Disentangled Representation Learning via Flow Matching

Jinjin Chi[1,2,3] **Taoping Liu**[1] **Mengtao Yin**[1] **Ximing Li**[1,3*] **Yongcheng Jing**[2]
Leszek Rutkowski[4,5,6] **Jialie Shen**[7] **Dacheng Tao**[2]
[1]College of Computer Science and Technology, Jilin University
[2]College of Computing and Data Science, Nanyang Technological University
[3]Key Laboratory of Symbolic Computation and Knowledge Engineering, MoE, Jilin University
[4]Systems Research Institute, Polish Academy of Sciences
[5]AGH University of Krakow
[6]SAN University
[7]City St George's, University of London
{chijinjin616@gmail.com, liximing86@gmail.com}

Abstract

Disentangled representation learning aims to capture the underlying explanatory factors of observed data, enabling a principled understanding of the data-generating process. Recent advances in generative modeling have introduced new paradigms for learning such representations. However, existing diffusion-based methods encourage factor independence via inductive biases, yet frequently lack strong semantic alignment. In this work, we propose a flow-matching–based framework for disentangled representation learning, which casts disentanglement as learning factor-conditioned flows in a compact latent space. To encourage semantic separation among factor representations, we introduce a non-overlap (orthogonality) regularizer that suppresses cross-factor interference and reduces information leakage between factors. Extensive experiments across multiple datasets demonstrate consistent improvements over representative baselines, yielding higher disentanglement scores as well as improved controllability and sample fidelity.

1 Introduction

Disentangled representation learning has emerged as a fundamental objective in modern machine learning, playing a pivotal role in advances across computer vision, natural language processing (NLP), and reinforcement learning [Bengio et al., 2013, Wang et al., 2024]. Motivated by the observation that the physical world is governed by a set of independent underlying factors, such as object shape, color, and size, this paradigm aims to encode high-dimensional observations into compact latent representations in which these semantic factors are explicitly separated. As illustrated in Figure 1, a single observation can be explained by multiple distinct factors of variation, here exemplified by seven interpretable generative factors. Such structurally disentangled representations provide a powerful inductive bias for downstream tasks, enabling controllable image synthesis and editing in vision [Xu et al., 2023, Wang et al., 2023], facilitating style manipulation and semantic control in NLP [Cheng et al., 2020, Hu et al., 2022], and supporting robust policy generalization and transfer in robotics [Higgins et al., 2018, Wu et al., 2025]. Consequently, disentangled representation learning has become an active and rapidly growing area within representation learning.

A wide range of disentangled representation learning methods have been proposed to advance this line of research. Among them, generative modeling–based methods have demonstrated par-

*Corresponding author.

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

ticular effectiveness in learning disentangled representations from high-dimensional image data, owing to their ability to explicitly model the data generation process. On the one hand, Variational Autoencoder (VAE)–based methods, such as β-VAE [Higgins et al., 2017] and FactorVAE [Kim and Mnih, 2018], encourage disentanglement by imposing structural regularization on the latent space. However, these methods suffer from a trade-off between reconstruction fidelity and disentanglement strength [Chen et al., 2018, Burgess et al., 2018]. On the other hand, Generative Adversarial Network (GAN)–based methods, including InfoGAN [Chen et al., 2016] and InfoGAN-CR [Lin et al., 2020], pursue disentanglement from a different perspective, typically by inducing semantic structure through controllable latent variables. Despite their remarkable generative capability, GAN-based models generally lack a reversible inference mechanism between data and latent spaces, which makes them less flexible than VAEs for representation learning [Wang et al., 2024].

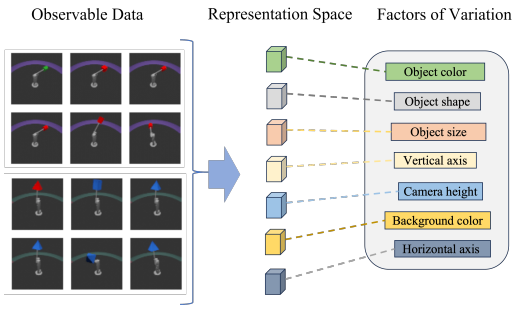


Figure 1: Illustration of the MPI3D-toy dataset [Gondal et al., 2019]. The seven rectangles represent the underlying factors of variation in the scene, including object color, shape, size, camera height, background color, vertical axis and horizontal axis.

More recently, modern generative models, particularly diffusion models, have attracted increasing attention due to their strong modeling capacity and stable training dynamics, opening new opportunities for disentangled representation learning. Existing diffusion-based methods [Wu and Zheng, 2024, Yang et al., 2023b, 2024, Jun et al., 2025] commonly introduce inductive biases to diffusion models to encourage statistical independence across latent dimensions. However, disentanglement in practice requires not only independence but also a semantic alignment criterion: *each latent unit (or subspace) should correspond to a distinct, interpretable generative factor*. When semantic alignment is weak, latent dimensions may remain statistically decorrelated yet still mix factors, limiting interpretability and undermining fine-grained, factor-wise interventions.

To address these limitations, we argue that the deterministic formulation of *flow matching* provides a more principled geometric basis for semantic alignment than stochastic diffusion trajectories. Flow matching learns continuous-time generative dynamics by directly matching probability flow fields, avoiding iterative denoising objectives and the associated noise-schedule design choices. It enables efficient inference through ordinary differential equation (ODE) integration. Building on these structural advantages, we develop a flow-matching framework that encourages semantic separation among factor representations by regularizing token-associated velocity components to reduce overlap. To the best of our knowledge, this is the first study to investigate flow matching for general disentangled representation learning. Our key contributions are three-fold:

- We formulate disentangled representation learning as a factor-conditioned flow in a compact latent space. This enables deterministic generation through ODE-based sampling.
- We introduce an orthogonality regularizer on token-associated velocity components to discourage overlap and redundant factor attributions. This regularization encourages semantic separation among the learned factor representations.
- Experiments on multiple datasets show consistent improvements in disentanglement metrics over representative baselines. Qualitative results further demonstrate improved semantic controllability and sample fidelity.

2 Related Work

VAE-based methods. VAEs [Kingma and Welling, 2014] learn latent-variable generative models by maximizing the evidence lower bound (ELBO), enabling joint inference and generation. Disentanglement is typically achieved by modifying the ELBO to encourage statistical independence among latent dimensions. For example, β-VAE [Higgins et al., 2017] increases the weight of the KL term to promote a factorized prior, while FactorVAE [Kim and Mnih, 2018] and β-TCVAE [Chen et al., 2018] explicitly penalize total correlation. DIP-VAE [Kumar et al., 2018] instead aligns moments of

the aggregated posterior with the prior. Overall, these methods balance reconstruction fidelity and disentanglement through objective reweighting or additional regularization.

GAN-based methods. GAN-based methods achieve disentanglement by introducing structured latent codes within adversarial training [Goodfellow et al., 2014]. InfoGAN [Chen et al., 2016] maximizes mutual information between latent codes and generated samples to capture interpretable factors. Subsequent methods improve controllability via explicit factorization or structured latent spaces, such as separating content and style [Kazemi et al., 2019, Varur et al., 2025]. Architectural advances, e.g., StyleGAN [Karras et al., 2019, 2020], further enable semantically meaningful control. However, these methods often suffer from limited reconstruction ability due to the difficulty of GAN inversion [Wang et al., 2022].

