ENCODER–DECODER MANIFOLD ALIGNMENT FOR IDEMPOTENT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, several learning paradigms have been introduced to enforce idempotency in generative models. The goal of this property is to ensure that repeated application of a model leaves samples unchanged once they lie on the target data manifold. In this work, we explore enforcing idempotency in encoder–decoder models. We argue that a key reason for the failure of generative models on some tasks, such as editing, is a geometric mismatch between the manifolds learned by the encoder and decoder. The encoder projects inputs onto one latent manifold, while the decoder implicitly learns to reconstruct data from a different manifold. This discrepancy prevents the model from learning truly idempotent mappings. To address this issue, we propose a new training framework that explicitly closes this gap by forcing the encoder and decoder to learn consistent representations of the same underlying data manifold. By aligning the geometry of these components, our method encourages stable projections. Empirically, we show that our approach achieves significantly lower idempotency error and consistently regenerates identical outputs under repeated application, compared to existing methods. We demonstrate the effectiveness of the proposed framework on both image generation and image editing tasks. Finally, we show that enforcing idempotency in this manner improves identity preservation and information stability, leading to more realistic and controllable generative editing models. We additionally release an open source reference implementation of the objective.[1]

1 INTRODUCTION

Generative modeling has become a foundational tool across a wide range of data modalities, from images and speech to video (Rombach et al., 2022; Shen et al., 2023; Ho et al., 2022). Modern high-performing generative models, including latent diffusion models (Rombach et al., 2022; Wu et al., 2025b), flow matching (Lipman et al., 2022; Hu et al., 2024), and rectified flows (Liu et al., 2022; Lupaşcu & Stupariu, 2026; Kim et al., 2025), share a common architectural principle: rather than operating directly in high-dimensional data spaces such as pixel grids or audio waveforms, they learn a compact latent representation and perform generation within it. By compressing data through a learned encoder and reconstructing it through a learned decoder, these models delegate the complexity of the data distribution to a well-behaved latent space. The encoder and decoder therefore bound what the rest of the pipeline can achieve, and their quality directly determines the fidelity and controllability of the outputs (Rombach et al., 2022; Yao et al., 2025).

This dependence is most visible in editing, where the goal is to take an input such as an image, speech recording, or video and modify specific attributes while leaving everything else intact (Abdal et al., 2021; Anastassiou et al., 2024; Avrahami et al., 2025; Yang et al., 2025). Editing is typically performed in latent space by encoding the input, modifying its latent representation, and decoding it back to data space. This process requires the encoder and decoder to share a consistent data manifold.

When they are not aligned, latent edits lose localization, and target modifications corrupt unedited features such as identity or fine details (Tov et al., 2021; Roich et al., 2022; Rout et al., 2024; Pan & Bareinboim, 2025; Wang et al., 2026). This raises a fundamental question: what property must a model satisfy to ensure stable, details preserving editing?

[1] https://anonymous.4open.science/r/idem_align-5FBE

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.

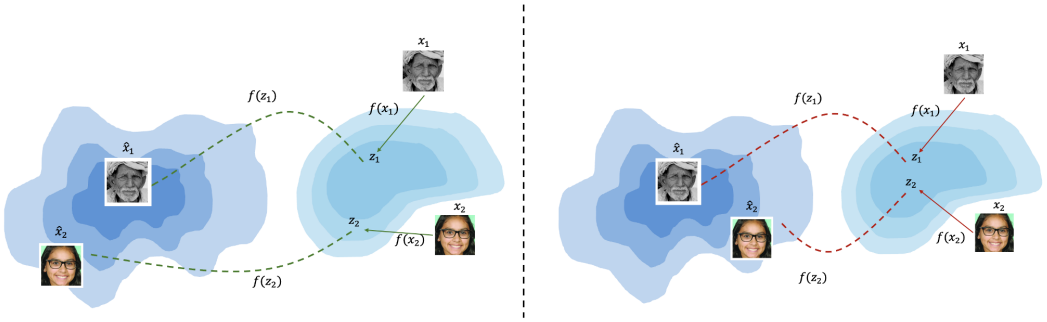


Figure 1: Comparison between aligned and unaligned encoder–decoder manifolds. **Left:** Aligned manifold, where the encoder and decoder learn consistent representations of the same data manifold, leading to stable fixed points. **Right:** Unaligned manifold, where geometric mismatch between the encoder and decoder causes drift under repeated generation.

A common answer is disentangled representation learning (Bengio et al., 2013), where the encoder separates factors of variation into distinct latent components and the decoder recombines them to generate data. Editing can then be performed by manipulating specific latent variables in supervised settings (Chen et al., 2016) or by interpolating the latent space in unsupervised ones (Lin et al., 2024).

However, enforcing disentanglement can come at a cost. These objectives often encourage models to prioritize a restricted set of explanatory factors, potentially discarding subtle or hard-to-model variations. As a result, the learned representation may fail to preserve all information in the original input, leading to degraded identity and fine-detail preservation after editing (Burgess et al., 2018; Shukor et al., 2022; Dalva et al., 2023).

We argue that a more basic requirement is *idempotency*: for an encoder–decoder model $f = D_\theta \circ E_\phi$, applying the model twice should give the same result as applying it once, i.e., $f(f(x)) = f(x)$. This property exposes a failure that can occur even after a single pass. Specifically, if $E(D(E(x))) \neq E(x)$, the decoder changes the representation produced by the encoder, causing the reconstructed sample to shift in latent space when it is re-encoded. This shift can alter attributes that were not meant to be edited, leading to identity loss (Zhou et al., 2025; Wang et al., 2026). Projection operators are idempotent by definition (Taylor & Lay, 1980; Deutsch & Deutsch, 2001; Federer, 2014), so idempotency is a necessary condition for f to act as a projection onto the learned data manifold. The reconstruction objective does not enforce it, since it only asks that $D(E(x)) \approx x$ on training samples.

As illustrated in Figure 1, this degradation comes from a geometric mismatch between the encoder and decoder manifolds. When misaligned, re-encoding $D(E(x))$ projects the reconstructed data onto a shifted latent representation, causing progressive drift across iterations. Conversely, aligning these manifolds ensures that the composition $D_\theta \circ E_\phi$ forms a stable projection whose fixed points lie directly on the learned manifold. Because projection operators are idempotent by definition (Taylor & Lay, 1980; Deutsch & Deutsch, 2001; Federer, 2014), we argue that idempotency is a necessary condition for an optimal information preserving autoencoder.

To verify the misalignment hypothesis, we evaluate the Jacobians of the encoder and decoder at points on the manifold (Figure 3). At a code $z = E(x)$ we take $J_D = \partial D/\partial z$, and at the decoded point $D(z)$ we take $J_E = \partial E/\partial x$, so their product $J_E J_D$ is the Jacobian of the latent round trip $E \circ D$. If the encoder inverts the decoder, this round trip is the identity, and so is its Jacobian, meaning every perturbation of a latent code returns unchanged. We therefore report the singular values of $J_E J_D$, which give the gain in the best- and worst-preserved directions and are invariant to a rotation of the latent coordinates. The columns of J_D span the decoder's tangent plane, while the singular vectors of J_E separate the directions the encoder is sensitive to from those it discards (Rifai et al., 2011; Bengio et al., 2013); $\sigma_{\min} \ll 1$ therefore indicates that the decoder moves the surface along a direction the encoder treats as off-manifold. For the baseline, $\sigma_{\max}$ stays near one while $\sigma_{\min}$

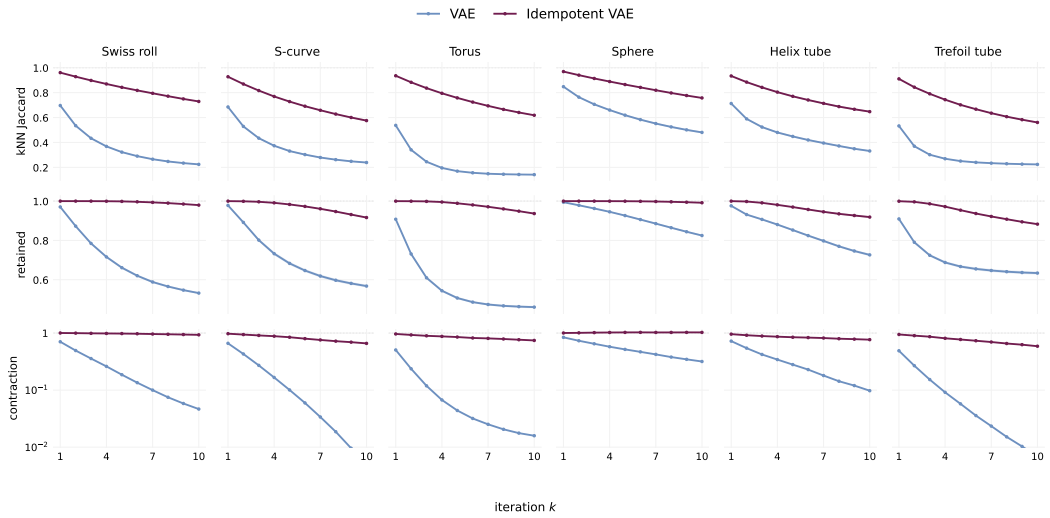


Figure 2: A comparison of latent-space neighbourhood structure after k applications of $f = D \circ E$, measured against the latents of the original data. **(top)** Jaccard overlap, **(middle)** Retained neighbours and **(bottom)** Contraction on a log scale.

falls by up to two orders of magnitude, whereas the idempotent model remains near the identity in both directions. In Figure 2, we show how nearby points and their distances to multiple anchor points collapse after 10 iterations of reconstruction, and Figure 4 shows how the points collapse for the helix tube. All these figures show empirically that the encoder is not able to encode the decoded point back to the same latent point, even when the reconstruction looks correct, and that this is because the encoder and decoder learn different geometries of the manifold, which reconstruction error alone does not reveal. To close this gap, we propose a latent consistency objective that trains the encoder and decoder to agree. A decoded sample must re-encode to its own latent code, and the same must hold for codes sampled around the encoder's outputs, so the constraint covers a neighbourhood around the data and extends beyond the points the encoder produces. In this paper, we show that idempotency is necessary for an encoder–decoder model to act as a projection onto the learned data manifold, introduce a simple method to align encoder and decoder representations, and demonstrate that this alignment improves identity preservation and editing success across different generative architectures.

