ones. The discriminator is trained by minimizing the classification loss

$$\mathcal{L}_{\text{clf}}(\psi) = -\mathbb{E}\left[\log D_\psi(\hat{x}^{\text{single}}) + \log\left(1 - D_\psi(\hat{x}^{\text{recomb}})\right)\right].$$

The generator and encoder are trained adversarially to make recombined samples indistinguishable from single-source samples by minimizing

$$\mathcal{L}_{\text{adv}}(\theta, \phi; \psi) = -\mathbb{E}\left[\log D_\psi(\hat{x}^{\text{recomb}})\right].$$

We train the model using alternating minimization. The generator and encoder are optimized by minimizing

$$\mathcal{L}_{\text{gen}}(\theta, \phi) = \mathcal{L}_{\text{MSE}}(\theta, \phi) + \lambda \mathcal{L}_{\text{adv}}(\theta, \phi; \psi),$$

while the discriminator is trained to minimize $\mathcal{L}_{\text{clf}}(\psi)$. Here $\mathcal{L}_{\text{MSE}}$ denotes the standard diffusion reconstruction loss

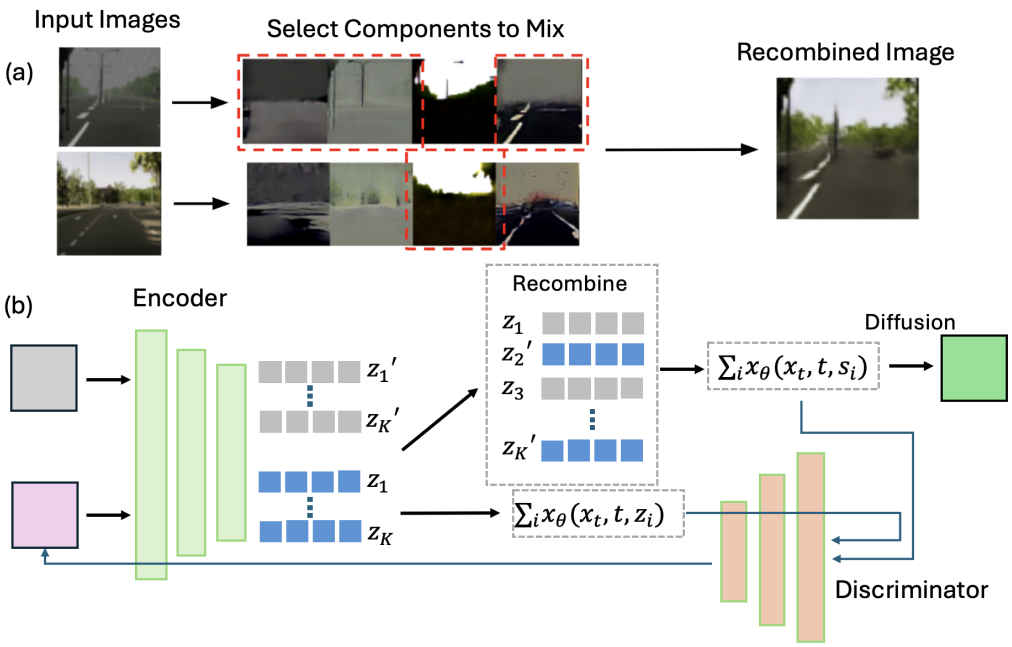


Figure 1. (a) We encode two input images into $K = 4$ latent components. The middle panels show component-wise reconstructions: for each k, we decode using only z_k, with all other components set to zero. Red dashed boxes mark which components are chosen from each source to form a hybrid latent code $\tilde{z}$. Decoding $\tilde{z}$ yields the final image (right), which merges appearance/scene attributes from both sources. (b) During training, the discriminator learns to distinguish between predictions from latents from a single input and latents from multiple sources, while the model attempts to fool the discriminator. At inference, recombined latents are sampled via the standard diffusion denoising process.

used to train the base model. We discuss the computational and memory overhead of the adversarial recombination objective in Appendix B.4.

4. Experiments

In this section, we evaluate the effectiveness of our proposed approach across multiple settings[1]. We report the mean correlation coefficient (MCC), which is widely used in the nonlinear ICA literature (Hyvarinen & Morioka, 2016), and the mutual information gap (MIG) (Chen et al., 2018), to measure disentanglement with respect to the ground-truth factors.

4.1. Synthetic Setting

We first validate the proposed recombination mechanism in a controlled synthetic setting (Appendix C). Appendix C isolates the effect of the recombination discriminator in a synthetic setting where the data-generating factors are known and the decoder is fixed. We apply a fixed random mixing $z = Ms$ to entangle the true factors, and train only a linear reparameterization W using adversarial feedback from recombined samples. Thus, any improvement cannot be attributed to reconstruction learning or decoder

[1]We provide source code for our experiments here and here.

Algorithm 1 Compositional image training with single-source vs recombined discriminator

Input: dataset $\mathcal{D}$, encoder E_ϕ, generator/decoder G_θ, discriminator D_ψ, weight λ
while not converged **do**
  Sample two datapoints $x^A, x^B \sim \mathcal{D}$ and a recombination mask $S \in \{0,1\}^K$
  Encode latents: $z^A_{1:K} \leftarrow E_\phi(x^A)$, $z^B_{1:K} \leftarrow E_\phi(x^B)$
  Recombine latents: $\tilde{z}_k \leftarrow S_k z^A_k + (1 - S_k) z^B_k$ for $k = 1, \ldots, K$
  Sample $t \sim \mathrm{Unif}(\{1, \ldots, T\})$, $\epsilon \sim \mathcal{N}(0, I)$
  Forward diffusion: $x^A_t \leftarrow \sqrt{\bar{\alpha}_t}\, x^A + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$
  Single-source generation: $\hat{x}^{\text{single}} \leftarrow G_\theta(x^A_t, t, z^A_{1:K})$
  Recombined generation: $\hat{x}^{\text{recomb}} \leftarrow G_\theta(x^A_t, t, \tilde{z}_{1:K})$
  Compute diffusion reconstruction loss $\mathcal{L}_{\text{rec}} = \|x^A - \hat{x}^{\text{single}}\|_2^2$
  Discriminator update (single-source/recombined):
  $\mathcal{L}(\psi) = -\mathbb{E}\big[\log D_\psi(\hat{x}^{\text{single}}) + \log\big(1 - D_\psi(\hat{x}^{\text{recomb}})\big)\big]$
    $\psi \leftarrow \psi - \eta_\psi \nabla_\psi \mathcal{L}$
  Generator/encoder update:
    $\mathcal{L}_{\text{adv}}(\theta, \phi) = -\mathbb{E}\big[\log D_\psi(\hat{x}^{\text{recomb}})\big]$
    $\theta, \phi \leftarrow \theta, \phi - \eta \nabla_{\theta,\phi} (\mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{adv}})$

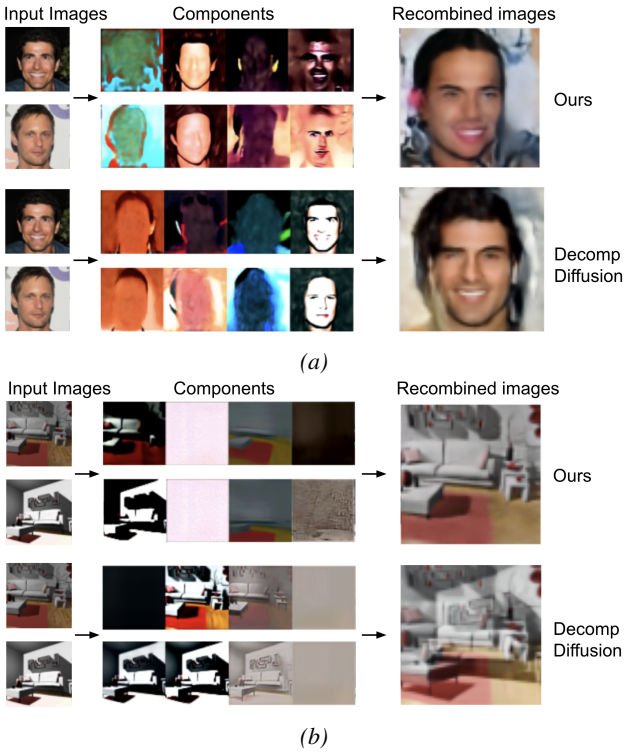


Figure 2. The unsupervised training of the baseline compositional model lacks an explicit mechanism to encourage compositional generalization. Recombining latent factors from two source images can therefore produce visually and physically inconsistent artifacts. For example, in (b), camera-related information is distributed across multiple latents, leading to inconsistencies under recombination.

capacity. Discriminator feedback reduces the squared Mahalanobis distance of recombined samples from 7.637 ± 0.459 to 4.101 ± 0.133, close to the real-data value 3.99 ± 0.002, while also improving disentanglement and correlation metrics, shown in Table 4. This suggests a discriminator supplies a useful signal for reshaping latent coordinates toward recombination-consistent factorizations.

Table 1. Synthetic recombination metrics (mean $\pm$ std over 3 seeds). Lower is better (↓), higher is better (↑). For reference, real samples have Mahalanobis^2 $= 3.99 \pm 0.002$.

Metric	Naive recomb.	+ discriminator
Mahalanobis^2 ↓	7.637 ± 0.459	4.101 ± 0.133
MCC ↑	0.603 ± 0.018	0.757 ± 0.018
MIG ↑	0.060 ± 0.008	0.122 ± 0.043
TC ↓	1.0946 ± 0.0405	0.8409 ± 0.3163
Block corr ↓	0.214 ± 0.007	0.039 ± 0.060

4.2. Image Settings

To evaluate the effectiveness of our proposed approach in image recombination, we compare our discriminator-driven method against the baseline Decomp Diffusion model across the image datasets Falcor3D (Nie, 2019), CelebA-HQ (Karras et al., 2018), VKITTI (Gaidon et al., 2016), and CLEVR (Johnson et al., 2017). Our evaluation primarily focuses on the image quality of recombined samples. Then, we discuss the empirical findings of image quality and disentanglement using the Falcor3D dataset. We compare against a reimplementation of Decomp Diffusion under our evaluation pipeline. Here, all main image experiments use $K = 4$ latent components for both our method and the Decomp Diffusion baseline. We additionally report a $K = 3$ latent-capacity ablation on Falcor3D in Table 3; further discussion of choosing latent settings are provided in Appendix B.5.

Perceptual Quality of Recombined Images. We evaluate the perceptual quality of recombined images using qualitative inspection, Fréchet Inception Distance (FID), and Kernel Inception Distance (KID). Table 2 reports FID and KID scores for our method and Decomp Diffusion across datasets, showing consistent improvements over the baseline. We further visualize compositional faithfulness on CLEVR in Fig. 4a, whose object-centric structure enables clear interpretation of factor-level effects.

Disentanglement Evaluation. We evaluate disentanglement using MIG and MCC on the Falcor3D dataset, a synthetic benchmark with ground-truth factor labels that enables precise assessment of factor separation. Table 4b shows that our method improves both MIG and MCC over the baseline. Results are consistent across repeated evaluations of a fixed model. These findings suggest that discriminator feedback improves the compositional consistency of the learned representation, which correlates with stronger disentanglement metrics. Qualitative results in Figs. 2, 3, and 7 show cleaner and more disentangled components compared to Decomp Diffusion.

Our approach optimizes both disentanglement and the compositional generation process itself, by regularizing recombined samples to be indistinguishable from single-source generations. This contrasts with prior work that studies disentanglement or compositional generalization separately, and aligns with findings that disentangled representations alone may still fail to compose correctly (Montero et al., 2022; Duan et al., 2020).

4.3. Videos for Robotics Setting

Robotic demonstration datasets are typically narrow: they contain a limited set of task instances, object placements, and action styles. Yet, downstream planning and control

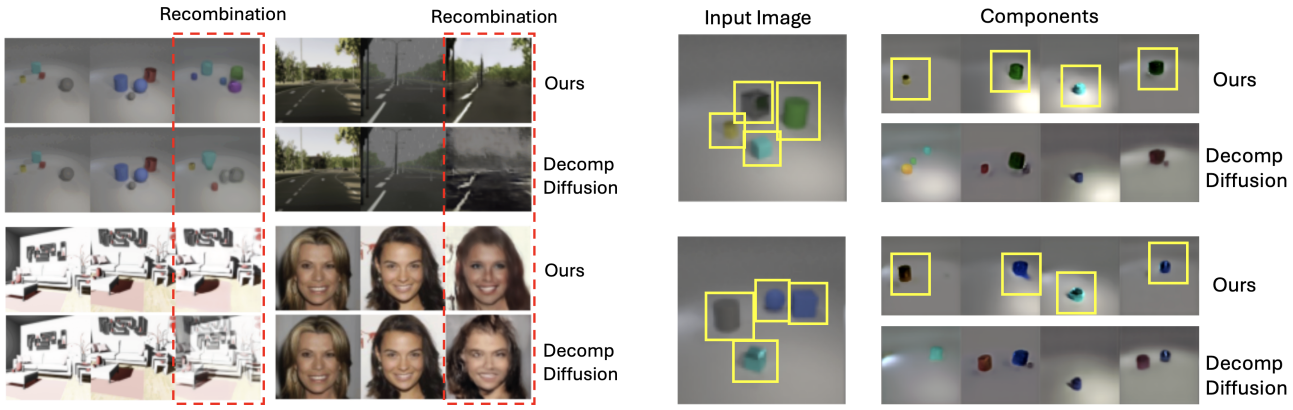


Figure 3. **Qualitative comparison of recombination and decomposition.** Left: for each group, the first two images are source samples drawn from the dataset, and the third image is a recombined generation obtained by mixing latent factors from the two sources. Across CLEVR, Virtual KITTI, Falcor3D, and CelebA-HQ, our method produces recombined samples that remain visually coherent and structurally consistent, while Decomp Diffusion often exhibits recombination artifacts. Right: given CLEVR input images containing four objects, our model decomposes each scene into disentangled, object-aligned components, where each component localizes a single object. By contrast, Decomp Diffusion exhibits substantial entanglement across components, often capturing multiple objects or incomplete object regions.

