MoG-Diff: Mixture-of-Gaussians Latent Structuring for Controllable Diffusion-Bridge Generation

Anonymous authors
Paper under double-blind review

Abstract

Diffusion-bridge autoencoding connects representation learning, reconstruction, and generation, yet its latent space need not be semantically organized or provide explicit coordinates for traversing generative modes. We introduce MoG-Diff, a unified framework that structures the latent space with a mixture of Gaussians and represents each sample through a component and a component-relative residual. Variational mixture inference learns the component organization, while a task-adaptive interface connects the structured latent to the endpoint-conditioned bridge. A pointwise Doob h-transform analysis establishes compatibility with the bridge construction. The resulting geometry supports component-conditional generation, residual-preserving and responsibility-controlled pairwise transport, and multi-component responsibility simplex control. On MNIST and Fashion-MNIST, MoG-Diff achieves the best clustering metrics among the evaluated methods, complete component-majority class coverage, and the lowest Conditional Feature FID. On CelebA, it performs on par with or slightly better than DBAE in reconstruction and provides qualitative natural-image extensions of component generation and transport. Responsibility control reduces pairwise MAE from 0.2065 to 0.0356 and three-component simplex MAE from 0.0644 to 0.0227. These results show that Gaussian-mixture structure can serve not only as a representation prior but also as an operational geometry for controllable diffusion-bridge generation.

1 Introduction

Diffusion models have established a flexible framework for high-quality generation by learning to reverse a gradual corruption process (Ho et al., 2020; Song et al., 2021b). Diffusion bridges extend this paradigm from noise-to-data generation to stochastic transport between informative endpoints (De Bortoli et al., 2021; Zhou et al., 2024). Diffusion-bridge autoencoding further introduces a compact latent that specifies a sample-dependent bridge endpoint, connecting representation learning, reconstruction, and generation (Kim et al., 2025). However, a latent that is sufficient for reconstruction need not be organized into semantically meaningful regions, nor does ordinary interpolation explicitly control which generative modes are traversed.

Deep clustering methods jointly learn representations and cluster assignments (Xie et al., 2016; Jiang et al., 2017), while Gaussian-mixture latent models associate discrete components with regions of a continuous representation space (Dilokthanakul et al., 2016; Jiang et al., 2017). Their mixture structure, however, is typically optimized for representation learning or clustering and does not directly provide coordinates for diffusion-bridge generation and cross-component control. Conversely, externally provided labels enable conditional generation but do not discover semantic structure without supervision. These limitations motivate a framework in which a learned mixture geometry organizes the diffusion-bridge latent space, supports generative control, and preserves reconstruction fidelity.

We address this question with MoG-Diff, a unified framework that equips a latent-conditioned diffusion bridge with a mixture-of-Gaussians latent structure. Each latent is expressed in component–residual coordinates, $z_{\mathrm{mix}} = \mu_c + \sigma_c \odot r$, where c identifies a mixture-level semantic region and the component-whitened residual r describes relative variation within that region. A task-adaptive latent interface connects this structured representation to the endpoint decoder and diffusion bridge. We

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

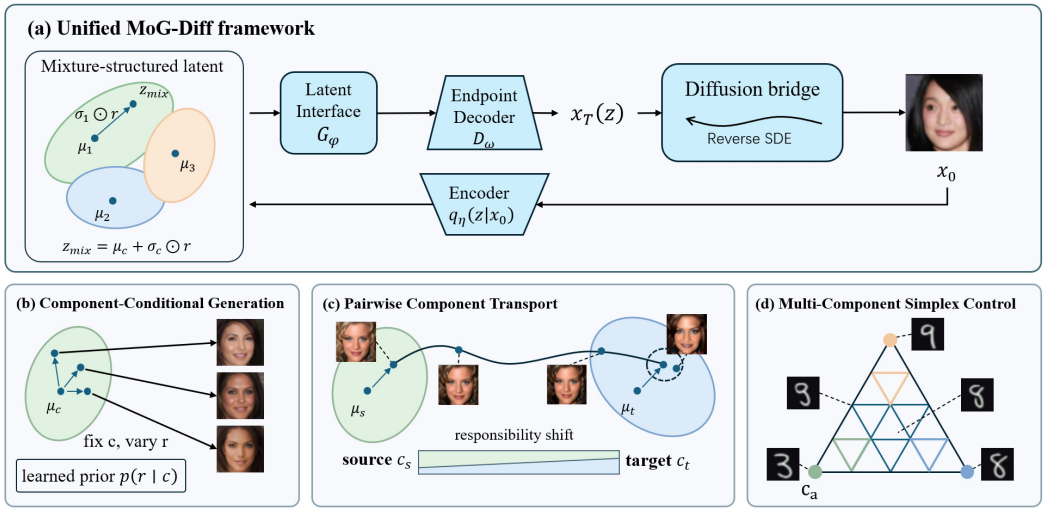

Figure 1: **Overview of MoG-Diff.** (a) MoG-Diff represents each mixture-structured latent as a component and a component-relative residual, and connects it to the endpoint decoder and diffusion bridge through a task-adaptive latent interface. (b) Learned components provide conditions for generation while residuals capture within-component variation. (c) The projected pairwise variant first transports a source latent while preserving its residual coordinate and then applies a target-conditioned Residual-DPM projection tail. (d) Multi-component simplex control extends pairwise transport to controllable mixture responsibilities over three or more components.

formulate variational inference over mixture assignments and provide a pointwise Doob h-transform analysis establishing compatibility between the mixture prior and the endpoint-conditioned bridge. As summarized in Fig. 1, the same component–residual geometry supports component-conditional generation, pairwise component transport with optional residual adaptation, and multi-component simplex control.

Experiments examine semantic organization, reconstruction, conditional generation, and transport on MNIST, Fashion-MNIST, and CelebA. MoG-Diff obtains the highest ACC, NMI, and ARI among the evaluated clustering methods on MNIST and Fashion-MNIST. Under a matched reconstruction protocol, it reduces CelebA MSE by 1.88% and yields small improvements in SSIM and LPIPS, indicating that mixture structuring does not compromise reconstruction fidelity. For component-conditional generation, it covers all ten component-majority classes on both MNIST and Fashion-MNIST and achieves the lowest conditional feature FID among the evaluated methods. On MNIST, responsibility control reduces pairwise responsibility MAE from 0.2065 to 0.0356. In the evaluated three-component setting, it reduces simplex responsibility MAE from 0.0644 to 0.0227. CelebA further provides a qualitative natural-image demonstration of projected component transport. We interpret the simplex experiment as evidence of controllability over mixture responsibilities rather than unrestricted visual blending of several semantic classes.

Our main contributions are:

- MoG-Diff is introduced as a diffusion bridge framework that connects a Gaussian-mixture latent space to the endpoint generator through a task-adaptive interface. Variational mixture inference learns the component structure, and a pointwise Doob h-transform analysis establishes compatibility with endpoint-conditioned bridge generation.
- We develop a component–residual coordinate system for component-conditional generation, mixture-aware pairwise transport, target-conditioned residual adaptation, and multi-component simplex control.
- Experiments on MNIST, Fashion-MNIST, and CelebA assess clustering, reconstruction, component-conditional generation, and transport. The results show improved semantic

organization and preserved reconstruction fidelity, together with quantitative component control on MNIST and qualitative extensions to natural images.

2 Related Work

Diffusion bridges and latent representations. Diffusion bridges generalize conventional data-to-noise diffusion to finite-time transport between informative endpoint distributions. Diffusion Schrödinger Bridge formulates this process as an entropy-regularized path-space problem (De Bortoli et al., 2021), while Denoising Diffusion Bridge Models and I^2SB learn bridge dynamics for conditional generation, translation, and restoration (Zhou et al., 2024; Liu et al., 2023). Related representation models introduce compact latents into diffusion generation. Diffusion Autoencoders separate semantic codes from stochastic detail, while latent score-based models learn generative priors over encoded representations (Preechakul et al., 2022; Vahdat et al., 2021). NoiseCLR discovers unsupervised semantic editing directions in pretrained diffusion models (Dalva & Yanardag, 2024). Diffusion Bridge AutoEncoders further use a latent-dependent endpoint for reconstruction and generation (Kim et al., 2025). MoG-Diff adopts this endpoint-conditioned setting but organizes its conditioning latent through mixture components and component-relative residuals.

Deep clustering and mixture-structured latents. Deep clustering jointly optimizes representations and cluster assignments. DEC refines soft assignments in the embedding space, while IDEC additionally preserves local structure through reconstruction (Xie et al., 2016; Guo et al., 2017). VaDE introduces a Gaussian-mixture variational prior for cluster inference and conditional generation (Jiang et al., 2017). ClusterGAN combines discrete and continuous latent variables with an inverse mapping (Mukherjee et al., 2019), while GM-GAN uses a Gaussian-mixture GAN prior for unsupervised clustering (Ben-Yosef & Weinshall, 2018). TreeVAE learns hierarchical generative clusters (Manduchi et al., 2023), whereas TreeDiffusion conditions a diffusion generator on learned hierarchical cluster representations (da Silva Gonçalves et al., 2026). ClusterDDPM combines an EM-based mixture prior with a conditional DDPM for clustering and generation (Yan et al., 2025). A multimodal diffusion VAE has also been used for generative clustering (Palumbo et al., 2024). These methods use cluster variables for representation learning, assignment, direct decoding, or generator conditioning. MoG-Diff uses mixture coordinates to organize a diffusion-bridge latent space, condition generation, and define inter-component transport.

Conditional generation and latent-space control. Classifier guidance steers diffusion sampling using classifier gradients (Dhariwal & Nichol, 2021), while composable diffusion combines conditional energies to express multiple concepts (Liu et al., 2022). Other methods obtain editing directions from pretrained generators. InterFaceGAN identifies semantic boundaries (Shen et al., 2020), GANSpace extracts interpretable principal directions (Härkönen et al., 2020), and NoiseCLR discovers unsupervised directions in diffusion models (Dalva & Yanardag, 2024). MoG-Diff uses learned mixture statistics to define component-conditional generation and residual-based transport. User-specified responsibility targets can further refine pairwise and simplex paths.

3 Method

MoG-Diff organizes the conditioning latent of a diffusion bridge with a Gaussian mixture and connects it to the endpoint generator through a task-adaptive interface. We next define its generative process, mixture inference, learning objectives, and control operations.