2 Related Work

2.1 Learning a Representation for Editing

Editing in generative models aims to modify specific attributes of a sample while preserving all other underlying factors (Abdal et al., 2021; Kawar et al., 2023; Radford, 2015; Sheng et al., 2025). A common approach is to learn a representation in which different factors of variation are disentangled, allowing controlled manipulation of selected attributes while keeping the remaining content unchanged. Many editing methods build on this idea by enforcing factorized latent spaces (Lin et al., 2024; Wu et al., 2021; Karras et al., 2019) or attribute-specific directions (Abdal et al., 2021; Anastassiou et al., 2024), enabling edits through latent traversal or conditional manipulation. In practice, however, perfect disentanglement is difficult to achieve, and overly constrained representations may limit reconstruction quality or expressiveness(Burgess et al., 2018; Shukor et al., 2022; Dalva et al., 2023). Recent editing methods therefore, focus less on strict disentanglement and more on preserving identity and content consistency under transformation (Chen et al., 2023; Esser et al., 2024; Rout et al., 2024). A key challenge in these approaches is maintaining stability, because repeated edits or re-encoding of edited samples often lead to drift, identity degradation, or accumulation of artifacts. This instability suggests that representation quality alone is insufficient and that the interaction between the encoder and decoder plays a critical role in reliable editing.

2.2 IDEMPOTENT MODELS

An operator is idempotent if applying it more than once produces no further change; for a generative model, a sample already on the data manifold should be left unchanged. IGN (Shocher et al., 2023) uses this property to learn a global projector: a single network $f : \mathcal{X} \to \mathcal{X}$ that maps any input onto the learned data manifold in one step. The model is trained with two terms added to the reconstruction loss,

$$\mathcal{L}_{\text{idem}} = \left\| f_{\theta'}(f_\theta(z)) - f_\theta(z) \right\|_1, \qquad \mathcal{L}_{\text{tight}} = -\left\| f_\theta(f_{\theta'}(z)) - f_{\theta'}(z) \right\|_1, \tag{1}$$

where $f_{\theta'}$ is a frozen copy of f_θ. The idempotency term makes generated samples fixed points, and the tightening term prevents arbitrary points from becoming fixed points. Although IGN is trained only on noise and real data, the learned projector performs zero-shot tasks such as denoising corrupted images, colorizing grayscale images, and mapping sketches to realistic images, all by projecting them onto the data manifold. Zaman et al. (2025) replace the tightening term with objectives derived from the probability flow ODE of a diffusion model, treating noisy points along the trajectory as structured off-manifold inputs. Jensen & Vicary (2025) enforce idempotency through a modified backpropagation update, motivated by perturbation theory (Kato, 2013), that takes small steps toward an idempotent operator using the corrector $g(K) = 3K^2 - 2K^3$. Durasov et al. (2024) use idempotency as a domain-agnostic objective for test-time training. All of these methods enforce idempotency in data space, as a property of a single map $f : \mathcal{X} \to \mathcal{X}$. We enforce it in latent space, as a property of the encoder and decoder pair.

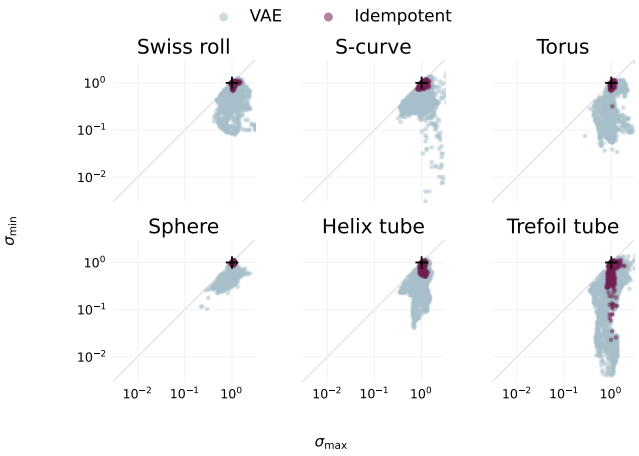


Figure 3: A comparison of the Jacobian $J_E J_D$ between the baseline VAE and the idempotent model, evaluated at 200 points on each surface. Each dot represents one point, plotted using its largest and smallest singular values. The cross marks $(1, 1)$, where perturbations are preserved in both directions. Dots below the diagonal indicate that one direction is contracted more than the other. The baseline shows a clear difference between $\sigma_{\max}$ and $\sigma_{\min}$, with $\sigma_{\max}$ remaining close to one while $\sigma_{\min}$ decreases. This indicates that some directions are preserved while others are contracted. In contrast, the idempotent model concentrates near $(1, 1)$, indicating that perturbations are better preserved in both directions.

2.3 CYCLE CONSISTENCY IN LATENT SPACE

Cycle consistency requires that mapping a sample forward and back returns it to its starting point (Zhu et al., 2017a). An autoencoder has two such cycles: the data cycle $D(E(x)) \approx x$, which reconstruction enforces, and the latent cycle $E(D(z)) \approx z$, which it does not. The latent cycle is sufficient for idempotency: if $E(D(z)) = z$ for every z in the image of E, then $(D \circ E) \circ (D \circ E) = D \circ E$. Several methods add the latent cycle explicitly. BicycleGAN (Zhu et al., 2017b), MUNIT (Huang et al., 2018), and DRIT (Lee et al., 2018) use it to keep generators from ignoring their latent or style codes, and InfoGAN, BiGAN, and ALI (Chen et al., 2016; Donahue et al., 2016; Dumoulin et al., 2016) learn to recover latent codes from generated samples. Chen & Konukoglu (2018) apply it to reconstructions for unsupervised lesion detection, and H3AE (Wu et al., 2025a) uses it to improve reconstruction in video autoencoders. These methods apply the latent cycle to prior samples or to training codes, for output diversity, anomaly detection, or reconstruction quality. We use it to align the encoder and decoder manifolds, apply it to training codes and to a neighbourhood around them.



Figure 4: Idempotency test on the helix tube: the same samples are reconstructed once and ten times, each round applying $f = D \circ E$. Both models look correct after one round, but by the 10th round the VAE has contracted the surface onto a thin subset, while the idempotent VAE keeps both the shape and the spacing between points. More surfaces are shown in Figure 15 in Appendix C.

3 Latent Idempotency

We consider a general encoder–decoder generative model composed of an encoder $E_\phi : \mathcal{X} \to \mathcal{Z}$ and a decoder $D_\theta : \mathcal{Z} \to \mathcal{X}$. Given a dataset $\{x_i\}_{i=1}^n$ of examples drawn independently from a data distribution $p_\mathcal{X}$ over $\mathcal{X}$, the encoder maps inputs to a latent space $\mathcal{Z}$, and the decoder maps latent codes back to $\mathcal{X}$ (Figure 9, Appendix A). The overall mapping is

$$f(x) = D_\theta(E_\phi(x)), \tag{2}$$

and the standard reconstruction objective is

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x\sim p_\mathcal{X}}\big[\, \|D_\theta(E_\phi(x)) - x\| \,\big]. \tag{3}$$

Optimizing this objective alone does not guarantee that the encoder and decoder learn compatible representations. The encoder may learn a latent manifold that differs from the one the decoder assumes, resulting in a geometric mismatch between the two components (Figure 1, right). Under this mismatch, small changes in latent space do not correspond to controlled or semantically smooth changes in the output.

An ideal autoencoder acts as a projection onto a learned data manifold $\mathcal{M}$. Projection operators are idempotent by definition (Taylor & Lay, 1980; Deutsch & Deutsch, 2001; Federer, 2014), so applying f twice must give the same output as applying it once, $f \circ f = f$:

$$D_\theta\big(E_\phi\big(D_\theta(E_\phi(x))\big)\big) = D_\theta(E_\phi(x)). \tag{4}$$

If Eq. equation 4 does not hold, f is not a projection, since the output changes after a second pass through the model. A sufficient condition for Eq. equation 4 is that the latent code is unchanged by a round trip through the decoder and encoder:

$$E_\phi\big(D_\theta(E_\phi(x))\big) = E_\phi(x). \tag{5}$$

When Eq. equation 5 holds only approximately, the output drift is bounded by the latent drift. If D_θ is L-Lipschitz, then

$$\big\|D_\theta\big(E_\phi(D_\theta(E_\phi(x)))\big) - D_\theta(E_\phi(x))\big\| \;\le\; L\,\big\|E_\phi\big(D_\theta(E_\phi(x))\big) - E_\phi(x)\big\|. \tag{6}$$

We therefore enforce Eq. equation 5 directly in latent space.

