LEARNING MANIFOLD-FAITHFUL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoencoders learn latent representations for reconstructing observed data. However, reconstruction training constrains the decoder only at encoded data points, providing no guarantee that perturbations of these codes will decode to samples that remain on the data manifold. Consequently, even a small step away from an encoded point can push the model into an unconstrained region of the latent space, causing the decoded output to degrade. To address this limitation, we introduce manifold-faithful traversal, which constrains movement through the latent space to follow the learned data manifold. By keeping the trajectory on the manifold, latent codes can be repeatedly perturbed while producing plausible intermediate samples without accumulating off-manifold drift over successive steps. We propose a two-phase training framework that first learns a reconstructive representation and then trains the decode–encode mapping to correct perturbed latent codes toward the manifold. Because the true latent manifold is unknown, we approximate this projection using the nearest clean latent code in each training batch. Unlike disentanglement, which seeks latent directions associated with identifiable semantic factors, manifold-faithful traversal does not require specifying what changes along a trajectory; it only requires that the trajectory remain on the data manifold. Experiments on synthetic data and Colored MNIST show that the proposed technique substantially reduces accumulated off-manifold drift and produces coherent trajectories in both random and target-directed walks. We further show that a learned traversal policy together with a memory-efficient training scheme lets the approach scale to larger latent widths and longer trajectories, remaining feasible and effective where the original mechanism is not. Such stable multi-step traversal potentially enables latent space exploration for generative modeling and downstream tasks such as data augmentation that depend on outputs remaining realistic.

1 INTRODUCTION

Autoencoders are commonly trained by minimizing the reconstruction error,

$$\mathcal{L}_{\text{recon}}(E, D) = \mathbb{E}_{x \sim p_x}\left[\|x - D(E(x))\|_2^2\right], \tag{1}$$

where $E : \mathcal{X} \to \mathbb{R}^d$ maps an observation x to a latent representation $z = E(x)$, and $D : \mathbb{R}^d \to \mathcal{X}$ reconstructs the observation from that representation. By minimizing Eq. 1, the composition $D \circ E$ is encouraged to preserve the information needed to recover observed data. Such representations are useful not only for reconstruction and compression, but also as spaces in which data can potentially be manipulated. Ideally, nearby movements in the latent space should correspond to plausible variations of the underlying data, allowing one to start from an observed sample, perturb its representation, and obtain another plausible sample. While reconstruction training encourages the decoder to behave correctly on $\mathcal{M}$, it provides no corresponding guarantee for nearby points outside this set. Consequently, a small latent perturbation

$$\bar{z} = z + \eta u, \qquad z \in \mathcal{M}, \tag{2}$$

may move the representation into a region where the decoder is unconstrained, even when $\bar{z}$ is close to z. The resulting decode $D(\bar{z})$ may therefore deviate from the data manifold despite accurate



Figure 1: A targeted latent walk (blue) under our method, moving toward a target code (green star) while staying on the data manifold (gray).

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.

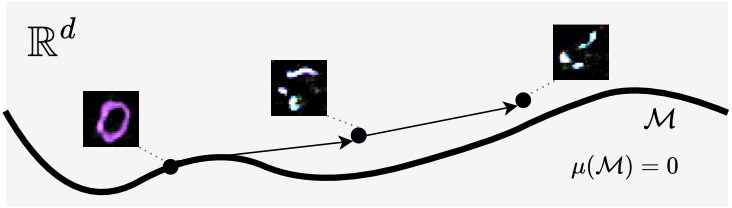


Figure 2: Reconstruction training constrains D only on $\mathcal{M}$, which has Lebesgue measure zero. As the walk moves off $\mathcal{M}$, the decoder's real output degrades into unstructured noise.

reconstruction at the original code. Under repeated perturbations, these deviations can accumulate, progressively pushing the trajectory farther from regions supported by the data.

This motivates the central question of this work: *can we repeatedly move through a learned latent space while remaining on the data manifold?* Starting from an observation x and its latent representation $z_0 = E(x)$, we ask whether a small perturbation $z_0 + \eta u$ can still decode to a plausible sample, and whether this process can be repeated to produce a long trajectory of plausible, smoothly varying samples without accumulating off-manifold drift. We refer to this property as **manifold-faithful traversal**: movement through the latent space should follow the learned data manifold, such that every intermediate point decodes to a plausible sample throughout the trajectory.

This question is distinct from disentanglement. Disentanglement concerns the semantic meaning of latent variables or directions, whereas manifold-faithful traversal concerns whether movement along those directions remains valid. We therefore do not require a latent direction to correspond to an identifiable semantic factor; we require only that traversing it remains supported by the data manifold.

As a solution, we propose a two-phase training framework that first learns a reconstructive representation and then trains the decode–encode mapping to correct perturbed latent codes toward the data manifold in the latent space. Because the true latent manifold is unknown, we approximate this projection using the nearest clean latent code in each training batch. Experiments on synthetic data and Colored MNIST show that the proposed technique substantially reduces accumulated off-manifold drift and produces coherent trajectories in both random and target-directed walks.

Contributions. We summarize our contributions as follows: (1) We formalize manifold-faithful latent traversal as an unsupervised task distinct from disentanglement and from counterfactual editing, and position it relative to identifiability impossibility results (Sec. 2). (2) We propose a two-phased objective whose second phase adds a nearest-neighbour latent projection loss, training $E \circ D$ to behave as a retraction onto the latent manifold (Sec. 3 and 4). (3) We give a traversal algorithm alternating a small random step with a decode-encode projection, and show on a synthetic dataset and Colored MNIST that trajectories stay on-manifold and transition smoothly (Sec. 5). (4) We show the approach scales, pairing a learned traversal policy with a memory-efficient training scheme to handle larger latent widths and longer trajectories on Colored MNIST, at settings where the original mechanism becomes infeasible (Sec. 6).

2 The Task: Manifold-Faithful Traversal

Let p_x be supported on a low-dimensional set $\mathcal{X} \subset \mathbb{R}^D$ and let

$$\mathcal{M} = E(\text{supp}(p_x)) \subset \mathbb{R}^d \tag{3}$$

be the induced latent manifold, the set of codes the decoder has ever been asked to invert. Reconstruction training (Eq. 1) only constrains D on $\mathcal{M}$, which is typically of measure zero in $\mathbb{R}^d$ (Figure 2). We therefore ask for the following property, which equation 1 does not imply.

Definition 1 (Manifold-faithful traversal.)**.** An encoder-decoder pair supports (η, ϵ) manifold-faithful traversal if $\forall z \in \mathcal{M}$ and unit direction u, $d(D(z + \eta u), \text{supp}(p_x)) \leq \epsilon$, and the property is closed under iteration, so that T successive perturbations do not accumulate off-manifold drift.

Three things are worth separating. Let $x = g(s_1, \ldots, s_k)$ be a generative model with true factors s_i. Disentanglement asks that the learned encoder satisfies $E(x) = (r_1(x), \ldots, r_d(x))$ where each r_j depends on at most one s_i, which is unidentifiable from p_x alone without side information Locatello et al. (2019). Counterfactual editing assumes a structural causal model $s_i := f_i(\mathrm{pa}_i, u_i)$ and asks for $x_{s_i \leftarrow a} = g(s_1, \ldots, s_{i-1}, a, s_{i+1}, \ldots, s_k)$ given observation x, which is the image that would have been observed under intervention $\mathrm{do}(s_i := a)$, requiring both the causal graph and recovery of the exogenous u_i Pearl et al. (2003); Pan & Bareinboim (2024). Manifold faithful traversal (Def. 1) only asks that $D(z + \eta u) \in \mathrm{supp}(p_x)$ for $z \in \mathcal{M}$, closed under iteration, without specifying which factor changes or why.

3 METHOD

This section develops the two-phase training objective (Sec. 3.1 and Sec. 3.2) and the traversal recursion used at inference time (Sec. 3.3). Sec. 4 shows both are well-designed.

3.1 PHASE I: RECONSTRUCTION WARM-UP

Training the projection loss from scratch admits a degenerate solution, where a constant encoder makes every code identical, so every perturbation projects back to the same point and the loss vanishes. We therefore train for T_{warm} steps on $\mathcal{L}_{rec}$ alone, obtaining an encoder-decoder pair whose latent set already carries the information content of the data.