Diffusion- and flow-based methods. Diffusion models have emerged as a leading family of generative models, achieving state-of-the-art fidelity and mode coverage. They generate samples by reversing a gradual noising process through learned denoising or score estimation [Ho et al., 2020, Yang et al., 2023a]. Unlike VAEs and GANs, diffusion-based disentanglement remains relatively underexplored, as diffusion architectures do not naturally yield compact, factorized latent representations. Consequently, existing methods typically rely on strong inductive biases to enforce factor-wise independence [Wu and Zheng, 2024, Yang et al., 2023b, 2024, Jun et al., 2025]. Specifically, Wu and Zheng [2024] decomposes images into content and mask groups to improve interpretability; Yang et al. [2023b] structures the denoising process to separate factors across time steps; Yang et al. [2024] uses cross-attention to route distinct factors; and Jun et al. [2025] enhances separation via dynamic Gaussian anchoring.

Despite these advances, such methods remain constrained by the stochastic nature of diffusion trajectories and the complexity of iterative denoising, leading to high computational cost and limited geometric structure for precise semantic alignment. Recent work explores flow-based formulations for disentanglement. SCFlow [Ma et al., 2025] leverages flow matching for implicit style–content disentanglement via invertible mappings. However, this method is tailored to specific settings and does not provide a general framework. In contrast, our method formulates disentanglement within a general flow-matching framework with explicit factor-level modeling.

3 Background

Let $p_0(\mathbf{x})$ denote a tractable source distribution (typically a standard Gaussian) and let $p_1(\mathbf{x})$ denote the target data distribution. Flow matching is a simulation-free generative modeling framework that learns a time-dependent vector field to transport probability mass from p_0 to p_1 [Lipman et al., 2022].

Probability flow ODE. Let $\mathbf{v}_t : \mathbb{R}^n \to \mathbb{R}^n$ be a time-dependent vector field for $t \in [0, 1]$. Under mild regularity conditions, $\mathbf{v}_t$ defines a deterministic flow map ϕ_t as the solution to the Ordinary Differential Equation (ODE),

$$\frac{d}{dt}\phi_t(\mathbf{x}) = \mathbf{v}_t(\phi_t(\mathbf{x})), \qquad \phi_0(\mathbf{x}) = \mathbf{x}. \tag{1}$$

Starting from an initial draw $\mathbf{x}_0 \sim p_0$, the ODE generates a trajectory $\mathbf{x}_t = \phi_t(\mathbf{x}_0)$. The distribution of $\mathbf{x}_t$ is therefore the pushforward of p_0 through ϕ_t,

$$q_t \coloneqq (\phi_t)_{\#} p_0. \tag{2}$$

In generative modeling, we parameterize the vector field $\mathbf{v}_t$ by a neural network $\mathbf{v}_\theta(\mathbf{x}, t)$ and aim to choose θ so that transporting p_0 to time $t = 1$ matches the data distribution, i.e., $q_1 \approx p_1$. At inference time, we sample $\mathbf{x}_0 \sim p_0$ and numerically integrate Eq. (1) from $t = 0$ to 1 to obtain $\mathbf{x}_1$.

Conditional Flow Matching (CFM). Directly matching q_1 to p_1 is typically intractable. Conditional Flow Matching (CFM) [Lipman et al., 2022] instead trains $\mathbf{v}_\theta$ using supervised regression targets constructed from pairs of endpoint samples. Concretely, we draw $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$, then sample a time $t \sim \mathcal{U}[0, 1]$ and define an interpolation (a "bridge") between the endpoints,

$$\mathbf{x}_t = \psi_t(\mathbf{x}_0, \mathbf{x}_1). \tag{3}$$

The key observation is that ψ_t induces a *conditional* velocity along this bridge,

$$\mathbf{u}_t \coloneqq \frac{d}{dt}\psi_t(\mathbf{x}_0, \mathbf{x}_1), \tag{4}$$

which can be computed analytically once ψ_t is chosen. CFM trains the model to predict this velocity from the intermediate point $\mathbf{x}_t$ and time t by minimizing

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\, \mathbf{x}_1,\, t}\Big[\big\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{u}_t\big\|_2^2\Big]. \tag{5}$$

Intuitively, this objective encourages $\mathbf{v}_\theta$ to agree with the local direction of transport implied by the chosen bridge between source and data samples, without requiring explicit likelihood evaluation or solving an optimal transport problem [Dao et al., 2023].

Linear interpolation. In this work, we use the simple linear bridge

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \tag{6}$$

which yields a constant velocity with respect to t. Differentiating Eq. (6) gives

$$\mathbf{u}_t = \frac{d}{dt}\mathbf{x}_t = \mathbf{x}_1 - \mathbf{x}_0. \tag{7}$$

Plugging Eq. (7) into the CFM objective in Eq. (5) results in the training loss used in our method,

$$\mathcal{L}_{\mathrm{Linear}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\, \mathbf{x}_1,\, t}\Big[\big\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\big\|_2^2\Big]. \tag{8}$$

4 Our Method

In this work, we propose a disentangled representation learning framework based on a *factor-conditioned* latent-space flow. Given an input image, we extract N factor embeddings, each intended to summarize a distinct, controllable semantic attribute. We then learn a conditional flow in latent space that transports samples from a simple source prior to the data distribution. Crucially, we decompose the predicted transport velocity into N token-associated components and regularize them to reduce overlap. This design encourages factor-wise separation in the learned representation and enables targeted interventions at generation time by modifying individual factor tokens used to condition the flow.

Overview. Given an image I, we encode it into a latent target $\mathbf{z}_1 = E(I)$ using a pretrained VQ-GAN encoder [Esser et al., 2021]. This mapping induces a compact, semantically meaningful latent space and provides a stable training target, improving both efficiency and optimization stability. We then cast image generation as a *transport problem* in this latent space: starting from a simple prior sample, we learn dynamics that move the latent state toward $\mathbf{z}_1$. To provide structured semantic cues that can steer this transport, we extract a set of N factors with a trainable factor encoder $T_\gamma(\cdot)$,

$$S_\gamma(I) = \{\mathbf{s}_i\}_{i=1}^N = T_\gamma(I). \tag{9}$$

Rather than treating the factors as a global condition, we inject them into the flow network via *cross-attention*, where intermediate features attend to $S_\gamma(I)$ across multiple U-Net layers, enabling factor-aware conditioning throughout the network. This design provides a structured interface between the encoder and the flow, and has been shown to improve semantic alignment [Rombach et al., 2022, Yang et al., 2024].

We then learn a factor-conditioned vector field $\mathbf{v}_\theta(\mathbf{z}, S_\gamma(I), t)$ that governs the latent dynamics. Since the flow-matching objective constrains only the total velocity, it does not enforce factor-wise disentanglement and may lead to redundant explanations. To address this, we decompose the velocity into N token-associated components and regularize them to reduce overlap, encouraging semantic separation among factor representations. During training, we jointly optimize θ and γ to learn disentangled and controllable representations. Figure 2 illustrates the overall framework.