Latent consistency loss. Let $\mathrm{sg}(\cdot)$ denote the stop-gradient operator. For an input x, let $z = E_\phi(x)$ and $\hat{z} = E_\phi\big(D_\theta(\mathrm{sg}(z))\big)$. We define

$$\mathcal{L}_{\text{data}} = \mathbb{E}_{x\sim p_\mathcal{X}}\big[\, \|\hat{z} - \mathrm{sg}(z)\|_2^2 \,\big]. \tag{7}$$

Let $r = \hat{z} - z$, and let J_D denote the decoder Jacobian at z and J_E the encoder Jacobian at $D_\theta(z)$. With the stop gradient, the gradients are

$$\nabla_\theta \mathcal{L}_{\text{data}} = 2\Big(\tfrac{\partial D_\theta}{\partial \theta}\Big)^\top J_E^\top r, \qquad \nabla_\phi \mathcal{L}_{\text{data}} = 2\Big(\tfrac{\partial E_\phi}{\partial \phi}\Big|_{D_\theta(z)}\Big)^\top r, \tag{8}$$

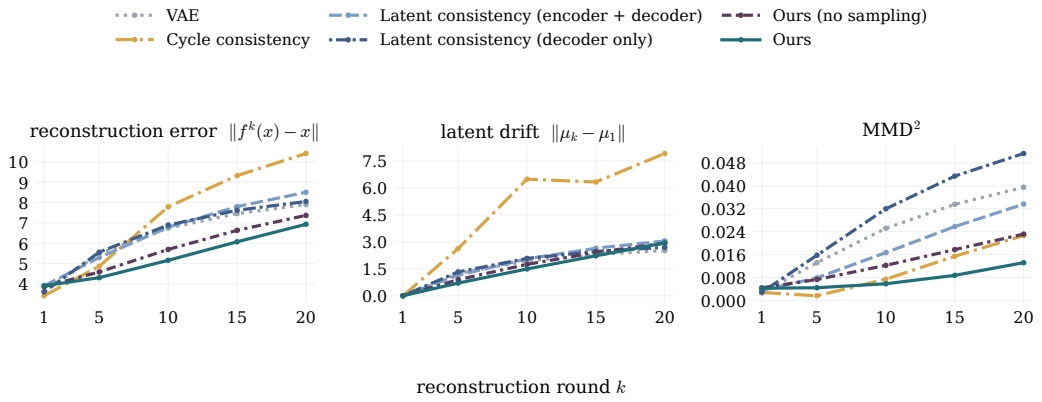


Figure 5: A comparison of training objectives on Colored MNIST under repeated reconstruction. Each curve applies $f = D \circ E$ a total of k times to held-out images and measures, against the source, the reconstruction error, the latent drift $\|\mu_k - \mu_1\|$ between posterior means, and the MMD^2 between the resulting images and the data. Curves are means over three seeds. Our objective degrades most slowly in reconstruction error and stays closest to the data distribution, while cycle consistency preserves the distribution early but drifts furthest in the latent space.

so both networks are updated only through the second pass, where the mismatch occurs. Without the stop gradient, z also depends on ϕ and the encoder gradient gains the term $2\,(\partial E_\phi/\partial\phi|_x)^\top (J_E J_D - I)^\top r$, which moves the codes themselves. This opens a degenerate descent direction: for any $\alpha > 0$, the reparameterization $E_\phi \mapsto \alpha E_\phi$, $D_\theta(\cdot) \mapsto D_\theta(\cdot/\alpha)$ leaves both the reconstruction and the round-trip Jacobian $J_E J_D$ unchanged, while scaling $\mathcal{L}_{\text{data}}$ by α^2. Shrinking the latent codes therefore lowers the loss without improving alignment. Freezing the target removes this direction, so the loss falls only when the round trip $E_\phi \circ D_\theta$ returns z.

Sampling. Eq. equation 7 constrains only the codes the encoder produces from training data. To extend the constraint beyond them, we also apply it to codes sampled around the encoded codes $\mathcal{Z}_\phi = \{E_\phi(x) : x \sim p_\mathcal{X}\}$. Let ν denote the distribution these samples are drawn from. The idempotency loss is

$$\mathcal{L}_{\text{idem}} = \mathbb{E}_{x\sim p_\mathcal{X}}\big[\,\|E_\phi(D_\theta(z)) - z\|_2^2\,\big] + \mathbb{E}_{\tilde{z}\sim\nu}\big[\,\|E_\phi(D_\theta(\tilde{z})) - \tilde{z}\|_2^2\,\big], \qquad z = \text{sg}(E_\phi(x)), \tag{9}$$

where the codes drawn from ν are treated as constants. We take ν to be a Gaussian matched to the per-dimension mean c and standard deviation s of the encoded codes in the batch, with the scale controlled by a hyperparameter κ,

$$\tilde{z} = c + \kappa\, s \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{10}$$

which requires no extra networks and adapts to the latent scale as it changes during training.

Objective. The regularizer is model-agnostic and can be added to any encoder–decoder objective $\mathcal{L}_{\text{base}}$:

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda\,\mathcal{L}_{\text{idem}}, \qquad \lambda > 0. \tag{11}$$

Appendix B proves that idempotency is necessary for $D_\theta \circ E_\phi$ to act as a projection onto the learned manifold and that latent consistency is sufficient to enforce it, and Appendix A reports the implementation details and design choices.

4 Experimental Setup

Synthetic surfaces. To test the manifold misalignment hypothesis in a controlled setting, we conduct experiments on six synthetic surfaces in $\mathbb{R}^3$. We train a baseline VAE and an idempotent VAE with a two-dimensional latent space. We then evaluate whether repeated reconstruction preserves

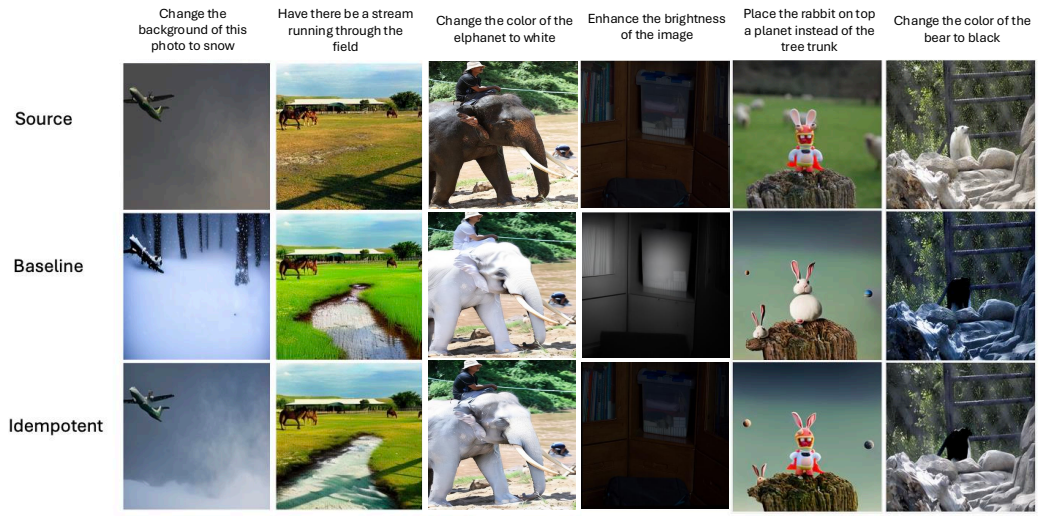


Figure 6: Instruction-guided editing using a latent diffusion model (Brooks et al., 2023) trained with a standard VAE (Baseline) and our idempotent VAE (Idempotent). Our method better follows editing instructions while preserving source image structure and details.

the local structure of the latent representation by measuring neighborhood preservation and changes in pairwise distances (Figure 2). Specifically, we use the kNN Jaccard index, retained neighbors, and contraction factor, defined in Appendix A.2. Additional implementation details are provided in Appendix A.

Objective comparison. We compare six consistency objectives on Colored MNIST, summarized in Table 1. All variants use the same convolutional VAE architecture and training setup and differ only in the consistency objective and its weight. We evaluate reconstruction error, latent drift, and distributional discrepancy with repeated reconstruction iterations on held-out images. Results are shown in Figure 5.

Table 1: Consistency objectives compared in our experiments.

Objective	$\mathcal{L}_{\text{cons}}$
VAE	0
Cycle	$\|f^2(x) - x\|_2^2$
Latent (both)	$\|E(D(z)) - z\|_2^2$
Latent (decoder)	$\mathrm{KL}(q\|q')$
Ours (no sampling)	$\|E(D(\mathrm{sg}[z])) - \mathrm{sg}[z]\|_2^2$
Ours	$\mathbb{E}_{z' \sim p_{\text{aug}}} \|E(D(\mathrm{sg}[z'])) - \mathrm{sg}[z']\|_2^2$

Image editing. We evaluate the effect of idempotent representations on image editing by training a latent diffusion model following the InstructPix2Pix framework (Brooks et al., 2023), replacing its standard VAE with our idempotent VAE. We evaluate both editing and reconstruction quality on I2EBench (Ma et al., 2024), MagicBrush (Zhang et al., 2023), and BM-Bench (Chang et al., 2025). Quantitative results are reported in Table 2, with qualitative results shown in Figure 6.

Idempotent generation. We further evaluate reconstruction quality on natural images using CelebA and MNIST. We train a VQ-VAE-2 (Razavi et al., 2019) on CelebA (Liu et al., 2018) and evaluate its stability on out-of-distribution samples from LFW (Zheng et al., 2017) (Figure 8). Following Shocher et al. (2023), we also train a DCGAN (Radford, 2015) on CelebA and compare our alignment framework with the IGN baseline with different input perturbations including Gaussian noise, grayscale, and sketching (Figure 11). Additional results, qualitative visualizations, and ablation studies are provided in Appendix C.

Table 2: Main results on image editing and reconstruction fidelity. Baseline and Idempotent InstructPix2Pix are evaluated across three benchmarks: I2EBench (500), MagicBrush (528), and BM-Bench (613). Bold values indicate better performance within each pair. MSE values are reported in $\times 10^{-3}$.