3.2 PHASE II: NEAREST-NEIGHBOUR LATENT PROJECTION

Let $B = \{x_i\}_{i=1}^n$ be a mini-batch, $z_i = E(x_i)$, and $Z_B = \{z_1, ..., z_n\}$ the batch's clean codes, which we treat as an empirical sample of $\mathcal{M}$. We perturb

$$\bar{z}_i = z_i + \sigma \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I_d) \tag{4}$$

and define the empirical projection of $\bar{z}_i$ as its nearest clean code,

$$z_i^{proj} = \hat{\Pi}_{\mathcal{M}}(\bar{z}_i) = \arg\min_{z \in Z_B} \|z - \bar{z}_i\|_2 \tag{5}$$

We then decode and re-encode the perturbed code and pull the result toward the projection:

$$\mathcal{L}_{proj} = \frac{1}{n} \sum_{i=1}^{n} \|E(D(\bar{z}_i)) - \mathrm{sg}[z_i^{proj}]\|_2^2 \tag{6}$$

where sg[.] is a stop-gradient. Without it, gradients flow through z_i^{proj} into E, which can reduce $\mathcal{L}_{\mathrm{proj}}$ by pulling all z_i together, collapsing Z_B to a point rather than learning a projection. The full Phase-II objective is:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{proj}, \quad \lambda = 1 \tag{7}$$

The Phase-II algorithm is provided as Algorithm 1 in Appendix A.

Why nearest neighbour and not the original code? Replacing z_i^{proj} with z_i recovers a latent-space denoising objective, which asks the model to undo the perturbation, which pulls every trajectory back to its start and makes traversal impossible by construction. Projection to the nearest code instead asks the model to keep the displacement while snapping it onto the data manifold. The two coincide in the small-noise regime, when σ is below the typical nearest-neighbour distance in Z_B, the $\arg\min$ in equation 5 usually returns z_i itself, and separate as σ grows, which is exactly the regime where traversal happens. A similar intuition is also seen in latent quantization in VQ-VAE Van Den Oord et al. (2017), but there the code-book is learned and finite, whereas ours is the batch itself and is re-drawn every step, making it a nonparametric plug-in estimator of $\Pi_{\mathcal{M}}$.

Interpretation: $E \circ D$ as a retraction. At the optimum of equation 6, $E \circ D$ maps a neighbourhood of $\mathcal{M}$ back onto $\mathcal{M}$ while approximately fixing $\mathcal{M}$ point-wise, i.e.

$$(E \circ D)(z) \approx \Pi_{\mathcal{M}}(z) \text{ for } d(z, \mathcal{M}) \leq \sigma, \quad \text{hence} \quad (E \circ D)^2 \approx (E \circ D) \tag{8}$$

an idempotency condition of the kind targeted by Shocher et al. (2024); Alharthi et al. (2026).

This closes the argument for why Phase-II training is well-behaved. The objective being optimized is a faithful proxy for the population-level projection loss. The remaining question is a separate one. Granting a well-trained $E \circ D$, does repeatedly applying it at inference time stay stable, or can small per-step errors compound into unbounded drift over a long walk? Sec. 4 answers this, showing the walk is stable at inference.

3.3 Inference: Latent Traversal

Traversal iterates a small random step followed by the learned projection:

$$z_{t+1} = E(D(z_t + \eta u_t)), \quad x_{t+1} = D(z_{t+1}), \tag{9}$$

with $z_0 = E(x)$ and u_t a unit direction. This projection step is what prevents the accumulated drift that ordinarily makes a random walk in the latent space degenerate.

The full training algorithm, the framework diagram, the noise-calibration procedure, and the geometric-intuition diagram for why the nearest-neighbour target respects the manifold's curvature are detailed in Appendix A.

4 Theoretical Guarantees

Sec. 3 defined the training objective and the traversal recursion built on it; this section proves both are well-designed. Sec. 4.1 shows the training objective is sound, and Sec. 4.2 shows the resulting walk is stable at inference.

4.1 Soundness of the Projection Objective

Lemma 1 (Finite-sample distortion of batch projection)**.** *Let $\mathcal{M} = S^1$ and $Z_n = \{z_1, \ldots, z_n\}$ i.i.d uniform on $\mathcal{M}$. For any $\delta \in (0,1)$, with probability $\geq 1 - \delta$ over the draw of Z_n, for every radius $r > 0$,*

$$\sup_{z : d(z,\mathcal{M}) \leq r} \|\hat{\Pi}_{\mathcal{M}}(z) - \Pi_{\mathcal{M}}\|_2 \leq \ell_n := \frac{C(\log n + \log(1/\delta))}{n}$$

for an absolute constant C.

This lemma is proved for $\mathcal{M} = S^1$ ($k = 1$). For general k, covering numbers degrade the rate from $O(1/n)$ to $O(1/n^{1/k})$, so we make no claim there beyond the qualitative behavior it supports, which the experiments below support past this case.

The projection loss is only useful if it avoids the degenerate solutions a simpler denoising target admits. Denoising has a degenerate global optimum that kills the traversal walk outright (Proposition 2), while the nearest-neighbor target provably does not (Proposition 3).

Proposition 2 (Degeneracy of the denoising target)**.** *Suppose E, D have enough capacity to represent any measurable map on a neighborhood of $\mathcal{M}$. For $\mathcal{L}_{den}(\theta) = \mathbb{E}[\|E \circ D(z + \epsilon) - z\|_2^2]$, the map $g(z, \epsilon) = z$ is realizable and achieves $\mathcal{L}_{den}(\theta) = 0$, the global minimum. Under this minimizer, the traversal recursion $z_{t+1} = (E \circ D)(z_t + \eta u_t)$ satisfies $z_{t+1} = z_t$, resulting in zero displacement.*

Proposition 2 shows the failure mode is not a training artifact but the global optimum of the denoising objective itself, so no amount of extra training fixes it. We now show the projection target used in $\mathcal{L}_{\text{proj}}$ escapes this failure mode with constant probability.

Proposition 3 (Non-degeneracy of the nearest-neighbor target)**.** *Fix i and, without loss of generality, place z_i at angle 0 on $\mathcal{M} = S^1$. Let $L_i, R_i > 0$ denote the gap from z_i to its nearest sampled neighbor counterclockwise and clockwise. Let $\bar{z}_i = z_i + \sigma\epsilon_i (\mod 2\pi), \epsilon_i \sim \mathcal{N}(0,1)$, independent of the sample locations, and let $z_i^{\text{proj}} = \hat{\Pi}_{\mathcal{M}}(\bar{z}_i)$ as in Equation 5. Set $\sigma = c\pi/n$ for a constant $c > 0$. Then*

$$\mathbb{P}(z_i^{\text{proj}} \neq z_i) \geq \Phi(2/c)$$

where $\Phi(x) = \mathbb{P}(\epsilon \geq x)$ for $\epsilon \sim \mathcal{N}(0,1)$. In particular, $\Phi(2/c) \to \frac{1}{2}$ as $c \to \infty$.

Having established that the nearest-neighbor target is both well-estimated from a batch (Lemma 1) and non-degenerate (Propositions 2 and 3), we put the two together. Does training against the finite-batch estimator $\hat{\Pi}_{\mathcal{M}}$ actually control the loss we care about, the one defined against the true projection $\Pi_{\mathcal{M}}$? Theorem 4 answers yes, up to two error sources, the finite-sample distortion from Lemma 1 and a fixed floor from the probability mass of the perturbation that escapes the σ-band entirely, where the batch estimator gives no guarantee.

Theorem 4 (Empirical objective controls the population objective)**.** *Let $\mathcal{M} = S^1$ with uniform density ν. Define*

$$\mathcal{L}^\star(\theta) = \mathbb{E}_{z\sim v,\epsilon}[\|(E \circ D)(z+\epsilon) - \Pi_{\mathcal{M}}(z+\epsilon)\|_2^2],$$

$$\mathcal{L}_n(\theta) = \mathbb{E}_{z\sim v,\epsilon}[\|(E \circ D)(z+\epsilon) - \hat{\Pi}_{\mathcal{M}}(z+\epsilon)\|_2^2]$$

with $\epsilon \sim N(0, \sigma^2 I)$, and suppose $E\circ D$ has outputs confined to a region of diameter D for every input in $\mathbb{R}^d$, not merely near $\mathcal{M}$ (this holds whenever D has a bounded output). Let $p_{tail} := \mathbb{P}(\|\epsilon\| > \sigma)$. Then with probability $\geq 1-\delta$ over the batch draw,

$$|L_n(\theta) - L^\star(\theta)| \;\leq\; 2D \cdot \frac{C(\log n + \log(1/\delta))}{n} \;+\; 2D^2\, p_{\text{tail}} \qquad \textit{for all } \theta.$$

The bound does not vanish as $n \to \infty$, as the term $2D^2 p_{\text{tail}}$ is a fixed floor set by how much Gaussian mass escapes the σ-band, independent of n. With the training objective's soundness established this way, we turn to whether repeatedly applying the trained composition at inference time is stable.