3.1 Generative Process with a Mixture-of-Gaussians Prior

Mixture-structured latent. Let $c \in \{1, \dots, K\}$ denote a mixture-component index and $z \in \mathbb{R}^d$ the structured latent, where d is the latent dimension. The symbol z_{mix} in Fig. 1 refers to the same latent z. We define

$$\begin{aligned} c &\sim \mathrm{Cat}(\boldsymbol{\pi}), \qquad p_\theta(z \mid c) = \mathcal{N}\big(\mu_c, \mathrm{diag}(\sigma_c^2)\big), \\ r &\sim \mathcal{N}(0, I), \qquad z = F_c(r) := \mu_c + \sigma_c \odot r. \end{aligned} \tag{1}$$

Here $\boldsymbol{\pi} \in \Delta^{K-1}$ and $\sigma_c > 0$. The component c identifies a region of the learned geometry, while r records the sample's relative position within it. Components are learned without class supervision and need not correspond one-to-one to semantic classes.

Latent interface and endpoint. The structured latent determines the bridge endpoint through

$$\begin{aligned} g &= G_\phi(z), \qquad x_T = D_\omega(g), \\ x_T &\xrightarrow[\text{conditioned on } g]{\mathcal{B}_\psi} x_0. \end{aligned} \tag{2}$$

Here G_ϕ is either the identity or a learned adapter A_ϕ, representing task-specific configurations of the same framework. Conversely, $q_\eta(z \mid x_0)$ maps an observation to the structured latent. Thus, z provides mixture organization, while g interfaces with the endpoint generator.

Structural prior and residual sampler. In addition to Gaussian residual sampling, MoG-Diff can learn the aggregated within-component variation:

$$r \sim p_\xi(r \mid c), \qquad z = \mu_c + \tau\sigma_c \odot r, \tag{3}$$

where $\tau > 0$ controls variation. Standard mixture sampling is recovered when $p_\xi(r \mid c) = \mathcal{N}(0, I)$ and $\tau = 1$. The learned sampler retains (μ_c, σ_c) as its coordinate system, although the induced conditional distribution of z need not remain Gaussian. Its density and moments are derived in Appendix A.3.

3.2 Endpoint Preservation under Doob's h-Transform

For a fixed latent z, let $y(z) = D_\omega(G_\phi(z))$ and $h_z(x,t) = p_{t,T}(y(z) \mid x)$, where $p_{t,T}$ is the transition density of the reference diffusion $dX_t = f(X_t,t)\,dt + \beta(t)\,dW_t$. Doob's h-transform gives the endpoint-conditioned process

$$dX_t = \left[f(X_t,t) + \beta^2(t)\nabla_{X_t} \log h_z(X_t,t)\right] dt + \beta(t) dW_t. \tag{4}$$

For each admissible fixed z, the ideal bridge satisfies $X_T = y(z)$ almost surely. Changing the prior over z changes the distribution of endpoints, while this conditional pinning property holds pointwise. The proof and reverse-time dynamics are in Appendix A.1. Learned scores and finite-step solvers approximate the ideal bridge.

3.3 Variational Inference over Mixture Components

For an input x_i, the stochastic structured encoder defines

$$q_\eta(z \mid x_i) = \mathcal{N}(z; m_i, \mathrm{diag}(v_i))\,. \tag{5}$$

The coordinate-optimal component assignment uses the expected log density:

$$\begin{aligned} a_{ik} &:= \mathbb{E}_{q_\eta(z|x_i)}[\log p_\theta(z \mid c = k)], \\ \gamma_{ik} &:= q_\eta(c = k \mid x_i) = \mathrm{softmax}_k(a_{ik} + \log \pi_k). \end{aligned} \tag{6}$$

The resulting mixture regularizer is

$$\begin{aligned} \mathcal{L}_{\mathrm{mix},i} = & -\mathcal{H}[q_\eta(z \mid x_i)] \\ & -\sum_{k=1}^{K} \gamma_{ik}\left(a_{ik} + \log \pi_k - \log \gamma_{ik}\right). \end{aligned} \tag{7}$$

Appendix A.2 gives the closed form of a_{ik} and derives the regularizer. The expectation accounts for encoder uncertainty. Evaluating the density only at m_i would omit v_i.

At inference time, clustering and transport operate on a deterministic latent z and use the pointwise responsibility

$$\rho_k(z) := \frac{\pi_k \mathcal{N}(z; \mu_k, \mathrm{diag}(\sigma_k^2))}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(z; \mu_\ell, \mathrm{diag}(\sigma_\ell^2))}. \tag{8}$$

Thus, γ_{ik} integrates encoder uncertainty during training, whereas $\rho_k(z)$ evaluates component membership at a specific latent point.

3.4 Learning Objectives and Numerical Computation

Identity-interface objective. When $g = z$, the bridge and endpoint decoder are trained with the mixture regularizer. Let $\widehat{x}_0$ denote the bridge prediction and $u_k = B^{-1}\sum_{i=1}^{B}\gamma_{ik}$ the mean assignment to component k over a minibatch of size B. We use $\mathcal{L}_{\mathrm{bal}} = D_{\mathrm{KL}}(u\|\,\mathrm{Uniform}(K))$ and

$$\mathcal{L}_{\mathrm{bridge}} = \mathbb{E}\big[w(t)\|\widehat{x}_0(x_t, t, z, D_\omega(z)) - x_0\|_2^2\big]\,, \tag{9}$$

$$\mathcal{L}_{\mathrm{id}} = \mathcal{L}_{\mathrm{bridge}} + \lambda_{\mathrm{mix}}\mathcal{L}_{\mathrm{mix}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}. \tag{10}$$

The bridge prediction term follows DBAE. The mixture regularizer and usage balance organize the learned components.

Learned-interface objective. When $g = A_\phi(z)$, a frozen bridge encoder provides the deterministic posterior-mean target $z_{\mathrm{tgt}} = m_{\mathrm{bridge}}(x)$, rather than a reparameterized sample. We optimize

$$\mathcal{L}_{\mathrm{align}} = \|A_\phi(z) - z_{\mathrm{tgt}}\|_2^2, \tag{11}$$

$$\mathcal{L}_{\mathrm{end}} = \|D_\omega(A_\phi(z)) - x_0\|_2^2, \tag{12}$$

$$\mathcal{L}_{\mathrm{learned}} = \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{end}}\mathcal{L}_{\mathrm{end}}. \tag{13}$$

We update the interface and endpoint decoder while keeping the semantic and bridge encoders, mixture prior, and bridge denoiser frozen. Generation then reuses the pretrained bridge denoiser with the adapted latent and updated endpoint decoder. This task-adaptive optimization preserves a single framework without requiring all modules to share one training loss.

Residual-prior learning. With the mixture geometry fixed, high-confidence component assignments provide whitened residuals $r_0 = (z - \mu_c) \oslash \sigma_c$. A component-conditional denoiser learns their distribution through noise prediction. DDIM sampling (Song et al., 2021a) then supplies $r \sim p_\xi(r \mid c)$ in Eq. equation 3. The same model can refine an existing residual for projected transport. The filtering rule, forward noising, and training objective are specified in Appendix B.1. This auxiliary sampler leaves the representation-learning mixture regularizer unchanged.

Numerical computation. Mixture likelihoods are evaluated in log space, and images are obtained from a numerical reverse bridge. Solver and guidance settings are given in Appendix B.

3.5 Component Transport and Multi-Component Simplex Control

Residual-preserving pairwise transport. For a source latent z_s assigned to component s, we define

$$r_s = (z_s - \mu_s) \oslash \sigma_s, \tag{14}$$

$$\begin{aligned}\mu(\alpha) &= (1-\alpha)\mu_s + \alpha\mu_t,\\ \log\sigma(\alpha) &= (1-\alpha)\log\sigma_s + \alpha\log\sigma_t,\\ z_{\mathrm{cov}}(\alpha) &= \mu(\alpha) + \sigma(\alpha)\odot r_s.\end{aligned} \tag{15}$$

where $\alpha \in [0, 1]$. The path interpolates component statistics while preserving the source residual. At $\alpha = 1$, it reaches $\mu_t + \sigma_t \odot r_s$: a target-component latent that retains the source sample's relative within-component coordinate.

Analytic cycle. The residual-preserving endpoint map has an analytic inverse, so $T_{t\to s}(T_{s\to t}(z)) = z$ in latent space. Appendix E.2 gives the inverse and evaluation protocol. This identity does not imply image-identity preservation.

Responsibility-controlled pairwise paths. Because α need not equal the realized MoG responsibility, we optimize each latent from $z_{\mathrm{cov}}(\alpha)$ toward

$$\rho^*(\alpha) = (1-\alpha)e_s + \alpha e_t, \tag{16}$$

$$\mathcal{L}_{\mathrm{ctrl}}(z;\alpha) = \lambda_\rho\|\rho(z) - \rho^*(\alpha)\|_2^2 \tag{17}$$

$$+ \lambda_r\|(z - \mu(\alpha)) \oslash \sigma(\alpha) - r_s\|_2^2 \tag{18}$$

$$+ \lambda_p\frac{-\log p_\theta(z)}{d}. \tag{19}$$

Table 1: Semantic organization and reconstruction fidelity. (a) Clustering on MNIST and Fashion-MNIST (F-MNIST). (b) CelebA reconstruction on the complete official test split.

(a) Semantic Organization

	MNIST			F-MNIST		
Method	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
VaDE	0.9512	0.8868	0.8959	0.5301	0.5347	0.3556
IDEC	0.8089	0.7882	0.7326	0.5700	0.6087	0.4503
ClusterGAN	0.8061	0.7729	0.7008	0.5584	0.5498	0.4216
TreeVAE	0.8565	0.8448	0.7921	0.5492	0.5958	0.4338
MoG-Diff	**0.9532**	**0.8908**	**0.9000**	**0.6245**	**0.6125**	**0.4721**

(b) Reconstruction Fidelity

Dataset	Method	MSE ↓	SSIM ↑	LPIPS ↓
CelebA	DBAE	0.001278	0.910858	0.025766
CelebA	**MoG-Diff**	**0.001254**	**0.911984**	**0.025742**

The terms enforce responsibility accuracy, residual preservation, and mixture density. Model parameters remain frozen.