		Editing (Source → Edited)				*VAE Reconstruction (Source vs. VAE(Source))*				
Benchmark	Method	CLIP-I ↑	DINO ↑	SSIM ↑	LPIPS ↓	CLIP-I ↑	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓
I2EBench	Baseline	0.856	0.760	0.620	0.357	**0.951**	27.63	0.743	3.32	0.142
	Idempotent	**0.866**	**0.789**	**0.629**	**0.322**	0.949	**27.68**	**0.744**	**3.28**	**0.140**
MagicBrush	Baseline	0.870	0.780	0.702	0.299	**0.957**	26.43	0.805	2.88	0.119
	Idempotent	**0.872**	**0.782**	**0.705**	**0.297**	0.954	**26.51**	**0.807**	**2.84**	**0.117**
BM-Bench	Baseline	0.875	0.816	0.739	0.271	**0.940**	27.80	0.809	2.43	0.119
	Idempotent	**0.903**	**0.864**	**0.772**	**0.228**	0.938	**27.87**	**0.811**	**2.39**	**0.118**

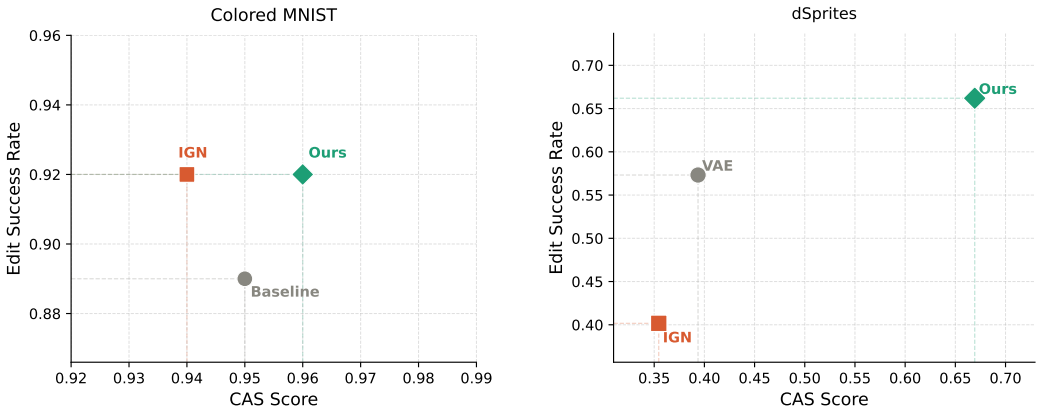


Figure 7: Attribute editing evaluation on Colored MNIST and dSprites. Each point represents a VAE-based variant (VAE, VAE + IGN, and VAE + Ours) scored by Classification Accuracy Score (CAS; Ravuri & Vinyals, 2019) and Edit Success Rate (ESR); higher is better for both.

5 Results

Idempotency preserves latent structure. We first evaluate whether multiple reconstruction iterations of an autoencoder preserve the geometric structure of its latent representation. Figure 2 tracks 1,000 samples over 10 reconstruction iterations and measures both neighborhood structure and pairwise distances. Without an explicit idempotency constraint, repeated reconstruction progressively distorts the geometry, with neighborhood relationships changing and points contracting toward a smaller region of the latent space. In contrast, the idempotent VAE substantially preserves both neighborhood relationships and relative distances across iterations.

Comparing consistency objectives. We compare six consistency objectives on Colored MNIST (Figure 5). Both cycle consistency and latent consistency improve stability relative to the standard VAE, reducing reconstruction error under repeated application. The proposed latent idempotency objective produces lower drift across iterations, indicating that constraining the encoder–decoder round trip provides a more effective mechanism for preserving the representation.

Idempotency improves generative stability. We further evaluate repeated reconstruction on natural images. Figure 8 and Table 3 show that the idempotent model preserves more information from the original input over successive reconstruction rounds, while achieving lower reconstruction error and reduced variance. This improvement persists beyond the synthetic settings, suggesting that the benefits of alignment extend to more complex image distributions.

Application to image editing. Figure 7 shows the results for image editing on Colored MNIST and dSprites. The idempotent VAE achieves a better editing success rate than the two baselines. Finally, we incorporate the idempotent VAE into a latent diffusion model following InstructPix2Pix. Figures 6 and Table 2 show that replacing standard VAE with our idempotent VAE improves the

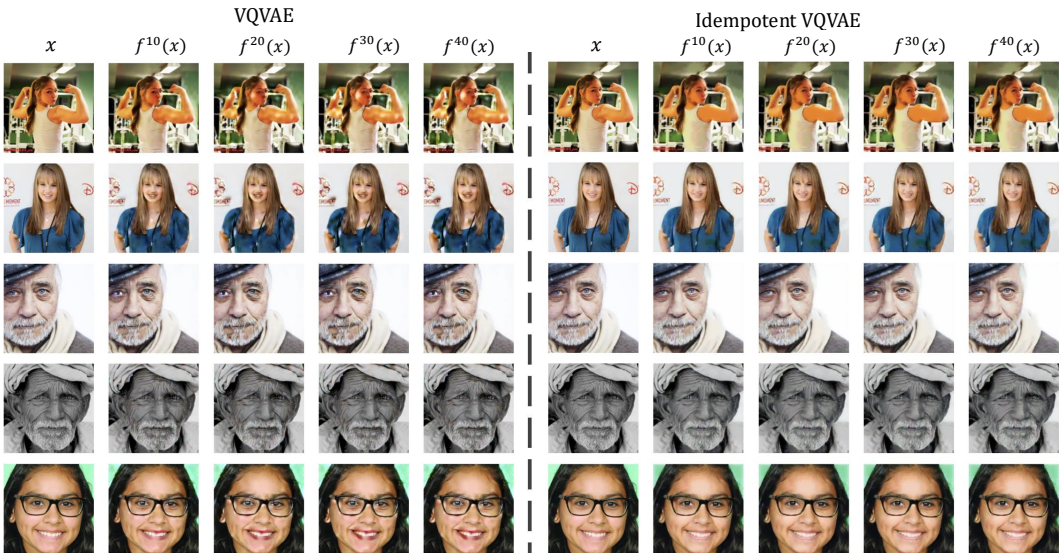


Figure 8: Idempotency test. Images show outputs after 10, 20, 30, and 40 iterations ($f^{10}(x)$, $f^{20}(x)$, $f^{30}(x)$, $f^{40}(x)$). **Left:** VQ-VAE baseline exhibits progressive drift in identity and details. **Right:** Our idempotent model preserves identity and visual fidelity across iterations.

Table 3: Reconstruction stability after 30 encoder–decoder iterations. We report mean ± standard deviation over 30 iterations for each metric. Lower values are better for all metrics.

Dataset	Method	FID ↓	MSE ↓	LPIPS ↓
MNIST	VAE	1.70 ± 1.06	0.157 ± 0.075	0.191 ± 0.041
	VAE + Ours	$\mathbf{1.12 \pm 0.65}$	$\mathbf{0.140 \pm 0.062}$	$\mathbf{0.173 \pm 0.031}$
CelebA	IGN (Shocher et al., 2023)	54.68 ± 16.71	0.464 ± 0.149	0.273 ± 0.036
	Ours	$\mathbf{14.31 \pm 6.74}$	$\mathbf{0.349 \pm 0.099}$	$\mathbf{0.226 \pm 0.030}$
LFW	VQ-VAE-2 (Razavi et al., 2019)	$\mathbf{16.33 \pm 1.49}$	0.0073 ± 0.0026	$\mathbf{0.1046 \pm 0.0081}$
	VQ-VAE-2 + Ours	17.27 ± 0.14	$\mathbf{0.0038 \pm 0.0001}$	0.1252 ± 0.0018

editing metrics across the evaluated benchmarks while maintaining comparable reconstruction fidelity. Qualitatively, the idempotent model better preserves non-target image content during editing, such as the shadow shown in the second example in Figure 6. Additional results and ablations are provided in Appendix C.

6 Conclusion

We introduced a simple framework for enforcing idempotency in encoder–decoder models by aligning the manifolds learned by the encoder and decoder. Our analysis shows that this alignment provides a direct mechanism for stabilizing the encoder–decoder round trip, while latent-space consistency offers an effective way to enforce the desired alignment. Across synthetic surfaces, image reconstruction, generation, and editing, our method consistently reduces drift under repeated application and better preserves the structure and information of the original input. These results suggest that encoder–decoder alignment is a useful objective for improving the stability of generative models and their downstream applications.

AI use statement

We used generative AI tools to help refine the writing, to set up the evaluation scripts, and to monitor experiments. For writing, we used them to edit prose for clarity and to draft LaTeX for tables and figures, working from results we produced. For engineering, we used them to help write evaluation and plotting code and to help locate related work, which we then read and cited from the primary

sources. The core implementation and experimental design are done by the authors. We did not use generative AI to generate research ideas or to produce any experimental result. All code was reviewed and tested by the authors, every number in the paper comes from our own training and evaluation runs, and every citation was checked against the cited paper. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Code and implementation details are released.

REFERENCES

Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021.

Philip Anastassiou, Zhenyu Tang, Kainan Peng, Dongya Jia, Jiaxin Li, Ming Tu, Yuping Wang, Yuxuan Wang, and Mingbo Ma. Voiceshop: A unified speech-to-speech framework for identity-preserving zero-shot voice editing. *arXiv preprint arXiv:2404.06674*, 2024.

Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7877–7888, 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.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Di Chang, Mingdeng Cao, Yichun Shi, Bo Liu, Shengqu Cai, Shijie Zhou, Weilin Huang, Gordon Wetzstein, Mohammad Soleymani, and Peng Wang. Bytemorph: Benchmarking instruction-guided image editing with non-rigid motions. *arXiv preprint arXiv:2506.03107*, 2025.

Hong Chen, Yipeng Zhang, Simin Wu, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023.

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

Xiaoran Chen and Ender Konukoglu. Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders. *arXiv preprint arXiv:1806.04972*, 2018.

Yusuf Dalva, Hamza Pehlivan, Oyku Irmak Hatipoglu, Cansu Moran, and Aysegul Dundar. Image-to-image translation with disentangled latent vectors for face editing. *IEEE transactions on pattern analysis and machine intelligence*, 45(12):14777–14788, 2023.

Frank Deutsch and F Deutsch. *Best approximation in inner product spaces*, volume 7. Springer, 2001.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Nikita Durasov, Assaf Shocher, Doruk Oner, Gal Chechik, Alexei A Efros, and Pascal Fua. It $\hat{3}$: Idempotent test-time training. *arXiv preprint arXiv:2410.04201*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Herbert Federer. *Geometric measure theory*. Springer, 2014.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Vincent Tao Hu, Wei Zhang, Meng Tang, Pascal Mettes, Deli Zhao, and Cees Snoek. Latent space editing in transformer-based flow matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 2247–2255, 2024.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European conference on computer vision*, pp. 179–196. Springer, 2018.

Nikolaj Banke Jensen and Jamie Vicary. Enforcing idempotency in neural networks. In *Forty-second International Conference on Machine Learning*, 2025.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6007–6017, 2023.