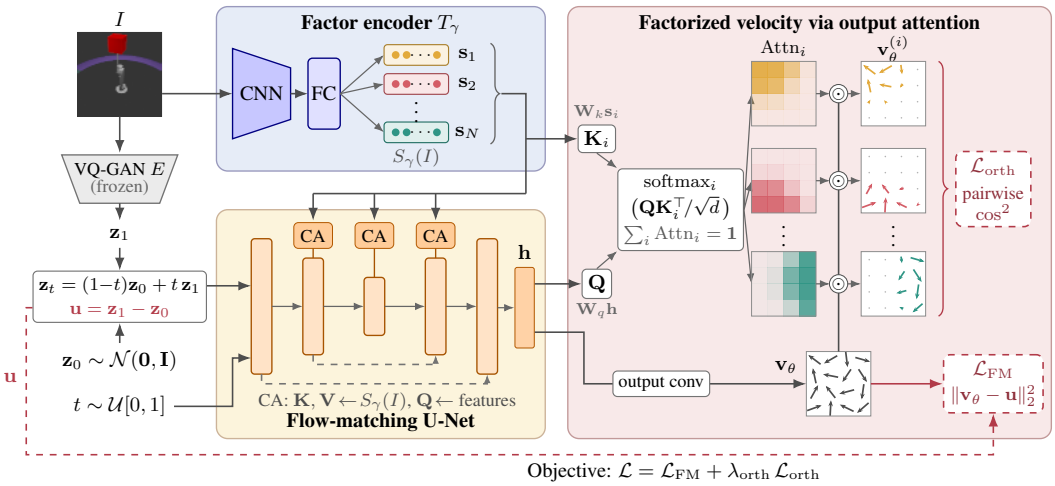


Figure 2: Training pipeline of the proposed framework. A frozen VQ-GAN encoder E maps I to $\mathbf{z}_1$, which is linearly interpolated with Gaussian noise $\mathbf{z}_0$ to obtain $\mathbf{z}_t$. Factor tokens $S_\gamma(I) = \{\mathbf{s}_i\}_{i=1}^N$ from T_γ condition the U-Net through cross-attention (CA). Output attention computes soft routing masks Attn_i from $\mathbf{h}$ and the tokens (Eqs. (17)–(18)). These masks decompose the predicted velocity into factor-specific components $\mathbf{v}_\theta^{(i)}$ that sum to $\mathbf{v}_\theta$ (Eqs. (14) and (19)). $\mathcal{L}_{\mathrm{FM}}$ supervises the aggregate velocity, while $\mathcal{L}_{\mathrm{orth}}$ discourages overlap between components. Masks and velocity fields are schematic; dots indicate small velocity magnitudes.

4.1 Flow Matching Objective

To train the factor-conditioned vector field $\mathbf{v}_\theta$, we adopt the flow matching framework and construct a regression target along a predefined interpolation path connecting a source prior sample to the data latent, conditioned on the factors $S_\gamma(I)$. Concretely, we sample $\mathbf{z}_0 \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \mathcal{U}[0, 1]$, and define the linear interpolation (as in Eq. (6)):

$$\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}_1. \tag{10}$$

This path induces a constant target velocity (cf. Eq. (7)):

$$\mathbf{u} = \frac{d\mathbf{z}_t}{dt} = \mathbf{z}_1 - \mathbf{z}_0. \tag{11}$$

We implement $\mathbf{v}_\theta$ with a U-Net backbone, and inject the conditioning factors $S_\gamma(I)$ into its intermediate feature representations via cross-attention. Concretely, for a spatial feature map in the U-Net, we form queries from the spatial features and use the factors as keys and values. The cross-attention is defined as

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \tag{12}$$

where d is the key/query dimensionality. Substituting the target velocity into the flow matching loss in Eq. (8) yields our factor-conditioned regression objective:

$$\mathcal{L}_{\mathrm{FM}}(\theta, \gamma) = \mathbb{E}_{t, \mathbf{z}_0, \mathbf{z}_1}\left[\left\|\mathbf{v}_\theta(\mathbf{z}_t, S_\gamma(I), t) - (\mathbf{z}_1 - \mathbf{z}_0)\right\|_2^2\right]. \tag{13}$$

Why factorization is needed. Although the cross-attention mechanism injects encoder features into $\mathbf{v}_\theta$ as semantic context to associate input cues with latent transport, *the flow-matching objective does not encourage disentanglement*. It constrains only the aggregate velocity to match the target direction $(\mathbf{z}_1 - \mathbf{z}_0)$, without prescribing how individual factors should contribute. Consequently, different factors may redundantly explain the same transport dynamics (e.g., through channel redundancy or cross-factor leakage), resulting in ambiguous attribution and limited controllability. To obtain a factor-aligned and non-redundant decomposition, we explicitly decompose the predicted velocity into factor-specific components and introduce a regularizer that promotes diversity among them.

4.2 Factorized Velocity via Output Attention

We decompose the aggregate velocity field into factor-specific components and encourage them to be non-redundant via an orthogonality regularizer. In practice, we implement this factorization with an output-attention routing mask that, at each spatial location, distributes the predicted velocity across factors by assigning each factor a proportional share.

Factorized velocity and regularization. We decompose the factor-conditioned velocity field as

$$\mathbf{v}_\theta(\mathbf{z}_t, S_\gamma(I), t) \;=\; \sum_{i=1}^{N} \mathbf{v}_\theta^{(i)}(\mathbf{z}_t, S_\gamma(I), t), \tag{14}$$

where $\mathbf{v}_\theta^{(i)}$ denotes the component attributed to the i-th factor. The flow-matching loss $\mathcal{L}_{\mathrm{FM}}(\theta, \gamma)$ in Eq. (13) is applied to the aggregate field $\mathbf{v}_\theta$. The regularizer below specifies how the aggregate should be partitioned across factors.

To discourage redundant attributions, we penalize pairwise alignment between factor-specific components. For each training sample (and each sampled t), we flatten each component into a vector

$$\boldsymbol{\alpha}_i \;=\; \mathrm{vec}\Big(\mathbf{v}_\theta^{(i)}(\mathbf{z}_t, S_\gamma(I), t)\Big), \tag{15}$$

and define the orthogonality loss as the average squared cosine similarity over all factor pairs,

$$\mathcal{L}_{\mathrm{orth}} \;=\; \frac{1}{N(N-1)} \sum_{i \neq j} \left(\frac{\boldsymbol{\alpha}_i^\top \boldsymbol{\alpha}_j}{\|\boldsymbol{\alpha}_i\|_2 \, \|\boldsymbol{\alpha}_j\|_2 + \varepsilon} \right)^2, \tag{16}$$

where ε is a small constant for numerical stability. Minimizing $\mathcal{L}_{\mathrm{orth}}$ discourages overlap among the spatial allocations of the predicted velocity to different tokens.