4.2 Stability of the Traversal Walk

Measuring how much a single such step can move a point requires a way to quantify how sharply $\mathcal{M}$ curves at the point it projects onto.

Definition 2 (Local concavity coefficient)**.** For a closed set $C \subset \mathbb{R}^d$ with ℓ_2-projection Π_C, the local concavity coefficient $\gamma_x(C)$ at $x \in C$ is the smallest γ such that for all $y \in C$ and all z with $\Pi_C(z) = x$

$$\langle y - x, z - x\rangle \leq \gamma \|z - x\|_2 \|y - x\|_2^2$$

Equivalently, for any z with $\Pi_C(z) = x$ and any $y \in C$,

$$(1 - \gamma_x(C)\|z - x\|_2) \cdot \|x - y\|_2 \leq \|z - y\|_2$$

We write $\gamma_{\max} = \sup_x \gamma_x(\mathcal{M})$ over the region of interest.

Theorem 5 (Barber & Ha (2018), Lemma 11, specialized to $n = 2$, $r = 1$)**.** *Barber & Ha (2018)'s Lemma 11 gives $\gamma_X(C) = \frac{1}{2}$ for $C = \{X \in \mathbb{R}^{n\times r} : X^\top X = I_r\}$. At $n = 2$, $r = 1$, C is S^1, so*

$$\gamma_x(S^1) = \frac{1}{2} \qquad \textit{for every } x \in S^1,$$

$\gamma_{\max} = \frac{1}{2}$ for the manifold used throughout this document. We omit the proof, which is given in Barber & Ha (2018).

$\gamma_x(\mathcal{M})$ is largest exactly where $\mathcal{M}$ curves most sharply, so the traversal step size η is effectively bottlenecked by the most curved region of $\mathcal{M}$, not its average curvature. The following bound makes this precise for a single step.

Theorem 6 (One-step displacement bound under exact projection)**.** *Let $\gamma_{\max} = \sup_x \gamma_x(\mathcal{M})$ over the relevant region, $z \in \mathcal{M}$, u a unit vector, and η with $\gamma_{\max}\eta < 1$. Then the exact projection $x' = \Pi_{\mathcal{M}}(z + \eta u)$ satisfies*

$$\|x' - z\| \leq \frac{\eta}{1 - \gamma_{\max}\eta}.$$

For small η, using $\frac{1}{1-x} = 1 + x + x^2 + \dots$ with $x = \gamma_{\max}\eta$, the right-hand side above becomes $\eta + \gamma_{\max}\eta^2 + O(\eta^3)$, so the projected displacement is the step size plus a curvature-driven second-order correction that vanishes as $\mathcal{M}$ flattens. This bounds how far a single point can move under projection. The walk of Eq. 9 instead needs to compare two different points evolving under related perturbations, which the following two-point form gives directly.

Lemma 7 (Two-point contraction, Barber & Ha (2018), Lemma 4)**.** *For any $z, w \in \mathbb{R}^d$, writing $x = \Pi_{\mathcal{M}}(z)$ and $y = \Pi_{\mathcal{M}}(w)$,*

$$\Big(1 - \gamma_x(\mathcal{M})\|z - x\|_2 - \gamma_y(\mathcal{M})\|w - y\|_2\Big) \cdot \|x - y\|_2 \le \|z - w\|_2.$$

The proof follows directly from Lemma 4 of Barber & Ha (2018).

Specializing to z, w each within η of $\mathcal{M}$ (so $\|z - x\|_2, \|w - y\|_2 \le \eta$) and using $\gamma_x(\mathcal{M}), \gamma_y(\mathcal{M}) \le \gamma_{\max}$,

$$\big(1 - 2\gamma_{\max}\eta\big) \cdot \|x - y\|_2 \le \|z - w\|_2,$$

so $\Pi_{\mathcal{M}}$ is L_2-Lipschitz on this region with $L_2 = 1/(1 - 2\gamma_{\max}\eta)$, given that $2\gamma_{\max}\eta < 1$.

Theorem 6 bounds the displacement of a single perturb-and-project step. The next corollary extend this one-step bound to a full T-step trajectory for the finite-sample estimator actually used in training.

Corollary 8 (T-step drift, finite-sample estimator)**.** *Consider the iteration $z_{t+1} = \hat{\Pi}_{\mathcal{M}}(z_t + \eta u_t)$ using the finite-sample estimator, with $z_0 \in Z_n \subset \mathcal{M}$, and the shadow trajectory $z_t^\star = \Pi_{\mathcal{M}}(z_{t-1}^\star + \eta u_{t-1})$ using the exact projection with the same noise sequence, $z_0^\star = z_0$. Let $e_t = \|z_t - z_t^\star\|$ and $L_2 = 1/(1 - 2\gamma_{\max}\eta)$, with $2\gamma_{\max}\eta < 1$. Then, for all $t \ge 0$,*

$$e_t \le \ell_n \cdot \frac{1 - L_2^t}{1 - L_2}.$$

5 Experiments

We evaluate different training variants on two datasets. All autoencoder architectures are similar per dataset and differ only in the training objective, so differences in traversal behavior are attributable to the objective rather than to capacity. All evaluation is on an independent held-out split.

Datasets. *Synthetic:* a two dimensional swiss roll ($k = 2$) embedded in $\mathbb{R}^{50}$ by a fixed random smooth MLP, with additive Gaussian noise. Latent width $N \in \{2, 4, 8\}$. Since the generating manifold is known in closed form, off-manifold distance $d(x_t, \mathcal{X})$ can be measured exactly. *Colored MNIST:* digits colored with one of six fixed colors with additive noise, encoded by a convolutional autoencoder with $N \in \{16, 32, 64\}$, where the true intrinsic dimension is unknown.

Variants. $\mathcal{L}_{rec}$ is the uncorrected baseline, with plain reconstruction error and a walk with no on-manifold correction. $\mathcal{L}_{rec} + \mathcal{L}_{den}$ fine-tunes E, D to denoise a perturbed code back to its own origin, the ablation that isolates our contribution, since it differs from ours in the target of a denoising objective. $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ is our method, with the nearest code in the batch as the target, acting as an approximate projection. $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$ is a latent cycle consistency loss, proposed in Alharthi et al. (2026). All corrected variants use the same walk mechanism at inference: step, decode, re-encode.



Figure 3: Targeted walk on synthetic dataset using $\mathcal{L}_{rec} + \mathcal{L}_{prj}$.

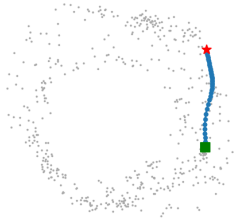


Figure 4: Undirected walk on synthetic dataset using $\mathcal{L}_{rec} + \mathcal{L}_{prj}$.

5.1 Evaluation Modes

We evaluate two complementary modes, both starting from a real held-out code $z_0 = E(x)$ and applying Eq. 9 iteratively. *Undirected walk:* step in a random unit direction in $\mathbb{R}^d$, with no destination. This directly tests Def. 1, does the trajectory stay near the manifold when pushed in an arbitrary direction with nothing guiding it? *Targeted walk:* step toward a second real held-out code z_{target}, re-aiming at the target after every step. This tests a strictly harder question, does the corrected walk reach a specific state while having a meaningful trajectory?