Jimyeong Kim, Jungwon Park, Yeji Song, Nojun Kwak, and Wonjong Rhee. Reflex: Text-guided editing of real images in rectified flow via mid-step feature extraction and attention adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15939–15948, 2025.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. *Neural Networks*, 10(6):1117–1119, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 35–51, 2018.

Weiwei Lin, Chenhang He, Man-Wai Mak, Jiachen Lian, and Kong Aik Lee. Voxgenesis: Unsupervised discovery of latent speaker manifold for speech synthesis. *arXiv preprint arXiv:2403.00529*, 2024.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

Marian Lupaşcu and Mihai Sorin Stupariu. Optimal transport for rectified flow image editing: Unifying inversion-based and direct methods. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6764–6774, 2026.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516, 2024.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Yushu Pan and Elias Bareinboim. Counterfactual image editing with disentangled causal latent space. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 833–840, 2011.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 42(1):1–13, 2022.

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

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Jiang Bian, et al. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*, 2023.

Zheng-Yan Sheng, Li-Juan Liu, Yang Ai, Jia Pan, and Zhen-Hua Ling. Voice attribute editing with text prompt. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei A Efros. Idempotent generative network. *arXiv preprint arXiv:2311.01462*, 2023.

Mustafa Shukor, Xu Yao, Bharath Bushan Damodaran, and Pierre Hellier. Semantic unfolding of stylegan latent space. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 221–225. IEEE, 2022.

Angus E. Taylor and David C. Lay. *Introduction to Functional Analysis*. John Wiley & Sons, New York, 2nd edition, 1980. ISBN 0471846465.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on graphics (TOG)*, 40(4):1–14, 2021.

Xiaoce Wang, Sifan Zhou, Kaifei Wang, Leli Xu, Xuerui Qiu, Tao He, and Ming Li. Why do dit editors drift? plug-and-play low frequency alignment in vae latent space. *arXiv preprint arXiv:2605.08250*, 2026.

Yushu Wu, Yanyu Li, Ivan Skorokhodov, Anil Kag, Willi Menapace, Sharath Girish, Aliaksandr Siarohin, Yanzhi Wang, and Sergey Tulyakov. H3ae: High compression, high speed, and high quality autoencoder for video diffusion models. *arXiv preprint arXiv:2504.10567*, 2025a.

Zilong Wu, Hideki Murata, Nayu Takahashi, Qiyu Wu, and Yoshimasa Tsuruoka. Latentps: Image editing using latent representations in diffusion models. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 167–176, 2025b.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12863–12872, 2021.

Xiangpeng Yang, Linchao Zhu, Hehe Fan, and Yi Yang. Videograin: Modulating space-time attention for multi-grained video editing. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 2025.

Shehtab Zaman, Chengyan Liu, and Kenneth Chiu. Score-based idempotent distillation of diffusion models. *arXiv preprint arXiv:2509.21470*, 2025.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

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

Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15792–15801. IEEE, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017b.

APPENDIX CONTENTS

Appendix A. Implementation Details 16
A.1. Synthetic-Data Experiments 16
A.2. Manifold-Alignment Analysis 18
A.3. VQ-VAE-2 18
A.4. DCGAN: IGN vs. Ours 18
A.5. InstructPix2Pix 19
A.6. Evaluation Metrics 19
Appendix B. Proofs 19
Idempotency is Necessary for Projection Optimality 19
Latent Consistency is Sufficient for Output Idempotency 19
Latent Error Bounds Output-Space Drift 20
Appendix C. Additional Results 20

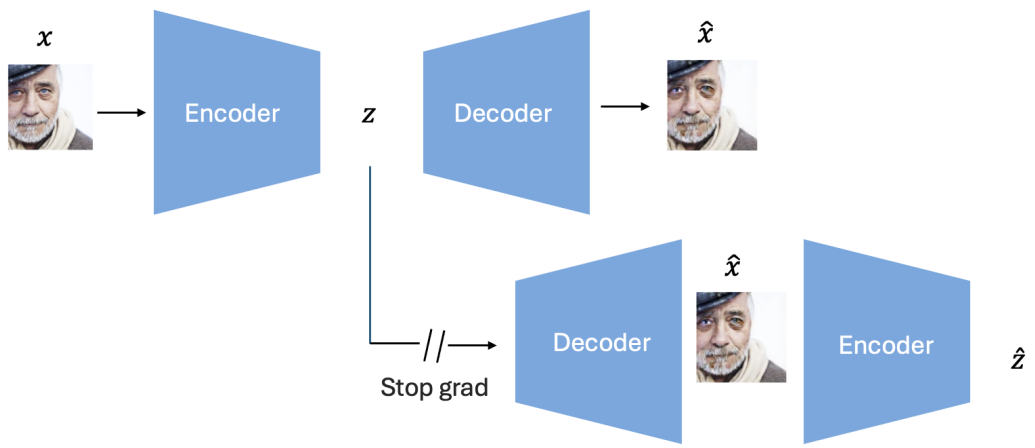


Figure 9: Idempotent training via encoder-decoder consistency. An input sample x is encoded into a latent representation z and decoded to produce a reconstruction $\hat{x}$. The reconstruction is then passed through a second decode-encode cycle with gradients stopped, encouraging the encoder and decoder to learn consistent representations.

A Implementation Details

Figure 9 illustrates the objective. An input x is encoded to a latent code $z = E(x)$ and decoded to a reconstruction $\hat{x} = D(z)$; the latent code is then passed through a second decode–encode cycle, $\hat{z} = E(D(\mathrm{sg}[z]))$, and we penalise the discrepancy between z and $\hat{z}$. The stop-gradient prevents this term from updating the encoder through the second pass, so the constraint is enforced in latent space rather than in input space as in IGN (Shocher et al., 2023).

We implement the proposed objective on four autoencoders: a small VAE trained on synthetic data, a VQ-VAE trained on CelebA, a DCGAN model that allows direct comparison with IGN (Shocher et al., 2023), and the InstructPix2Pix VAE (Brooks et al., 2023), which we fine-tune on the InstructPix2Pix dataset. Our implementations build on the publicly available code for IGN,[2] InstructPix2Pix,[3] and VQ-VAE-2.[4] Each setting is described in a separate subsection below.

A.1 Synthetic-Data Experiments

In these experiments we use three data sources: **Colored MNIST** (LeCun et al., 1998) and **dSprites** (Matthey et al., 2017), which have labelled generative factors and let us measure attribute editing, and **six 2-D manifolds** embedded in $\mathbb{R}^3$, which have an exact data manifold and let us isolate the geometric effect of the loss and to test if repeatedly applying the model $f = D \circ E$ keeps points on the manifold. Table 4 summarises the data.

Table 4: Synthetic datasets.

Dataset	Sample	Factors	Size
Colored MNIST	$3 \times 28 \times 28$	digit (10), color (10)	60k / 10k
dSprites	$1 \times 64 \times 64$	shape, scale, orient., posX, posY	90/10 split
Six manifolds	$\mathbb{R}^3$ (2-D)	swiss roll, S-curve, torus, sphere, helix tube, trefoil tube	6,000 pts

[2] https://github.com/Alokia/Idempotent-Generative-Network
[3] https://github.com/timothybrooks/instruct-pix2pix
[4] https://github.com/rosinality/vq-vae-2-pytorch

Idempotency objective. All variants optimise the VAE ELBO augmented with a latent-consistency term enforced *in latent space* (Algorithm 1). The term has two parts: (i) a data term that requires re-encoding a decoded posterior sample to return its own code, and (ii) a synthetic term that samples codes from the posterior statistics at several scales and requires the encode–decode round trip to fix them. Gradients through the second pass are stopped, so the constraint shapes the map $E \circ D$.

Algorithm 1 Idempotency-regularised VAE training step

Require: batch x, KL weight β, idempotency weight λ, term weights $\lambda_{\text{data}}, \lambda_{\text{syn}}$, scales $\mathcal{S} = \{0.5, 1, 2\}$

$\mu, \log \sigma^2 \leftarrow E(x), \quad z \sim \mathcal{N}(\mu, \text{diag}(\sigma^2))$ ▷ clean pass

$\mathcal{L}_{\text{ELBO}} \leftarrow \text{recon}(D(z), x) + \beta \, \text{KL}(q(z \mid x) \| p(z))$

$\hat{\mu}, \widehat{\log \sigma^2} \leftarrow E(D(\text{sg}[z]))$ ▷ re-encode reconstruction

$\mathcal{L}_{\text{data}} \leftarrow \|\hat{\mu} - \text{sg}[\mu]\|_2^2 + \|\widehat{\log \sigma^2} - \text{sg}[\log \sigma^2]\|_2^2$

for $s \in \mathcal{S}$ **do**

  $z_s \sim \mathcal{N}(\text{sg}[\mu], s^2 \, \text{diag}(\sigma^2))$

  $\mathcal{L}_s \leftarrow \|E(D(z_s)) - \text{sg}[z_s]\|_2^2$

end for

$\mathcal{L}_{\text{syn}} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathcal{L}_s$ ▷ off-manifold latents

$\mathcal{L} \leftarrow \mathcal{L}_{\text{ELBO}} + \lambda(\lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{syn}} \mathcal{L}_{\text{syn}})$

update θ, ϕ by $\nabla \mathcal{L}$

Attribute editing (Colored MNIST, dSprites). Editing follows a two-stage pipeline. We first train an unconditional VAE, then freeze the encoder and train a conditional FFJORD flow g_c on the encoder means using learnable per-factor embeddings as the condition c. An edit maps a latent to the base density under the source attributes and inverts under the target attributes (Algorithm 2). Configurations are in Table 5. The results for this experiment are shown in Figure 13 and 7.

Algorithm 2 Attribute editing via the conditional flow

Require: image x, source attributes c_s, target attributes c_t

$\mu \leftarrow E(x); \quad \epsilon \leftarrow g_{c_s}(\mu)$ ▷ flow forward under source

$\mu' \leftarrow g_{c_t}^{-1}(\epsilon)$ ▷ flow inverse under target

return $D(\mu')$

Table 5: Model and training configuration for the editing datasets.