Dataset	Metric	Decomp Diffusion	Ours
CelebA-HQ	FID	82.70 ± 19.39	$\mathbf{43.98 \pm 0.78}$
	KID	0.076 ± 0.014	$\mathbf{0.055 \pm 0.019}$
Falcor3D	FID	157.11 ± 9.53	$\mathbf{130.19 \pm 3.62}$
	KID	0.177 ± 0.033	$\mathbf{0.161 \pm 0.005}$
VKITTI	FID	88.46 ± 3.81	$\mathbf{84.22 \pm 0.19}$
	KID	0.086 ± 0.005	$\mathbf{0.079 \pm 0.003}$
CLEVR	FID	25.70 ± 3.75	$\mathbf{24.16 \pm 0.16}$
	KID	0.023 ± 0.007	$\mathbf{0.020 \pm 0.001}$

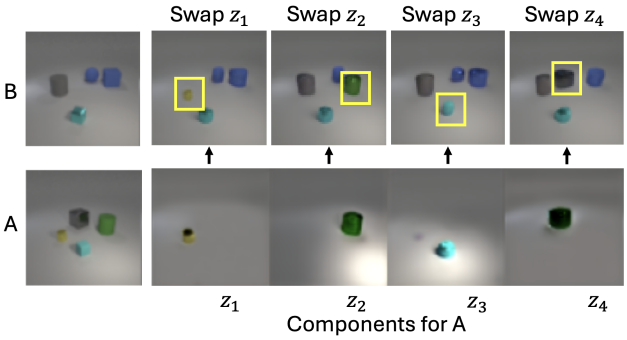
Table 2. **FID and KID comparison on *recombined samples*.** Samples are generated from Decomp Diffusion and from our model trained with discriminator feedback. Results are reported as mean $\pm$ standard deviation across 3 training runs. For each dataset, we generate 10K samples from recombined latents and compute FID and KID against the original dataset distribution as proxies for perceptual quality. Lower is better for both metrics.

requires coverage: a policy must succeed across many initial states and subtle variations that were never demonstrated. However, many tabletop manipulation tasks are approximately compositional in time: a trajectory can be viewed as a sequence of reusable action primitives (reach, grasp, lift, transport, place). If a video model learns latent factors that isolate these primitives and their relevant object interactions, then recombining latents across demonstrations can synthesize task-relevant rollouts that are novel yet physically plausible. We demonstrate that unsupervised components of a video diffusion model can be used for *directed exploration* in robotic environments by generating and executing recombined rollouts that expand state-space coverage beyond demonstrations.

We evaluate our approach in LIBERO (Liu et al., 2023), a simulated environment designed for tabletop manipulation using a Franka Panda robot, featuring a variety of dexterous manipulation tasks. Each task requires the agent to reach a final state, defined by a descriptive sentence indicating the target object and the desired outcome. The action space includes changes in the end effector's position and orientation, as well as the gripper's applied force, resulting in a 7-dimensional control space.

Problem setup. We consider a finite-horizon partially observed control setting with observation space $\mathcal{O} \subset \mathbb{R}^{H \times W \times C}$ and action space $\mathcal{A} \subset \mathbb{R}^7$ (3D translation, 3D rotation, gripper open/close). A video of length N is a sequence $x_{1:N} = (x_1, \ldots, x_N)$ with $x_n \in \mathcal{O}$. We assume access to a dataset of demonstration videos $\mathcal{D} = \{x^{(m)}_{1:N_m}\}_{m=1}^{M}$ recorded in a table-top manipulation benchmark. We evaluate on two manipulation scenes from the LIBERO benchmark, denoted Scene 5 and Scene 6. Scene 5 involves placing mugs of varying appearance onto designated plates (e.g., "put red mug on left plate"), while Scene 6 focuses on placing different objects relative to a plate (e.g., "put chocolate pudding to the right of plate"). Each scene contains multiple task variants that differ in object type, color, and target relation; the complete list of task specifications is provided in Appendix E.3.

Video recombination objective. Let K denote the number of latent factors. An encoder E_ϕ maps a video suffix to K latent components $z_{1:K} = E_\phi(x_{2:N})$ where $z_k \in \mathbb{R}^{d_k}$. Generation is performed by a diffusion model that is conditioned on the first frame. The denoiser $\bar{G}_\theta$ predicts the original video at every time-diffusion step t conditioned on (i) the noised video, (ii) the factor latents, and (iii) the initial frame $\epsilon_\theta(\tilde{x}^{(t)}_{1:N}, t \mid x_1, z_{1:K})$. First-frame conditioning stabilizes



(a)

Model	MIG ↑	MCC ↑
InfoGAN	0.025 ± 0.010	0.527 ± 0.019
β-VAE	0.090 ± 0.010	0.517 ± 0.016
MONet	0.139 ± 0.020	0.657 ± 0.008
COMET*	0.010 ± 0.002	0.573 ± 0.003
Decomp Diffusion*	0.065 ± 0.019	0.657 ± 0.025
Ours* ($\lambda = 0.005$)	0.118 ± 0.012	$\mathbf{0.707 \pm 0.011}$
Ours* ($\lambda = 0.01$)	0.116 ± 0.058	0.685 ± 0.020
Ours* ($\lambda = 0.02$)	$\mathbf{0.141 \pm 0.031}$	0.640 ± 0.038

(b)

Figure 4. **Compositional behavior via latent swapping and disentanglement evaluation.** In (a), swapping a single latent component z_k from image A into B generates recombined samples, while the bottom row shows components of A decoded individually. Compositional generation is faithful: swapping object-level factors replaces the corresponding object while preserving the rest of the scene. In (b), we compare MIG and MCC disentanglement metrics on Falcor3D. Asterisks denote models reimplemented and evaluated in our codebase, with mean ± standard deviation over three independent training runs; unasterisked entries report previously published results and are included for reference (Su et al., 2024).

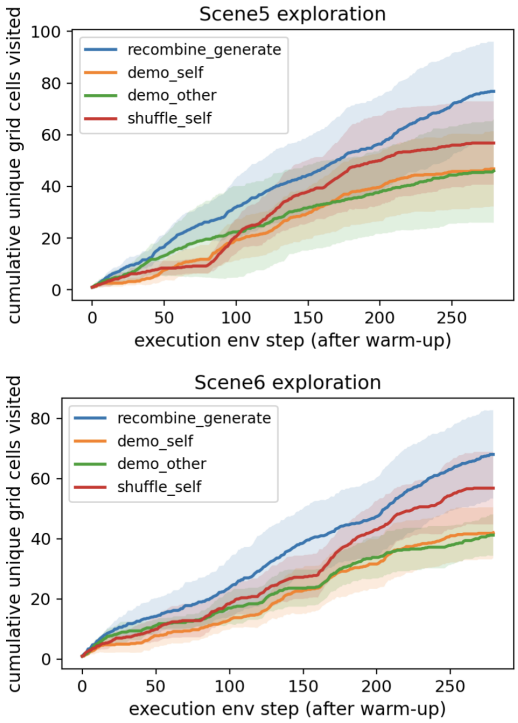


Figure 5. Mean ± std joint-state coverage over 10 paired runs for Scene5 (top) and Scene6 (bottom). Recombined videos yield action trajectories with consistently higher exploration than non-compositional baselines under identical warm-up conditions.

training and focuses the representation on action primitives and object interactions rather than the initial scene layout.

The base diffusion objective uses an ℓ_2 reconstruction loss on the predicted clean sample:

$$\mathcal{L}_{\text{diff}}(\theta, \phi) = \mathbb{E}\Big[\big\|x_{1:N} - \hat{x}_\theta\big(\tilde{x}^{(t)}_{1:N}, t \big| x_1, z_{1:K}\big)\big\|_2^2\Big],$$

where $\tilde{x}^{(t)}_{1:N}$ denotes the noised trajectory at diffusion step t. Given source videos x^A and x^B and timestep t, we introduce a discriminator D_ψ that distinguishes single-source generations $\hat{x}^{\text{single}}_{1:N} = \bar{G}_\theta(x^{A,(t)}, t, x^A_1, z_{1:K})$ from recombined generations $\hat{x}^{\text{recomb}}_{1:N} = \bar{G}_\theta(x^{A,(t)}, t, x^A_1, \tilde{z}_{1:K})$ where $\bar{G}_\theta(x, t, x_1, z_{1:K}) = \frac{1}{K}\sum_{j=1}^{K} \bar{G}_\theta(x, t, x_1, z_j)$. The discriminator is trained with the objective

$$L_{\text{clf}}(\psi) = -\mathbb{E}\Big[\log D_\psi\Big(\hat{x}^{\text{single}}_{1:N}\Big)\Big] - \mathbb{E}\Big[\log\big(1 - D_\psi\big(\hat{x}^{\text{recomb}}_{1:N}\big)\big)\Big],$$

while the generator and encoder are trained adversarially to make recombined generations indistinguishable from single-source ones: $\mathcal{L}_{\text{adv}}(\theta, \phi) = -\mathbb{E}\big[\log D_\psi(\hat{x}^{\text{recomb}}_{1:N})\big]$. The generator and encoder are optimized with the combined objective $\mathcal{L}_{\text{diff}} + \lambda\, \mathcal{L}_{\text{adv}}$, while the discriminator is optimized with L_{clf}. Training alternates between these objectives. See Fig. 15 for visualization of recombination.

Evaluation protocol and baselines. We evaluate generated rollouts using two complementary metrics: a *physical consistency* score L_{phys}, which measures disagreement between actions inferred from generated frames and those executed in the environment, and a *coverage* metric Cov_Δ, which quantifies exploration breadth in discretized joint-state space. We provide further details in Appendix E.3. To compare exploration fairly across planning strategies, we use a paired-episode protocol. For each episode we fix the task instance, demonstration index, environment seed, and a short warm-up horizon during which ground-truth demonstration actions are executed before the generated plan takes over. Details and a discussion of the warm-up's role are in Appendix E.3.

Qualitative results. Fig. 6 shows a recombined robotic rollout generated by mixing latent factors from two training videos. Figs. 19 and 20 visualize the learned video

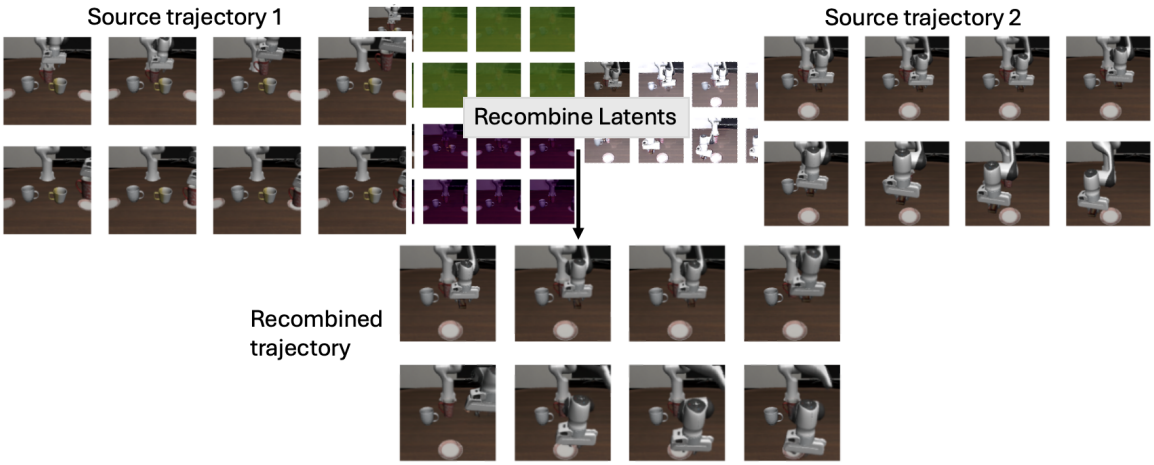


Figure 6. Recombined video plan from Scene 5 and Scene 6 from Libero. The two source scenes are "Red mug on right plate" from Scene 5 and "Chocolate pudding left of plate" from Scene 6. The generated video plan shows the robot attempting to place the pudding on the edge of the plate. This specific action sequence does not appear in the original demonstrations, which only include placements beside the plate. The influence of Scene 5 can be observed in the trajectory as the robot arm is pulled sideways, suggesting interactions between the recombined behaviors.

decomposition, showing that individual components capture interpretable aspects of the trajectory.

Ablations and trade-offs. Sweeping λ shows an inverted-U trend: moderate discriminator feedback improves L_{phys} and Cov_{Δ}, while excessive feedback over-regularizes and harms reconstruction. Table 5 shows that the best variant increases explored grid cells by 35.8% in Scene5 and 73.8% in Scene6 over the baseline.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J. B., Jaakkola, T. S., and Agrawal, P. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023.

Anand, E. T. and Liaw, S. Feel-good thompson sampling for contextual bandits: a markov chain monte carlo showdown. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Bagautdinov, T., Wu, C., Saragih, J., Fua, P., and Sheikh, Y. Modeling facial geometry using compositional VAEs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3877–3886, 2018.

Bansal, A., Chu, H.-M., Schwarzschild, A., Sengupta, R., Goldblum, M., Geiping, J., and Goldstein, T. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives, 2014. URL https://arxiv.org/abs/1206.5538.

Black, K., Nakamoto, M., Atreya, P., Walke, H. R., Finn, C., Kumar, A., and Levine, S. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June 2023.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-VAE, 2018.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. In *International Conference on Learning Representations*, 2019.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations (ICLR)*, 2024.

Dai, X., Liang, K., and Xiao, B. Advdiff: Generating unrestricted adversarial examples using diffusion models, 2024.

Denton, E. and Fergus, R. Stochastic video generation with a learned prior. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1174–1183. PMLR, 10–15 Jul 2018.

Du, Y. and Kaelbling, L. Position: compositional generative modeling: a single model is not all you need. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Du, Y., Li, S., and Mordatch, I. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

Du, Y., Li, S., Sharma, Y., Tenenbaum, J. B., and Mordatch, I. Unsupervised learning of compositional energy concepts. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC, 2023a.

Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., and Abbeel, P. Learning universal policies via text-guided video generation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023b.

Duan, S., Matthey, L., Saraiva, A., Watters, N., Burgess, C., Lerchner, A., and Higgins, I. Unsupervised model selection for variational disentangled representation learning. In *International Conference on Learning Representations*, 2020.

Engelcke, M., Jones, O. P., and Posner, I. GENESIS-v2: Inferring unordered object representations without iterative refinement. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Fu, X., Wang, X., Liu, X., Bai, J., Xu, R., Wan, P., Zhang, D., and Lin, D. Learning video generation for robotic manipulation with collaborative trajectory control. In *The Fourteenth International Conference on Learning Representations*, 2026.

Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. Virtualworlds as proxy for multi-object tracking analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4340–4349, 2016. doi: 10.1109/CVPR.2016.470.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2023.

Gandikota, R., Orgad, H., Belinkov, Y., Materzynska, J., and Bau, D. Concept sliders: LoRA adaptors for precise control in diffusion models. In *European Conference on Computer Vision*, 2024.

Gaujac, B., Feige, I., and Barber, D. Learning disentangled representations with the wasserstein autoencoder. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III*, pp. 69–84, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-86522-1. doi: 10.1007/978-3-030-86523-8_5.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. GANSpace: Discovering interpretable GAN controls. In *Advances in Neural Information Processing Systems*, 2020.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Hyvarinen, A. and Morioka, H. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 17–23 Jul 2022.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. Variational autoencoders and nonlinear ica: A unifying framework. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2207–2217. PMLR, 26–28 Aug 2020.

Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kirilenko, D., Vorobyov, V., Kovalev, A., and Panov, A. Object-centric learning with slot mixture module. In *The Twelfth International Conference on Learning Representations*, 2024.

Ko, P.-C., Mao, J., Du, Y., Sun, S.-H., and Tenenbaum, J. B. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations (ICLR)*, 2024.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Lachapelle, S., Mahajan, D., Mitliagkas, I., and Lacoste-Julien, S. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, 2023.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Lin, Y., Preiss, J. A., Anand, E., Li, Y., Yue, Y., and Wierman, A. Online adaptive policy selection in time-varying systems: No-regret via contractive perturbations. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53508–53521. Curran Associates, Inc., 2023.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023.

Liu, N., Li, S., Du, Y., Tenenbaum, J. B., and Torralba, A. Learning to compose visual relations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 423–439, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-19789-5. doi: 10.1007/978-3-031-19790-1_26.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 09–15 Jun 2019.

Luo, Y. and Du, Y. Grounding video models to actions through goal conditioned exploration. *arXiv preprint arXiv:2411.07223*, 2024.

Ma, T., Zheng, J., Wang, Z., Jiang, C., Cui, A., Liang, J., and Yang, S. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control, 2026. URL https://arxiv.org/abs/2603.10448.

Misino, E., Marra, G., and Sansone, E. VAEL: Bridging variational autoencoders and probabilistic logic programming. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. H. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4667–4679, 2022.

Montero, M. L., Bowers, J., Costa, R. P., Ludwig, C. J., and Malhotra, G. Lost in latent space: Examining failures of disentangled models at combinatorial generalisation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Netanyahu, A., Du, Y., Bronars, A., Pari, J., Tenenbaum, J., Shu, T., and Agrawal, P. Few-shot task learning through inverse generative modeling. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 98445–98477. Curran Associates, Inc., 2024.

Nie, W. High resolution disentanglement datasets, 2019.

Park, T., Zhu, J.-Y., Wang, O., Lu, J., Shechtman, E., Efros, A. A., and Zhang, R. Swapping autoencoder for deep image manipulation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalainathan, D., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Sehgal, A., Grayeli, A., Sun, J. J., and Chaudhuri, S. Neurosymbolic grounding for compositional world models, 2024.

Shen, Y., Wei, F., Du, Z., Liang, Y., Lu, Y., Yang, J., Zheng, N., and Guo, B. VideoVLA: Video generators can be generalizable robot manipulators. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=UPHlqbZFZB.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Song, Y. and Kingma, D. P. How to train your energy-based models, 2021.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Squires, C., Seigal, A., Bhate, S. S., and Uhler, C. Linear causal disentanglement via interventions. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32540–32560. PMLR, 23–29 Jul 2023.

Su, J., Liu, N., Wang, Y., Tenenbaum, J. B., and Du, Y. Compositional image decomposition with diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46823–46842. PMLR, 21–27 Jul 2024.

Wang, C., Uehara, M., He, Y., Wang, A., Lal, A., Jaakkola, T., Levine, S., Regev, A., Wang, H., and Biancalani, T. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Wu, Z., Hu, J., Lu, W., Gilitschenski, I., and Garg, A. Slot-diffusion: Object-centric generative modeling with diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zekri, O. and Boulle, N. Fine-tuning discrete diffusion models with policy gradient methods. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zhang, Y., Tzeng, E., Du, Y., and Kislyuk, D. Large-scale reinforcement learning for diffusion models, 2024.

Zhou, S., Du, Y., Chen, J., Li, Y., Yeung, D.-Y., and Gan, C. Robodreamer: learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zhu, Y., Xie, J., Wu, Y. N., and Gao, R. Learning energy-based models by cooperative diffusion recovery likelihood. In *The Twelfth International Conference on Learning Representations*, 2024.

A. Detailed Background on Decomp Diffusion and COMET

This appendix reviews two closely related frameworks for unsupervised compositional representation learning that our work builds on top of: COMET (Du et al., 2021) and Decomp Diffusion (Su et al., 2024). Both methods aim to decompose images into independently manipulable latent factors and to support recombination by composing these factors at generation time. The primary difference lies in how composition is implemented and optimized.

A.1. COMET: Composable Energy-Based Representations

COMET represents each latent factor as an energy function over images. Given an image $x_i \in \mathbb{R}^D$, an encoder produces a set of latent variables $z_1, \dots, z_K = \mathrm{Enc}_\phi(x_i)$, and a shared energy network defines a family of energies $E_\theta(x, z_k) : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}$. A single latent z_k specifies an energy landscape whose minimum corresponds to an image consistent with that factor. Composition is defined by summing energies and jointly minimizing over the image:

$$x^\star(z_{1:K}) = \arg\min_x \sum_{k=1}^{K} E_\theta(x, z_k). \tag{1}$$

COMET is trained by reconstructing the input image as the minimum-energy configuration under the inferred latents. The learning objective minimizes the reconstruction error between the input and the result of energy minimization,

$$\mathcal{L}_{\mathrm{COMET}} = \left\| \arg\min_x \sum_{k=1}^{K} E_\theta\big(x, \mathrm{Enc}_\phi(x_i)[k]\big) - x_i \right\|_2^2. \tag{2}$$

Since the argmin is intractable to compute, it is approximated by unrolling several steps of gradient descent on x,

$$x^{(n)} = x^{(n-1)} - \lambda \nabla_x \sum_{k=1}^{K} E_\theta(x^{(n-1)}, z_k), \qquad n = 1, \dots, N, \tag{3}$$

with $x^{(0)}$ initialized from noise. Training therefore requires backpropagation through an unrolled optimization procedure, which can be computationally expensive and sensitive to hyperparameters. Despite this cost, COMET provides a conceptually clean formulation of compositionality: factors are combined by additive energies, and recombination is implemented by minimizing the same summed objective using latents drawn from different images.

A.2. Decomp Diffusion: Composing Denoising Components

Decomp Diffusion builds on diffusion probabilistic models and replaces explicit energy minimization with diffusion denoising dynamics. For the diffusion training in Decomp Diffusion (Su et al., 2024), a noisy image

$$x_i^t = \sqrt{\bar{\alpha}_t}\, x_i + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

is passed to a denoiser $\epsilon_\theta(x_i^t, t)$ trained to predict the noise (or original image) via

$$\mathcal{L}_{\mathrm{DDPM}} = \left\| \epsilon - \epsilon_\theta(x_i^t, t) \right\|_2^2. \tag{4}$$

In our experiments, we have the denoiser predict the original image. When the denoiser is conditioned on a latent z, $\epsilon_\theta(x_t, t, z)$ corresponds to the gradient of a latent-dependent energy $E_\theta(x, z)$. To enable decomposition, Decomp Diffusion encodes each input image into K latents,

$$z_1, \dots, z_K = \mathrm{Enc}_\phi(x_i),$$

and trains a shared conditional denoiser such that the mean of per-latent denoising predictions matches the true noise (or original image):

$$\frac{1}{K} \sum_{k=1}^{K} \epsilon_\theta(x_i^t, t, z_k) \approx \epsilon. \tag{5}$$

The resulting training objective is

$$\mathcal{L}_{\mathrm{Decomp}} = \left\| \epsilon - \frac{1}{K} \sum_{k=1}^{K} \epsilon_\theta(x_i^t, t, z_k) \right\|_2^2. \tag{6}$$

Latent dimensionality constraints act as an information bottleneck, encouraging different z_k to capture distinct factors. At inference time, composition is performed by averaging denoising predictions from any chosen set of latents at each diffusion step, $\epsilon_{\text{pred}}(x_t, t) = \frac{1}{K}\sum_k \epsilon_\theta(x_t, t, z_k)$, and running the standard diffusion sampling procedure using ϵ_{pred}. This enables recombination without solving an inner optimization problem.

A.3. Conceptual Relationship Between COMET and Decomp Diffusion

Both COMET and Decomp Diffusion define composition through additive structure. COMET composes factors by summing energies and explicitly minimizing over the image, while Decomp Diffusion composes factors by averaging denoising (energy-gradient) contributions and relying on diffusion dynamics to perform implicit optimization. The latter avoids backpropagation through unrolled image updates during training and yields a more stable and scalable learning procedure, while preserving the same compositional intuition.

A.4. Other Baseline Methods

InfoGAN (Chen et al., 2016) extends GANs by maximizing mutual information between a subset of latent variables and generated samples, encouraging individual latent dimensions to capture interpretable factors of variation. β-VAE (Higgins et al., 2017) promotes disentangled representations by upweighting the KL divergence term in the variational objective, encouraging statistical independence among latent dimensions within a fixed-dimensional latent space. MONet (Burgess et al., 2019) is an object-centric generative model that decomposes scenes into multiple components using an iterative attention mechanism and soft segmentation masks, enabling pixel-level decomposition into object-like regions while primarily focusing on spatial structure rather than global scene factors.

B. Experiment Details

In this section, we provide dataset details in Appendix B.1, and present training and inference details of our method in Appendix B.2 and Appendix B.3.

B.1. Dataset Details

We evaluate our method on four image datasets commonly used for studying compositionality and disentangled representations: CelebA-HQ, Falcor3D, Virtual KITTI 2, and CLEVR.

CelebA-HQ. CelebA-HQ (Karras et al., 2018) is a high-quality face image dataset derived from CelebA, consisting of aligned human face images with rich variation in global attributes such as facial structure, hair style, hair color, skin tone, and expression. Following prior work, we use images resized to 64×64 resolution.

Falcor3D. Falcor3D (Nie, 2019) is a synthetic dataset rendered using a physically based graphics pipeline, designed specifically for evaluating disentanglement. Images depict simple 3D scenes with controlled generative factors including object geometry, material properties, lighting direction and intensity, and camera pose. Ground-truth factor annotations are available, enabling quantitative evaluation of disentanglement metrics such as MIG and MCC. We use the standard 128×128 image resolution.

Virtual KITTI 2. Virtual KITTI 2 (VKITTI2) (Gaidon et al., 2016) is a photorealistic synthetic dataset that mirrors real-world driving scenes from the KITTI benchmark. It provides rendered outdoor scenes with structured variation in camera viewpoint, lighting, weather conditions, background layout, and object configurations. In our experiments, we use VKITTI2 to evaluate decomposition of complex global scene factors and the plausibility of recomposed scenes under latent recombination.

CLEVR. CLEVR (Johnson et al., 2017) is a synthetic object-centric dataset consisting of images containing multiple simple 3D objects placed on a tabletop. Each scene varies along interpretable factors such as object shape, color, size, material, and spatial arrangement. Although originally designed for visual reasoning, CLEVR is widely used for evaluating object-level decomposition and compositional generalization. We use 64×64 rendered images and focus on recombination of object-centric latent factors.

B.2. Training Overview

We used standard denoising training to train our denoising networks, with 1000 diffusion steps and squared cosine beta schedule. In Decomp Diffusion and in our implementation, the denoising network ϵ_θ is trained to directly predict the original image x_0. To train our diffusion model that conditions on inferred latents z_k, we first utilize the latent encoder to encode input images into features that are further split into a set of latent representations $\{z_1, \ldots, z_k\}$. For each input image, we then train our model conditioned on each decomposed latent factor z_k using standard denoising loss. Each model is trained for 20k - 40k steps on an NVIDIA V100 32GB machine (roughly 16 - 32 hours), and we use a batch size of 16 when training.

Single-step discriminator training. Applying the discriminator to fully sampled outputs would require running an iterative denoising process (e.g., DDIM with tens of steps) during training, significantly increasing computational cost and complicating gradient-based optimization. Instead, we train the discriminator on denoised predictions obtained from a single diffusion step while sharing the same noisy input x_t. Although these predictions are not exact samples from the model, they provide a consistent and differentiable surrogate that allows the discriminator to detect incoherent latent recombinations. Because DDIM sampling repeatedly applies the same denoiser across timesteps, shaping single-step predictions across noise levels is a practical proxy for improving the quality of the final multi-step samples.

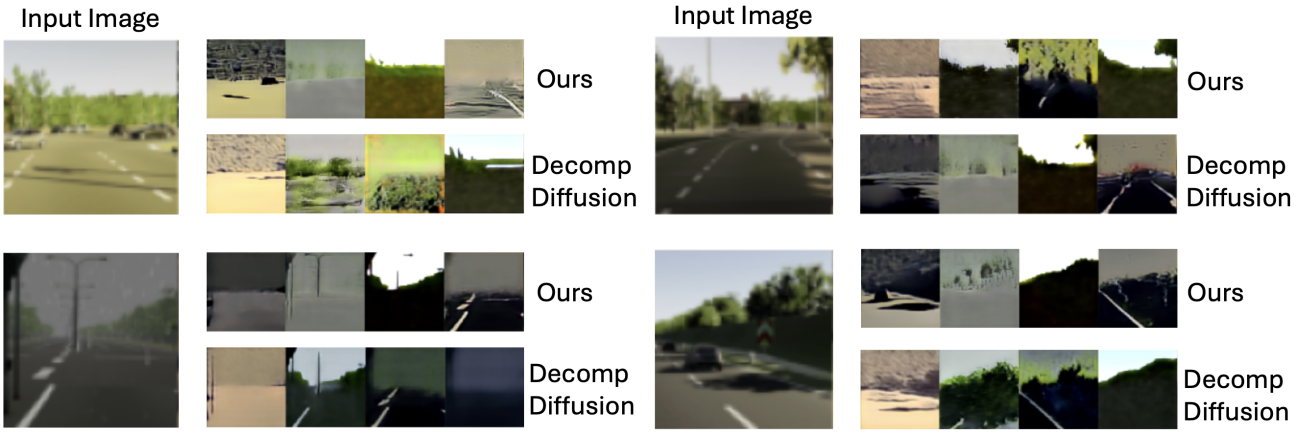


Figure 7. **Qualitative decomposition results on VKITTI.** Our model produces visually cleaner and semantically more disentangled decompositions than Decomp Diffusion. Our components isolate coherent scene factors such as road markings, object and shadow regions, tree-level details, sky boundaries, and road contours. In contrast, Decomp Diffusion often spreads the same factor across multiple components, such as the white road markings in the bottom-left example, and mixes tree, road, and sky-boundary information across components in the bottom-right example.