Practical implementation via output attention. We instantiate $\{\mathbf{v}_\theta^{(i)}\}_{i=1}^N$ using an output-attention routing mask. Let $\mathbf{h}$ denote the final hidden feature map produced by the U-Net backbone that predicts $\mathbf{v}_\theta$. We form per-location queries and token keys as

$$\mathbf{Q} = \mathbf{W}_q \mathbf{h}, \qquad \mathbf{K}_i = \mathbf{W}_k \mathbf{s}_i, \tag{17}$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are learned projections. The routing weight for factor i is

$$\mathrm{Attn}_i = \frac{\exp\Big(\frac{\mathbf{Q}\mathbf{K}_i^\top}{\sqrt{d}}\Big)}{\sum_{j=1}^{N} \exp\Big(\frac{\mathbf{Q}\mathbf{K}_j^\top}{\sqrt{d}}\Big)}, \tag{18}$$

which satisfies $\sum_{i=1}^{N} \mathrm{Attn}_i = 1$ at each spatial location. We then define factor-specific velocities by gating the aggregate prediction:

$$\mathbf{v}_\theta^{(i)}(\mathbf{z}_t, S_\gamma(I), t) = \mathrm{Attn}_i \odot \mathbf{v}_\theta(\mathbf{z}_t, S_\gamma(I), t), \tag{19}$$

where $\odot$ denotes element-wise multiplication (with Attn_i broadcast to match the shape of $\mathbf{v}_\theta$). By the simplex constraint in Eq. (18), the components sum to the aggregate field, recovering Eq. (14).

Why output attention? This design offers three practical advantages: (i) *Factor-aligned attribution*: Attn_i provides an explicit, spatially varying assignment of the velocity field to each factor. (ii) *Additive consistency*: the simplex constraint $\sum_i \mathrm{Attn}_i = 1$ guarantees that the factor-specific components sum exactly to the aggregate prediction, preserving the flow-matching supervision. (iii) *Efficiency and stability*: gating reuses the already-computed aggregate field and adds minimal overhead, while the orthogonality loss in Eq. (16) directly discourages redundant factor routes.

Overall objective. We optimize the combined objective

$$\mathcal{L}(\theta, \gamma) \;=\; \mathcal{L}_{\mathrm{FM}}(\theta, \gamma) \;+\; \lambda_{\mathrm{orth}} \, \mathcal{L}_{\mathrm{orth}}, \tag{20}$$

where λ_{orth} controls the strength of the regularization. The full algorithm is given in Appendix C.1.

Table 1: Quantitative comparison on Cars3D, Shapes3D, and MPI3D-toy datasets using FactorVAE score and DCI-D (mean±std).

	Method	Cars3D		Shapes3D		MPI3D-toy	
		FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑
VAE	FactorVAE	0.906 ± 0.052	0.161 ± 0.019	0.840 ± 0.066	0.611 ± 0.101	0.152 ± 0.025	0.240 ± 0.051
	β-TCVAE	0.855 ± 0.082	0.140 ± 0.019	0.873 ± 0.074	0.613 ± 0.114	0.179 ± 0.017	0.237 ± 0.056
	DAVA	0.940 ± 0.010	0.230 ± 0.040	0.820 ± 0.030	0.780 ± 0.030	0.410 ± 0.040	0.300 ± 0.030
GAN	ClosedForm	0.873 ± 0.036	0.243 ± 0.048	0.951 ± 0.021	0.525 ± 0.078	0.523 ± 0.056	0.318 ± 0.014
	GANSpace	0.932 ± 0.018	0.209 ± 0.031	0.788 ± 0.091	0.284 ± 0.034	0.465 ± 0.036	0.229 ± 0.042
	DisCo-GAN	0.855 ± 0.074	0.271 ± 0.037	0.877 ± 0.031	0.708 ± 0.048	0.371 ± 0.030	0.292 ± 0.024
Diffusion	DisDiff	0.976 ± 0.018	0.232 ± 0.019	0.902 ± 0.043	0.723 ± 0.013	0.617 ± 0.070	0.337 ± 0.057
	FDAE	0.918 ± 0.027	0.232 ± 0.041	0.987 ± 0.023	0.917 ± 0.038	0.779 ± 0.036	0.528 ± 0.042
	EncDiff	0.773 ± 0.060	0.279 ± 0.022	0.999 ± 0.000	0.969 ± 0.030	0.872 ± 0.049	0.685 ± 0.044
	DyGA	0.941 ± 0.002	0.414 ± 0.013	$\mathbf{1.000 \pm 0.000}$	0.938 ± 0.001	0.930 ± 0.004	0.627 ± 0.002
	Ours	$\mathbf{0.980 \pm 0.011}$	$\mathbf{0.487 \pm 0.034}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.977 \pm 0.032}$	$\mathbf{0.931 \pm 0.017}$	$\mathbf{0.710 \pm 0.022}$

Choice of λ_{orth}. In practice, we select λ_{orth} via a small validation sweep over $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0\}$ and choose the largest value that improves factor diversity without degrading $\mathcal{L}_{\text{FM}}$ or sample quality.

5 Experiments

We present qualitative and quantitative results that demonstrate our method's effectiveness on synthetic and real-world data. All experiments are conducted on a Linux server equipped with $8\times$ NVIDIA RTX 4090 GPUs. The code will be released at `https://github.com/jinjinchi/LDR-FM`.

5.1 Experimental Setup

Datasets. We evaluate on standard disentanglement datasets: Cars3D [Reed et al., 2015], Shapes3D [Burgess and Kim, 2018], and MPI3D-toy [Gondal et al., 2019]. Cars3D comprises 3D-rendered car images with viewpoint-related factors. Shapes3D consists of procedurally rendered 3D objects with multiple controlled generative factors (e.g., shape, size, orientation, and color). MPI3D-toy contains images rendered in a controlled environment with annotated factors of variation. To assess performance on real-world imagery, we additionally report results on CelebA [Liu et al., 2015], a large-scale face dataset with attribute annotations.

Baselines. We compare against three families of baselines: *VAE-based*: FactorVAE [Kim and Mnih, 2018], β-TCVAE [Chen et al., 2018], and DAVA [Estermann and Wattenhofer, 2023]; *GAN-based*: ClosedForm [Shen and Zhou, 2021], GANSpace [Härkönen et al., 2020], and DisCo-GAN [Ren et al., 2022]; *Diffusion-based*: DisDiff [Yang et al., 2023b], FDAE [Wu and Zheng, 2024], EncDiff [Yang et al., 2024], and DyGA [Jun et al., 2025]. We use official implementations when available, noting that several baselines also employ pre-trained encoders in their original setups.

Evaluation metrics. On synthetic datasets, we report the FactorVAE score [Kim and Mnih, 2018] and the disentanglement component of DCI [Eastwood and Williams, 2018], denoted DCI-D. These metrics assess factor-to-coordinate alignment and the specificity of individual representation coordinates, respectively. On CelebA, we report Total AUROC Difference (TAD) [Yeats et al., 2022], which measures attribute alignment through the gap between the two highest per-coordinate AUROC scores for each evaluated attribute. We use Fréchet Inception Distance (FID) [Heusel et al., 2017] to evaluate generation quality on CelebA. Following common practice, all metrics except FID are computed on PCA-reduced latent representations when the latent space is high-dimensional.

Implementation details. We use a batch size of 128 across all datasets. Unless otherwise stated, our model uses $N = 20$ latent factor tokens and an orthogonality weight of $\lambda_{\text{orth}} = 0.01$. During sampling, we solve the associated ODE using the adaptive-step Dopri5 solver [Akhtar, 2025]. Architecture details and additional experimental results are provided in Appendices C.2 and C.3, respectively.