	Colored MNIST	**dSprites**
VAE architecture / latent	ConvVAE / 32	ResVAE64 / 64
VAE lr, batch, epochs	10^{-3}, 2048, 200	10^{-3}, 1024, 20
Flow (FFJORD) blocks / hidden	4 / (256^3)	4 / (256^3)
Flow lr, batch, epochs	$5{\times}10^{-4}$, 8192, 30	$5{\times}10^{-4}$, 4096, 30
Condition embeddings	digit,color $\times 16$	5 factors $\times 16$
Condition vocabularies	10, 10	[3, 6, 40, 32, 32]

Manifold iteration (six manifolds). For each shape we sample 6,000 points, standardise them, and train an MLP-VAE (latent dim $= 2$, hidden 128, depth 3; Adam, lr 10^{-3}; 1200 epochs; $\beta = 0.4$, $\lambda = 10$). We then iterate the deterministic map $f^t(x) = (D \circ E)^t(x)$ and measure how far the iterates leave the manifold. Figure 15 shows the results for this experiment.

Evaluation metrics. We report **CAS** (Ravuri & Vinyals, 2019) (a classifier trained on generated samples, tested on real data), **ESR** (a real-data classifier's agreement that an edit hit its target attribute), **edit reversibility** (MSE/LPIPS between an image and its edit-then-revert), and **idempotency** (reconstruction MSE under $t \in \{10, 30, 50\}$ repeated encode–decode steps).

A.2 MANIFOLD-ALIGNMENT ANALYSIS

The reconstruction map $P = D \circ E$ is meant to project onto the data manifold, and a well-aligned encoder and decoder make P idempotent, so that iterating it stays put. We probe this alignment two ways: locally, through the Jacobian of the latent round trip, and globally, by iterating P and tracking what happens to the latent arrangement.

Round-trip Jacobian. Let $h = E \circ D$ and $J = \partial h / \partial z \big|_{z_0}$ at $z_0 = E(x)$. If the encoder inverts the decoder then h is the identity and $J = I$, so every latent perturbation returns unchanged. The singular values $\sigma_{\max}(J) \geq \sigma_{\min}(J)$ give the gain in the best- and worst-preserved directions; because they are invariant to rotation, a relabelling of the latent coordinates is correctly not counted as a failure. A singular value below one means part of a perturbation was lost on the round trip, which happens exactly when the decoder moves the surface in a direction the encoder does not respond to. We evaluate J at 200 points per surface, as it varies across the manifold, and report the joint distribution of $(\sigma_{\max}, \sigma_{\min})$ in Figure 3. Isotropic loss appears along the diagonal $\sigma_{\max} = \sigma_{\min}$; anisotropic loss, where one direction survives and its orthogonal complement is crushed, appears away from it.

Neighbourhood preservation under iteration. We apply P to the data t times and compare the resulting latents $z_t = E(P^t(x))$ against z_0. Let $N_k^t(i)$ be the index set of the k nearest neighbours of point i at iteration t ($k = 20$), and $\rho_t(i)$ its nearest-neighbour distance. Averaging over anchors,

$$\mathcal{J}(t) = \left\langle \frac{|N_k^0(i) \cap N_k^t(i)|}{|N_k^0(i) \cup N_k^t(i)|} \right\rangle_i, \qquad \mathcal{R}(t) = \left\langle \frac{|N_k^0(i) \cap N_{2k}^t(i)|}{k} \right\rangle_i, \qquad \mathcal{C}(t) = \frac{\mathrm{med}_i\, \rho_t(i)}{\mathrm{med}_i\, \rho_0(i)}.$$

$\mathcal{J}$ penalises both neighbours that leave and new ones that arrive; $\mathcal{R}$ widens the net to $2k$ after iteration and so penalises only those that leave, making the two separable. $\mathcal{C}$ is one when the latent cloud keeps its scale. Because nearest-neighbour sets are invariant to uniform rescaling, measuring in the latent space separates reordering from shrinkage rather than confounding them. Metrics are averaged over 10 anchor resamplings of 1,000 points, for $t = 1, \ldots, 10$.

For the idempotent model $\mathcal{J}$ and $\mathcal{R}$ stay high and $\mathcal{C}$ stays near one throughout, while the baseline degrades monotonically on both. The two analyses agree: the surfaces on which the baseline's spectra depart furthest from identity are the same ones whose neighbourhoods dissolve fastest under iteration, indicating that the drift observed in the iteration grids (Figures 2) originates in a local geometric mismatch rather than in accumulated numerical error.

A.3 VQ-VAE-2

We test the objective on a discrete-latent model, the two-level hierarchical VQ-VAE-2 (Razavi et al., 2019), trained on CelebA and evaluated on LFW at 256×256 (images normalised to $[-1, 1]$). The architecture uses 128 base channels, two residual blocks (32 channels), a 64-dim embedding with a 256-entry codebook per level, and EMA codebook updates (decay 0.99). The baseline minimises reconstruction MSE plus a commitment loss (weight 0.25); we train with Adam (lr 3×10^{-4}).

Our variant adds a latent-consistency term on the *quantised* codes. For each level, decoding the (detached) codes and re-encoding must return the same pre-quantisation features, and the same must hold for synthetic codes drawn from the per-channel mean and standard deviation of the batch's codes:

$$\mathcal{L}_{\text{idem}} = \underbrace{\left\| E(D(\mathrm{sg}[q])) - \mathrm{sg}[z_e] \right\|^2}_{\text{data codes}} + 0.5 \underbrace{\left\| E(D(q_{\text{syn}})) - q_{\text{syn}} \right\|^2}_{\text{synthetic codes}}, \qquad q_{\text{syn}} \sim \mathcal{N}(\bar{q},\, \sigma_q^2),$$

summed over the top and bottom levels and added to the baseline loss (all other settings unchanged). Both models are evaluated with the reconstruction metrics, the results are shown in (Table 3).

A.4 DCGAN: IGN VS. OURS

To compare directly with IGN (Shocher et al., 2023) we use its DCGAN encoder–decoder on CelebA at 64×64, batch size 256, AdamW (lr 10^{-4}, $\beta = (0.5, 0.999)$). **IGN** applies its losses in image

space: reconstruction, idempotency $f(f(z)) \approx f(z)$, and a smoothed tightness term, evaluated on noise sampled from the batch statistics (weights λ_{rec}=10, λ_{idem}=10, λ_{tight}=2.5, a=1.5).

Ours keeps the same backbone but enforces idempotency *in latent space*: it re-encodes a decoded latent and matches the encoder's posterior statistics,

$$\mathcal{L}_{\text{idem}} = \left\|\mu\big(E(D(\text{sg}[z]))\big) - \text{sg}[\mu(E(x))]\right\|^2 + \left\|\log\sigma^2(\cdot) - \text{sg}[\log\sigma^2(E(x))]\right\|^2,$$

trained alongside an ℓ_1 reconstruction loss, an adversarial loss with a discriminator, and feature matching. Both methods are evaluated with the reconstruction-stability protocol above (Table 3). Figure 11 shows generation results for this experiment.

A.5 InstructPix2Pix

We also fine-tune the VAE of InstructPix2Pix (Brooks et al., 2023) with the same latent-consistency objective; the implementation follows the setups above, adding the latent-space idempotency term to the VAE's reconstruction loss while leaving the backbone unchanged. Training uses the InstructPix2Pix generated editing dataset (input image, instruction, edited image), available from the authors' repository.[5]

A.6 Evaluation Metrics

For VAE reconstruction we report PSNR and SSIM (pixel-level fidelity), LPIPS (Zhang et al., 2018) (perceptual quality), and CLIP-I (Radford et al., 2021) (semantic preservation). For instruction-based editing we additionally report DINO (Caron et al., 2021) (fine-grained structural preservation) and FID (Heusel et al., 2017) (distributional realism of edits). For synthetic benchmarks we use CAS (Ravuri & Vinyals, 2019) (class fidelity of generated samples) and ESR (attribute edit accuracy). For idempotent generation we report FID, MSE, and LPIPS over repeated reconstruction iterations.

B Proofs

We provide formal proofs for the theoretical claims made in Section 3.

Idempotency is Necessary for Projection Optimality

Definition 1 (Projection Operator)**.** A map $P : \mathcal{X} \to \mathcal{X}$ is a projection onto a set $\mathcal{M} \subseteq \mathcal{X}$ if $P(x) \in \mathcal{M}$ for all x, and $P(p) = p$ for all $p \in \mathcal{M}$.

Proposition 1. *Let $f = D_\theta \circ E_\phi : \mathcal{X} \to \mathcal{X}$ be an encoder–decoder composition that implements a projection onto a learned manifold $\mathcal{M}$, so that $f(x) = \text{proj}_{\mathcal{M}}(x)$. Then f must be idempotent, $f(f(x)) = f(x)$ for all $x \in \mathcal{X}$.*

Proof. Since f is a projection onto $\mathcal{M}$, we have $f(x) \in \mathcal{M}$ for all $x \in \mathcal{X}$. By definition of a projection, any point already on $\mathcal{M}$ is fixed, so $f(p) = p$ for all $p \in \mathcal{M}$. Setting $p = f(x) \in \mathcal{M}$ gives

$$f(f(x)) = f(x). \qquad \square$$

Remark 1. Contrapositive. If $f(f(x)) \neq f(x)$ for some x, then f does not implement a valid projection onto any manifold, and the learned representation is suboptimal in the projection sense.

Latent Consistency is Sufficient for Output Idempotency

Proposition 2. *Let $f = D_\theta \circ E_\phi$. If the encoder satisfies the latent consistency condition*

$$E_\phi(D_\theta(E_\phi(x))) = E_\phi(x) \quad \forall\, x \in \mathcal{X}, \tag{12}$$

then f is idempotent, so that $D_\theta(E_\phi(D_\theta(E_\phi(x)))) = D_\theta(E_\phi(x))$ for all $x \in \mathcal{X}$.