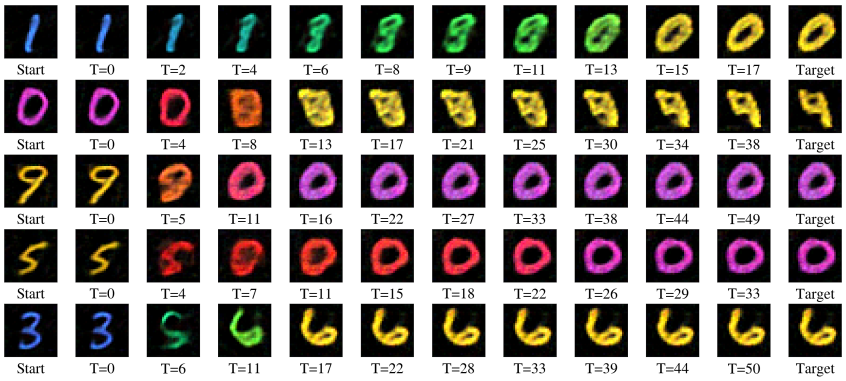


Figure 5: Targeted walks using $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ between 5 different held-out start-target pairs, chosen to differ in both color and digit identity.

5.2 RESULTS

Synthetic data: trajectory geometry Figure 3 shows the latent trajectory under the targeted walk for $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{proj}}$. The trajectory follows the manifold geometry toward the target, staying on-manifold throughout. In contrast, $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cyc}}$ leave the manifold within a few steps, drifting into regions with no data support, while $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{den}}$ stays on-manifold but resists displacement (full results in Figure 8). The same pattern holds for undirected walks. Figure 4 shows $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{proj}}$ tracing the manifold's curvature with no target to steer toward (all variants in Figure 9), confirming that the non-compounding behavior comes from the learned projection, not from the guidance of a target.

Quantitative: Off-Manifod Distance Table 1 reports exact distance to the true manifold at each traversal step. $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cyc}}$ compound drift linearly, while both $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{den}}$ and $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{proj}}$ plateau, empirical evidence of the retraction in Eq. 8. The denoising variant plateaus lower ($\approx$1.2 vs. $\approx$2.9) but at the cost of resisting displacement entirely, as the visual results below make clear (expanded analysis in Appendix C.3).

Variant	t=0	t=20	t=40
$\mathcal{L}_{rec}$	0.706	14.699	33.464
$\mathcal{L}_{rec}$ cyclic	0.699	12.706	175.796
$\mathcal{L}_{rec} + \mathcal{L}_{den}$	0.674	**1.204**	**1.211**
$\mathcal{L}_{rec} + \mathcal{L}_{prj}$	0.604	2.851	2.964
$\mathcal{L}_{rec} + \mathcal{L}_{cyc}$	0.747	16.608	41.599

Table 1: Distance to the true manifold $d(x_t, \mathcal{X})$ ($\downarrow$) at traversal step t on synthetic data.

Colored MNIST: Visual Quality. The targeted walk using $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ produces recognizable, smoothly varying digits, with visible changes in color, style, and digit between consecutive frames (full variant comparison for targeted walks in Figure 10, undirected walks in Figure 11). $\mathcal{L}_{rec}$ collapses into unstructured patterns within the first few steps in both modes. $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$ arrives at the destination with blurry intermediate steps in the first mode, but results in incoherent images in the undirected walk. $\mathcal{L}_{rec} + \mathcal{L}_{den}$ stays relatively coherent, but produces nearly identical frames, showing resistance to displacement. Notice that in the undirected walk setting, $\mathcal{L}_{rec} + \mathcal{L}_{proj}$, which has no target to steer toward, stays equally coherent while successfully traversing, confirming the stability is a property of the learned projection.

Color and digit shape interpolate together. Figure 5 shows targeted walks between 5 different start-target pairs using $\mathcal{L}_{rec} + \mathcal{L}_{proj}$. Two patterns hold across all five. The color at each step moves along a single consistent direction around the hue wheel toward the target color, rather than jumping directly or drifting through unrelated hues. The digit shape passes through intermediate forms that share structure with both endpoints rather than through unrelated digits, pairs that are already both loop digits need comparatively little shape change. Neither color nor digit identity is supervised anywhere in training, so this joint, monotone interpolation is a property the model discovered under $\mathcal{L}_{proj}$ alone.

6 TOWARD LARGE SCALE

Equation 9 leaves the choice of unit direction u_t open. A natural instantiation is the Euclidean direction toward a target, $u_t \propto z_{\text{target}} - z_t$, recomputed at every step, but this is a heuristic, not a learned quantity, and a straight line in $\mathbb{R}^d$ has no particular reason to track $\mathcal{M}$'s geometry once $\mathcal{M}$ has appreciable curvature. We describe a learned alternative, and two mechanisms that make training it tractable at larger scale. Appendix A gives the corresponding details and algorithms.

Learned direction. We replace Equation 9's rule with

$$\begin{aligned} &\pi_\theta : \mathbb{R}^d \times \mathbb{R}^d \times [0,1] \times [0,\infty) \;\to\; \mathbb{S}^{d-1} \times [0,1], \\ &\qquad \pi_\theta\big(z_t, z_{target}, t/T_{\max}, \|z_t - z_{target}\|_2/m\big) = (d_t, p_t), \qquad u_t = p_t \cdot m \cdot d_t \end{aligned} \tag{10}$$

with $m = f \cdot \text{median}_{z_i, z_j} \|z_i - z_j\|_2$ calibrated per dataset for the same reason as σ (Appendix A), since median pairwise distance, unlike a covariance-based statistic, neither scales pathologically with d nor assumes $\mathcal{M}$'s intrinsic dimension equals d. Whether $p_t \cdot m$ carries z_t past z_{target} is a property of what π_θ learns, governed entirely by the training objective. Training against the full roll-out rather than the first-arrival step, since an arrival-only loss scores nothing past the first crossing of the target and so bounds nothing about behavior afterward.

Memory bank. Theorem 4's finite-sample term tightens as n grows, but a training batch at resolution $\rho \times \rho$ costs $\Theta(n\rho^2 c)$ in memory, so a fixed budget bounds $n = O(\rho^{-2})$, shrinking exactly the quantity the theorem wants larger. A $\text{Bank} = \{z_1, \ldots, z_n\}$, $n \gg |B|$, computed under no gradient and refreshed periodically, supplies a fresh random subsample $S_t \subset \text{Bank}$, $|S_t| = s \leq n$, at each projection step in place of Equation 9's batch search (Algorithm 2). Lemma 1 applies unchanged with $n = s$, so a single projection's accuracy is governed by s, not $|\text{Bank}|$, and training batch size is chosen purely by the memory bound above, independent of s.

Hierarchical training. Training π_θ over a full roll-out of length T costs $\Theta(T \cdot A)$, where A is the per-step activation cost of $E \circ D$, growing large for a distant target. This is bounded by training on short sub-trajectories using real points from the bank along a shortest path between any sampled (z_0, z_{target}) (Algorithm 3), with π_θ's target input remaining z_{target} throughout while the loss is computed locally, per edge, with the gradient detached at each point.

6.1 EMPIRICAL VALIDATION

We validate the mechanisms of this section against the original one from Sec. 3, which we call unscaled, as latent width N, trajectory length $T_{\max}$, and pool size n grow together, on the same synthetic swiss-roll and Colored MNIST datasets and evaluation protocol as Sec. 5. At the largest Colored MNIST scale we test, the unscaled mechanism is not merely slower, a feasibility probe projects it would exceed a fixed wall-clock budget and the run is not attempted, while the scaled mechanism completes and produces a working walk. This is the central claim of this section. The unscaled mechanism does not just get more expensive with scale, it stops being attemptable, while the scaled mechanism keeps running, using substantially less peak memory throughout and, past the smallest scale we test, less wall-clock time as well.

Figure 14 shows the scaled mechanism on Colored MNIST at the larger scale cell, reaching crisp, recognizable targets with color and digit shape interpolating smoothly, the same qualitative behavior as Sec. 5, now at a scale where the unscaled mechanism cannot run at all. Quality is comparable to the unscaled mechanism wherever both can be measured, and in some cases better, so the scaled mechanism is not trading quality for feasibility. Appendix E reports the full comparison at every scale tested, including the synthetic dataset, a discussion of how quality changes with scale, and additional qualitative examples.