Target-conditioned residual projection. For natural-image transport, the analytic path reaches $z_{\text{arr}} = z_{\text{cov}}(1) = \mu_t + \sigma_t \odot r_s$. A frozen target-conditioned Residual-DPM then refines the source residual to r_p. The second stage interpolates within the target component:

$$z_{\text{tail}}(\beta) = \mu_t + \sigma_t \odot \text{SLERP}(r_s, r_p; \beta), \qquad \beta \in [0, 1]. \tag{20}$$

The first stage preserves r_s while changing component statistics. The tail fixes the target statistics while adapting the residual, without a separately encoded target latent. Its noising, denoising, and path settings are given in Appendix E.3. Refinement changes the residual, so the complete path is not analytically invertible.

Multi-component simplex control. Pairwise transport extends to selected components $\mathcal{C} = \{c_1, \ldots, c_M\}$ with weights $w \in \Delta^{M-1}$. Interpolating component means and log-scales while preserving the source residual gives

$$z_{\text{ana}}(w) = \sum_{i=1}^{M} w_i \mu_{c_i} + \exp\left(\sum_{i=1}^{M} w_i \log \sigma_{c_i}\right) \odot r_s. \tag{21}$$

The target responsibility assigns weight w_i to component c_i and zero mass elsewhere. Responsibility refinement reuses Eq. equation 19 with the corresponding interpolated statistics and target weights.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate semantic organization and component-conditional generation on MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017), reconstruction and qualitative natural-image extensions on CelebA (Liu et al., 2015), and transport and simplex control on MNIST. We use $K = 10$ components for MNIST and Fashion-MNIST and $K = 5$ for CelebA. Clustering baselines include VaDE, IDEC, ClusterGAN, and TreeVAE (Jiang et al., 2017; Guo et al., 2017; Mukherjee et al., 2019; Manduchi et al., 2023). Generation baselines include DBAE + KMeans, ClusterGAN, and TreeDiffusion (da Silva Gonçalves et al., 2026). Transport strategies are compared within the same frozen MoG-Diff geometry. Evaluation configurations and checkpoint lineage are documented in Appendix C.

Protocols and metrics. MNIST clustering uses the combined official splits, while the remaining evaluations use official test splits. DBAE and MoG-Diff share the same reconstruction protocol.

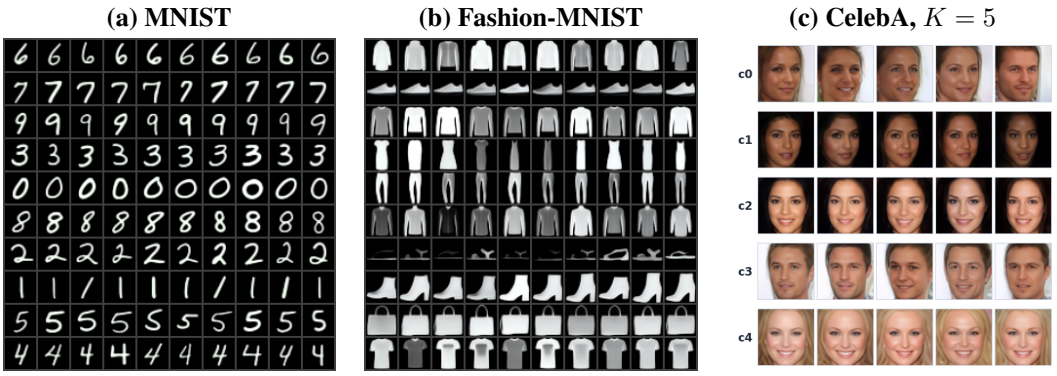

Figure 2: **Component-conditional generation with MoG-Diff.** Each row fixes a component and columns show independent samples. Panels (a,b) use Residual-DPM, whereas (c) shows a task-specific CelebA latent-refinement configuration. All images are final 40-step bridge outputs. CelebA components are interpreted through post-hoc attribute profiles rather than mutually exclusive class labels.

Other image results use final 40-step Heun outputs unless stated otherwise. Labels are used only for post-hoc evaluation, and generation deviations are computed over five sampling repetitions from a fixed checkpoint. We report ACC, NMI, and ARI for clustering; MSE, SSIM, and LPIPS for reconstruction; consistency, coverage, diversity, and Conditional Feature FID for generation; and endpoint success, responsibility MAE, density percentile, and adjacent LPIPS for transport. Conditional Feature FID is computed in dataset-specific classifier spaces rather than the standard Inception space, and diversity is interpreted jointly with consistency and coverage.

4.2 Semantic Organization and Reconstruction Fidelity

We evaluate semantic organization on MNIST and Fashion-MNIST and reconstruction fidelity on CelebA.

Semantic organization. As summarized in Table 1(a), MoG-Diff achieves the best ACC, NMI, and ARI on both datasets, slightly surpassing the strong generative-clustering baseline VaDE on MNIST. The improvement is more pronounced on Fashion-MNIST, where MoG-Diff increases ACC from the best baseline result of 0.5700 to 0.6245 while also obtaining the highest NMI and ARI.

Reconstruction fidelity. For CelebA, Table 1(b) reports comparable reconstruction fidelity for MoG-Diff and DBAE, with small numerical improvements for MoG-Diff in MSE, SSIM, and LPIPS. This suggests that mixture-based latent organization preserves reconstruction quality.

4.3 Component-Conditional Generation with Residual Modeling

We quantitatively evaluate component-conditioned Residual-DPM generation on MNIST and Fashion-MNIST and use CelebA as a qualitative natural-image extension.

Component-conditioned generation. Following Eq. equation 3, MoG-Diff samples a component-conditioned residual and decodes it through the complete bridge. The component specifies a semantic region, while the residual describes within-component variation. Sampling hyperparameters are listed in Appendix C.4.

Class-based generation. For MNIST, MoG-Diff achieves perfect consistency, complete coverage, and the lowest Conditional Feature FID among the methods compared in Table 2. On Fashion-MNIST, it is the only method to cover all ten classes and reduces the best baseline Conditional Feature FID from 366.25 to 210.54. ClusterGAN and TreeDiffusion obtain higher raw diversity, but this coincides with lower consistency and incomplete coverage, indicating greater semantic mix-

Table 2: Component-conditional generation on MNIST and Fashion-MNIST (F-MNIST). MoG-Diff uses component-whitened Residual-DPM. Baselines use their own generators. Conditional Feature FID uses evaluation-only Hungarian matching and dataset-specific classifier features (not Inception-FID). Values are mean ± standard deviation over five sampling repetitions from a fixed checkpoint.

Dataset	Method	Consistency ↑	Coverage ↑	Raw Diversity ↑	Conditional Feature FID ↓
MNIST	VaDE	0.9998±0.0004	**10.0±0.0**	10.73±0.13	330.36±3.78
	DBAE + KMeans	**1.0000±0.0000**	8.0±0.0	3.08±0.03	1109.87±1.45
	ClusterGAN	0.8264±0.0112	9.0±0.0	34.31±0.47	465.23±14.21
	TreeDiffusion	0.7324±0.0048	9.2±0.45	**36.65±0.18**	374.09±6.69
	MoG-Diff	**1.0000±0.0000**	**10.0±0.0**	18.16±0.08	**199.40±2.28**
F-MNIST	VaDE	0.7908±0.0039	7.0±0.0	15.36±0.12	479.69±14.69
	DBAE + KMeans	0.9236±0.0066	7.4±0.5	2.93±0.05	828.78±2.21
	ClusterGAN	0.6314±0.0099	7.8±0.4	26.36±0.20	366.25±10.91
	TreeDiffusion	0.5178±0.0186	7.2±0.45	**26.96±0.27**	449.06±26.16
	MoG-Diff	**0.9252±0.0053**	**10.0±0.0**	15.58±0.23	**210.54±4.29**

Table 3: Control within a fixed MoG-Diff latent geometry. All methods use the same frozen checkpoint and differ only in their inference-time paths. Endpoint and corner success measure predicted classes, not image identity.

(a) Pairwise Component Transport

Path	Endpoint Succ. ↑	Resp. MAE ↓	Min. Density Pct. ↑	Adjacent LPIPS ↓
Linear	0.9800	0.2073	0.3308	0.00535
Cov.-aware	0.9800	0.2065	0.3381	**0.00534**
Resp. control	**0.9960**	**0.0356**	**0.3557**	0.00802

(b) Three-Component Simplex Control

Construction	Resp. MAE ↓	Density Pct. ↑	Corner Succ. ↑	Edge LPIPS ↓
Cov. barycentric	0.0644	**0.5167**	1.0000	**0.0147**
Resp. control	**0.0227**	0.4380	1.0000	0.0198

ing. MoG-Diff instead retains useful within-component variation while preserving component-level semantics, as illustrated by the MNIST and Fashion-MNIST samples in Fig. 2(a,b).

Natural-image extension. CelebA attributes overlap rather than form mutually exclusive classes, so Fig. 2(c) serves as a qualitative natural-image extension. It uses a task-specific raw latent refiner with the same frozen $K = 5$ MoG-Diff representation and bridge, producing distinguishable attribute profiles with within-component variation. Because this sampler differs from Table 2, the panel is interpreted qualitatively. Implementation details and additional Residual-DPM results are provided in Appendix D.

4.4 Component Transport and Multi-Component Simplex Control

We evaluate pairwise transport on MNIST, provide a qualitative CelebA extension, and generalize the same component–residual geometry to three-component simplex control.

Pairwise component transport. Following Sec. 3.5, we compare linear, covariance-aware, and responsibility-controlled paths on MNIST using repeated real-source selections. All models remain frozen, and labels are used only for post-hoc evaluation. The complete trajectory protocol is specified in Appendix E.1.

Pairwise control accuracy and path quality. Table 3(a) reports the pairwise results, with representative MNIST trajectories shown in Fig. 3(a). Linear and covariance-aware paths attain the same classifier endpoint success and smoothness. Responsibility control reduces MAE from 0.2065 to 0.0356 and raises endpoint success to 0.9960, at the cost of higher adjacent LPIPS. This reveals a trade-off between analytic smoothness and posterior-control accuracy. Analytic forward–reverse transport recovers the latent up to 10^{-7} numerical error without implying image-identity preservation. Complete cycle results are reported in Table 9 of Appendix E.2.