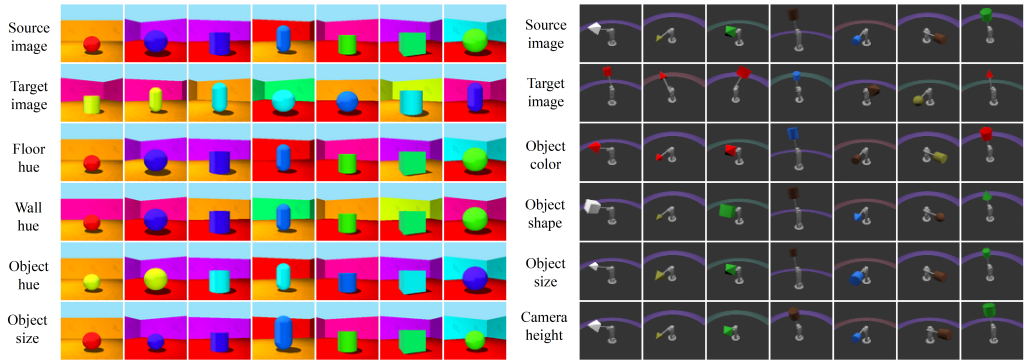


Figure 3: Factor swapping on Shapes3D (left) and MPI3D-toy (right). The top two rows show source and target images. Subsequent rows show images generated after replacing one source factor token with the corresponding target token; each row corresponds to a different exchanged factor.

5.2 Comparison with State-of-the-Art Methods

Results: Cars3D, Shapes3D, and MPI3D-toy. Table 1 reports disentanglement results on three synthetic datasets using the FactorVAE score and DCI-D. Our method achieves the highest mean DCI-D and the highest or tied highest mean FactorVAE score on all three datasets. The improvements over VAE- and GAN-based baselines are substantial. On MPI3D-toy, our method achieves a FactorVAE score of 0.931 and a DCI-D score of 0.710, exceeding the strongest VAE baseline, DAVA (0.410/0.300), and the strongest GAN baseline, ClosedForm (0.523/0.318). Compared with diffusion-based baselines, the gains in DCI-D are smaller but consistent, while FactorVAE scores are matched or improved. Since diffusion models already provide strong generative backbones, further improvements in disentanglement can benefit from explicitly encouraging semantic structure. Our method promotes such structure through factor-aligned representations, improving DCI-D without degrading the FactorVAE score.

Results: CelebA. Table 2 reports disentanglement-related behavior (TAD↑) and sample quality (FID↓) on the CelebA dataset. Our method achieves the highest mean TAD and the lowest FID among the evaluated methods, indicating stronger attribute alignment and improved generation quality. These results demonstrate that the benefits of our method extend beyond synthetic datasets to real-world face images.

Table 2: Attribute alignment (TAD) and generation quality (FID) on CelebA (mean±std).

Metric	FactorVAE	DisDiff	FDAE	EncDiff	DyGA	**Ours**
TAD ↑	0.137 ± 0.043	0.305 ± 0.010	0.275 ± 0.104	0.638 ± 0.008	0.905 ± 0.021	$\mathbf{0.913 \pm 0.012}$
FID ↓	96.3 ± 1.6	18.2 ± 2.1	18.8 ± 1.5	14.8 ± 2.3	12.1 ± 0.9	$\mathbf{7.8 \pm 0.1}$

5.3 Visualization Results

A simple yet effective way to evaluate whether the feature extractor learns *well-disentangled* representations is to actively intervene on the inferred factors and inspect the resulting generations. If changing a single factor induces a predictable, semantically coherent modification in the output while leaving other attributes largely unchanged, this indicates that the representation separates underlying generative factors [Bengio et al., 2013]. We therefore perform a *factor swapping* experiment on Shapes3D and MPI3D-toy. Given a source image and a target image, we encode each into its set of factors, replace one source factor with the corresponding target factor, and synthesize an image conditioned on the modified representation. Figure 3 shows that the corresponding attributes, such as wall hue and object shape, are transferred from the target image while the source image's other attributes are preserved. These results provide qualitative evidence of disentanglement and factor-level controllability. Additional factor-swapping results for object shape and orientation on Shapes3D are provided in Figure 5 in the Appendix.

Table 3: Ablation of the conditioning mechanism and orthogonality regularizer (mean±std).

Variant	Cars3D		Shapes3D		MPI3D-toy	
	FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑
AdaGN	0.954 ± 0.028	0.308 ± 0.084	0.865 ± 0.024	0.684 ± 0.040	0.695 ± 0.039	0.440 ± 0.040
w/o $\mathcal{L}_{\text{orth}}$	0.914 ± 0.002	0.448 ± 0.046	0.999 ± 0.001	0.946 ± 0.021	0.804 ± 0.073	0.637 ± 0.041
Full model	$\mathbf{0.980 \pm 0.011}$	$\mathbf{0.487 \pm 0.034}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.977 \pm 0.032}$	$\mathbf{0.931 \pm 0.017}$	$\mathbf{0.710 \pm 0.022}$

Table 4: Sensitivity analysis with respect to the number of latent factor tokens N on the Cars3D, Shapes3D, and MPI3D-toy datasets (mean±std).

	Cars3D		Shapes3D		MPI3D-toy	
N	FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑	FactorVAE ↑	DCI-D ↑
5	0.909 ± 0.059	0.195 ± 0.086	0.795 ± 0.045	0.710 ± 0.074	0.443 ± 0.040	0.282 ± 0.069
10	0.964 ± 0.013	0.431 ± 0.034	1.000 ± 0.000	0.973 ± 0.020	0.907 ± 0.017	0.649 ± 0.024
15	0.934 ± 0.027	0.521 ± 0.045	1.000 ± 0.000	0.962 ± 0.032	0.926 ± 0.013	0.687 ± 0.004
20	0.980 ± 0.011	0.487 ± 0.034	1.000 ± 0.000	0.977 ± 0.032	0.931 ± 0.017	0.710 ± 0.022
30	0.963 ± 0.036	0.582 ± 0.040	1.000 ± 0.000	0.946 ± 0.037	0.943 ± 0.035	0.757 ± 0.045

5.4 Ablation Study

We conduct ablation studies to assess the contributions of the conditioning mechanism and the orthogonality regularizer to disentanglement. Cross-attention allows spatial features in the U-Net to attend to individual factor tokens, providing a structured interface between the factor encoder and the flow network. To examine whether this interaction benefits disentanglement, we replace cross-attention conditioning with adaptive group normalization (*AdaGN conditioning*). In addition, the flow-matching objective supervises the aggregate velocity without explicitly constraining how individual factors contribute to it. This motivates the orthogonality regularizer, which discourages redundancy among factor-specific velocity components. We therefore evaluate a variant without this regularizer (*w/o* $\mathcal{L}_{\text{orth}}$), setting $\lambda_{\text{orth}} = 0$, to assess whether explicit factor separation provides additional benefits beyond the base flow-matching objective. Table 3 compares both variants with the full model on Cars3D, Shapes3D, and MPI3D-toy, using $N = 20$.