7 RELATED WORK

Structure of the latent space. The latent space induced by Eq. 1 has become an object of study in its own, motivating work on disentanglement Higgins et al. (2017); Locatello et al. (2019), identifiability

under auxiliary variables Khemakhem et al. (2020) and multiple distributions Zhang et al. (2024), latent-space steering of auto-regressive models Li et al. (2023), and conditional diffusion in learned latent spaces Rombach et al. (2022).

Editing as a counterfactual question. One particular use of this latent space is editing. Generative modeling is usually split into unconditional generation, which maps a sample from a simple base distribution to an image, and conditional generation, which additionally conditions on side information, such as a discrete attribute (e.g. eye color) or an open ended text description Rombach et al. (2022). Editing instead asks a counterfactual question about an observed sample, given this photograph, how would it look had the person been ten years younger? The question spans causal inference, representation learning, identifiability, and generative modeling Pearl et al. (2003); Schölkopf et al. (2021); Pan & Bareinboim (2024).

Supervised approaches to editing. In a fully unsupervised setting it is not hard but ill-posed. Impossibility results show that without additional information beyond the observations themselves, such as labels, auxiliary variables, heterogeneous environments, or temporal structure, the latent factors are not identifiable up to anything stronger than an arbitrary invertible mixing Hyvärinen & Pajunen (1999); Locatello et al. (2019). If the factors cannot be separated, one cannot know which coordinate to move to realize a specific edit. Methods that do achieve controllable editing accordingly import supervision, such as image sets with domain labels and a cycle-consistency objective Zhu et al. (2017a), attribute classifiers used to find directions in a pretrained GAN latent space Shen et al. (2020); Härkönen et al. (2020), or instruction supervision Brooks et al. (2023).

Repeated self-application. Idempotent generative networks train f to satisfy $f(f(z)) = f(z)$, so repeated application converges to a fixed point on the data manifold Shocher et al. (2024), and Alharthi et al. (2026) align the encoder and decoder manifolds under the same repeated map. Closest to our perspective, Fumero et al. (2026) show that iterating the encode-decode map of an already-trained autoencoder implicitly defines a vector field on the latent space, and study its attractors. All three treat $E \circ D$ or f as a dynamical system and ask where repeated application lands. We train Phase II so this attractor set coincides with $\mathcal{M}$ itself, with formal guarantees for when it does (Sec. 4). A structurally similar loss, encode a decoded or generated code and penalize its discrepancy from the original, also appears elsewhere for mode diversity Zhu et al. (2017b); Huang et al. (2018); Lee et al. (2018), disentanglement Chen et al. (2016), adversarial joint matching Donahue et al. (2017); Dumoulin et al. (2017), and diffusion self-consistency Luo et al. (2023), none with our retraction guarantees.

Where we differ. Our task is orthogonal to the identifiability obstruction rather than blocked by it. Identifiability results constrain what a latent direction means, they do not constrain whether a perturbed code decodes to a valid sample. The first is a question about semantics and requires supervision, while the second is a question about the geometry of the learned latent set, and is answerable without labels. We discuss what this task does and does not claim relative to interventional editing and Pearl's causal hierarchy in Appendix F.

8 Conclusion

We asked whether a latent code can be perturbed repeatedly and still decode to something real, without labels. The property is not implied by reconstruction quality and is not blocked by identifiability results, which constrain what a latent direction means rather than whether a perturbed code decodes to a valid sample. Training $E \circ D$ to project onto the nearest observed code, rather than back onto the perturbed point's own origin, is enough to obtain walks that stay coherent over tens of steps where an uncorrected baseline collapses. What the correction provides is stability under displacement, not control over what changes, and the gap between those is where we expect interesting work to be. The batch nearest neighbor is a coarse estimate of $\Pi_{\mathcal{M}}$ and degrades when the batch covers $\mathcal{M}$ sparsely and results are on small-scale data. In addition, scaling to natural images is a natural next step.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and software development, including improving the clarity and presentation of the manuscript, developing evaluation and visualization scripts, and identifying relevant literature. We have not used generative AI tools to develop the research questions or ideas, design the experiments, implement the core method, or generate or alter experimental results. Other uses of generative AI are not applicable to this work.

All AI-assisted material was reviewed by the authors before inclusion. AI-assisted code was reviewed and tested by the authors for correctness, and literature identified with AI assistance was checked against the primary sources to verify the accuracy and relevance of the citations. All reported results and numerical values were obtained from experiments conducted by the authors. We take responsibility for the final content of this work, including all text, code, claims, and other artifacts produced with the assistance of generative AI.

REFERENCES

Dareen Alharthi, Abdul Waheed, and Bhiksha Raj. Encoder-decoder manifold alignment for idempotent generation. *arXiv preprint arXiv:2606.22304*, 2026.

Rina Foygel Barber and Wooseok Ha. Gradient descent with non-convex constraints: local concavity determines convergence. *Information and Inference: A Journal of the IMA*, 7(4):755–806, 12 2018. ISSN 2049-8764. doi: 10.1093/imaiai/iay002. URL https://doi.org/10.1093/imaiai/iay002.

Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and causal inference: the works of judea pearl*, pp. 507–556. 2022.

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

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017.

Marco Fumero, Luca Moschella, Emanuele Rodolà, and Francesco Locatello. Navigating the latent space dynamics of neural models. *arXiv preprint arXiv:2505.22785*, 2026.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020.

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

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

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.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

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

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Yushu Pan and Elias Bareinboim. Counterfactual image editing. *arXiv preprint arXiv:2403.09683*, 2024.

Judea Pearl et al. Causality: models, reasoning, and inference. *Econometric Theory*, 19(675-685):46, 2003.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9240–9249. IEEE, 2020.

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei Efros. Idempotent generative network. In *International Conference on Learning Representations*, volume 2024, pp. 35440–35460, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from multiple distributions: A general setting. *arXiv preprint arXiv:2402.05052*, 2024.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 2242–2251. Ieee, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, volume 30, 2017b.

APPENDIX

A Extended Method **12**

A.1 Training Objective and Design Choices 12

A.2 Geometric Intuition 13

B Proofs **13**

C Additional Results **16**

C.1 Targeted walk on synthetic data 16

C.2 Undirected walk on synthetic data 16

C.3 Extended Quantitative Analysis: Off-Manifold Distance 17

C.4 Targeted walk on Colored MNIST 17

C.5 Undirected walk on Colored MNIST 17

D Extended Scaling Methodology **18**

E Extended Scaling Results **19**

F Extended Discussion **20**

A Extended Method

A.1 Training Objective and Design Choices

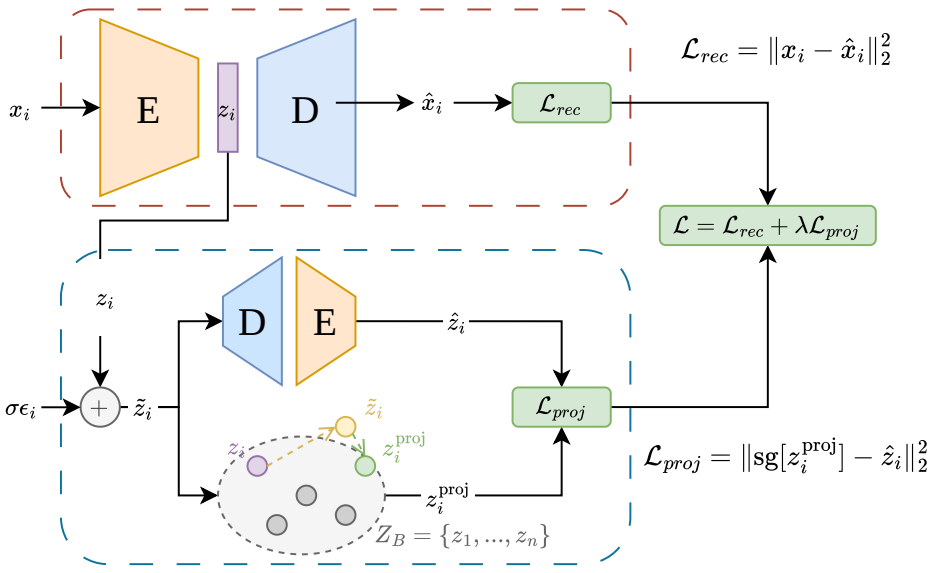


Figure 6: Training framework. Phase 1 (top): standard reconstruction autoencoder. Phase 2 (bottom): a code is perturbed, decoded, re-encoded, and pulled towards a target z_i^{proj} on the batch's own code set Z_B