[5] https://github.com/timothybrooks/instruct-pix2pix

Proof. Let $z = E_\phi(x)$ and $\hat{x} = D_\theta(z)$. By assumption equation 12,

$$E_\phi(\hat{x}) = E_\phi(D_\theta(z)) = z.$$

Applying D_θ to both sides,

$$D_\theta(E_\phi(\hat{x})) = D_\theta(z) = \hat{x}.$$

Substituting back $\hat{x} = D_\theta(E_\phi(x))$,

$$D_\theta(E_\phi(D_\theta(E_\phi(x)))) = D_\theta(E_\phi(x)). \qquad \square$$

LATENT ERROR BOUNDS OUTPUT-SPACE DRIFT

Proposition 3. *Suppose D_θ is L-Lipschitz. Then the output-space idempotency error is bounded by the latent-space idempotency error,*

$$\|D_\theta(E_\phi(D_\theta(E_\phi(x)))) - D_\theta(E_\phi(x))\| \;\le\; L \cdot \|E_\phi(D_\theta(E_\phi(x))) - E_\phi(x)\|.$$

Proof. Let $z = E_\phi(x)$ and $\hat{z} = E_\phi(D_\theta(z))$. Since D_θ is L-Lipschitz,

$$\|D_\theta(\hat{z}) - D_\theta(z)\| \le L\,\|\hat{z} - z\| = L\,\|E_\phi(D_\theta(E_\phi(x))) - E_\phi(x)\|. \qquad \square$$

Corollary 1. *Under the assumptions of Proposition 3, the latent idempotency loss $\mathcal{L}_{\text{idem}} = \mathbb{E}_x\left[\|E_\phi(D_\theta(\text{sg}(z))) - \text{sg}(z)\|^2\right]$ controls output-space drift through the upper bound*

$$\mathbb{E}_x\,\|D_\theta(\hat{z}) - D_\theta(z)\|^2 \;\le\; L^2\,\mathcal{L}_{\text{idem}}, \qquad z = E_\phi(x),\;\; \hat{z} = E_\phi(D_\theta(z)).$$

At a fixed Lipschitz constant, reducing $\mathcal{L}_{\text{idem}}$ tightens this bound, which justifies minimizing the latent loss as a practical surrogate for output-space idempotency.

Remark 2. The bound does not imply a proportional reduction in drift. The true drift may fall by a different factor, or remain unchanged while the bound tightens, and L itself varies over training and across regions of the latent space. The guarantee is one-sided, so a small latent loss certifies small output drift, while a large latent loss does not certify the converse. Figure 5 reports measured output drift directly for this reason.

C ADDITIONAL RESULTS

Idempotency loss weight. Table 6 ablates the idempotency loss weight λ on MNIST. Performance is stable across a wide range of values, with higher λ generally improving FID while maintaining competitive LPIPS.

Latent interpolation. Figure 10 compares latent interpolations across VAE, VAE + IGN, and VAE + Ours on MNIST. Encoder–decoder alignment enables smooth, semantically meaningful transitions: digits change gradually from 7 to 2 with clear intermediate samples. Baseline and IGN interpolations contain noisy or ambiguous intermediates, indicating latent space instability. This suggests alignment improves not only idempotency but also the global geometry of the latent space.

Editing quality. Figure 12 shows qualitative editing results on Colored MNIST. VAE + Ours preserves digit identity and color fidelity under repeated application, whereas Baseline and IGN exhibit color drift and digit distortion. Figure 6 compares a latent diffusion model trained on idempotent versus standard VAE representations. The idempotent representation preserves finer details, including object positions, grass color, and sky color across edited scenes.

Robustness to input transformations. Figure 11 evaluates DCGAN under input transformations (Gaussian noise, grayscale, and sketching) following Shocher et al. (2023). Our method produces more consistent outputs and better preserves facial structure under repeated application compared to IGN.

Per-factor CAS. Figure 13 breaks down CAS per generative factor on dSprites and Colored MNIST. Our method achieves competitive CAS across most factors, with the largest remaining gap on orientation in dSprites. Figure 14 shows qualitative reconstruction comparisons across VAE, VAE + IGN, and VAE + Ours, confirming that our method produces more consistent and stable reconstructions.

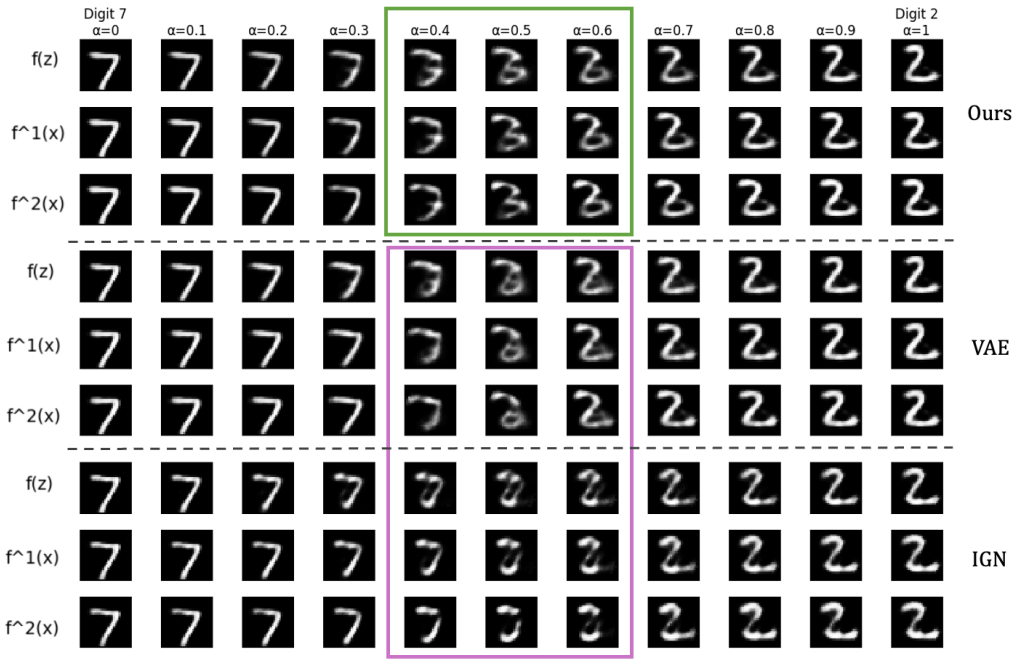


Figure 10: Latent interpolation under repeated generation on MNIST. Each column shows interpolation coefficient α between digits 7 and 2; each row shows repeated application of the same generator. Our method produces smooth, stable interpolations compared to VAE and IGN.

Table 6: Ablation study of the idempotency loss weight (`Idem W`) on MNIST reconstruction using VAE + Ours. FID, MSE, and LPIPS are reported as mean $\pm$ std over 30 iterations. Bold values indicate the best result per metric. Higher idempotency weights generally improve FID while maintaining competitive MSE and LPIPS.

Idem W	0.0	0.2	0.5	0.8	1	2	5	10	20	100
FID	1.70 ± 1.06	1.18 ± 0.67	1.20 ± 0.73	1.08 ± 0.66	1.11 ± 0.68	1.21 ± 0.75	1.32 ± 0.79	1.12 ± 0.65	1.10 ± 0.67	$\mathbf{0.93 \pm 0.54}$
MSE	0.157 ± 0.075	0.144 ± 0.064	0.146 ± 0.066	0.141 ± 0.062	0.142 ± 0.063	0.145 ± 0.065	0.147 ± 0.066	0.140 ± 0.062	0.139 ± 0.062	$\mathbf{0.137 \pm 0.059}$
LPIPS	0.191 ± 0.041	0.177 ± 0.033	0.176 ± 0.034	0.175 ± 0.033	0.176 ± 0.034	0.178 ± 0.035	0.180 ± 0.035	0.173 ± 0.031	0.176 ± 0.034	$\mathbf{0.171 \pm 0.030}$

Manifold iteration grids. Figure 15 visualises what repeated reconstruction does to six synthetic surfaces in $\mathbb{R}^3$. After a single round both the baseline and our model reproduce each surface faithfully, so a one-shot comparison would find little to separate them. The difference appears only under repetition: by the tenth round the baseline has collapsed each surface onto a thin subset, losing both its extent and its internal ordering, while our model returns a shape close to the original. Because colour encodes position on the source manifold and is carried through the iterations, the mixing visible in the baseline rows shows that points have moved relative to one another rather than the cloud merely shrinking uniformly.

Round-trip Jacobian. Figure 3 locates the cause of that collapse in the local geometry. If the encoder inverts the decoder, the Jacobian of the latent round trip $E \circ D$ is the identity and both its singular values equal one, so every perturbation returns unchanged. The baseline instead sits away from $(1, 1)$ with $\sigma_{\max}$ near one and $\sigma_{\min}$ up to two orders of magnitude smaller, meaning one direction survives the round trip while its orthogonal complement is destroyed: the decoder moves the surface in a direction the encoder does not respond to. The loss is therefore anisotropic rather than a uniform shrinkage, which is what folding a non-trivially embedded surface into a two-dimensional latent produces. This measurement requires no iteration, and the surfaces on which the spectra depart furthest from identity are those whose neighbourhoods dissolve fastest in Figure 2, indicating that the drift originates in a local geometric mismatch rather than in accumulated numerical error.