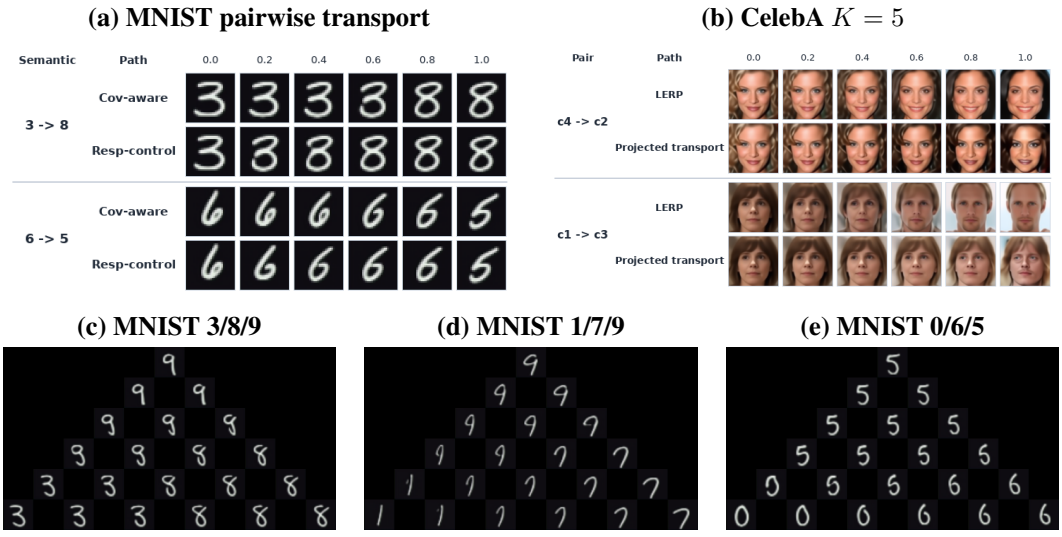

Figure 3: **Pairwise and multi-component control.** (a) MNIST pairwise transport; (b) CelebA LERP and projected transport; (c–e) covariance-barycentric MNIST simplexes.

Projected natural-image transport on CelebA. Figure 3(b) compares real-endpoint LERP with projected transport. LERP connects two encoded real images, whereas projected transport starts from the same source, moves analytically to the target component, and applies the refinement in Eq. equation 20 without using the LERP target latent. Its endpoint is therefore a source-associated target-component realization. We report this result qualitatively because no matched identity metric or external baseline is available. Projection settings are given in Appendix E.3.

Three-component simplex control. Table 3(b) evaluates Eq. equation 21 on the MNIST triples 3/8/9, 1/7/9, and 0/6/5. Figure 3(c–e) shows their covariance-barycentric simplex lattices, respectively. Both constructions succeed at all simplex corners. Responsibility control reduces MAE from 0.0644 to 0.0227, whereas the analytic construction achieves higher density and smoother edges. This extends control from pairwise paths to a three-component responsibility simplex. The simplex interior represents mixture responsibilities rather than literal visual fusion. Complete lattice settings are provided in Appendix E.4.

Summary. Overall, analytic transport favors residual preservation and smoothness, whereas responsibility optimization improves posterior control. The learned mixture therefore serves as both a clustering prior and an operational geometry for controllable generation.

5 Conclusion

We introduced MoG-Diff, which represents diffusion-bridge latents through a Gaussian-mixture component and a component-relative residual. Experiments show that MoG-Diff improves semantic organization, maintains comparable reconstruction fidelity, supports component-conditional generation, and enables pairwise transport and three-component simplex control. The control results reveal a trade-off among responsibility accuracy, latent density, and local smoothness, and simplex interiors represent mixture responsibilities rather than literal visual fusion. Overall, MoG-Diff shows that the learned mixture geometry can serve as a shared coordinate system for representation learning and controllable diffusion-bridge generation. Future work will assess transport with independent semantic measures and extend simplex control beyond three components.

AI use statement

In this work, we used generative AI tools to translate portions of the manuscript, assist with parts of the mathematical proofs, support qualitative analysis and interpretation of experimental results,

and provide feedback on the experimental evaluation. Additionally, we used generative AI tools to draft and revise portions of the manuscript and to find and summarize related literature. The authors reviewed AI-assisted material before incorporating it into the paper and take responsibility for the final text, mathematical arguments, results, and citations.

REFERENCES

Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. *arXiv preprint arXiv:1808.10356*, 2018.

Jorge da Silva Gonçalves, Laura Manduchi, Moritz Vandenhirtz, and Julia E. Vogt. TreeDiffusion: Hierarchical generative clustering for conditional diffusion. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, volume 16013 of *Lecture Notes in Computer Science*, pp. 447–462. Springer, 2026. doi: 10.1007/978-3-032-05962-8_26. URL https://link.springer.com/chapter/10.1007/978-3-032-05962-8_26.

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

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/940392f5f32a7ade1cc201767cf83e31-Abstract.html.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. doi: 10.48550/arXiv.1611.02648. URL https://arxiv.org/abs/1611.02648.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1753–1759. International Joint Conferences on Artificial Intelligence Organization, 2017. doi: 10.24963/ijcai.2017/243. URL https://www.ijcai.org/proceedings/2017/243.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9841–9850. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6fe43269967adbb64ec6149852b5cc3e-Abstract.html.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1965–1972. International Joint Conferences on Artificial Intelligence Organization, 2017. doi: 10.24963/ijcai.2017/273. URL https://www.ijcai.org/proceedings/2017/273.

Yeongmin Kim, Kwanghyeon Lee, Minsang Park, Byeonghu Na, and Il-chul Moon. Diffusion bridge autoencoders for unsupervised representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL

https://proceedings.iclr.cc/paper_files/paper/2025/hash/d368aba36f74776cc7a1079332a31973-Abstract-Conference.html.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://proceedingsoftheieee.ieee.org/gradient-based-learning-applied-to-document-recognition/.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image schrödinger bridge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22042–22062. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23ai.html.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision – ECCV 2022*, volume 13677 of *Lecture Notes in Computer Science*, pp. 423–439. Springer, 2022. doi: 10.1007/978-3-031-19790-1_26. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/6940_ECCV_2022_paper.php.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Liu_Deep_Learning_Face_ICCV_2015_paper.html.

Laura Manduchi, Moritz Vandenhirtz, Alain Ryser, and Julia E. Vogt. Tree variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 36, pp. 54952–54986. Curran Associates, Inc., 2023. doi: 10.52202/075280-2398. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac58b418745b3e5f10c80110c963969f-Abstract-Conference.html.

Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. ClusterGAN: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4610–4617, 2019. doi: 10.1609/aaai.v33i01.33014610. URL https://ojs.aaai.org/index.php/AAAI/article/view/4385.

Emanuele Palumbo, Laura Manduchi, Sonia Laguna, Daphné Chopard, and Julia E. Vogt. Deep generative clustering with multimodal diffusion variational autoencoders. In *International Conference on Learning Representations*, 2024.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Preechakul_Diffusion_Autoencoders_Toward_a_Meaningful_and_Decodable_Representation_CVPR_2022_paper.html.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Shen_Interpreting_the_Latent_Space_of_GANs_for_Semantic_Face_Editing_CVPR_2020_paper.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11287–11302. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5dca4c6b9e244d24a30b4c45601d9720-Abstract.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. doi: 10.48550/arXiv.1708.07747. URL https://arxiv.org/abs/1708.07747.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 478–487. PMLR, 2016. URL https://proceedings.mlr.press/v48/xieb16.html.

Jie Yan, Jing Liu, and Zhong-yuan Zhang. Clusterddpm: an em clustering framework with denoising diffusion probabilistic models. *Information Sciences*, pp. 122518, 2025.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/20e45668fefa793bd9f2edf19be12c4b-Abstract-Conference.html.

APPENDIX

A MATHEMATICAL DETAILS OF MOG-DIFF

A.1 ENDPOINT PRESERVATION AND REVERSE-TIME BRIDGE DYNAMICS

Fix a latent z and write $y(z) = D_\omega(G_\phi(z))$. For a reference diffusion with transition density $p_{t,T}$, define

$$\begin{aligned} dX_t &= f(X_t, t)\, dt + \beta(t)\, dW_t, \\ h_z(x, t) &= p_{t,T}(y(z) \mid x). \end{aligned} \tag{22}$$

Assume $h_z > 0$ and that the fixed-endpoint bridge is well defined. The backward transition equation makes $h_z(X_t, t)$ a positive martingale for $t < T$. Its Doob transform has drift $f + \beta^2 \nabla_x \log h_z$, as stated in Eq. equation 4.

Pointwise endpoint compatibility. For a fixed starting state $X_0 = x$ and any bounded function F of the path up to time $t < T$, the transformed expectation satisfies

$$\mathbb{E}_x^{h_z}[F] = \frac{\mathbb{E}_x[F\, p_{t,T}(y(z) \mid X_t)]}{p_{0,T}(y(z) \mid x)} = \mathbb{E}_x[F \mid X_T = y(z)]. \tag{23}$$

The equality follows from the Markov property and the conditional-density identity. Under the regularity conditions for the conditioned bridge, its terminal value therefore satisfies

$$X_T = y(z) \qquad \text{almost surely.} \tag{24}$$

This argument holds for every admissible fixed z and does not depend on the prior used to draw z. Sampling z changes the endpoint distribution through the pushforward $D_\omega \circ G_\phi$; that image-space distribution need not be a Gaussian mixture.

Reverse dynamics. Write

$$b_z(x, t) = \nabla_x \log h_z(x, t), \qquad s_\psi^z(x, t) := s_\psi(x, t; y(z), G_\phi(z)).$$

With s_ψ^z approximating the conditional bridge score, the reverse SDE and probability-flow ODE in decreasing time are

$$dX_t = \big[f - \beta^2(s_\psi^z - b_z)\big]dt + \beta\, d\overline{W}_t, \tag{25}$$

$$dX_t = \big[f - \beta^2(\tfrac{1}{2}s_\psi^z - b_z)\big]dt. \tag{26}$$

The arguments (X_t, t) are suppressed for compactness. These dynamics follow fixed-endpoint diffusion bridges, and $y(z)$ is supplied by the diffusion-bridge autoencoder (**?**Kim et al., 2025). Endpoint pinning is a property of the ideal continuous-time conditioned process; learned scores and finite-step numerical solvers approximate it.