B.3. Inference Details

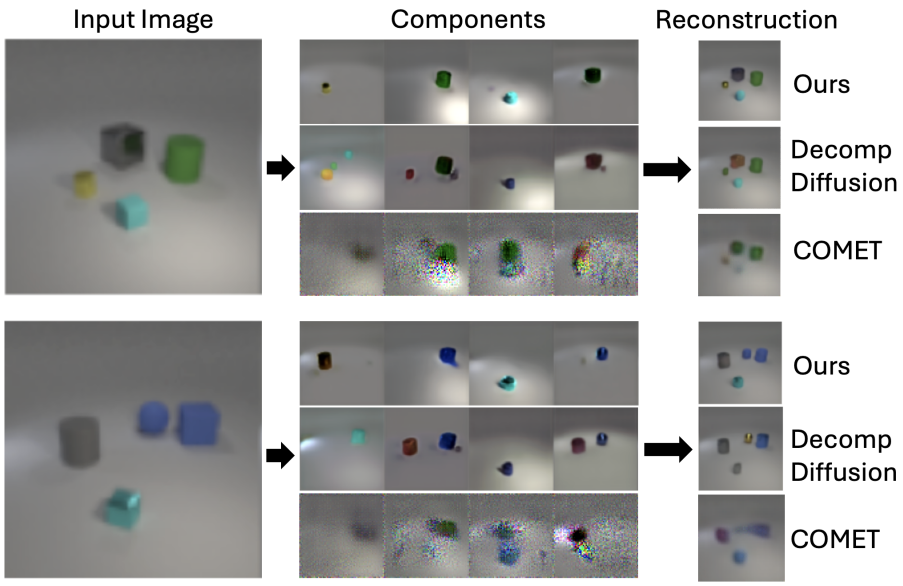
When generating images, we use DDIM with 50 steps for faster image generation.

Decomposition. To decompose an image x, we first pass it into the latent encoder Enc_ϕ to extract out latents $\{z_1, \cdots, z_K\}$. For each latent z_k, we generate an image corresponding to that component by running the image generation algorithm on z_k.

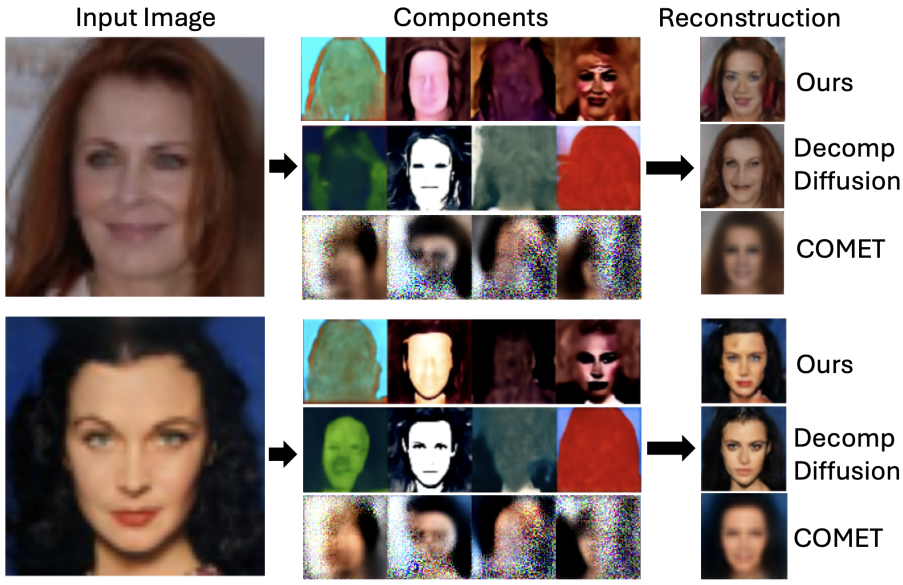
Reconstruction. To reconstruct an image x given latents $\{z_1, \cdots, z_K\}$, in the denoising process, we predict ϵ by averaging the model outputs conditioned on each individual z_k. The final result is a denoised image which incorporates all inferred components, i.e., reconstructs the image.

Recombination. To recombine images x and x', we recombine their latents $\{z_1, \ldots, z_K\}$ and $\{z'_1, \cdots, z'_K\}$. We select the desired latents from each image and condition on them in the image generation process, i.e., predict ϵ in the denoising process by averaging the model outputs conditioned on each individual latent.

To additively combine images x and x' so that the result has all components from both images, e.g., combining two images with 4 objects to generate an image with 8 objects, we modify the generation procedure. In the denoising process, we assign the predicted ϵ to be the mean over all $2 \times K$ model outputs conditioned on individual latents in $\{z_1, \cdots, z_K\}$ and $\{z'_1, \cdots, z'_K\}$. This results in an image with all components from both input images.



(a) CLEVR



(b) CELEBA-HQ

Figure 8. We extend the qualitative comparison between Decomp Diffusion and COMET on CLEVR from Fig. 3, and include CELEBA-HQ as well. We visualize decomposition and reconstruction results. Our method outperforms Decomp Diffusion and COMET across object separation, qualitative disentanglement, and reconstruction quality.

B.4. Computational Overhead

Relative to Decomp Diffusion, our method adds one recombined prediction and one discriminator update per training step. Both the single-source prediction $\hat{x}^{\text{single}}$ and the recombined prediction $\hat{x}^{\text{recomb}}$ are computed by averaging K factor-conditioned denoiser evaluations. Thus, the denoiser compute is approximately doubled during training, from K to $2K$ conditional denoiser evaluations per step, with an additional discriminator forward/backward pass on the predicted clean samples.

For fixed K, the asymptotic training complexity remains the same order as Decomp Diffusion, but with a constant-factor

overhead from the second denoiser evaluation and the discriminator update. The memory overhead is mainly due to storing activations for the additional recombined prediction during the generator/encoder update, along with the discriminator activations during its update. At inference time, the discriminator is discarded, and recombined samples are generated using the same averaged denoising procedure as Decomp Diffusion, so the inference cost is unchanged.

B.5. Ablations

As λ increases, recombined sample quality initially improves on both datasets, indicating that discriminator feedback helps constrain generations toward the data manifold, as seen in Fig 9. However, overly strong guidance leads to worse FID, suggesting that excessive discriminator pressure can conflict with the diffusion reconstruction objective. We also include a qualitative example in Fig. 10 highlighting a potential degenerate behavior of the adversarial objective when the specific discriminator weight is too high. This motivates using a moderate discriminator weight in all subsequent experiments.

Latent capacity. Unless otherwise stated, all main image experiments use $K = 4$ latent components for both our method and the Decomp Diffusion baseline. The $K = 3$ results in Table 3 are a separate latent-capacity ablation on Falcor3D and should not be directly compared to the main Falcor3D results in Table 4b, which use $K = 4$. We did not include $K = 5$ experiments in the main comparison due to memory constraints of training on V100s. Moreover, the Decomp Diffusion baseline with $K = 5$ produced much lower-quality recombinations on these datasets.

B.6. Metric computation (MIG, MCC).

Let $r(x) \in \mathbb{R}^D$ denote the representation returned by our encoder and let y denote the ground-truth generative factors. We construct $r(x)$ by concatenating the K component vectors produced by the encoder and use the encoder mean for each component. We compute both MIG and MCC over the individual coordinates of $r(x)$, treating each coordinate as a scalar latent variable. We do not use a learned probe or regression for either metric.

MIG. We sample 4,000 observations and factors, compute representations, and discretize each latent coordinate independently using a histogram discretizer with 20 bins. We estimate mutual information using `sklearn.metrics.mutual_info_score`. For each factor, MIG is the difference between the largest and second-largest mutual informations across latent coordinates, normalized by the factor entropy. We report the mean and standard deviation across factors.

MCC. We compute Spearman correlations between each representation coordinate and each ground-truth factor, apply Hungarian matching to the absolute correlation matrix to assign coordinates to factors, and report the mean absolute matched correlation.

Model	MIG ↑	MCC ↑
Decomp Diffusion*	0.014 ± 0.003	0.415 ± 0.017
Ours* ($\lambda = 0.005$)	0.014 ± 0.005	0.411 ± 0.019
Ours* ($\lambda = 0.01$)	$\mathbf{0.020 \pm 0.009}$	0.418 ± 0.016
Ours* ($\lambda = 0.02$)	0.010 ± 0.002	$\mathbf{0.428 \pm 0.001}$

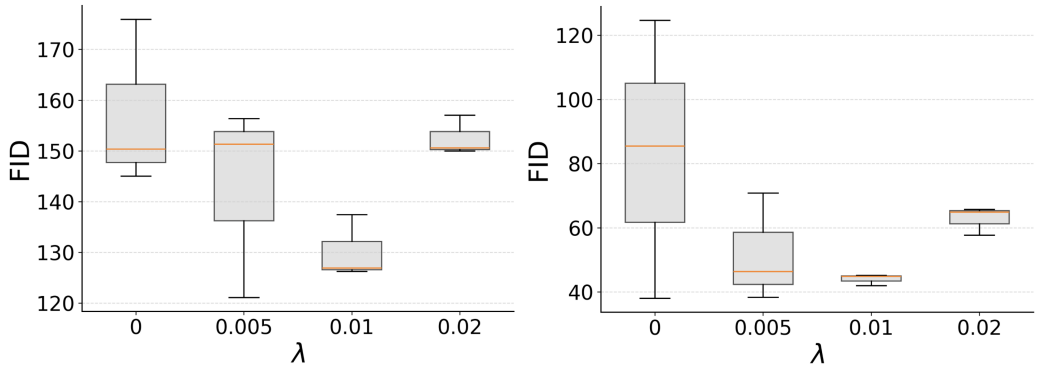
Table 3. For $K = 3$, our method achieves a higher mean MIG at $\lambda = 0.01$ and a higher mean MCC at $\lambda = 0.02$ compared to Decomp Diffusion.

C. Synthetic Recombination Experiment

C.1. Synthetic Recombination Experiment: Isolating Discriminator Feedback

We first study a synthetic setting designed to isolate the effect of discriminator feedback on latent recombination, independent of diffusion training or high-capacity generators. The goal of this experiment is to verify our method's core mechanism: that penalizing recombined samples provides a signal that reshapes representations toward a recombination-friendly factorization.

Setup. We construct data from three independent latent factors $s = (s_1, s_2, s_3)$, for $s_i \in \mathbb{R}^2$ drawn from distinct non-Gaussian distributions. Observations $x \in \mathbb{R}^4$ are generated by a fixed decoder with explicit cross-factor inter-



(a) Falcor3D. *(b)* CelebA-HQ.

Figure 9. **Effect of discriminator strength (λ) on sample quality.** We plot the FID (lower is better) as a function of discriminator guidance weight λ for two datasets: (a) **Falcor3D** and (b) **CelebA-HQ**. Increasing λ initially improves FID, indicating that the discriminator effectively pulls recombined samples closer to the data manifold. However, overly strong guidance ($\lambda = 0.02$) slightly degrades FID.

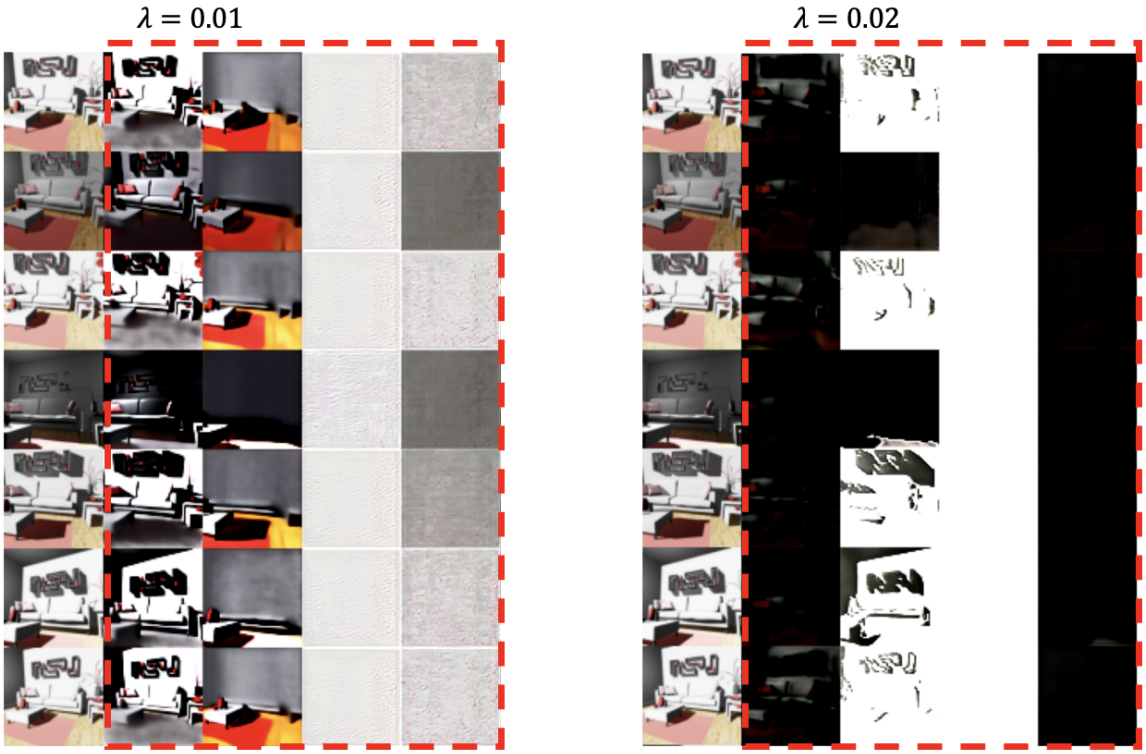


Figure 10. Qualitative visualization of learned latent components at two discriminator weights. **Left:** Non-collapsed representation at $\lambda = 0.01$. **Right:** partial latent collapse observed in a single example at $\lambda = 0.02$. In each panel, the leftmost column shows input images from the dataset, while the red dashed region visualizes the outputs associated with individual latent components. Notably, for $\lambda = 0.02$, the last two latent columns produce nearly identical outputs across different inputs, indicating under-utilized or collapsed latent dimensions.

actions; concretely, writing $s_i = (s_{i,x}, s_{i,y})$, we define $x_0 = \langle s_1, s_2 \rangle$, $x_1 = \langle s_2, s_3 \rangle$, $x_2 = \|s_1\|_2^2 - \|s_3\|_2^2$, and $x_3 = s_{1,x}s_{3,y} - s_{1,y}s_{3,x}$. To simulate entangled representations, we apply a random linear mixing $z = Ms$.