Algorithm 1 Phase-II training step (nearest-neighbor latent projection)

Require: batch $\{x_i\}_{i=1}^n$, noise scale σ, weight λ

1: $z_i \leftarrow E(x_i), \quad \hat{x}_i \leftarrow D(z_i)$ ▷ clean pass
2: $\tilde{z}_i \leftarrow z_i + \sigma\varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, I)$
3: $z_i^{\text{proj}} \leftarrow \arg\min_{z \in \{z_1,\ldots,z_n\}} \|z - \tilde{z}_i\|_2^2$ ▷ batch nearest neighbor
4: $\hat{z} \leftarrow E(D(\tilde{z}_i))$ ▷ decode–encode
5: $\mathcal{L} \leftarrow \frac{1}{n}\sum_i \|x_i - \hat{x}\|_2^2 + \lambda \frac{1}{n}\sum_i \|\hat{z}_i - \text{sg}[z_i^{\text{proj}}]\|_2^2$
6: update θ, ϕ by $\nabla\mathcal{L}$

Figure 6 shows the full training objective. A standard autoencoder is trained to minimize the reconstruction loss. Phase 2 adds a projection loss, where each clean code $z_i = E(x_i)$ is perturbed to $\tilde{z}_i = z_i + \sigma\epsilon_i$, decoded and then re-encoded to $\hat{z}_i = E(D(\tilde{z}_i))$, and pulled towards a target on Z_B. The total loss is: $\mathcal{L}_{rec} + \lambda\mathcal{L}_{proj}$

Noise scale is calibrated, not fixed The perturbation scale in Eq. 4 has no reason to transfer between datasets of different intrinsic dimension and sample density. We calibrate it per $(dataset, N)$ cell by sweeping σ and selecting the value at which the snap rate (the fraction of perturbed codes whose nearest clean neighbor is a different point) falls in $15 - 30\%$.

Cost. The batch nearest-neighbour search is $\mathcal{O}(n^2 d)$ per step, in addition to the decode-encode pass, which raises the cost of training. For large n or d, Eq. 5 could be replaced by an approximate search.

A.2 Geometric Intuition

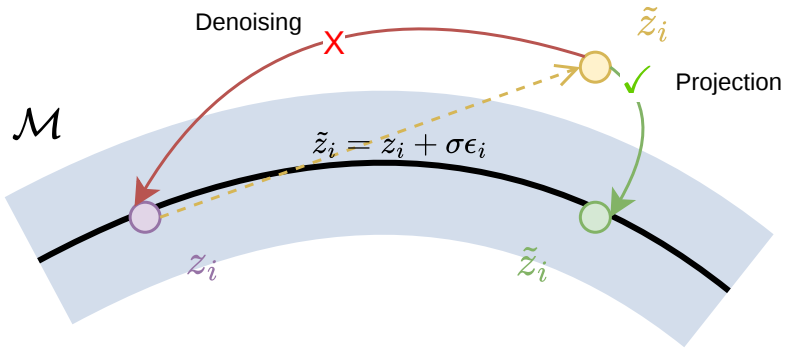


Figure 7: Self-denoising fights the manifold's curvature, while projection does not.

Figure 7 states the geometric reason the projection target is the nearest clean code rather than z_i itself. On a curved manifold, the nearest point to a perturbed $\tilde{z}_i$ is generally not z_i. Training towards z_i learns to invert the noise, not to project. Training towards the nearest code learns a projection onto the manifold. This is exactly the difference between $\mathcal{L}_{rec} + \mathcal{L}_{den}$ and $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ compared earlier.

B Proofs

Restatement of Lemma 1.

Lemma (Finite-sample distortion of batch projection)**.** *Let $\mathcal{M} = S^1$ and $Z_n = \{z_1, \ldots, z_n\}$ i.i.d uniform on $\mathcal{M}$. For any $\delta \in (0, 1)$, with probability $\geq 1 - \delta$ over the draw of Z_n, for every radius*

$r > 0$,

$$\sup_{z:d(z,\mathcal{M})\le r} \|\hat{\Pi}_{\mathcal{M}}(z) - \Pi_{\mathcal{M}}\|_2 \le \ell_n := \frac{C(\log n + \log(1/\delta))}{n}$$

for an absolute constant C.

Proof. Cover S^1 with $m = \lceil 2\pi/\ell \rceil$ disjoint arcs of length ℓ. For a fixed arc, $\mathbb{P}(\text{empty}) \le e^{-n\ell/2\pi}$ (from $(1-\ell/2\pi)^n \le e^{-n\ell/2\pi}$). Union bound over m arcs, $\mathbb{P}(\exists\text{empty arc}) \le \frac{2\pi}{\ell} e^{-n\ell/2\pi}$. Substituting $\ell = 2\pi(\log n + \log(1/\delta))/n$ gives $\frac{2\pi}{\ell} e^{-n\ell/2\pi} = \delta/(\log n + \log(1/\delta)) \le \delta$ using $\log n + \log(1/\delta) \ge 1$, which holds whenever $n/\delta \ge e$ (in particular for every $n \ge 2$ and every $\delta < 1$). So $\mathbb{P}(\exists\text{empty arc}) \le \delta$, i.e. with probability $\ge 1-\delta$ every arc of length ℓ contains a sample. For any z with $d(z,\mathcal{M}) \le r$, its true nearest point $x = \Pi_{\mathcal{M}}(z)$ lies in some such arc, which contains a sample point within ℓ of x, hence the batch nearest is within ℓ of x, for every such z. □

Restatement of Proposition 2.

Proposition (Degeneracy of the denoising target)**.** *Suppose E, D have enough capacity to represent any measurable map on a neighborhood of $\mathcal{M}$. For $\mathcal{L}_{den}(\theta) = \mathbb{E}[\|E \circ D(z+\epsilon) - z\|_2^2]$, the map $g(z,\epsilon) = z$ is realizable and achieves $\mathcal{L}_{den}(\theta) = 0$, the global minimum. Under this minimizer, the traversal recursion $z_{t+1} = (E \circ D)(z_t + \eta u_t)$ satisfies $z_{t+1} = z_t$, resulting in zero displacement.*

Proof. $\mathcal{L}_{den} \ge 0$ pointwise, g achieves 0 pointwise, hence is a global minimizer. Substituting into the recursion gives the claim directly. □

Restatement of Proposition 3.

Proposition (Non-degeneracy of the nearest-neighbor target)**.** *Fix i and, without loss of generality, place z_i at angle 0 on $\mathcal{M} = S^1$. Let $L_i, R_i > 0$ denote the gap from z_i to its nearest sampled neighbor counterclockwise and clockwise. Let $\bar{z}_i = z_i + \sigma\epsilon_i (\mod 2\pi), \epsilon_i \sim \mathcal{N}(0,1)$, independent of the sample locations, and let $z_i^{\text{proj}} = \hat{\Pi}_{\mathcal{M}}(\bar{z}_i)$ as in Equation 5. Set $\sigma = c\pi/n$ for a constant $c > 0$. Then*

$$\mathbb{P}(z_i^{\text{proj}} \ne z_i) \ge \Phi(2/c)$$

where $\Phi(x) = \mathbb{P}(\epsilon \ge x)$ for $\epsilon \sim \mathcal{N}(0,1)$. In particular, $\Phi(2/c) \to \frac{1}{2}$ as $c \to \infty$.

Proof. Local flip condition. If $0 \le \sigma\epsilon_i \le R_i$, then $z_i^{\text{proj}} = z_i$ iff $\sigma\epsilon_i < R_i/2$, the distance from $\bar{z}_i$ to z_i is $\sigma\epsilon_i$ and to the clockwise neighbor is $R_i - \sigma\epsilon_i$, and z_i is closer iff $\sigma\epsilon_i < R_i\sigma\epsilon_i$. If $\sigma\epsilon_i > R_i$, $\bar{z}_i$ has passed the neighbor, so $z_i^{\text{proj}} \ne z_i$. The mirror statement holds for $\epsilon < 0$ with L_i in place of R_i. Since the two directions are disjoint events,

$$\{z_i^{\text{proj}} \ne z_i\} \supseteq \{\epsilon \ge R_i/(2\sigma)\} \sqcup \{\epsilon \le -L_i/(2\sigma)\}.$$

Gap means. The n gaps between n i.i.d. uniform points on the circle sum to 2π and are exchangeable, so $\mathbb{E}[R_i] = \mathbb{E}[L_i] = 2\pi/n$.