A.2 EXPECTED COMPONENT LOG DENSITY AND KL

For each observation x_i, factor the variational joint distribution as $q_\eta(z, c \mid x_i) = q_\eta(z \mid x_i)\, q_\eta(c \mid x_i)$, with $q_\eta(c = k \mid x_i) = \gamma_{ik}$. The reference mixture joint is $p_\theta(z, c = k) = \pi_k p_\theta(z \mid c = k)$. Since the encoder covariance is $\mathrm{diag}(v_i)$, where v_{ij} is a variance, the expected component log density has the closed form

$$\begin{aligned} a_{ik} &:= \mathbb{E}_{q_\eta(z \mid x_i)}[\log p_\theta(z \mid c = k)] \\ &= -\frac{1}{2} \sum_{j=1}^{d} \left[\log(2\pi\sigma_{kj}^2) + \frac{v_{ij} + (m_{ij} - \mu_{kj})^2}{\sigma_{kj}^2} \right]. \end{aligned} \tag{27}$$

The v_{ij} term comes from $\mathbb{E}_{q_\eta}[(z_j - \mu_{kj})^2] = v_{ij} + (m_{ij} - \mu_{kj})^2$; evaluating a Gaussian density only at m_i would omit it.

Expanding the joint KL term gives

$$D_{\mathrm{KL}}(q_\eta(z, c \mid x_i) \| p_\theta(z, c)) = -\mathcal{H}[q_\eta(z \mid x_i)] + \sum_{k=1}^{K} \gamma_{ik}(\log \gamma_{ik} - \log \pi_k - a_{ik}), \tag{28}$$

which is the regularizer in Eq. equation 7. For fixed encoder and mixture parameters, let $\widetilde{\gamma}_{ik} = \pi_k e^{a_{ik}} / \sum_\ell \pi_\ell e^{a_{i\ell}}$. The assignment-dependent term equals a KL to $\widetilde{\gamma}_i$ plus a constant, yielding the coordinate-optimal assignment in Eq. equation 6. The Gaussian encoder entropy is $\mathcal{H}[q_\eta(z \mid x_i)] = \frac{1}{2} \sum_j \log(2\pi e v_{ij})$.

For a minibatch of size B, let $u_k = B^{-1} \sum_{i=1}^{B} \gamma_{ik}$ denote the average component assignment. The separate balancing term is

$$\mathcal{L}_{\mathrm{bal}} = D_{\mathrm{KL}}(u \| \operatorname{Uniform}(K)) = \sum_{k=1}^{K} u_k \log(K u_k). \tag{29}$$

The variational γ_{ik} integrates encoder uncertainty during training, whereas $\rho_k(z)$ in Eq. equation 8 evaluates membership at a deterministic latent used for clustering or transport.

A.3 Distribution Induced by Conditional Residual Sampling

This subsection characterizes the latent distribution induced by the component-conditional residual sampler used in MoG-Diff. Here z_{mix} denotes the structured latent z in Sec. 3.1.

For a component c, let $\sigma_c \in \mathbb{R}^d_+$ denote its component-wise scale vector, and let $r \in \mathbb{R}^d$ denote the corresponding component-relative residual coordinate. Given a temperature $\tau > 0$, the residual sampler constructs the structured latent as

$$r \sim p_\xi(r \mid c), \qquad z_{\mathrm{mix}} = \mu_c + \tau \sigma_c \odot r, \tag{30}$$

where τ controls the magnitude of within-component variation.

Let

$$S_c = \operatorname{diag}(\sigma_c). \tag{31}$$

For a fixed component c, Eq. equation 30 defines the invertible affine transformation

$$z_{\mathrm{mix}} = \mu_c + \tau S_c r. \tag{32}$$

Assuming that the residual distribution admits a density, the induced conditional density in the structured latent space follows from the change-of-variables formula:

$$p_{\xi,\tau}(z_{\mathrm{mix}} \mid c) = \frac{p_\xi\big((z_{\mathrm{mix}} - \mu_c) \oslash (\tau \sigma_c) \mid c\big)}{\tau^d \prod_{j=1}^{d} \sigma_{cj}}. \tag{33}$$

The corresponding marginal latent distribution is

$$p_{\xi,\tau}(z_{\mathrm{mix}}) = \sum_{c=1}^{K} \pi_c \, p_{\xi,\tau}(z_{\mathrm{mix}} \mid c), \tag{34}$$

where $\boldsymbol{\pi}$ denotes the mixture weights.

The original diagonal Gaussian-mixture prior is recovered as a special case. Specifically, if

$$p_\xi(r \mid c) = \mathcal{N}(0, I), \qquad \tau = 1, \tag{35}$$

then

$$p_{\xi,\tau}(z_{\mathrm{mix}} \mid c) = \mathcal{N}\big(z_{\mathrm{mix}}; \mu_c, \operatorname{diag}(\sigma_c^2)\big), \tag{36}$$

and Eq. equation 34 reduces to the original mixture-of-Gaussians prior.

When $p_\xi(r \mid c)$ is learned by a conditional Residual-DPM, the residual distribution is not restricted to be standard Gaussian. Consequently, the induced conditional distribution $p_{\xi,\tau}(z_{\mathrm{mix}} \mid c)$ is generally non-Gaussian. Nevertheless, (μ_c, σ_c) continues to define the reference location and scale of

the component-relative coordinate system. The Residual-DPM therefore models within-component variation without replacing the mixture-level organization.

More precisely, if the corresponding moments exist, the induced conditional moments satisfy

$$\begin{aligned} \mathbb{E}[z_{\mathrm{mix}} \mid c] &= \mu_c + \tau S_c \mathbb{E}[r \mid c], \\ \mathrm{Cov}(z_{\mathrm{mix}} \mid c) &= \tau^2 S_c \,\mathrm{Cov}(r \mid c) S_c. \end{aligned} \tag{37}$$

Thus, for a learned non-Gaussian residual distribution, μ_c and σ_c remain the reference parameters of the coordinate transformation, but they need not coincide with the actual mean and standard deviation of the induced conditional distribution.

For component-conditional generation, the component index c is fixed to the desired component, and a residual is sampled according to

$$r \sim p_\xi(r \mid c). \tag{38}$$

For unconditional generation, the component is additionally sampled as

$$c \sim \mathrm{Cat}(\boldsymbol{\pi}), \qquad r \sim p_\xi(r \mid c). \tag{39}$$

After constructing z_{mix} using Eq. equation 30, MoG-Diff applies the task-dependent latent interface

$$g = G_\phi(z_{\mathrm{mix}}), \tag{40}$$

where G_ϕ is either the identity or the learned adapter A_ϕ, depending on the task configuration. The resulting g conditions the endpoint decoder and image bridge as in Eq. equation 2. The component index controls the coarse latent region, whereas the conditional residual sampler controls continuous variation within that region.

The temperature τ leaves the learned mixture parameters $(\boldsymbol{\pi}, \mu_c, \sigma_c)$ unchanged, but rescales the effective variation of the induced conditional distribution. As $\tau \to 0$, the generated structured latents collapse toward the reference location μ_c, whereas larger values of τ produce broader within-component variation. In practice, the Residual-DPM is used as a generative mechanism for sampling residuals; an explicit closed-form likelihood for its learned residual distribution is not required.

B Training and Numerical Implementation

This appendix specifies residual-prior training, numerical parameterization, and sampling procedures. Latent residual sampling and image-space diffusion-bridge sampling are separate stages.

Training-stage attribution. The endpoint decoder and image bridge, including $\mathcal{L}_{\mathrm{bridge}}$, follow the DBAE backbone. The MoG component parameters, variational mixture regularizer $\mathcal{L}_{\mathrm{mix}}$, and usage balance $\mathcal{L}_{\mathrm{bal}}$ provide the added latent organization. The learned-interface configuration subsequently optimizes the adapter and endpoint decoder while the bridge and mixture geometry stay frozen. Residual-DPM is trained after this representation stage as an auxiliary sampler; the transport paths and simplex controls are inference-time constructions.

B.1 Conditional Residual-DPM Training and Refinement

After fitting the mixture representation, we keep latents satisfying $\max_k \rho_k(z) \geq \delta$, assign $c = \arg\max_k \rho_k(z)$, and express each selected latent in the coordinates of its assigned component. With $\epsilon \sim \mathcal{N}(0, I)$, the residual denoiser is trained by

$$r_0 = (z - \mu_c) \oslash \sigma_c, \tag{41}$$

$$r_t = \sqrt{\overline{\alpha}_t}\, r_0 + \sqrt{1 - \overline{\alpha}_t}\, \epsilon, \tag{42}$$

$$\mathcal{L}_{\mathrm{res}} = \mathbb{E}\|\epsilon_\xi(r_t, t, c) - \epsilon\|_1. \tag{43}$$

The trained conditional Residual-DPM generates a component-specific residual by DDIM denoising from Gaussian noise. That residual is mapped back through Eq. equation 3 before the image bridge. The generator thus models within-component variation in the fixed MoG coordinate system; the structural mixture prior used in representation learning is unchanged. The same frozen residual model can denoise an existing source residual from a shorter forward-noising horizon during projected transport, as specified in Appendix E.3. The raw-latent refiner used in the qualitative CelebA generation panel is a separate task-specific configuration detailed in Appendix D.

B.2 Stable Mixture Computation

The mixture weights are parameterized by unconstrained logits $\ell_1, \ldots, \ell_K$:

$$\pi_k = \frac{\exp(\ell_k)}{\sum_{j=1}^{K} \exp(\ell_j)}. \tag{44}$$

The component variances are represented in log space. Let v_{kj} denote the log-variance of dimension j in component k. Then

$$\sigma_{kj}^2 = \exp(v_{kj}), \qquad \sigma_{kj} = \exp\left(\frac{1}{2} v_{kj}\right). \tag{45}$$

For a deterministic latent z, the unnormalized log responsibility of component k is

$$\begin{aligned} a_k(z) = \log \pi_k - \frac{1}{2} \sum_{j=1}^{d} \Big[& \log(2\pi) + v_{kj} \\ & + (z_j - \mu_{kj})^2 \exp(-v_{kj}) \Big]. \end{aligned} \tag{46}$$

The normalized responsibilities are evaluated in log space:

$$\rho_k(z) = \exp\left(a_k(z) - \operatorname{logsumexp}_\ell a_\ell(z)\right). \tag{47}$$

This computation avoids explicitly normalizing very small component probabilities. The same log-space representation is used when evaluating mixture densities and component assignments.

B.3 Image-Space Diffusion-Bridge Solver

For a fixed latent, the bridge starts from $x_T = D_\omega(G_\phi(z))$. The continuous-time formulation admits both a reverse-time stochastic differential equation and a deterministic probability-flow ordinary differential equation, given in Eqs. equation 25 and equation 26.