We introduce a discriminator trained to distinguish single-source samples from recombined samples. A linear reparameterization W of the latent space is then trained adversarially to fool the discriminator on recombined samples, while the decoder remains fixed. This setting corresponds to an idealized regime where the reconstruction objective is trivially

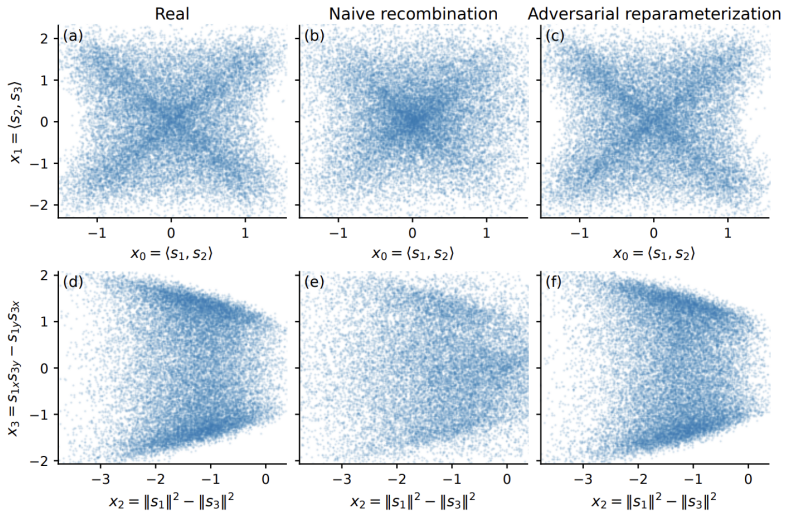


Figure 11. Synthetic recombination experiment. Left: real samples. Middle: naive recombination in entangled coordinates produces off-manifold samples. Right: discriminator feedback restores the geometry of real data. Top row shows (x_0, x_1) projections; bottom row shows (x_2, x_3) projections.

Table 4. Synthetic recombination experiment. The decoder and mixing matrix are fixed; only W is trained using discriminator feedback. Real samples have Mahalanobis$^2 = 3.99 \pm 0.002$.

Metric	Naive recomb.	+ Discriminator
Mahalanobis^2 ↓	7.637 ± 0.459	4.101 ± 0.133
MCC ↑	0.603 ± 0.018	0.757 ± 0.018
MIG ↑	0.060 ± 0.008	0.122 ± 0.043
TC ↓	1.0946 ± 0.0405	0.8409 ± 0.3163
Block corr. ↓	0.214 ± 0.007	0.039 ± 0.060

satisfied, allowing us to isolate the influence of the adversarial recombination loss.

Results. As shown in Fig. 11, naive recombination in the entangled coordinates produces samples that deviate substantially from the data manifold, while discriminator feedback restores the geometry of real data. Quantitatively, recombined samples exhibit a large increase in off-manifold distance relative to real data, which collapses back to near-real levels after discriminator training (Table 4). Importantly, the learned reparameterization need not recover the true generative factors; instead, it reduces statistical dependence between latent components, enabling approximate closure under coordinate-wise recombination. This toy experiment demonstrates that feedback from recombined samples alone can induce a representation that supports compositional recombination, even without semantic identifiability or diffusion training. We provide intuition for this behavior with our theoretical analysis (Appendix F), which shows that recombination closure implies a Cartesian-product structure in latent space and that penalizing recombined samples contracts cross-component dependence.

While this synthetic setting does not admit semantic identifiability, we can quantify improved factorization using standard disentanglement-style metrics, leveraging access to ground-truth factors. Separately, we report Gaussian total correlation (TC; a log-det covariance proxy) as a measure of statistical dependence among coordinates. To assess block-level independence, we also report the mean absolute off-diagonal correlation of per-block ℓ_2 norms. We compute these metrics on the raw entangled latents z before training and on the discriminator-induced reparameterization $\hat{s} = Wz$ after training.

We construct independent latent factors $s = (s_1, s_2, s_3) \in \mathbb{R}^6$, with each $s_k \in \mathbb{R}^2$ following a distinct non-Gaussian distribution (ring, uniform square, and bimodal Gaussian). Observed variables $x \in \mathbb{R}^4$ are generated deterministically via

$$x_0 = \langle s_1, s_2 \rangle, \quad x_1 = \langle s_2, s_3 \rangle, \tag{7}$$

$$x_2 = \|s_1\|^2 - \|s_3\|^2, \quad x_3 = s_{1x}s_{3y} - s_{1y}s_{3x}, \tag{8}$$

which imposes explicit cross-factor constraints that characterize the data manifold.

To model entangled latent coordinates, we apply a fixed invertible linear mixing $z = Ms$, where M is a random orthogonal matrix. Here, z denotes the entangled latent representation. Naive recombination swaps 2D blocks directly in z-space between independent samples and then decodes using M^{-1}. Because blockwise recombination is not aligned with the true factorization, this operation generically violates the interaction constraints above, producing off-manifold samples (Fig. 11, middle column).

We introduce a learnable linear reparameterization $\hat{s} = Wz$, perform recombination in $\hat{s}$-coordinates, and map back via W^{-1} before decoding. The matrix W is trained adversarially using a discriminator on x, which distinguishes real samples from recombined ones; a weak regularizer encourages $W^\top W \approx I$. This setup mirrors discriminator feedback used in diffusion-based recombination, while remaining analytically transparent. Although this does not strictly guarantee invertibility, W does not become ill-conditioned in practice in our toy setting. We leave parameterizations with guaranteed invertibility (e.g., orthogonal or flow-based constructions) to future work.

To quantify deviation from the data manifold, we compute the squared Mahalanobis distance $d_M(x) = (x-\mu)^\top \Sigma^{-1}(x-\mu)$, where μ and Σ are the empirical mean and covariance of real samples in x-space. This metric measures how inconsistent a sample is with the second-order statistics of real data, independent of marginal scaling. Discriminator feedback substantially reduces the mean Mahalanobis distance of recombined samples, indicating recovery of a recombination-consistent coordinate system.

Architecture and training details. The discriminator $D : \mathbb{R}^4 \to [0, 1]$ is implemented as a lightweight multilayer perceptron with two hidden layers of width 64, ReLU activations, and a sigmoid output. It is trained using binary cross-entropy to distinguish real samples $x = f(s)$ from recombined samples obtained by blockwise recombination in latent space. We emphasize that D has limited capacity and operates purely in observation space, serving only to detect violations of the data manifold induced by recombination.

The reparameterization matrix $W \in \mathbb{R}^{6\times 6}$ is trained to minimize

$$\mathcal{L}(W) = \mathbb{E}_{\text{recomb}} \left[-\log D \left(f \left(M^{-1} W^{-1} \text{recombine}(Wz) \right) \right) \right] + \lambda \| W^\top W - I \|_F^2, \tag{9}$$

where the expectation is taken over randomly paired samples and random blockwise recombinations. The decoder f and mixing matrix M are fixed throughout training. We omit the reconstruction loss to isolate the adversarial recombination signal. Although the decoder $x = f(s)$ is deterministic, it is not globally invertible because it maps $s \in \mathbb{R}^6$ to $x \in \mathbb{R}^4$. Thus, this experiment tests whether adversarial feedback favors recombinations that stay on the data manifold.

Recombination is performed by swapping entire two-dimensional blocks corresponding to latent components between independently sampled latent vectors within a minibatch. All models are trained using Adam with learning rate 10^{-3}, batch size 256, and trained for 5,000 iterations. Results are stable across random initializations of M and W.

D. Additional Experimental Details

D.1. Image Training Details

Hyperparameters and examples of recombined images are provided.
Baseline models have the same settings as the one from Decomp Diffusion. For each dataset, we trained for 20k steps using a batch size of 16; for Falcor3D, we trained for 40k steps due to the higher 128×128 image resolution

Discriminator is adapted from DCGAN architecture. The discriminator is a convolutional neural network that maps an RGB image $x \in \mathbb{R}^{3\times H\times W}$ to a scalar in $(0, 1)$. Images are resized to $H = W = 64$ (except for Falcor3D, where we use $H = W = 128$). The network consists of four strided convolutional blocks followed by a linear discriminator:

- **Conv1:** 4×4 convolution, $3 \to 64$ channels, stride 2, padding 1, followed by LeakyReLU(0.2).
- **Conv2:** 4×4 convolution, $64 \to 128$ channels, stride 2, padding 1, followed by BatchNorm and LeakyReLU(0.2).
- **Conv3:** 4×4 convolution, $128 \to 256$ channels, stride 2, padding 1, followed by BatchNorm and LeakyReLU(0.2).
- **Conv4:** 4×4 convolution, $256 \to 512$ channels, stride 2, padding 1, followed by BatchNorm and LeakyReLU(0.2).

Each convolution reduces the spatial resolution by a factor of 2, yielding a final feature map of size $512 \times (H/16) \times (W/16)$. For 64×64 inputs, this corresponds to a $512 \times 4 \times 4$ tensor. The final feature map is flattened and passed through a fully connected layer of size $512 \cdot (H/16) \cdot (W/16) \rightarrow 1$, followed by a sigmoid activation to produce a scalar output.

An interesting future direction is doing in-depth comparisons of how the model architectures change the effectiveness of the discriminator.

D.2. Additional Qualitative Results

We show side-by-side comparisons between recombined samples produced by our method and those from Decomp Diffusion in Fig. 3a. Overall, our method yields recombined images that are more physically plausible and visually coherent. In contrast, Decomp Diffusion can exhibit off-manifold artifacts under recombination; for example, in Falcor3D, wall textures are inconsistently overlaid, suggesting that global scene factors such as layout and appearance are not coherently composed.

Fig. 2 illustrates a representative failure mode of Decomp Diffusion under latent recombination. While the baseline produces visually reasonable samples when generating from a single latent source, recombining latent factors from multiple images often leads to inconsistent global structure. In particular, in Falcor3D, camera and layout information are entangled across multiple latent dimensions, causing wall textures and scene geometry to be inconsistently overlaid when latents are mixed. This highlights the lack of an explicit mechanism in the baseline to enforce compositional consistency across global scene factors. For CLEVR and CELEBA-HQ, we further do a comparison with COMET in Fig. 8, showing our method's superior performance.

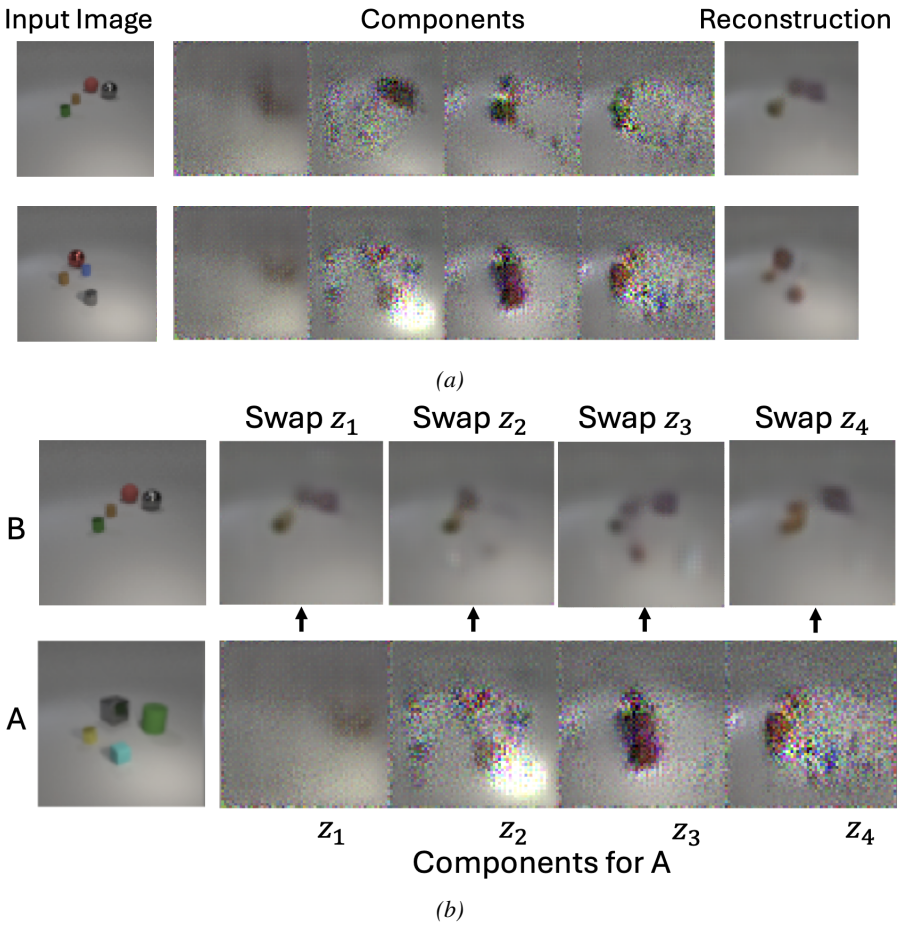


Figure 12. **COMET decomposition and recombination on CLEVR.** We visualize the decomposition learned by COMET on CLEVR. The resulting components are highly entangled, making individual objects difficult to identify, and the reconstruction quality is poor. Consequently, recombination is difficult to interpret: when swapping one latent z_k from image A into image B at a time, the recombined images do not show clear, localized, or object-level changes.

(a) COMET

(b) Decomp Diffusion

Figure 13. **Swapping individual latent components across images.** In each of the subfigures, we fix a reference image A (top left) and a set of other images (top). Below, each row shows the result of replacing a single latent component of A with the corresponding component from another image, while keeping all remaining components fixed. The results illustrate how changing individual latent components affects specific aspects of the generated image. The COMET and Decomp Diffusion subfigures exhibit recombinations that are physically incoherent. See Fig. 14 for a visual comparison illustrating how our method behaves under the same latent component swaps.