Conditioning mechanism. Replacing cross-attention with AdaGN reduces both metrics on all three datasets. The DCI-D score decreases from 0.487 to 0.308 on Cars3D, from 0.977 to 0.684 on Shapes3D, and from 0.710 to 0.440 on MPI3D-toy. These results support cross-attention as an effective interface between factor tokens and the latent flow network.

Orthogonality regularization. Adding $\mathcal{L}_{\text{orth}}$ improves both metrics across the three datasets. On MPI3D-toy, the FactorVAE score increases from 0.804 to 0.931, and DCI-D increases from 0.637 to 0.710. This ablation supports the contribution of the regularizer to learning more disentangled representations beyond cross-attention conditioning alone.

5.5 Sensitivity to the Number of Tokens N

We vary $N \in \{5, 10, 15, 20, 30\}$ to examine how token capacity affects disentanglement and report the results in Table 4. On Shapes3D and MPI3D-toy, increasing N from 5 to 10 makes a dedicated token for each ground-truth factor possible and yields substantial gains. Cars3D also benefits despite $N = 5$ already exceeding its three factors, suggesting that capacity requirements are not determined solely by factor count. Further increases in N improve both the FactorVAE score and DCI-D on MPI3D-toy, whereas DCI-D varies non-monotonically on Shapes3D and Cars3D.

These scores alone do not reveal how extra tokens are used under over-specification. Output attention in Eqs. (18) and (19) routes velocity softly without explicit pruning. Extra tokens may receive little output attention or participate in encoding existing factors. To assess the effects of over-specification on the learned representations, we report DCI informativeness (DCI-I) and completeness (DCI-C) for $N \in \{10, 20, 30\}$ in Appendix Table 8. DCI-I measures factor predictability, while DCI-C measures the concentration of each factor's predictive importance across representation components.

Table 5: Relative downstream accuracy with limited training labels on Shapes3D and MPI3D-toy. Acc denotes accuracy with 10,000 training examples. Higher ratios indicate better retention of this accuracy.

Method	Shapes3D		MPI3D-toy	
	$\mathrm{Acc}_{1000}/\mathrm{Acc}$	$\mathrm{Acc}_{100}/\mathrm{Acc}$	$\mathrm{Acc}_{1000}/\mathrm{Acc}$	$\mathrm{Acc}_{100}/\mathrm{Acc}$
DisDiff	0.928 ± 0.001	0.732 ± 0.002	0.862 ± 0.001	0.700 ± 0.002
FDAE	0.983 ± 0.004	0.746 ± 0.004	0.847 ± 0.011	0.745 ± 0.012
EncDiff	0.975 ± 0.000	0.772 ± 0.002	0.853 ± 0.005	0.701 ± 0.001
DyGA	0.990 ± 0.000	0.863 ± 0.001	0.888 ± 0.002	0.757 ± 0.001
Ours	$\mathbf{0.994 \pm 0.001}$	$\mathbf{0.890 \pm 0.003}$	$\mathbf{0.899 \pm 0.002}$	$\mathbf{0.759 \pm 0.012}$

DCI-I is maintained or improved across the three datasets, whereas DCI-C decreases on Shapes3D and MPI3D-toy and increases on Cars3D. On MPI3D-toy, increasing N from 10 to 30 raises DCI-D from 0.649 to 0.757, while DCI-C falls from 0.609 to 0.573. This pattern is consistent with the definition of DCI-D. The score down-weights unimportant components and can remain high when multiple components encode the same factor, provided each remains factor-specific. Additional token capacity can thus preserve or improve disentanglement and predictability while making factor information less concentrated.

5.6 Downstream Tasks

To assess the practical utility of our learned representations under limited supervision, we follow the evaluation protocol of Jun et al. [2025] and evaluate downstream classification on Shapes3D and MPI3D-toy using gradient-boosted trees (GBT). We train classifiers on the learned representations with different numbers of labeled samples. Let Acc_{1000} and Acc_{100} denote the accuracies obtained using 1,000 and 100 training samples, respectively, and let Acc denote the reference accuracy obtained using 10,000 training samples. We report $\mathrm{Acc}_{1000}/\mathrm{Acc}$ and $\mathrm{Acc}_{100}/\mathrm{Acc}$, which measure the fraction of each method's reference accuracy retained under limited supervision. This evaluation assesses relative sample efficiency when labeled data are scarce.

We compare our method with DisDiff, FDAE, EncDiff, and DyGA. Table 5 shows that our method achieves the highest mean accuracy ratios at both sample budgets on both datasets. For example, with 100 labeled samples on Shapes3D, our method retains 89.0% of its reference accuracy, compared with 86.3% for DyGA. These results indicate better retention of downstream classification performance as the number of labeled samples decreases, supporting the relative sample efficiency of our learned representations under this evaluation protocol.

6 Conclusion

We present a flow-matching framework for disentangled representation learning that models generation as factor-conditioned transport in a compact latent space. An orthogonality regularizer discourages overlap among token-associated velocity components, encouraging semantic separation among the learned factor representations. Experiments show improved mean DCI-D scores on the synthetic benchmarks and lower FID on CelebA compared with the evaluated diffusion-based baselines. Factor-swapping results further illustrate semantic controllability. These findings support structured latent transport as a promising approach to learning disentangled and controllable representations.

Acknowledgments and Disclosure of Funding

This research is supported by the Scientific and Technological Developing Scheme of Jilin Province (No. 20250102213JC), the National Natural Science Foundation of China (NSFC) (Nos. 62006094 and 62276113), and the National Science Centre, Poland (NCN) (No. 2025/59/B/ST6/01950). It is also supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

S. W. Akhtar. On tuning neural ODE for stability, consistency and faster convergence. *SN Computer Science*, 6(4):318, 2025.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Chris Burgess and Hyunjik Kim. 3D shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

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

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Neural Information Processing Systems*, pages 2610–2620, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems*, pages 2172–2180, 2016.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving disentangled text representation learning with information-theoretic guidance. In *Association for Computational Linguistics*, pages 7530–7541, 2020.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

Benjamin Estermann and Roger Wattenhofer. Dava: Disentangling adversarial variational autoencoder. In *International Conference on Learning Representations*, 2023.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Neural Information Processing Systems*, volume 32, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Neural Information Processing Systems*, volume 33, pages 9841–9850, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, volume 30, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. Text style transfer: A review and experimental evaluation. *ACM SIGKDD Explorations Newsletter*, 24(1):14–45, 2022.

Youngjun Jun, Jiwoo Park, Kyobin Choo, Tae Eun Choi, and Seong Jae Hwang. Disentangling disentangled representations: Towards improved latent units via diffusion models. In *Winter Conference on Applications of Computer Vision*, pages 3559–3569, 2025.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 848–856, 2019.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, pages 6127–6139, 2020.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, pages 3730–3738, 2015.

Pingchuan Ma, Xiaopei Yang, Yusong Li, Ming Gui, Felix Krause, Johannes Schusterbauer, and Björn Ommer. Scflow: Implicitly learning style and content disentanglement with flow models. In *International Conference on Computer Vision*, pages 14919–14929, 2025.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Neural Information Processing Systems*, volume 28, 2015.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Computer Vision and Pattern Recognition*, pages 1532–1540, 2021.