The reported image samples use the deterministic probability-flow form. Let X_i denote the state at solver level i, and let $v_\psi(X_i, \sigma_i; x_T, z)$ denote the bridge-conditioned probability-flow drift. For a descending noise schedule

$$\sigma_0 > \sigma_1 > \cdots > \sigma_{N-1} > \sigma_N = 0,$$

one Heun step is computed as

$$d_i = v_\psi(X_i, \sigma_i; x_T, z), \tag{48}$$

$$\widetilde{X}_{i+1} = X_i + (\sigma_{i+1} - \sigma_i) d_i, \tag{49}$$

$$\widetilde{d}_{i+1} = v_\psi(\widetilde{X}_{i+1}, \sigma_{i+1}; x_T, z), \tag{50}$$

$$X_{i+1} = X_i + (\sigma_{i+1} - \sigma_i) \frac{d_i + \widetilde{d}_{i+1}}{2}. \tag{51}$$

The noise levels follow a power-law schedule. For $i = 0, \ldots, N-1$, define

$$\begin{aligned} \Delta_\sigma &= \sigma_{\min}^{1/\rho_{\text{sched}}} - \sigma_{\max}^{1/\rho_{\text{sched}}}, \\ \sigma_i &= \left[\sigma_{\max}^{1/\rho_{\text{sched}}} + \frac{i}{N-1} \Delta_\sigma \right]^{\rho_{\text{sched}}}. \end{aligned} \tag{52}$$

Unless otherwise stated, the image-bridge experiments use $\rho_{\text{sched}} = 7$, $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, and zero churn. With zero churn, the image-space trajectory is deterministic conditional on the endpoint, latent condition, and initial state. Additional stochasticity therefore comes only from the latent sampler or from explicitly enabled sampling noise.

A nominal N-step Heun trajectory uses $2N - 1$ denoiser evaluations because the final step does not require a second correction evaluation. Thus, the 40-step bridge used by the component-generation and transport experiments uses 79 function evaluations, while the 140-step bridge used by the CelebA reconstruction comparison uses 279 function evaluations.

B.4 Guidance and Endpoint Conditioning

The bridge drift has a denoising term and an endpoint conditioning term. A scalar guidance parameter controls the weight of the endpoint term:

$$v_\psi^{(w)} = v_{\text{denoise}} - w\, v_{\text{endpoint}}, \tag{53}$$

where w denotes the bridge guidance strength.

Setting $w = 0$ removes the additional analytical endpoint-score contribution, but does not remove the bridge. The endpoint x_T remains the initial state of the bridge trajectory and remains available to the denoiser throughout sampling. Therefore, a zero-guidance run still executes the complete image-space bridge.

The principal image-generation configurations use 40-step Heun sampling. The MNIST full-bridge generation result uses guidance 0.5, whereas the Fashion-MNIST full-bridge result uses guidance 0. The CelebA qualitative generation and transport demonstrations use guidance 1. These settings are part of the sampling configuration and do not modify the learned mixture parameters.

B.5 Latent Residual Sampling

The latent residual sampler is separate from the image-space bridge solver. For component-conditional generation, c is fixed to the requested component; for unconditional generation, $c \sim \mathrm{Cat}(\boldsymbol{\pi})$. Given c, the sampler constructs

$$r \sim p_\xi(r \mid c), \qquad z_{\text{mix}} = \mu_c + \tau\sigma_c \odot r. \tag{54}$$

The latent residual diffusion model uses a denoising diffusion process and DDIM sampling. The latent sampling trajectory is independent of the subsequent image-space Heun trajectory. In the reported component-conditional experiments, the latent residual sampler uses 100 latent denoising steps, followed by the image-space bridge.

The complete generation pipeline is therefore

$$\begin{aligned}(c, r) \longrightarrow z_{\text{mix}} \longrightarrow g = G_\phi(z_{\text{mix}}) \\ \longrightarrow x_T = D_\omega(g) \longrightarrow \widehat{x}_0.\end{aligned} \tag{55}$$

Here G_ϕ is the identity for an identity-interface configuration and the learned adapter A_ϕ for a learned-interface configuration. The latent sampler and the image bridge are consequently treated as two separate numerical stages.

B.6 Protocol Summary

Table 4 summarizes the numerical solvers used in the reported experiments.

Task	Dataset	Latent sampler	Image solver	NFE
Generation	MNIST	Residual-DPM, 100 steps	Heun, 40 steps	79
Generation	Fashion-MNIST	Residual-DPM, 100 steps	Heun, 40 steps	79
Transport	MNIST	Fixed latent paths	Heun, 40 steps	79
Simplex	MNIST	Fixed latent paths	Heun, 40 steps	79
Qualitative extension	CelebA	Component-conditioned latent sampler	Heun, 40 steps	79
Reconstruction	CelebA	Posterior mean	Heun, 140 steps	279

Table 4: Summary of the numerical solvers used in the reported experiments. Latent denoising steps and image-bridge solver steps are separate. NFE denotes the number of image-bridge denoiser evaluations for the deterministic Heun trajectory.

All numerical comparisons use the same solver configuration within the corresponding comparison. The solver settings are not treated as additional learned model parameters.

C Experimental Protocols and Checkpoint Lineage

This appendix specifies the dataset splits, evaluation conventions, task-specific checkpoints, and sampling configurations used in the main paper. Numerical results are reported in the main text and the subsequent appendices.

C.1 Evaluation Conventions

Training seeds determine model fitting, whereas sampling and source-selection seeds repeat evaluation from fixed trained checkpoints. Unless otherwise stated, generation results are summarized over five sampling repetitions from one fixed checkpoint. The reported mean and standard deviation therefore reflect sampling variability rather than variation across independent training runs. Clustering results are fixed-final evaluations from one training lineage per method.

For clustering, ACC is computed after Hungarian matching between predicted components and ground-truth labels. Component generation is evaluated using consistency, coverage, raw diversity, and Conditional Feature FID. Diversity and FID are computed in dataset-specific classifier feature spaces; the reported FID is therefore not standard Inception-FID and is not comparable across datasets. Transport is evaluated using endpoint success, responsibility MAE, latent-density percentile, and adjacent LPIPS. Reconstruction is evaluated using MSE, SSIM, and LPIPS.

C.2 Dataset Splits and Evaluation Protocols

Dataset	Task	Split / evaluation set	Main protocol
MNIST	Clustering	Combined official train and test splits	Transductive evaluation on 70,000 images
MNIST	Component generation	Official test split	Five sampling repetitions from one fixed checkpoint
MNIST	Transport / simplex	Fixed source latents	Frozen-checkpoint path evaluation; repeated source selection for pairwise transport
Fashion-MNIST	Clustering	Official training and test splits	Training-set fitting and official-test evaluation
Fashion-MNIST	Component generation	Official test split	Five sampling repetitions from one fixed checkpoint
CelebA	Reconstruction	Complete official test split	Matched reconstruction adaptation and evaluation
CelebA	Component generation	Official test reference distribution	Five sampling repetitions from one fixed checkpoint
CelebA	Transport	Fixed source and target anchors	Qualitative comparison under one frozen checkpoint

Table 5: Dataset splits and evaluation protocols used in the main text. Repeated sampling and source selection operate on fixed trained checkpoints and do not represent independent training runs.

As summarized in Table 5, MNIST clustering is the only transductive evaluation in the main text. Fashion-MNIST models are fitted using the official training data and evaluated on the official test split. CelebA reconstruction uses all 19,962 images in the official test split. Detailed transport source-selection procedures are given in Appendix E.1.

C.3 Checkpoint Lineage

Checkpoint suffixes denote the number of optimizer updates completed for the corresponding model, with "k" meaning one thousand. The 50k and 70k suffixes identify MoG-Diff representation checkpoints, whereas 48k, 30k, and 25k identify task-specific latent diffusion checkpoints.

MoG-Diff denotes a unified framework, while individual tasks may use task-specific latent interfaces, samplers, and fixed checkpoints. The checkpoint relations relevant to the reported results are summarized in Table 6.

For correspondence with experimental directories, legacy MoG-DBAE denotes MoG-Diff with an identity interface, whereas legacy Hybrid or CRDB denotes MoG-Diff with a learned interface.

Dataset / task	Fixed configuration	Scope
MNIST clustering	Complete learned-interface MoG-Diff checkpoint	Same representation line as the main generation result; fixed-final evaluation without label-based checkpoint selection
MNIST generation	The same MoG-Diff checkpoint with a component-conditioned Residual-DPM	Sampling seeds are evaluation repetitions from fixed model and sampler checkpoints
MNIST transport / simplex	Task-specific learned-interface MoG-Diff checkpoint	Separate from the main clustering and generation checkpoint; all transport constructions share this frozen geometry, and the simplex experiment is a single-seed pilot
Fashion-MNIST clustering	MoG-Diff with an identity interface	Task-specific clustering checkpoint, separate from the generation configuration
Fashion-MNIST generation	DINOv2 semantic front, learned interface, and component-conditioned Residual-DPM	Task-specific generation checkpoint; temperature and guidance belong to the sampling protocol
CelebA reconstruction	$K = 5$, 50k MoG-Diff checkpoint followed by matched reconstruction adaptation	Posterior-mean evaluation with the encoder and mixture prior frozen
CelebA generation / transport	$K = 5$, 70k identity-interface checkpoint with task-specific latent samplers or refiners	Separate from the reconstruction-adaptation line; generation and transport use fixed representations and task-specific inference configurations

Table 6: Checkpoint lineage of the MoG-Diff results. Comparisons within each reported table use fixed task-specific checkpoints; checkpoints from unrelated task rows should not be compared as a single training run.

Table 3(a) of the main text compares transport constructions within one frozen MNIST geometry. It does not assume that the same checkpoint also produces the clustering and component-generation results in Tables 1 and 2. The quantitative values in Table 3(b) use the same transport checkpoint and are interpreted as a mechanism-oriented simplex pilot.

For CelebA reconstruction, DBAE and MoG-Diff receive the same 500-step adaptation with learning rate 3×10^{-5}. Their encoders remain frozen, and the MoG-Diff mixture prior is additionally fixed. Both methods use posterior-mean latents and the same Heun-140 evaluation protocol on the complete official test split. The adaptation therefore does not modify the learned component organization.

C.4 Component-Conditional Generation Configuration

Component-conditional generation follows the complete MoG-Diff pipeline in Eq. equation 55. For MNIST and Fashion-MNIST, the quantitative experiments use component-conditioned Residual-DPM sampling with the configurations in Table 7. For each requested component, the sampler starts from Gaussian residual noise, applies DDIM denoising, maps the resulting residual back to the mixture-structured latent space, and generates the final image through the complete image-space bridge.