E. Video Planning

E.1. Video Training Details

The model comprises a 3D convolutional video encoder and a latent-conditioned 3D UNet diffusion model. The video encoder consists of an initial $3 \times 3 \times 3$ convolution followed by 3 encoding stages. Each stage applies a residual 3D convolutional block and performs spatial downsampling via a strided $3 \times 3 \times 3$ convolution with stride $(1, 2, 2)$, doubling the channel width at each stage. Starting from 64 channels, the encoder increases the channel dimension to 512 while

Figure 14. **Latent component swaps for our method.** Same visualization protocol as Fig. 13. In contrast, our recombinations remain physically coherent, with lighting and camera-related factors varying independently without introducing geometric or illumination inconsistencies.

preserving the temporal dimension. The resulting feature map is mean-pooled over time and flattened, then projected by a fully connected layer to a latent vector of dimension $K \cdot d$, where K is the number of latent components.

Video generation is performed by a 3D UNet diffusion model with channel multipliers $(1, 2, 4)$ and one residual block per resolution level. The UNet operates on inputs of shape (B, C, F, H, W) and includes attention blocks at selected spatial resolutions. Diffusion timestep embeddings are computed via sinusoidal embeddings followed by a two-layer MLP. The latent vector is reshaped into K components and concatenated with the timestep embedding, and this combined embedding is injected into every residual block of the UNet. For compositional decoding, the input and conditioning embeddings are replicated across components, denoising predictions are computed independently per component, and the final output is obtained by averaging predictions across components. First-frame conditioning is implemented by concatenating the initial RGB frame across the temporal dimension at the UNet input.

E.2. Discriminator Architecture Details

The discriminator model is a convolutional neural network designed for processing temporally stacked image frames. It employs a series of pseudo-3D convolutional layers that decompose spatiotemporal feature extraction into separate spatial and temporal convolutions. Each convolutional block consists of a spatial `Conv2D` operation followed by a temporal `Conv1D` operation, enabling efficient modeling of both spatial structures and temporal dependencies.

The input to the model is a sequence of $F = 7$ frames, each with $C = 3$ color channels, forming a five-dimensional tensor of shape (B, C, F, H, W). The frames are processed by four convolutional stages, where the number of feature channels increases progressively from 64 to 512. Each stage applies a spatial convolution with a 3×3 kernel and a stride of $(1, 2, 2)$, reducing the spatial resolution while maintaining the temporal dimension. The spatial convolutions are followed by temporal `Conv1D` operations with a kernel size of 3, capturing temporal dependencies across adjacent frames.

After the final convolutional stage, the feature map has 512 channels and a reduced spatial size of $(H/16, W/16)$. The output is then flattened and passed through a fully connected layer with $512 \times 7 \times (H/16) \times (W/16)$ input dimensions, mapping to a single scalar output. The model does not use batch normalization and employs `LeakyReLU` activations ($\alpha = 0.2$) after each convolutional layer. The final output represents a classification score, which can be used for adversarial

Algorithm 2 Compositional video training with single-source vs recombined discriminator

Input: dataset $\mathcal{D}$, encoder E_ϕ, denoiser ϵ_θ, discriminator D_ψ, weight λ
while not converged **do**
  Sample two trajectories $x^A_{1:N}, x^B_{1:N} \sim \mathcal{D}$ and a recombination mask $S \sim \mathrm{Unif}\big(\{0,1\}^K \setminus \{\mathbf{0},\mathbf{1}\}\big)$
  Encode latents $z^A_{1:K} \leftarrow E_\phi(x^A_{2:N}), z^B_{1:K} \leftarrow E_\phi(x^B_{2:N})$
  Recombine latents: $\tilde{z}_k \leftarrow S_k z^A_k + (1 - S_k) z^B_k$ for $k = 1, \ldots, K$
  Sample $t \sim \mathrm{Unif}(\{1, \ldots, T\})$, $\epsilon \sim \mathcal{N}(0, I)$
  Forward diffusion: $x^{A,(t)} \leftarrow \sqrt{\bar{\alpha}_t}\, x^A_{1:N} + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$
  Single-source: $\hat{x}^{\text{single}} \leftarrow \bar{G}_\theta(x^{A,(t)}, t, x^A_1, z^A_{1:K})$ by conditioning on $(x^A_1, z^A_{1:K})$
  Recombined: $\hat{x}^{\text{recomb}} \leftarrow \bar{G}_\theta(x^{A,(t)}, t, x^A_1, \tilde{z}_{1:K})$ conditioning on $(x^A_1, \tilde{z}_{1:K})$
  Compute diffusion loss $\mathcal{L}_{\text{diff}}$ on $(x^A_{1:N})$ {standard diffusion training}
  Discriminator update (single-source/recombined):
  $L_{\text{clf}}(\psi) = -\mathbb{E}\left[\log D_\psi(\hat{x}^{\text{single}}_{1:N}) + \log\big(1 - D_\psi(\hat{x}^{\text{recomb}}_{1:N})\big)\right]$
  $\psi \leftarrow \psi - \eta_\psi \nabla_\psi L_{\text{clf}}$
  Generator/encoder update:
    $\mathcal{L}_{\text{adv}}(\theta, \phi) = -\mathbb{E}\left[\log D_\psi(\hat{x}^{\text{recomb}}_{1:N})\right]$
    $\theta, \phi \leftarrow \theta, \phi - \eta \nabla_{\theta,\phi}\left(\mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{adv}}\right)$

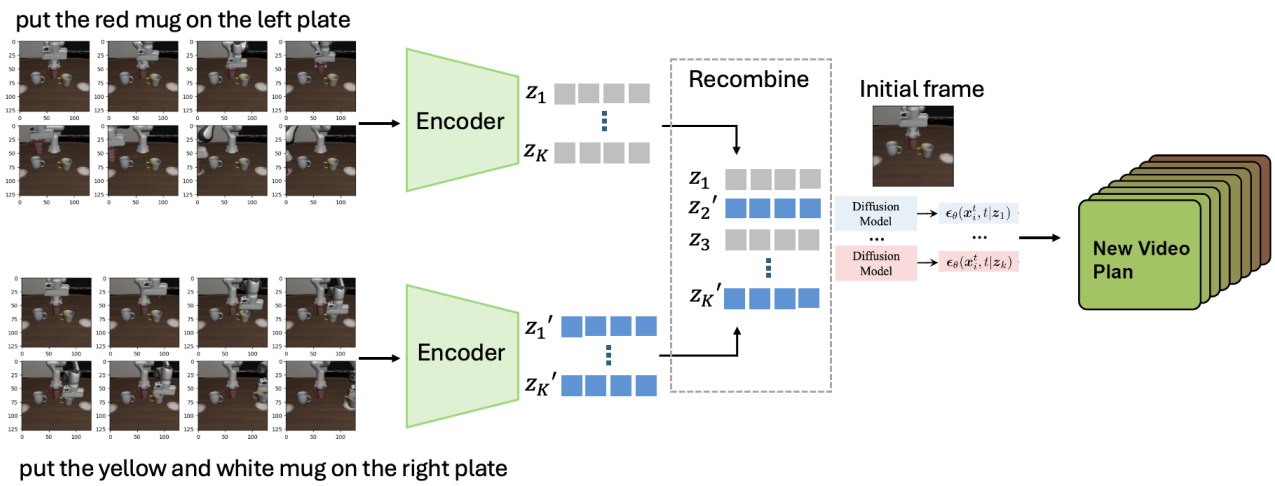


Figure 15. Demonstration videos are encoded into factorized latent components. Latents corresponding to different action primitives or object interactions are recombined, then decoded by a first-frame-conditioned diffusion model to generate novel, task-consistent video plans.

training or other video-based tasks.

E.3. Libero

Scene 5 and 6. Scene 5 contains mug placement tasks in which the robot is instructed to place a mug onto a specified plate. The task variants include: "Put red mug on left plate", "Put red mug on right plate", "Put the white mug on the left plate", and "Put the yellow and white mug on the right plate". Each task differs only in the target plate location and/or the mug's appearance, while the overall manipulation structure remains the same.

Scene 6 contains object placement tasks involving a plate as a reference target. The task variants include: "Put chocolate pudding to left of plate", "Put chocolate pudding to right of plate", "Put the red mug on the plate", and "Put the white mug on the plate". These tasks differ in object type and spatial relation relative to the plate.

Given a generated video sequence $\hat{X} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$, we extract a corresponding action sequence $a_t = \pi_{\text{diff}}(\hat{x}_t, \hat{x}_{t+1})$, for all $t \in \{1, \ldots, N-1\}$ where π_{diff} is the trained diffusion policy, and a_t is the extracted actions between consecutive

frames. We evaluate the quality of generated robotic video rollouts along two axes: physical consistency with the environment dynamics and diversity of explored behaviors. To assess physical realism, we measure the discrepancy between executed vs generated frames after executing inferred actions. These actions are executed in the Libero environment, producing an actual execution trajectory $X^E = (x_1^E, x_2^E, \ldots, x_N^E)$.

For each generated rollout, we apply an inverse dynamics or action-inference model to recover a sequence of actions from the generated frames, and compute a loss L_{phys} between the generated frames and the actual frames generated by these inferred actions (which we get from running the simulator). We use the pretrained diffusion policy model (inverse dynamics) from the AVDC codebase, trained on LIBERO Scenes 5–6 demonstration data, and keep it fixed across all methods. Architecturally, π_{diff} encodes the RGB observation with an adapted ResNet-18 with group normalization and spatial softmax pooling, and uses a 1D-convolutional FiLM U-Net diffusion backbone conditioned on the observation embedding (Ko et al., 2024). The loss between generated and executed frames is $L_{\text{phys}} = \frac{1}{N}\sum_{t=1}^{N} \|x_t^E - \hat{x}_t\|_2^2$ where x_t^E is the real frame from the Libero execution, $\hat{x}_t$ is the generated recombined frame, and N is the total number of frames. Lower values of L_{phys} indicate better adherence to the underlying physical dynamics. To quantify exploration breadth, we measure coverage in discretized joint-state space. We partition the continuous robot state into bins of resolution Δ and compute the number of bins visited by generated rollouts. The resulting metric Cov_Δ captures the diversity of reachable states induced by recombination, with higher values indicating broader exploration. Unless otherwise stated, all hyperparameters (including bin resolution Δ and action inference models) are held fixed across methods.

Warm-up phase. The warm-up ensures the robot reliably grasps the object before executing generated plans. Without it, recombined trajectories frequently fail to grasp the object and rollouts terminate early without meaningfully exploring the scene. This reflects a current limitation of our approach, which focuses on recombining and executing trajectories after object pickup; addressing reliable grasping jointly with compositional planning is a direction for future work.

E.4. Additional Results

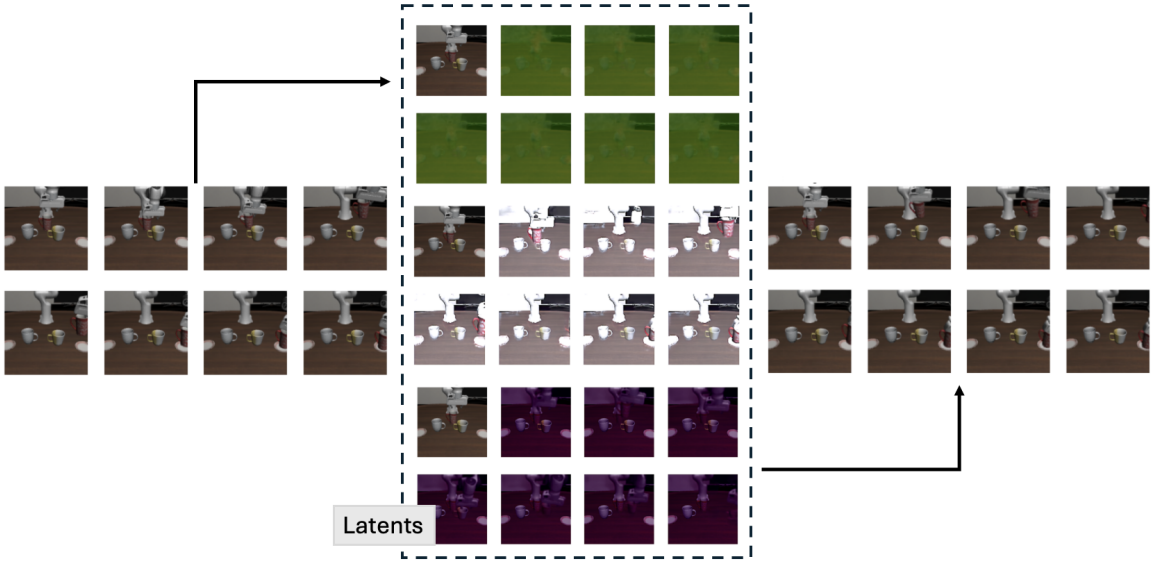


Figure 16. $\lambda = 0.003$: Good and meaningful representations. Reconstruction quality is fair. See decomposed components and reconstruction quality for large lambdas in Fig. 18b.

E.5. Qualitative assessment

In Fig. 6, we present the decomposition of an input sequence of video frames using a compositional diffusion generative model fine-tuned with a discriminator. By isolating individual components, the model effectively learns to conditionally disentangle distinct concepts based on the initial frame. When analyzing the visualizations of the latent components, we infer that the first latent primarily encodes the static elements of the scene—namely, the two plates and three mugs—while rendering the robot arm indistinct and blurry. The second latent component, in contrast, predominantly whitens the entire

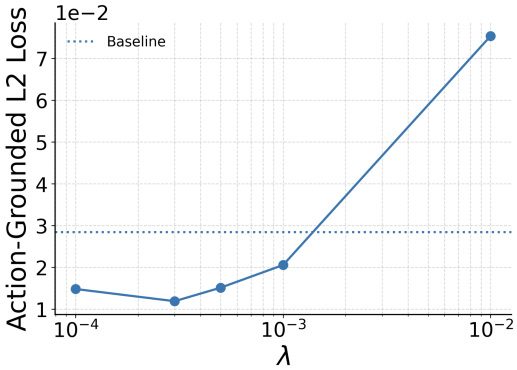


Figure 17. Effect of discriminator weight strength on Action-Grounded L2 Loss. The loss initially decreases as λ increases, reaching an optimal value before rising again, indicating a trade-off between classification guidance and physical realism.