Bounding one side. Since $\epsilon_i \perp R_i$, condition on R_i, conditioning on R_i $\mathbb{P}(\epsilon_i \ge R_i/(2\sigma)) = \mathbb{E}[\Phi(R_i/(2\sigma))] \ge \mathbb{E}[\bar{\Phi}(R_i/(2\sigma)) \cdot \mathbf{1}\{R_i < 4\pi/n\}]$. By Markov's inequality, $\mathbb{P}(R_i \ge 4\pi/n) \le \frac{\mathbb{E}[R_i]}{4\pi/n} = \frac{1}{2}$. On $\{R_i < 4\pi/n\}$, $R_i/(2\sigma) < 2/c$, and since Φ is strictly decreasing, $\Phi(R_i/(2\sigma)) > \Phi(2/c)$ there. Hence

$$\mathbb{P}(\epsilon_i \ge R_i/(2\sigma)) \ge \frac{1}{2}\Phi(2/c),$$

and by the identical argument on the other side, $\mathbb{P}(\epsilon_i \le -L_i/(2\sigma)) \ge \frac{1}{2}\Phi(2/c)$

Combine. The two events are disjoint, so their probabilities add

$$\mathbb{P}(z_i^{\text{proj}} \ne z_i) \ge \frac{1}{2}\Phi(2/c) + \frac{1}{2}\Phi(2/c) = \Phi(2/c)$$

Consequently, any global minimizer of $\mathcal{L}_{\text{proj}}$ satisfies $(E \circ D)(\bar{z}_i) = z_i^{\text{proj}}$ pointwise (by the same zero-loss argument as Proposition 2). On the $\Phi(2/c)$-probability event where $z_i^{\text{proj}} \neq z_i$, this forces disagreement with the degenerate identity solution of Proposition 2. □

Restatement of Theorem 4.

Theorem (Empirical objective controls the population objective)**.** *Let $\mathcal{M} = S^1$ with uniform density ν. Define*

$$\mathcal{L}^\star(\theta) = \mathbb{E}_{z\sim v,\epsilon}[\|(E \circ D)(z+\epsilon) - \Pi_{\mathcal{M}}(z+\epsilon)\|_2^2],$$

$$\mathcal{L}_n(\theta) = \mathbb{E}_{z\sim v,\epsilon}[\|(E \circ D)(z+\epsilon) - \hat{\Pi}_{\mathcal{M}}(z+\epsilon)\|_2^2]$$

with $\epsilon \sim N(0, \sigma^2, I)$, and suppose $E \circ D$ has outputs confined to a region of diameter D for every input in $\mathbb{R}^d$, not merely near $\mathcal{M}$ (this holds whenever D has a bounded output). Let $p_{tail} := \mathbb{P}(\|\epsilon\| > \sigma)$. Then with probability $\geq 1-\delta$ over the batch draw,

$$|L_n(\theta) - L^\star(\theta)| \;\leq\; 2D \cdot \frac{C(\log n + \log(1/\delta))}{n} \;+\; 2D^2\, p_{\text{tail}} \qquad \textit{for all } \theta.$$

Proof. Let $a = (E \circ D)(z+\varepsilon)$, $b = \Pi_{\mathcal{M}}(z+\varepsilon)$, $c = \hat{\Pi}_{\mathcal{M}.n}(z+\varepsilon)$,

$$\begin{aligned}\big|\|a-c\|^2 - \|a-b\|^2\big| &= \big|\|a-c\| - \|a-b\|\big| \cdot \big(\|a-c\| + \|a-b\|\big) \\ &\leq \|b-c\|_2 \cdot \big(\|a-c\| + \|a-b\|\big).\end{aligned}$$

Case $\|\epsilon\| \leq \sigma$: Then $d(z+\epsilon, \mathcal{M}) \leq \sigma$, so Lemma 1 applies (with radius σ), with probability $\geq 1-\delta$ over the batch draw, $\|b-c\|_2 \leq \ell_n = C(\log n + \log(1/\delta))/n$ uniformly over all such $z+\epsilon$. Combined with $\|a-c\| + \|a-b\| \leq 2D$, $\big|\|a-c\|^2 - \|a-b\|^2\big| \leq 2D\ell_n$ on this case.

Case $\|\epsilon\| > \sigma$: Lemma 1 does not apply, but b, c both lie in the diameter-D region, so $\|b-c\|_2 \leq D$ unconditionally, giving $\big|\|a-c\|^2 - \|a-b\|^2\big| \leq D \cdot 2D = 2D^2$.

Taking expectation over (z, ϵ) and splitting by the two cases,

$$|L_n(\theta) - L^\star(\theta)| \;\leq\; 2D\ell_n \cdot \mathbb{P}(\|\varepsilon\| \leq \sigma) + 2D^2 \cdot \mathbb{P}(\|\varepsilon\| > \sigma) \;\leq\; 2D\ell_n + 2D^2 p_{\text{tail}}.$$

The max-gap event from Lemma 1 does not depend on θ, so this bound holds uniformly over θ on that event, which has probability $\geq 1-\delta$. □

Restatement of Theorem 6.

Theorem (One-step displacement bound under exact projection)**.** *Let $\gamma_{\max} = \sup_x \gamma_x(\mathcal{M})$ over the relevant region, $z \in \mathcal{M}$, u a unit vector, and η with $\gamma_{\max}\eta < 1$. Then the exact projection $x' = \Pi_{\mathcal{M}}(z+\eta u)$ satisfies*

$$\|x' - z\| \leq \frac{\eta}{1-\gamma_{\max}\eta}.$$

Proof. Apply the second inequality from Definition 2 with $z' = z + \eta u$ in the role of "z", $x' = \Pi_{\mathcal{M}}(z')$ in the role of "x", and $y = z$ (valid since $z \in \mathcal{M}$):

$$\big(1 - \gamma_{x'}(\mathcal{M})\|z' - x'\|_2\big) \cdot \|x' - z\|_2 \leq \|z' - z\|_2 = \eta.$$

Since x' is the closest point of $\mathcal{M}$ to z' and $z \in \mathcal{M}$ is one particular point at distance η from z', $\|z' - x'\|_2 \leq \eta$. Since $\gamma_{x'}(\mathcal{M}) \geq 0$ and $\|z' - x'\|_2 \leq \eta$, $\gamma_{x'}(\mathcal{M})\|z' - x'\|_2 \leq \gamma_{x'}(\mathcal{M})\eta$, so

$$\big(1 - \gamma_{x'}(\mathcal{M})\|z' - x'\|_2\big) \cdot \|x' - z\|_2 \leq \|z' - z\|_2 = \eta.$$

Since $\gamma_{x'}(\mathcal{M}) \leq \gamma_{\max}$, the same monotonicity gives $(1 - \gamma_{\max}\eta)\|x' - z\| \leq \eta$. Since $\gamma_{\max}\eta < 1$, dividing gives the claim. □

Restatement of Corollary 8.

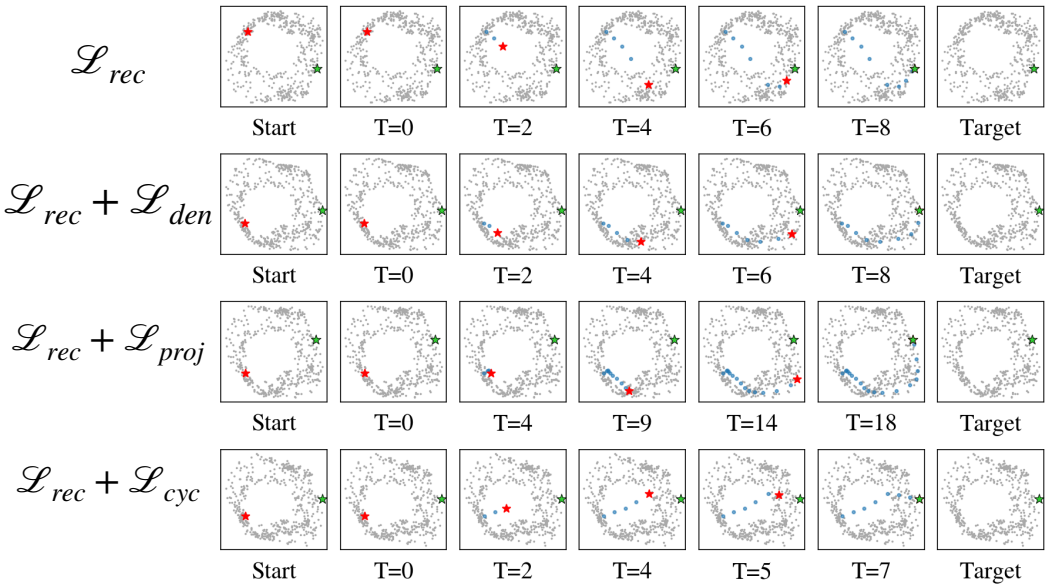


Figure 8: Targeted walk on synthetic dataset. Red: current position. Green: target. Blue dots: walk.