Guidance 0 on Fashion-MNIST removes the additional analytical endpoint-score contribution but does not skip the image bridge, as described in Appendix B.4. The decoded endpoint remains the initial state and conditioning input of the complete bridge trajectory.

Each sampling repetition generates 100 images for each of ten components. The quantitative results aggregate five repetitions, corresponding to 5,000 images per dataset and method. Sampling seed 0 is used to select the reported temperature and guidance configuration, while sampling seeds 1–5 are reserved for quantitative evaluation.

Dataset	Residual subset	Residual-DPM	τ	Latent solver	Image solver	Guidance	Samples / comp.
MNIST	$\max_k \rho_k(z) \geq 0.9$	48k	1.0	DDIM-100	Heun-40	0.5	100
Fashion-MNIST	High-confidence Core10 residuals	30k	1.4	DDIM-100	Heun-40	0	100

Table 7: Sampling configurations for the quantitative component-conditional generation experiments. Both datasets use $K = 10$. Latent DDIM sampling and image-space Heun integration are separate numerical stages.

Class labels are not used to train the MoG-Diff representation, component-conditioned Residual-DPM, or image bridge. An external classifier is applied only after generation to compute consistency, coverage, diversity, and Conditional Feature FID.

Baseline sampling. VaDE generates samples through its native component-conditional decoder. DBAE + KMeans samples from component-wise empirical diagonal latent distributions before applying the DBAE image bridge. ClusterGAN fixes the discrete cluster code and samples its continuous latent variables, whereas TreeDiffusion samples from the corresponding learned leaf condition. All methods use the same number of generated samples per component or leaf and the same number of sampling repetitions.

D Additional CelebA Generation Results

CelebA attributes overlap and do not define mutually exclusive semantic classes. We therefore interpret the $K = 5$ components through post-hoc attribute profiles rather than assigning one class label to each component. This appendix describes the raw latent-refinement configuration used in Fig. 2(c) and reports additional quantitative comparisons, including component-whitened Residual-DPM sampling. All configurations use the same frozen $K = 5$, 70k MoG-Diff representation and image bridge.

D.1 Sampler Configurations

Raw latent refinement in Fig. 2(c). For a requested component c, the qualitative panel initializes a latent from a scaled component Gaussian and applies short-horizon refinement with a component-conditional diffusion model trained in the raw latent space:

$$\begin{aligned} z^{(0)} &= \mu_c + 0.3\,\sigma_c \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \\ z_{\mathrm{ref}} &= \mathrm{DDIM}_\xi \left(z^{(0)}; c,\, t_{\mathrm{ref}} = 0.2 \to 0 \right), \\ z_{\mathrm{ref}} &\longrightarrow x_T \longrightarrow \widehat{x}_0. \end{aligned} \tag{56}$$

No forward noise is injected before refinement; $z^{(0)}$ is directly treated as the latent state at $t_{\mathrm{ref}} = 0.2$. The raw latent refiner uses the 25k latent-DPM checkpoint, followed by the complete 40-step Heun image bridge. It is a task-specific latent sampler within MoG-Diff rather than a separate proposed model.

Component-whitened Residual-DPM. The Residual-DPM configuration instead generates a component-whitened residual and maps it to the structured latent space:

$$r \sim p_\xi(r \mid c), \qquad z_{\mathrm{mix}} = \mu_c + 0.5\,\sigma_c \odot r. \tag{57}$$

The residual temperature is $\tau = 0.5$. The sampler uses a 25k Residual-DPM checkpoint trained from latents whose maximum component responsibility is at least 0.9. The resulting latent is decoded and passed through the same complete 40-step image bridge.

D.2 Quantitative Evaluation

For each sampler, we generate 100 images from each of the five components in every sampling repetition. Five repetitions produce 2,500 images per sampler. Reported deviations are standard deviations across these repetitions from fixed checkpoints and do not represent independent training runs.

Sampler	Max NMI ↑	Top-10 NMI ↑	Diversity ↑	PCA64 FID ↓
MoG direct	0.3567 ± 0.0041	0.1554 ± 0.0042	13.39 ± 0.11	709.5 ± 7.5
Raw latent-DPM, full sampling	0.1695 ± 0.0212	0.0714 ± 0.0067	$\mathbf{30.17 \pm 0.24}$	$\mathbf{242.7 \pm 15.6}$
Raw latent refinement, no forward noise	$\mathbf{0.4491 \pm 0.0128}$	$\mathbf{0.3412 \pm 0.0044}$	14.90 ± 0.13	608.8 ± 5.3
Component-whitened Residual-DPM	0.2699 ± 0.0128	0.0963 ± 0.0048	24.54 ± 0.28	447.4 ± 19.6

Table 8: CelebA $K = 5$ component-conditional generation results. Max NMI is the strongest association between component identity and one of the 40 predicted CelebA attributes; Top-10 NMI averages the ten strongest attribute associations. Diversity and PCA64 FID are computed in an external attribute-classifier feature space, and PCA64 FID is not standard Inception-FID. Deviations are standard deviations over five sampling repetitions from fixed checkpoints.

Table 8 shows that raw latent refinement yields the strongest component–attribute association, whereas full raw latent-DPM sampling achieves the highest diversity and lowest PCA64 FID. Component-whitened Residual-DPM provides an intermediate operating point: relative to direct MoG sampling, it increases diversity and improves feature-distribution proximity, while retaining stronger attribute association than full raw latent-DPM sampling. These results indicate a trade-off among component-level attribute organization, within-component variation, and feature-distribution proximity rather than a single sampler dominating every metric.

Because CelebA attributes overlap, the five components should not be interpreted as mutually exclusive facial categories. Instead, they capture recurring attribute profiles in the latent representation. The CelebA results therefore provide a natural-image extension of component-conditional generation rather than a class-conditional benchmark.

E Additional Details for Component Transport and Simplex Control

E.1 Pairwise Component Transport Protocol

Evaluation setting and source selection. Pairwise transport is evaluated in the frozen ten-dimensional mixture-structured latent space of the task-specific learned-interface MNIST MoG-Diff checkpoint used in Table 3(a). This checkpoint is separate from the main clustering and component-generation line. No model parameter is updated during path construction or image generation.

Real source latents are selected from the official MNIST training split. For each source component, candidates must be assigned to that component and have maximum responsibility of at least 0.99. For a candidate z_i in component s, we compute

$$r_i = (z_i - \mu_s) \oslash \sigma_s. \tag{58}$$

Candidates whose residual norms lie between the 25th and 75th percentiles are retained, and ten sources are selected using the evaluation seed. We use five source-selection seeds, $\{0, 1, 2, 3, 4\}$, representing repeated source selection from one frozen checkpoint. Ground-truth labels are used only for post-hoc component interpretation and external image evaluation.

Evaluated component pairs. The five directed component pairs and their post-hoc majority-digit interpretations are

Source s	Target t	Digit direction
0	8	$6 \to 5$
1	2	$7 \to 9$
2	9	$9 \to 4$
3	5	$3 \to 8$
7	5	$1 \to 8$

All five directions contribute to Table 3(a). Fig. 3(a) displays only the representative $3 \to 8$ and $6 \to 5$ directions; this qualitative selection does not change the evaluation protocol or reported statistics. Each trajectory contains eleven uniformly spaced points,

$$\alpha_j = \frac{j}{10}, \qquad j = 0, \dots, 10. \tag{59}$$

The evaluation contains 50 trajectories per source-selection seed and 250 trajectories per path construction.

Path constructions. Given source latent z_s in component s, we extract

$$r_s = (z_s - \mu_s) \oslash \sigma_s, \qquad z_t = \mu_t + \sigma_t \odot r_s. \tag{60}$$

The linear baseline interpolates between the residual-matched endpoints:

$$z_{\mathrm{lin}}(\alpha) = (1-\alpha)z_s + \alpha z_t. \tag{61}$$

The covariance-aware path interpolates component means and log-scales while preserving r_s:

$$\mu(\alpha) = (1-\alpha)\mu_s + \alpha\mu_t, \tag{62}$$

$$\log \sigma(\alpha) = (1-\alpha)\log \sigma_s + \alpha \log \sigma_t, \tag{63}$$

$$z_{\mathrm{cov}}(\alpha) = \mu(\alpha) + \sigma(\alpha) \odot r_s. \tag{64}$$

For responsibility control, the covariance-aware points initialize latent optimization toward

$$\rho^*(\alpha) = (1-\alpha)e_s + \alpha e_t. \tag{65}$$

Define the path-relative residual

$$\widehat{r}_j = (z_j - \mu(\alpha_j)) \oslash \sigma(\alpha_j). \tag{66}$$

The optimization objective is

$$\mathcal{L} = \frac{1}{11}\sum_{j=0}^{10} \Bigg[10\|\rho(z_j) - \rho^*(\alpha_j)\|_2^2 + \frac{0.1}{d}\|\widehat{r}_j - r_s\|_2^2 + \frac{0.01}{d}\big[-\log p_\theta(z_j)\big] \Bigg], \tag{67}$$

where $d = 10$. Adam is run for 200 steps with learning rate 0.05. All MoG and generative parameters remain frozen.

Image generation and metrics. Each trajectory point is mapped through the learned latent interface and endpoint generator, followed by the standard 40-step Heun bridge described under *Numerical Solvers and Stable Computation*, with guidance 1.0. Decoder-direct images are retained for diagnosis, while all reported transport metrics use final bridge outputs.

Endpoint success is the fraction of endpoint images classified as the post-hoc majority digit of the target component. Responsibility MAE is

$$\frac{1}{11}\sum_{j=0}^{10} \left|\rho_t(z_j) - \alpha_j\right|. \tag{68}$$

Density percentile is computed relative to the training-latent MoG log-density distribution, taking the minimum value along each trajectory. Adjacent LPIPS averages the perceptual distance between consecutive bridge images. Metrics are first averaged within each source-selection seed and then over the five seed-level means.

E.2 Analytic Cycle Consistency

Construction and protocol. For each pairwise trajectory, we take its endpoint at $\alpha = 1$. Given source component s, target component t, and transported endpoint z_t, the source-side latent is recovered by the inverse affine map

$$\widetilde{z}_s = \mu_s + \sigma_s \odot ((z_t - \mu_t) \oslash \sigma_t) . \tag{69}$$

For linear and covariance-aware transport, $z_t = \mu_t + \sigma_t \odot r_s$, so $\widetilde{z}_s \approx z_s$ up to numerical precision. Responsibility-controlled endpoints have been modified by latent optimization; their cycles therefore measure the resulting change in the within-component coordinate rather than exact analytic invertibility.