Model	**Scene5** ↑	**Scene6** ↑
Dataset	1603	1759
Baseline	9440	5400
Baseline + Discriminator (λ=0.0001)	9677	6994
Baseline + Discriminator (λ=0.001)	**12816**	**9385**
Baseline + Discriminator (λ=0.01)	7939	6358

Table 5. Comparison of state-space explored through discretization length = 0.03 from 640 generations. Here, baseline refers to training without discriminator feedback.

scene, selectively preserving only the color of the object being manipulated by the robot arm. The third latent component, beyond capturing variations in shadowing and color, accentuates the dynamic aspects of the scene, particularly tracing the trajectory of the robot arm and its direct interaction with the manipulated object, while diminishing emphasis on the surrounding environment.

E.6. State-space coverage and the inverted-U effect

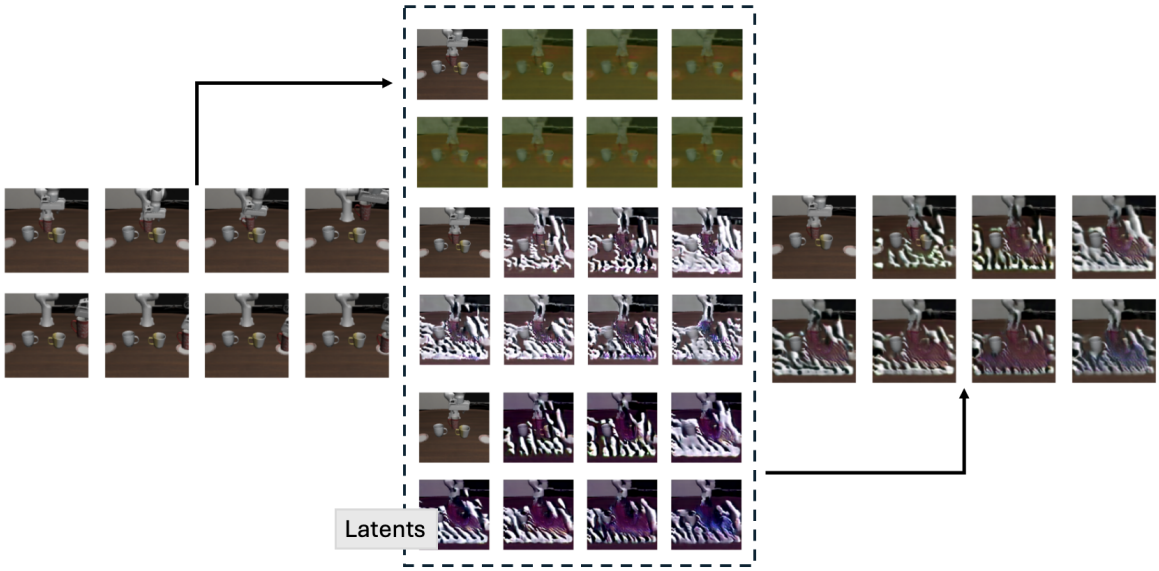
We discretize executed outcomes into a finite set of state bins $\mathcal{B}$ and let $q_\lambda(b)$ denote the probability that an executed rollout falls in bin b when sampling is performed with discriminator weight λ. If we execute n i.i.d. rollouts, the expected number of distinct bins visited is

$$\mathbb{E}[\mathrm{Cov}_n] \;=\; \sum_{b\in\mathcal{B}} \Big(1 - \big(1 - q_\lambda(b)\big)^n\Big). \tag{10}$$

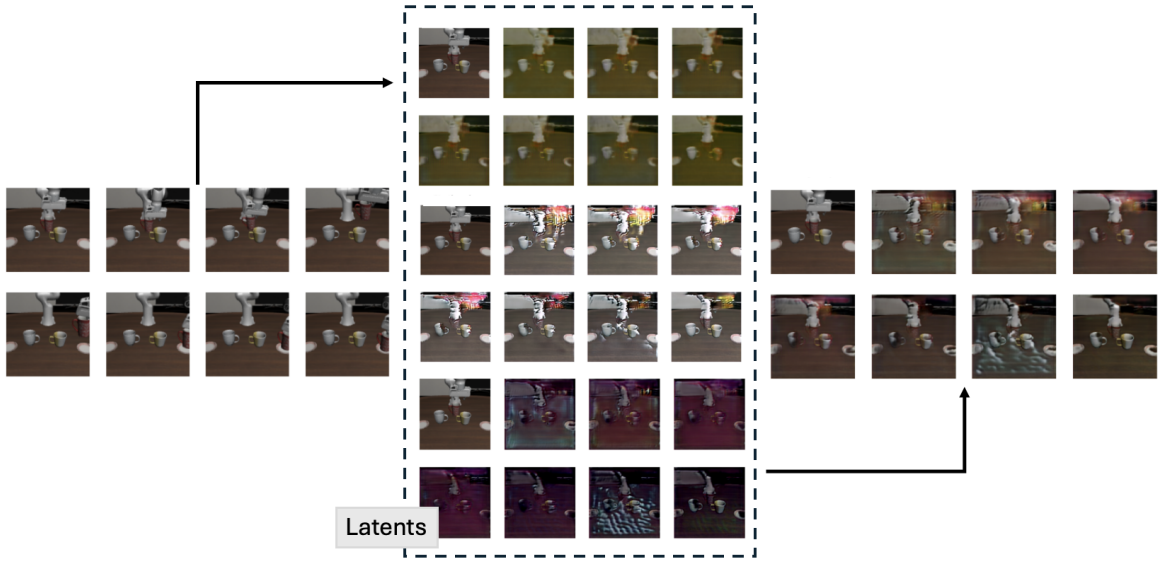
For $n \geq 2$, the function $u \mapsto 1-(1-u)^n$ is increasing and concave on $[0,1]$, so (10) is larger when q_λ spreads mass across many bins and smaller when q_λ concentrates on a few.

To relate this to discriminator feedback, partition $\mathcal{B} = \mathcal{F} \cup \mathcal{S}$ into a small set of *failure* bins $\mathcal{F}$ (e.g., resets/timeouts/trivial end states) and a typically much larger set of *successful/executable* bins $\mathcal{S}$. In practice, unguided recombination often produces many off-manifold videos that, after action extraction and execution, collapse into a small number of failures in $\mathcal{F}$. As λ increases from 0, guidance preferentially suppresses these off-manifold generations, shifting mass in q_λ away from the low-cardinality failure set and into the diverse successful set. This "filtering" effect tends to increase (10) at small-to-moderate λ.

For overly large λ, the sampling distribution can become sharply concentrated on a small set of very low-energy ("safe"/repetitive) videos. Execution then concentrates q_λ on only a few bins inside $\mathcal{S}$, decreasing (10) by concavity. The combination of initial filtering (less mass on failures) and eventual concentration among successes yields an inverted-U dependence of coverage on λ, with a maximum at an intermediate discriminator weight.



(a) $\lambda = 0.01$: Decomposed components lose meaning.



(b) $\lambda = 0.1$: Decomposed components become noisier.

Figure 18. Decomposition and Reconstruction Quality for different lambda values. For large lambdas, the decomposition degrades, leading to noisier reconstructed video plans.

F. A Theory of Recombination

F.1. Additive Energies of Features is Equivalent to Product of Experts

We begin by formalizing a standard equivalence used throughout compositional energy-based modeling. An energy-based model (EBM) defines an unnormalized density $p(x) \propto \exp(-E(x))$ such that $p(x) = \frac{\exp(-E(x))}{Z}$, where $Z = \int \exp(-E(x))\mathrm{d}x$ denotes the normalizing constant (or the partition function) and $E(x)$ (the energy) is a nonlinear regression function with some latent parameter θ (LeCun et al., 2006; Hinton, 2002; Song & Kingma, 2021; Du et al., 2020). We show that additive energies yield a product-of-experts distribution.

A product-of-experts (PoE) model combines K non-negative expert potentials $\{\tilde{p}_k(x)\}_{k=1}^{K}$ over the same variable $x \in \mathcal{X}$, via

$p_{PoE}(x) := \frac{1}{Z}\prod_{k=1}^{K}\tilde{p}_k(x)$, where $Z = \int_{\mathcal{X}}\prod_{k=1}^{K}\tilde{p}_k(u)\mathrm{d}u$ is the partition function. We show that our discriminator-based feedback can be interpreted as shaping the experts so that recombined expert sets still yield sufficient probability mass in the PoE intersection, which improves the fidelity of the recombination.

Lemma F.1 (Additive energies form a PoE distribution)**.** *Let $\{E_k(x)\}_{k=1}^{K}$ be a collection of energy functions and $\{\alpha_k\}_{k=1}^{K}$ be a collection of non-negative coefficients. Define the weighted aggregate energy $E(x) := \sum_{k=1}^{K}\alpha_k E_k(x)$. Next, for $i \in \{1,\dots,K\}$, we define the sequence of unnormalized densities $\tilde{p}_i(x) = \exp(-E_i(x))$. Then, the induced aggregate*

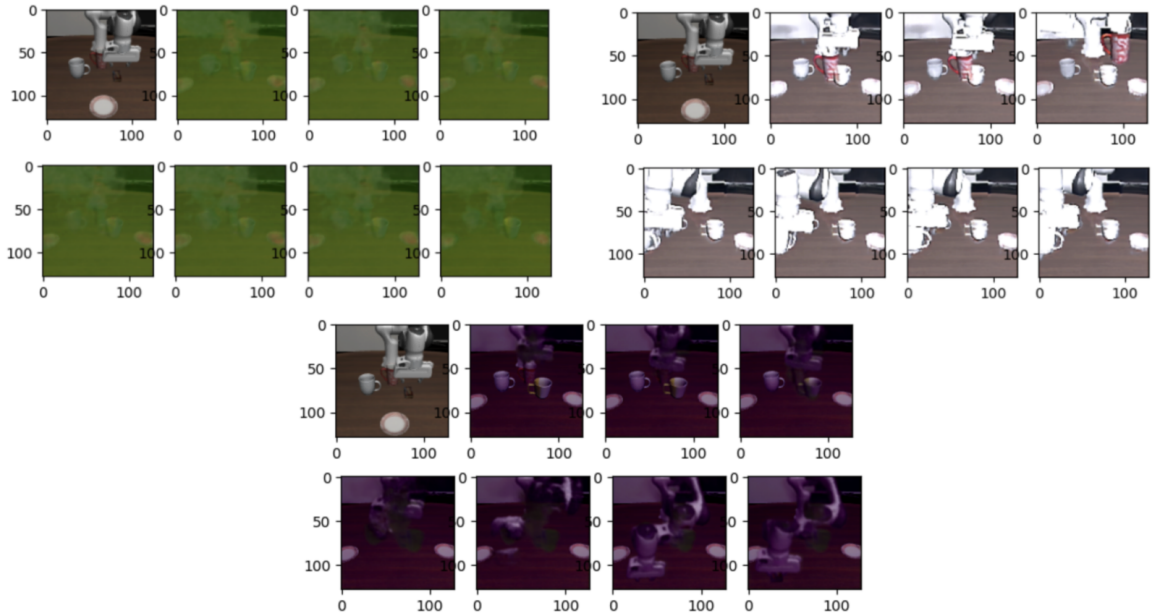


Figure 19. Visualizations of latent factors for the task 'put the red mug on the left plate'. Each row corresponds to a different recombined trajectory, and columns show selected frames from the rollout. Our interpretation is as follows – (top left): the robot arm is not visible and only the object configuration is shown. (top right): beginning and end of the executed trajectory. (bottom): intermediate frames from the middle of the trajectory.

Figure 20. Visualizations of latents from model trained on four factors. The top right and bottom left emphasize actions near the initial and final interactions between the robotic arm and the red mug. The bottom-right frame primarily captures the static configuration of objects that remain untouched during execution. The top right frame highlights intermediate stages of the trajectory, where the robot arm is actively manipulating the red mug, while background objects are less prominently represented.

model is a product of experts

$$\tilde{p}(x) = \prod_{k=1}^{K} \tilde{p}_k(x)^{\alpha_k}.$$

Proof. Note that

$$\begin{aligned}\tilde{p}(x) &= \exp\Big(-\sum_{k=1}^{K} \alpha_k E_k(x)\Big) \\ &= \prod_{k=1}^{K} \exp(-\alpha_k E_k(x)) \\ &= \prod_{k=1}^{K} \tilde{p}_k(x)^{\alpha_k}.\end{aligned}$$

To show that $\tilde{p}(x)$ is a realizable unnormalized density, for $k \in [K]$ let p_k be the normalized density corresponding to $\tilde{p}_k$ such that $p_k = \tilde{p}_k(x)/\int \tilde{p}_k(x)\mathrm{d}x := \tilde{p}_k(x)/Z_k$ where Z_k is the partition function of $\tilde{p}_k$. Then, $\tilde{p}_k^{\alpha_k} = (Z_k p_k)^{\alpha_k} = Z_k^{\alpha_k} p_k^{\alpha_k}$. Then, the constants $\prod_{k=1}^{K} Z_k^{\alpha_k}$ are absorbed into the partition function $Z = \int \prod_k \tilde{p}_k(x)^{\alpha_k}\mathrm{d}x$, yielding

$$p(x) = \frac{\prod_{k=1}^{K} \tilde{p}_k(x)^{\alpha_k}}{\prod_{k=1}^{K} Z_k^{\alpha_k}} = \frac{\prod_{k=1}^{K} \tilde{p}_k(x)^{\alpha_k}}{Z},$$

which proves the lemma. □

Remark F.2 (Naïve recombination artifacts). A PoE assigns high probability only to points that satisfy all experts simultaneously. If two experts encode incompatible constraints (for instance, if the camera pose information is split across multiple components) their product can assign near-zero density to the recombined configuration, which introduces artifacts in the recombination. This matches the empirical observation that naïve recombination can create inconsistent samples.

Remark F.3 (Additive diffusion). If each expert induces a differentiable density $p_k(x)$, then the PoE score decomposes additively $\nabla_x \log p(x) := \sum_{k=1}^{K} \alpha_k \nabla_x \log p_k(x)$ since

$$\log p(x) = \sum_k \alpha_k \log p_k(x) + \sum_k \alpha_k \log Z_k - \log Z.$$

Therefore, the parameterization in which the denoising network is a sum of per-component contributions can be viewed as learning a PoE over component-conditioned experts.