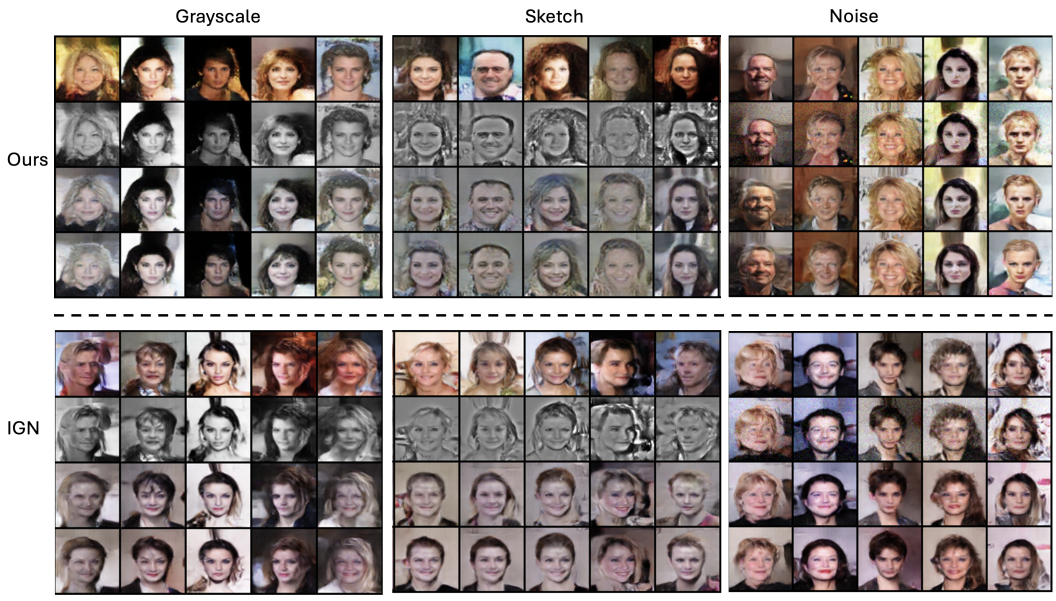


Figure 11: Idempotency under input transformations on CelebA using DCGAN. First row: generated images. Second row: transformed inputs (grayscale, sketch, noise) following Shocher et al. (2023). Third and fourth rows: first and second re-applications of the model. Our method produces more consistent outputs and preserves facial structure better than IGN.

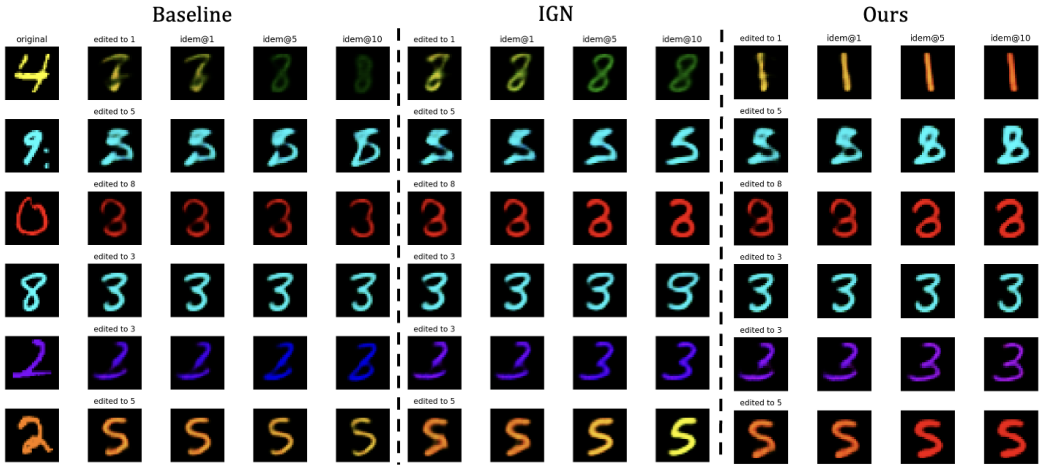


Figure 12: Qualitative editing results on Colored MNIST. Each row shows an original image, the edited result, and repeated applications at $f^1(x)$, $f^5(x)$, and $f^{10}(x)$. Baseline and IGN exhibit color drift and digit distortion, whereas our method preserves both digit identity and color fidelity across iterations.

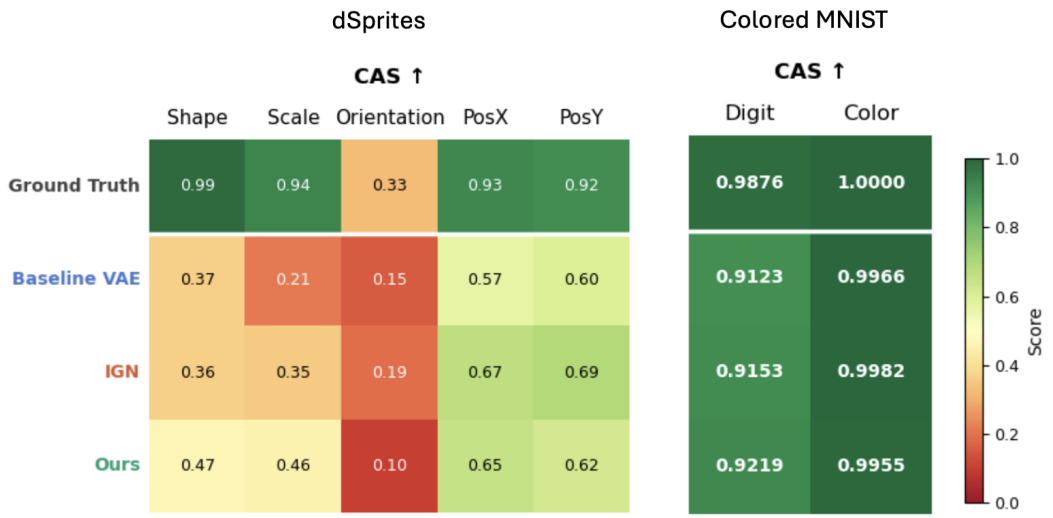
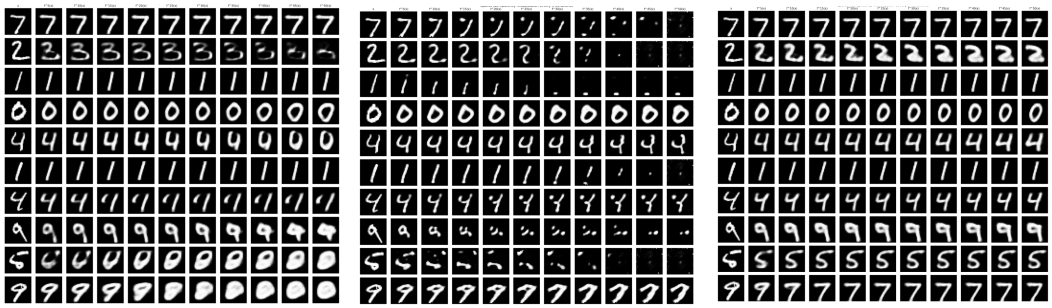


Figure 13: Per-factor CAS breakdown on dSprites and Colored MNIST. Ground Truth serves as an upper bound. Our method is competitive across most factors, with the largest remaining gap on orientation in dSprites.



(a) VAE (b) VAE + IGN (c) VAE + Ours

Figure 14: Qualitative reconstruction comparison on MNIST. Our method produces more consistent and stable reconstructions across iterations.

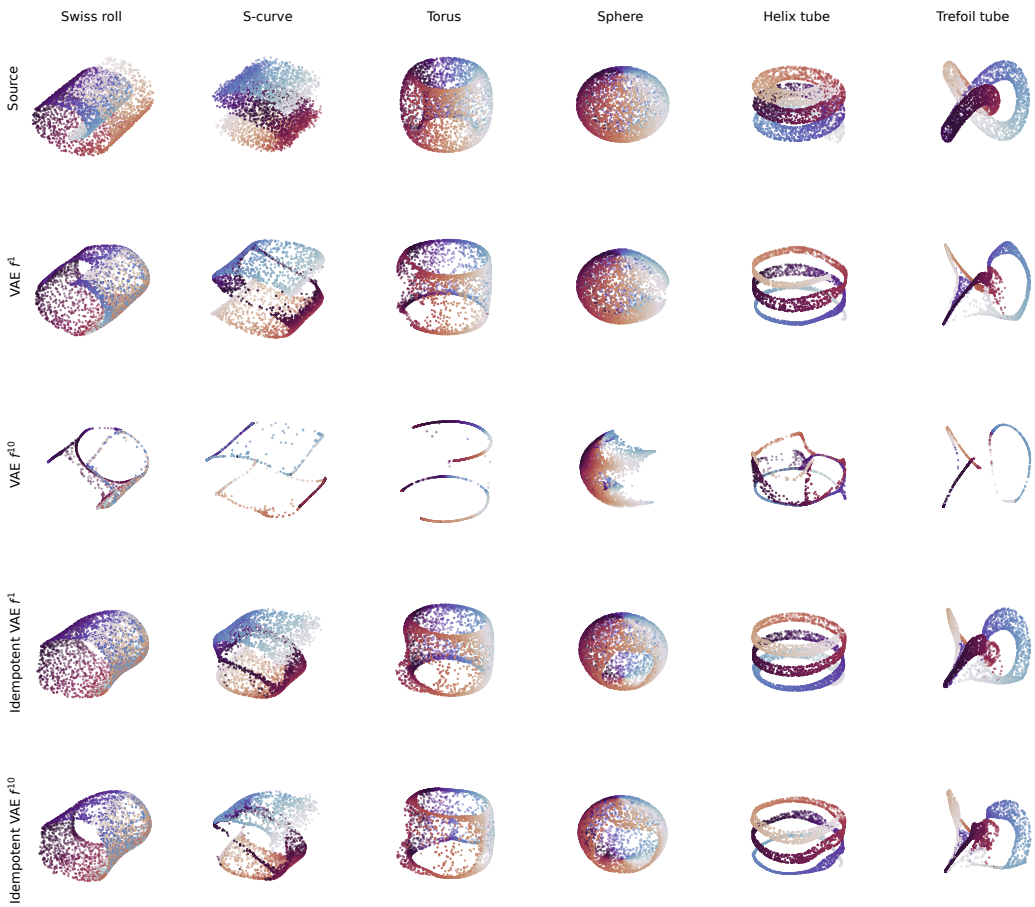


Figure 15: A comparison of the baseline VAE and the idempotent model after repeated reconstruction rounds, each applying $f = D \circ E$. Columns are the six surfaces; the top row is the source data, and the remaining rows show the output after one and ten rounds for each model. Colour encodes position on the source manifold and is carried through the iterations, so mixing indicates that points have moved relative to one another. After a single round both models reproduce the surface, but by the tenth the baseline has collapsed onto a thin subset while the idempotent model preserves the shape.