The cycle experiment uses the same checkpoint, component pairs, source latents, and five source-selection seeds as the pairwise experiment, resulting in 250 cycles per path construction. Each cycle retains the source, transported target, and inverse-transported images. Bridge outputs are not re-encoded before comparison, so the experiment evaluates latent transport together with the fixed endpoint generator and bridge rather than an encoder–decoder reconstruction loop.

Metrics and results. Latent cycle error is

$$\|\widetilde{z}_s - z_s\|_2, \tag{70}$$

and residual change is

$$\|(z_t - \mu_t) \oslash \sigma_t - (z_s - \mu_s) \oslash \sigma_s\|_2 . \tag{71}$$

Direct and bridge recovery measure whether the corresponding cycle image is classified as the post-hoc majority digit of the source component. Direct and bridge LPIPS compare each cycle image with its source realization; these are semantic and perceptual recovery measures rather than exact identity metrics.

Path	Latent L2	Residual Δ L2
Linear	1.25×10^{-7}	2.80×10^{-7}
Covariance-aware	1.11×10^{-7}	2.56×10^{-7}
Responsibility-controlled	0.0696	0.1876

Path	Direct recovery	Bridge recovery	Direct LPIPS	Bridge LPIPS
Linear	1.0000	1.0000	7.25×10^{-7}	3.81×10^{-6}
Covariance-aware	1.0000	1.0000	7.24×10^{-7}	3.81×10^{-6}
Responsibility-controlled	1.0000	1.0000	5.72×10^{-4}	8.46×10^{-4}

Table 9: Analytic cycle consistency on MNIST. Linear and covariance-aware transport are numerically invertible in latent space. Responsibility-controlled transport modifies the residual while preserving source-majority semantic recovery.

Linear and covariance-aware transport recover the original latent to approximately 10^{-7} numerical error, with bridge-cycle LPIPS close to zero. Responsibility control produces a larger cycle error because it modifies the residual toward a prescribed responsibility trajectory, while all evaluated cycle images retain the source component's majority digit.

E.3 CelebA Projected Transport Settings

Configuration and real-endpoint LERP. Fig. 3(b) uses the frozen $K = 5$, 70k identity-interface CelebA MoG-Diff checkpoint with a 512-dimensional latent space. Source latents are deterministic posterior means of real training images. The projection module is a 25k component-conditional Residual-DPM trained on component-whitened residuals from the same representation.

The LERP row connects independently encoded real-image latents:

$$z_{\mathrm{LERP}}(\gamma) = (1-\gamma)z_s + \gamma z_t^{\mathrm{real}}, \qquad \gamma \in [0,1]. \tag{72}$$

The displayed anchors are fixed high-responsibility real latents selected for qualitative visualization. Projected transport uses the same source latent as the corresponding LERP row but does not use its real target latent.

Two-stage projected transport. From

$$r_s = (z_s - \mu_s) \oslash \sigma_s, \tag{73}$$

we first construct an eleven-point covariance-aware path:

$$\mu(\alpha) = (1-\alpha)\mu_s + \alpha\mu_t, \tag{74}$$
$$\log\sigma(\alpha) = (1-\alpha)\log\sigma_s + \alpha\log\sigma_t, \tag{75}$$
$$z_{\mathrm{cov}}(\alpha) = \mu(\alpha) + \sigma(\alpha)\odot r_s, \qquad \alpha = 0, 0.1, \ldots, 1. \tag{76}$$

Its analytic arrival point is

$$z_{\mathrm{arr}} = \mu_t + \sigma_t \odot r_s, \tag{77}$$

which uses the target component statistics while preserving the source residual.

For a selected normalized projection time t_{proj}, forward noise is added to the source residual. A frozen target-conditioned Residual-DPM then applies 100 DDIM denoising steps:

$$r_{t_{\mathrm{proj}}} = \sqrt{\overline{\alpha}_{t_{\mathrm{proj}}}}\, r_s + \sqrt{1-\overline{\alpha}_{t_{\mathrm{proj}}}}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{78}$$
$$r_p = \mathrm{DDIM}_\xi(r_{t_{\mathrm{proj}}}; c = t, t_{\mathrm{proj}} \to 0). \tag{79}$$

At fixed target-component statistics, the projection tail is

$$z_{\mathrm{tail}}(\beta) = \mu_t + \sigma_t \odot \mathrm{SLERP}(r_s, r_p; \beta), \qquad \beta \in \{0, 0.2, \ldots, 1\}. \tag{80}$$

The complete path contains ten analytic points ($\alpha = 0, \ldots, 0.9$) followed by six projection-tail points. The analytic arrival at $\alpha = 1$ equals the tail point at $\beta = 0$ and is not duplicated. Thus, projected transport denotes analytic fixed-residual component transport followed by a target-conditioned residual-projection tail; the Residual-DPM is not applied independently at every analytic point.

Projection strength and displayed frames. Projection times

$$t_{\mathrm{proj}} \in \{0.01, 0.025, 0.05, 0.1\}$$

are evaluated for all 20 directed component pairs. For each pair, we select the largest value satisfying

$$\rho_t(z_{\mathrm{end}}) \geq 0.95 \tag{81}$$

and a maximum adjacent LPIPS no greater than that of the corresponding real-endpoint LERP. Both trajectories in Fig. 3(b) use $t_{\mathrm{proj}} = 0.1$. Selection uses mixture responsibility and path smoothness rather than CelebA attribute labels.

Every latent is decoded and passed through the standard 40-step Heun bridge at 64×64 resolution. Fig. 3(b) displays six frames from each path:

Progress	LERP	Projected transport
0.0	$\gamma = 0.0$	analytic, $\alpha = 0.0$
0.2	$\gamma = 0.2$	analytic, $\alpha = 0.3$
0.4	$\gamma = 0.4$	analytic, $\alpha = 0.6$
0.6	$\gamma = 0.6$	analytic, $\alpha = 0.9$
0.8	$\gamma = 0.8$	tail, $\beta = 0.4$
1.0	$\gamma = 1.0$	tail, $\beta = 1.0$

The horizontal labels show normalized display progress; the paths use different interpolation parameters. The projected position at 0.8 has already reached the target component and lies on the projection tail. The final LERP image is an encoded real target, whereas the final projected image is a source-associated target-component realization generated without that target image or latent.

E.4 THREE-COMPONENT SIMPLEX LATTICE

Scope and lattice. Table 3(b) and Fig. 3(c–e) use the same frozen task-specific MNIST checkpoint as the pairwise transport experiment. The simplex study uses one evaluation seed, three component triples, and two source residuals per triple, and is therefore interpreted as a mechanism-oriented pilot.

The evaluated triples are

Component triple	Majority-digit triple
$(3, 5, 2)$	$(3, 8, 9)$
$(7, 1, 2)$	$(1, 7, 9)$
$(4, 0, 8)$	$(0, 6, 5)$

Labels are used only for post-hoc component interpretation and corner evaluation.

For each triple $\mathcal{C} = (c_1, c_2, c_3)$, we use a resolution-five barycentric lattice:

$$w = \left(\frac{i}{5}, \frac{j}{5}, \frac{k}{5}\right), \qquad i + j + k = 5, \qquad i, j, k \geq 0. \tag{82}$$

The lattice contains 21 points. Vertices correspond to pure component weights, edges to two-component weights, and interior points to nonzero weights on all three components.

Source residuals and analytic construction. For each triple, two real source latents are selected from the first component using the same responsibility and residual-norm criteria as in the pairwise protocol. Selection uses evaluation seed 123 with a fixed triple-specific offset. These are six source selections from one frozen checkpoint, not independent training runs.

For source latent z_s in component c_1,

$$r_s = (z_s - \mu_{c_1}) \oslash \sigma_{c_1}. \tag{83}$$

For every lattice weight $w = (w_1, w_2, w_3)$, covariance-barycentric transport uses

$$\overline{\mu}(w) = \sum_{m=1}^{3} w_m \mu_{c_m}, \tag{84}$$

$$\log \overline{\sigma}(w) = \sum_{m=1}^{3} w_m \log \sigma_{c_m}, \tag{85}$$

$$z_{\mathrm{ana}}(w) = \overline{\mu}(w) + \overline{\sigma}(w) \odot r_s. \tag{86}$$

The component statistics vary over the simplex while the same source-relative residual is retained. The three corners are component-specific realizations of this shared residual rather than independently encoded real images.

Responsibility control. The analytic simplex initializes optimization toward

$$\rho_k^*(w) = \begin{cases} w_m, & k = c_m, \\ 0, & k \notin \mathcal{C}. \end{cases} \tag{87}$$

Define

$$\widehat{r}_w = (z_w - \overline{\mu}(w)) \oslash \overline{\sigma}(w). \tag{88}$$

All 21 points are jointly optimized using

$$\mathcal{L}_{\mathrm{simplex}} = \frac{1}{21} \sum_{w} \Bigg[10 \|\rho(z_w) - \rho^*(w)\|_2^2 + \frac{0.1}{d} \|\widehat{r}_w - r_s\|_2^2 + \frac{0.01}{d} \big[-\log p_\theta(z_w) \big] \Bigg], \tag{89}$$

where $d = 10$. Adam is run for 200 steps with learning rate 0.05, with all model and MoG parameters frozen.

Image generation, metrics, and interpretation. Each latent is mapped through the frozen learned interface, endpoint generator, and 40-step Heun bridge. Decoder-direct images are retained for diagnosis, while semantic and LPIPS metrics use final bridge outputs.

Responsibility MAE is

$$\frac{1}{21K}\sum_{w}\|\rho(z_w)-\rho^*(w)\|_1, \qquad K=10. \tag{90}$$

Density percentile is averaged over the lattice and source trajectories. Corner success measures whether each vertex produces the post-hoc majority digit of its component. Edge LPIPS averages perceptual distances over the 45 neighboring lattice edges and then over the six source trajectories.

Fig. 3(c–e) displays the covariance-barycentric construction for the first selected source residual of each triple, denoted by source slot $\mathrm{s}00$. Responsibility-controlled results and the second source residual are included in the evaluation records but are not displayed in the main figure.

A simplex interior point specifies a target mixture-responsibility configuration. It should not be interpreted as literal pixel-space superposition of three handwritten digits. The experiment tests whether the learned MoG geometry supports structured movement among multiple semantic components while retaining a shared within-component residual coordinate.