F.2. Recombining Subsets of Latent Representations

We consider the event where arbitrary subset recombinations stay on-manifold, and show that this implies a structural factorization in latent space. To begin, let $\mathcal{M} \subset \mathcal{X}$ denote the data manifold which is the support of p_{data}. For $i \in [K]$, let $Z_i \subseteq \mathbb{R}^{d_i}$ denote the latent space corresponding to the i'th latent factor. Let $\mathrm{Enc} : \mathcal{M} \to \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ be an encoder and $f : \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K \to \mathcal{X}$ be a decoder. Define the set of valid codes $\mathcal{Z} := \mathrm{Enc}(\mathcal{M})$ and projections $\pi_k(\mathcal{Z}) \subseteq \mathcal{Z}_k$. Next, define the coordinate-wise recombination operator for a mask $S \in \{0,1\}^K$ by

$$\mathrm{Mix}_S(z, z') := S \odot z + (1 - S) \odot z',$$

such that each coordinate is selected from one of the two codes. We then show that closure under coordinate-wise recombination implies that the latent support factorizes as a Cartesian product.

Proposition F.4 (Closure under subset recombination implies Cartesian-product support)**.** *Assuming (i) $f(\mathrm{Enc}(x)) = x$ for all $x \in \mathcal{M}$ and (ii) for all $z, z' \in \mathcal{Z}$ and all masks $S \in \{0,1\}^K$, $\mathrm{Mix}_S(z, z') \in \mathcal{Z}$, we have that the valid code set factorizes as a Cartesian product such that $\mathcal{Z} = \pi_1(\mathcal{Z}) \times \cdots \times \pi_K(\mathcal{Z})$. Hence, every recombination of per-component values that appears on the manifold can be recombined without leaving the manifold.*

Proof. Let $a_k \in \pi_k(\mathcal{Z})$ be arbitrary. By definition of the projection, for each k there exists a code $z^{(k)} \in \mathcal{Z}$ whose k'th coordinate is equal to a_k. Initialize $\bar{z} \leftarrow z^{(1)}$. For $k = 2, \ldots, K$, define a mask $S^{(k)}$ that selects coordinate k from $z^{(k)}$ and all other coordinates from $\bar{z}$. By closure, $\bar{z} \leftarrow \mathrm{Mix}_{S^{(k)}}(\bar{z}, z^{(k)}) \in \mathcal{Z}$ after each step. At the end, $\bar{z}$ contains coordinates $(a_1, \ldots, a_K)$ and lies in $\mathcal{Z}$, which gives $\pi_1(\mathcal{Z}) \times \cdots \times \pi_K(\mathcal{Z}) \subseteq \mathcal{Z}$. The reverse inclusion $\mathcal{Z} \subseteq \pi_1(\mathcal{Z}) \times \cdots \times \pi_K(\mathcal{Z})$ then follows by definition of our projection operators. Together, this proves the proposition. □

Remark F.5. The above formalizes a stronger notion of independent controllability, where each latent component can vary independently on the learned manifold. This is a structural condition that is consistent with higher disentanglement scores on datasets where ground-truth factor space has an approximate product structure.

F.3. Contraction of Pairwise Mutual Information through Recombination

We show that random subset recombination reduces statistical dependence between latent components, in the sense that pairwise mutual information contracts by a factor determined by the probability of selecting both components from the same source sample.

Definition F.6 (Mutual information)**.** For random variables X and Y with joint distribution $p_{X,Y}$ and marginals p_X and p_Y (respectively), the mutual information between X and Y is given by

$$I(X;Y) \coloneqq \mathrm{KL}(p_{X,Y} \| p_X p_Y) = \mathbb{E}_{(X,Y)\sim p_{X,Y}} \left[\log \frac{p_{X,Y}(X,Y)}{p_X(X) p_Y(Y)} \right].$$

To proceed, let $Z = (Z_1, \ldots, Z_K) \sim P_Z$ denote a random latent code (e.g., the aggregated posterior of $\mathrm{Enc}(X)$) (as defined in Appendix F.2), and let $Z^A, Z^B \sim P_Z$ (i.i.d.). Next, let $S \in \{0,1\}^K$ be a random mask independent of Z^A, Z^B. Define the recombined code $\tilde{Z}$ component-wise as $\tilde{Z}_k \coloneqq S_k Z_k^A + (1 - S_k) Z_k^B$ where $k = 1, \ldots, K$.

Lemma F.7 (Pairwise mutual information contracts under recombination)**.** *Fix two indices $i \neq j$. Let $\alpha_{ij} \coloneqq \Pr[S_i = S_j]$. Then, we have that*

$$\mathcal{I}(\tilde{Z}_i; \tilde{Z}_j) \leq \alpha_{ij} \cdot \mathcal{I}(Z_i; Z_j).$$

Proof. Since Z^A and Z^B are independent, the pair $(\tilde{Z}_i, \tilde{Z}_j)$ admits a mixture form where with probability α_{ij}, both coordinates come from the same draw (either A or B), yielding the correlated joint distribution P_{Z_i,Z_j}; with probability $1 - \alpha_{ij}$ they come from different draws, yielding independence $P_{Z_i} P_{Z_j}$. Thus, we write

$$P_{\tilde{Z}_i,\tilde{Z}_j} = \alpha_{ij} P_{Z_i,Z_j} + (1 - \alpha_{ij}) P_{Z_i} P_{Z_j}. \tag{11}$$

Since $P_{\tilde{Z}_i} = P_{Z_i}$ and $P_{\tilde{Z}_j} = P_{Z_j}$, we can write

$$\mathcal{I}(\tilde{Z}_i; \tilde{Z}_j) = \mathrm{KL}\left(P_{\tilde{Z}_i,\tilde{Z}_j} \| P_{Z_i} P_{Z_j} \right). \tag{12}$$

By convexity of $\mathrm{KL}(\cdot\|\cdot)$ in its first argument,

$$\begin{aligned}
\mathcal{I}(\tilde{Z}_i; \tilde{Z}_j) &= \mathrm{KL}\left(\alpha_{ij} P_{Z_i,Z_j} + (1 - \alpha_{ij}) P_{Z_i} P_{Z_j} \| P_{Z_i} P_{Z_j} \right) && (13)\\
&\leq \alpha_{ij} \cdot \mathrm{KL}\left(P_{Z_i,Z_j} \| P_{Z_i} P_{Z_j} \right) + (1 - \alpha_{ij}) \cdot \mathrm{KL}\left(P_{Z_i} P_{Z_j} \| P_{Z_i} P_{Z_j} \right) && (14)\\
&= \alpha_{ij} \cdot \mathcal{I}(Z_i; Z_j), && (15)
\end{aligned}$$

which proves the lemma. Therefore, random subset recombination mixes correlated latents with independent latents (from different samples), thereby shrinking dependence. Hence, enforcing that decoded samples from recombined latents remain on-manifold provides an implicit pressure toward latents that are more "factorial" (lower dependence), which aligns with common disentanglement objectives. □

F.4. A discriminator view of single-source noising

We analyze the adversarial term used in training. Recall that the recombined prediction is

$$\hat{x}_{\text{recomb}}(x_t, \tilde{z}) := \frac{1}{K} \sum_{k=1}^{K} G_\theta(x_t, t, \tilde{z}_k),$$

and the generator-side adversarial loss is

$$L_{\text{adv}} = -\mathbb{E}\big[\log D_\psi(\hat{x}_{\text{recomb}}(x_t, \tilde{z}))\big].$$

In our implementation, the recombined latent code $\tilde{z}$ is paired with the noised source input x_t^A, rather than with the unavailable noised recombined observation $x_t^\star$. The next result shows that this implemented objective remains close to the ideal one under mild stability assumptions.

Step 1: optimal discriminator as a density-ratio classifier. Fix t and the conditioning variables that determine the latent codes. Let p_{single} and p_{recomb} denote the densities of

$$\hat{x}_{\text{single}} := \hat{x}(x_t, z_{1:K}^A), \qquad \hat{x}_{\text{recomb}} := \hat{x}(x_t, \tilde{z})$$

under the implemented training distribution. For the logistic classification objective

$$L_{\text{clf}}(\psi) = -\mathbb{E}_{u\sim p_{\text{single}}}[\log D_\psi(u)] - \mathbb{E}_{u\sim p_{\text{recomb}}}[\log(1 - D_\psi(u))],$$

the pointwise optimal discriminator satisfies

$$D_\psi^*(u) = \frac{p_{\text{single}}(u)}{p_{\text{single}}(u) + p_{\text{recomb}}(u)}.$$

Hence

$$-\log D_\psi^*(u) = \log\left(1 + \frac{p_{\text{recomb}}(u)}{p_{\text{single}}(u)}\right).$$

Thus, under an optimal discriminator, the adversarial loss penalizes recombined predictions that fall in regions where their density is large relative to the single-source prediction density.

Proposition F.8 (Implemented adversarial loss is a controlled surrogate)**.** *Let*

$$\hat{x}(x_t, z_{1:K}) := \frac{1}{K} \sum_{k=1}^{K} G_\theta(x_t, t, z_k).$$

Assume:

(i) (stability in the noised input) *for every t and $z_{1:K}$,*

$$\|\hat{x}(x_t, z_{1:K}) - \hat{x}(x_t', z_{1:K})\|_2 \le L_x(t, z_{1:K})\|x_t - x_t'\|_2;$$

(ii) (approximate modularity in the latents) *for every t and x_t,*

$$\|\hat{x}(x_t, \tilde{z}) - \hat{x}(x_t, z^\star)\|_2 \le \frac{1}{K} \sum_{k=1}^{K} L_k(t, x_t)\|\tilde{z}_k - z_k^\star\|_2,$$

where $z^\star$ denotes the latent code that would be obtained from the unavailable clean recombined observation;

(iii) (smooth discriminator score) *the map $u \mapsto -\log D_\psi(u)$ is L_D-Lipschitz on the range of $\hat{x}$.*

Define the implemented and ideal discriminator-side losses

$$\ell_{\text{impl}} := -\log D_\psi\big(\hat{x}(x_t^A, \tilde{z})\big), \qquad \ell_{\text{id}} := -\log D_\psi\big(\hat{x}(x_t^\star, z^\star)\big).$$

Then

$$|\ell_{\text{impl}} - \ell_{\text{id}}| \le L_D \left[L_x(t,\tilde{z})\|x_t^A - x_t^\star\|_2 + \frac{1}{K}\sum_{k=1}^{K} L_k(t, x_t^\star)\|\tilde{z}_k - z_k^\star\|_2 \right].$$

Consequently,

$$\big|\mathbb{E}[\ell_{\text{impl}}] - \mathbb{E}[\ell_{\text{id}}]\big| \le \mathbb{E}\left[L_D \left(L_x(t,\tilde{z})\|x_t^A - x_t^\star\|_2 + \frac{1}{K}\sum_{k=1}^{K} L_k(t, x_t^\star)\|\tilde{z}_k - z_k^\star\|_2 \right)\right].$$

If the forward process uses the same Gaussian noise realization,

$$x_t^A = \sqrt{\bar{\alpha}_t}x^A + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad x_t^\star = \sqrt{\bar{\alpha}_t}x^\star + \sqrt{1-\bar{\alpha}_t}\,\epsilon,$$

then

$$\|x_t^A - x_t^\star\|_2 = \sqrt{\bar{\alpha}_t}\,\|x^A - x^\star\|_2,$$

and therefore

$$\big|\mathbb{E}[\ell_{\text{impl}}] - \mathbb{E}[\ell_{\text{id}}]\big| \le \mathbb{E}\left[L_D \left(L_x(t,\tilde{z})\sqrt{\bar{\alpha}_t}\,\|x^A - x^\star\|_2 + \frac{1}{K}\sum_{k=1}^{K} L_k(t, x_t^\star)\|\tilde{z}_k - z_k^\star\|_2 \right)\right].$$

Proof. By the triangle inequality,

$$\|\hat{x}(x_t^A, \tilde{z}) - \hat{x}(x_t^\star, z^\star)\|_2 \le \|\hat{x}(x_t^A, \tilde{z}) - \hat{x}(x_t^\star, \tilde{z})\|_2 + \|\hat{x}(x_t^\star, \tilde{z}) - \hat{x}(x_t^\star, z^\star)\|_2.$$

Apply assumption (i) to the first term and assumption (ii) to the second to obtain

$$\|\hat{x}(x_t^A, \tilde{z}) - \hat{x}(x_t^\star, z^\star)\|_2 \le L_x(t,\tilde{z})\|x_t^A - x_t^\star\|_2 + \frac{1}{K}\sum_{k=1}^{K} L_k(t, x_t^\star)\|\tilde{z}_k - z_k^\star\|_2.$$

Since $u \mapsto -\log D_\psi(u)$ is L_D-Lipschitz by assumption (iii),

$$|\ell_{\text{impl}} - \ell_{\text{id}}| \le L_D\,\|\hat{x}(x_t^A, \tilde{z}) - \hat{x}(x_t^\star, z^\star)\|_2,$$

which proves the first bound. Taking expectation gives the second.

Finally, under shared Gaussian forward noise,

$$x_t^A - x_t^\star = \sqrt{\bar{\alpha}_t}\,\big(x^A - x^\star\big),$$

hence

$$\|x_t^A - x_t^\star\|_2 = \sqrt{\bar{\alpha}_t}\,\|x^A - x^\star\|_2.$$

Substituting this identity yields the final result. □

Proposition F.8 separates two sources of approximation error: (i) an *input-surrogate term*, controlled by how much the clean recombined observation differs from the source observation after noising, and (ii) a *latent-modularity term*, controlled by how closely the recombined latent code $\tilde{z}$ matches the latent code $z^\star$ that would be inferred from the unavailable true recombined observation. When both terms are small, the implemented adversarial objective is a good surrogate for the ideal discriminator objective. Combined with the density-ratio interpretation above, this shows that our training rule approximately penalizes recombined predictions according to how atypical they are relative to single-source predictions, using the denoising architecture actually employed by the model.

Impact Statement

Our proposed approach poses no immediate social risks in its current form, as it is primarily evaluated on standard datasets and simulated environments. However, the ability to manipulate latent representations could be misused in ways that amplify biases inherent in the training data, leading to unintended consequences in decision-making systems. Ensuring responsible deployment is crucial to avoid potential ethical risks. Despite these concerns, our approach offers significant potential benefits across various domains, including robotics, automated planning, and creative content generation, by enabling improved generalization and compositional understanding of complex environments.