Sneha Varur, Anirudh R Hanchinamani, Tarun S Bagewadi, Uma Mudenagudi, Chaitra D Desai, Padmashree Desai, Sumit Meharwade, et al. Disc-gan: Disentangling style and content for cluster-specific synthetic underwater image generation. In *International Conference on Computer Vision*, pages 6844–6852, 2025.

Duomin Wang, Yu Deng, Zixin Yin, Heung-Yeung Shum, and Baoyuan Wang. Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In *Computer Vision and Pattern Recognition*, pages 17979–17989, 2023.

Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Computer Vision and Pattern Recognition*, pages 11379–11388, 2022.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9677–9696, 2024.

Ancong Wu and Wei-Shi Zheng. Factorized diffusion autoencoder for unsupervised disentangled representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5930–5939, 2024.

Kun Wu, Yichen Zhu, Jinming Li, Junjie Wen, Ning Liu, Zhiyuan Xu, and Jian Tang. Discrete policy: Learning disentangled action space for multi-task robotic manipulation. In *International Conference on Robotics and Automation*, pages 8811–8818, 2025.

Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *Computer Vision and Pattern Recognition*, pages 4402–4412, 2023.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023a.

Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. In *Neural Information Processing Systems*, volume 36, pages 69130–69156, 2023b.

Tao Yang, Cuiling Lan, Yan Lu, and Nanning Zheng. Diffusion model with cross attention as an inductive bias for disentanglement. In *Neural Information Processing Systems*, volume 37, pages 82465–82492, 2024.

Eric Yeats, Frank Liu, David Womble, and Hai Li. Nashae: Disentangling representations through adversarial covariance minimization. In *European Conference on Computer Vision*, pages 36–51, 2022.

A Broader Societal Impact

While this work aims to advance disentangled representation learning and controllable generative modeling, enabling applications in image editing, data analysis, and scientific understanding, it also raises potential concerns regarding misuse, such as unauthorized content manipulation or the creation of misleading media. Addressing potential biases in learned representations and ensuring fair and responsible deployment remain important directions for future research.

B Limitations

Our method uses a prespecified number of factor tokens N and does not automatically determine the required capacity. When N is below the ground-truth factor count, a dedicated token for each factor is impossible, which may limit token-level separation. When N is over-specified, output attention provides soft routing without explicitly pruning unnecessary components. Our experiments show that additional capacity can preserve or improve factor predictability while reducing completeness on Shapes3D and MPI3D-toy. Selecting N therefore remains an empirical design choice, motivating future work on adaptive token allocation and pruning.

C More Implementation Details

C.1 Algorithm

Algorithm 1 summarizes the training procedure for the proposed framework.

Algorithm 1 Overall Disentangled Representation Learning Framework

Input: pretrained VQ-GAN encoder $E(\cdot)$, networks T_γ and $\mathbf{v}_\theta$, number of tokens N, orthogonality weight λ_{orth}
Randomly initialize θ and γ
while not converged **do**
  Sample an image I from the training set
  Encode the image: $\mathbf{z}_1 \leftarrow E(I)$
  Extract factor tokens: $S_\gamma(I) = \{\mathbf{s}_i\}_{i=1}^N \leftarrow T_\gamma(I)$
  Sample $\mathbf{z}_0 \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \mathcal{U}[0, 1]$
  Construct $\mathbf{z}_t \leftarrow (1-t)\mathbf{z}_0 + t\mathbf{z}_1$
  Compute target velocity: $\mathbf{u} \leftarrow \mathbf{z}_1 - \mathbf{z}_0$
  Predict aggregate velocity using cross-attention conditioning: $\widehat{\mathbf{v}} \leftarrow \mathbf{v}_\theta(\mathbf{z}_t, S_\gamma(I), t)$
  Compute output-attention weights $\{\text{Attn}_i\}_{i=1}^N$ using Eq. (18)
  Obtain factor-specific velocities: $\mathbf{v}_\theta^{(i)} \leftarrow \text{Attn}_i \odot \widehat{\mathbf{v}}$
  Compute $\mathcal{L}_{\text{FM}} \leftarrow \|\widehat{\mathbf{v}} - \mathbf{u}\|_2^2$
  Compute $\mathcal{L}_{\text{orth}}$ using Eq. (16)
  Set $\mathcal{L} \leftarrow \mathcal{L}_{\text{FM}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}}$
  Update (θ, γ) using $\nabla_{\theta,\gamma}\mathcal{L}$
end while
Output: θ, γ

C.2 Model Architecture

We adopt the factor encoder architecture used in the official implementation of EncDiff [Yang et al., 2024]. The encoder consists of a convolutional backbone with progressive downsampling and residual blocks, followed by a fully connected layer that produces N scalar representations. Each scalar is mapped to a 16-dimensional factor token by a separate MLP, with no parameter sharing across factors. Table 6 summarizes the architecture.

Table 6: Architecture of the factor encoder T_γ (input $64\times64\times3$).

Layer	Configuration
Conv 4×4, $3 \to 128$, stride 2	BN, ReLU
Conv 4×4, $128 \to 128$, stride 2	BN, ReLU
Conv 4×4, $128 \to 128$, stride 2	BN, ReLU
Conv 4×4, $128 \to 128$, stride 2	BN, ReLU
$2\times$ residual block, 128 ch	Conv 3×3 + Conv 1×1, BN, ReLU
Flatten	$128 \times 4 \times 4 = 2048$
FC $2048 \to N$	factor scalars $c_1, \ldots, c_N$
Per-factor MLP (not shared across i)	c_i: FC $1 \to 64$, ELU, FC $64 \to 128$, ELU, FC $128 \to 16$
Output	$S_\gamma(I) = \{\mathbf{s}_i\}_{i=1}^N$, $\mathbf{s}_i \in \mathbb{R}^{16}$

For velocity prediction, we use a U-Net with a shared architecture across Shapes3D, Cars3D, and MPI3D-toy. Table 7 summarizes the architecture and dataset-specific training settings.

C.3 More Experiments

Informativeness and completeness under over-specification. To complement the sensitivity analysis, we report DCI informativeness (DCI-I) and completeness (DCI-C) for $N \in \{10, 20, 30\}$. These settings exceed the ground-truth factor counts of all three datasets. DCI-I measures factor

Table 7: U-Net architecture and training settings for velocity prediction on the three synthetic datasets.

Parameter	Shapes3D / Cars3D / MPI3D-toy
Input latent (VQ-GAN, f=4, 2048 codes)	$16 \times 16 \times 3$
Model channels	64
Channel multipliers	[1, 2, 4, 4]
Res. blocks per level	2
Attention resolutions	$16, 8, 4$
Attention heads	8
Cross-attention context dim	16 ($= \dim \mathbf{s}_i$)
Number of tokens N	20
Optimizer	AdamW
Batch size	128
Dataset size	0.48M / 0.018M / 1.04M

Table 8: DCI-I and DCI-C under over-specification of N (mean$\pm$std).

Dataset	$N = 10$		$N = 20$		$N = 30$	
	DCI-I ↑	DCI-C ↑	DCI-I ↑	DCI-C ↑	DCI-I ↑	DCI-C ↑
Cars3D	0.698 ± 0.136	0.251 ± 0.027	0.813 ± 0.101	0.279 ± 0.012	0.814 ± 0.110	0.289 ± 0.014
Shapes3D	0.999 ± 0.000	0.791 ± 0.017	0.999 ± 0.000	0.699 ± 0.038	0.999 ± 0.000	0.693 ± 0.043
MPI3D-toy	0.815 ± 0.010	0.609 ± 0.023	0.829 ± 0.036	0.584 ± 0.019	0.859 ± 0.004	0.573 ± 0.064

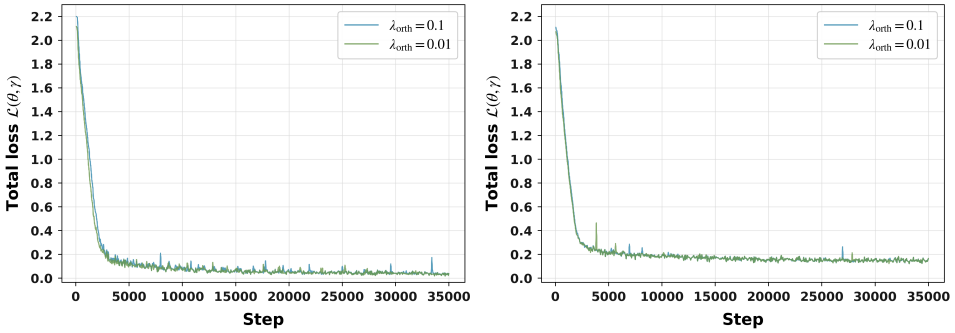


Figure 4: Training loss curves. Left: Shapes3D. Right: MPI3D-toy.

predictability, while DCI-C measures the concentration of each factor's predictive importance across representation components.

Table 8 shows that DCI-I remains unchanged on Shapes3D and increases on MPI3D-toy and Cars3D as N increases. DCI-C decreases on Shapes3D and MPI3D-toy but increases on Cars3D. The results on Shapes3D and MPI3D-toy are consistent with existing factor information becoming less concentrated across components while remaining predictable. They do not establish whether individual tokens are inactive or redundantly used.

Training loss curves. We visualize the training loss curves for different settings of the orthogonality weight λ_{orth}. As shown in Figure 4, our method exhibits stable and consistent convergence across the evaluated configurations.

Additional factor-swapping results. Figure 5 extends the qualitative evaluation on Shapes3D to all six generative factors, including object shape and orientation. For each source–target pair, we replace one factor token in the source representation with its counterpart from the target representation and generate the resulting image. The examples illustrate semantic transfer through individual factor interventions.

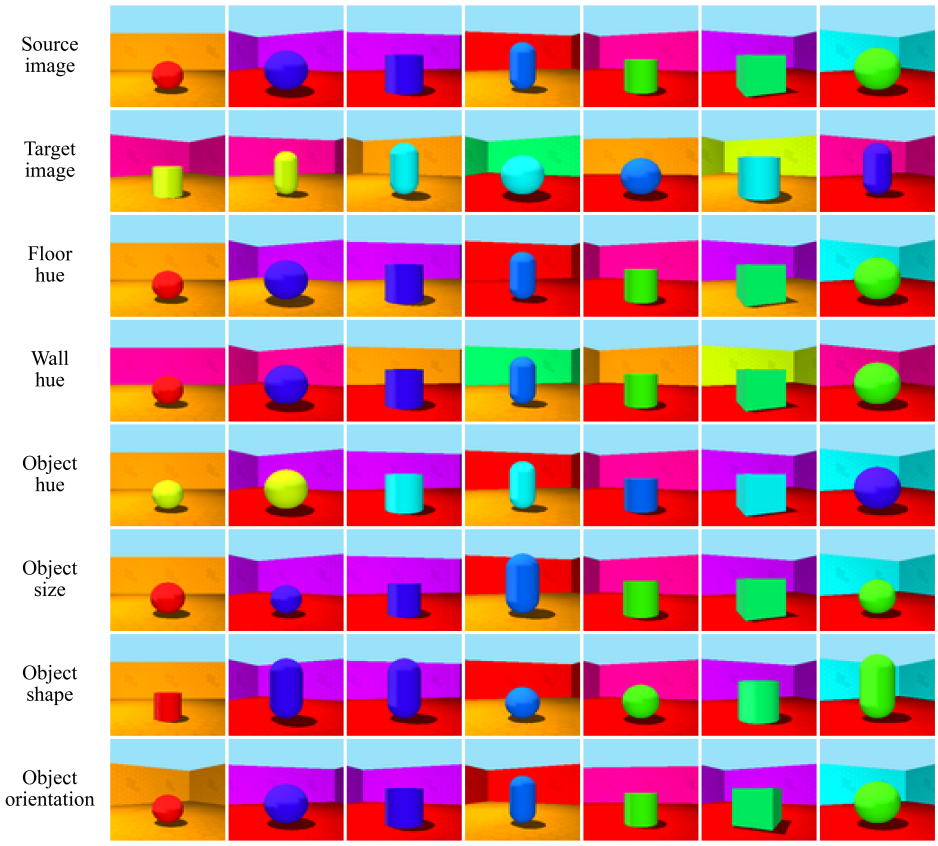


Figure 5: Factor swapping on Shapes3D. Each column shows a source–target pair. The first two rows contain the source and target images. The remaining rows show generations after replacing the source token associated with the indicated factor with its target counterpart. Results cover floor hue, wall hue, object hue, object size, object shape, and orientation.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are accurate and aligned with the contributions and scope of the paper. They clearly state the novelty of our proposed approach and the empirical findings that support our conclusions. Limitations are also appropriately acknowledged to ensure the claims do not overstate the generality or applicability of our results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations section discusses the need to prespecify the number of factor tokens N and the lack of explicit pruning of unnecessary components. It also describes the potential effects of insufficient or excessive token capacity.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not contain any theoretical results; it is entirely empirical. Therefore, this question is not applicable.

   Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 4 and Appendix C describe the model architecture and training procedure.

Guidelines:

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

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

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

Answer: [Yes]

Justification: The code and reproduction instructions are available at the GitHub repository linked in Section 5.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 and Appendix C describe the experimental settings.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report all quantitative results as mean $\pm$ standard deviation over multiple runs with different random seeds. The reported variability reflects randomness in model initialization and training. Unless otherwise specified, the error bars correspond to one standard deviation. All metrics reported in tables include these values, and we refer to the corresponding tables in the main text.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report compute details including GPU type.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted fully complies with the NeurIPS Code of Ethics. It does not involve human subjects, crowdsourcing, or sensitive data. The paper promotes transparency and reproducibility, and all experimental procedures adhere to ethical research standards.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both the positive potential of the method as well as the possible negative misuse of it in the supplementary.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work does not release new datasets or deploy models with a high risk of misuse. The proposed method focuses on representation learning and is evaluated on standard public datasets. Therefore, no additional safeguards for data or model release are required.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite the original papers corresponding to the code packages and datasets used in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: The GitHub repository includes the implementation and instructions for training and evaluation.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: This work does not use large language models (LLMs) as part of the proposed method. LLMs were not involved in the core methodology, experiments, or analysis.

Guidelines:

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