Corollary (T-step drift, finite-sample estimator)**.** *Consider the iteration $z_{t+1} = \hat{\Pi}_{\mathcal{M}}(z_t + \eta u_t)$ using the finite-sample estimator, with $z_0 \in Z_n \subset \mathcal{M}$, and the shadow trajectory $z_t^\star = \Pi_{\mathcal{M}}(z_{t-1}^\star + \eta u_{t-1})$ using the exact projection with the same noise sequence, $z_0^\star = z_0$. Let $e_t = \|z_t - z_t^\star\|$ and $L_2 = 1/(1 - 2\gamma_{\max}\eta)$, with $2\gamma_{\max}\eta < 1$. Then, for all $t \geq 0$,*

$$e_t \leq \ell_n \cdot \frac{1 - L_2^t}{1 - L_2}.$$

Proof. Let $w = \Pi_{\mathcal{M}}(z_t + \eta u_t)$. By triangle inequality,

$$e_{t+1} = \|z_{t+1} - z_{t+1}^\star\|_2 \leq \|z_{t+1} - w\|_2 + \|w - z_{t+1}^\star\|_2.$$

Since $z_t \in \mathcal{M}, d(z_t + \eta u_t, \mathcal{M}) \leq \eta$, so Lemma 1 gives $\|z_{t+1} - w\|_2 \leq \ell_n$ with the stated batch-draw confidence. For the second term, both w and $z_{t+1}^\star$ are the exact projection of $z_t + \eta u_t$ and $z_t^\star + \eta u_t$, respectively, are within η of $\mathcal{M}$. By Lemma 7, $\|w - z_{t+1}^\star\|_2 \leq L_2\, e_t$, using $\|(z_t + \eta u_t) - (z_t^\star + \eta u_t)\| = \|z_t - z_t^\star\| = e_t$. Combining the two gives $e_{t+1} \leq \ell_n + L_2\, e_t$. Unrolling with $e_0 = 0$,

$$e_t \leq \ell_n \sum_{k=0}^{t-1} L_2^k = \ell_n \cdot \frac{1 - L_2^t}{1 - L_2}.$$

□

C Additional Results

C.1 Targeted walk on synthetic data

Figure 8 shows the targeted walk on the synthetic dataset for all four variants. $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cyc}}$ leave the manifold within a few steps, where the decoded points scatter away from the data support. $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{den}}$ stays on the manifold. $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{proj}}$ follows the manifold geometry toward the target , successfully traversing while staying on-manifold throughout.

C.2 Undirected walk on synthetic data

Figure 9 shows the undirected walk on the synthetic dataset for all four variants, mirroring Figure 8's targeted-walk comparison. $\mathcal{L}_{rec}$ and $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$ again leave the manifold, $\mathcal{L}_{rec} + \mathcal{L}_{den}$ stays

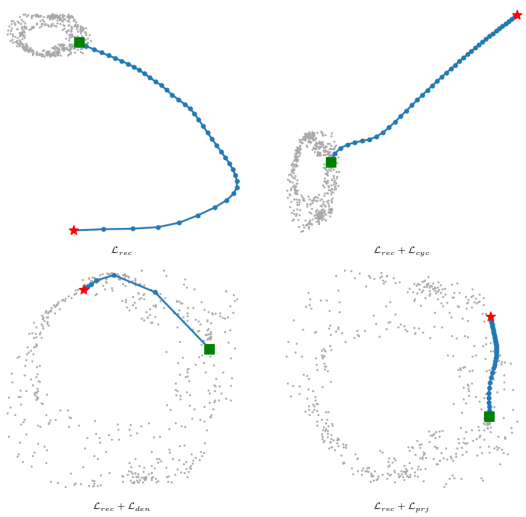


Figure 9: Undirected walk on synthetic dataset.

on-manifold but resists displacement, and $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ traces the manifold's curvature with no target to steer toward, confirming the non-compounding behavior comes from the learned projection rather than the guidance of a target.

C.3 Extended Quantitative Analysis: Off-Manifold Distance

Table 1 reports the distance to the true manifold as a function of traversal step. The key pattern is the distinction between the compounding and non-compounding drift. There are three main things to observe here: (1) *Compounding drift.* $\mathcal{L}_{rec}$ and $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$ grow in t, where each step adds error that the next step does not correct, and the trajectory leaves the manifold permanently. This is consistent with the mathematical observation in Sec. 2, that $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$'s fixed-point set is all of $\mathbb{R}^d$ rather than $\mathcal{M}$. (2) *Non-compounding drift.* Both $\mathcal{L}_{rec} + \mathcal{L}_{den}$ and $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ have a distance that rises a little and then plateaus, showing no further accumulation out to $t = 40$. This is the empirical evidence of the retraction property of Eq. 8, where the decode-encode composition absorbs each step's displacement rather than letting it compound. (3) *The denoising-projection gap.* $\mathcal{L}_{rec} + \mathcal{L}_{den}$ has a lower plateau than $\mathcal{L}_{rec} + \mathcal{L}_{proj}$, but at a cost that Table 1 does not show and the visual results, discussed later, make it clear. The denoising target pulls every step back towards its origin, so the walk resists displacement by design, while the projection target keeps the displacement while snapping it onto the manifold, making the model do more to allow manifold-faithful traversals.

C.4 Targeted walk on Colored MNIST

Figure 10 shows the targeted walk on Colored MNIST for all four variants. $\mathcal{L}_{rec}$ collapses into unstructured patterns within the first few steps, $\mathcal{L}_{rec} + \mathcal{L}_{cyc}$ reaches the destination but with blurry intermediate steps, $\mathcal{L}_{rec} + \mathcal{L}_{den}$ stays relatively coherent but produces nearly identical frames, and $\mathcal{L}_{rec} + \mathcal{L}_{proj}$ produces recognizable, smoothly varying digits throughout.

C.5 Undirected walk on Colored MNIST

Figure 11 shows undirected walks on Colored MNIST. Without a target to steer toward, the contrast between variants is clearer, $\mathcal{L}_{\text{rec}}$ produces grey frames (the walk has left the data support entirely), $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{den}}$ produces near-identical frames, resisting displacement, and $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{proj}}$ produces recognizable digits that change smoothly in color and writing style across frames. $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cyc}}$ produces visually incoherent images, consistent with its compounding drift on the synthetic data.

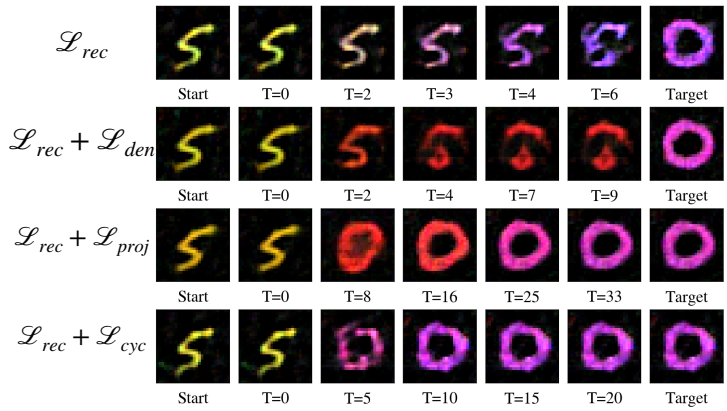


Figure 10: Targeted walks using latent traversal on Colored MNIST. Each row is a training variant, while each column shows decoded frames.

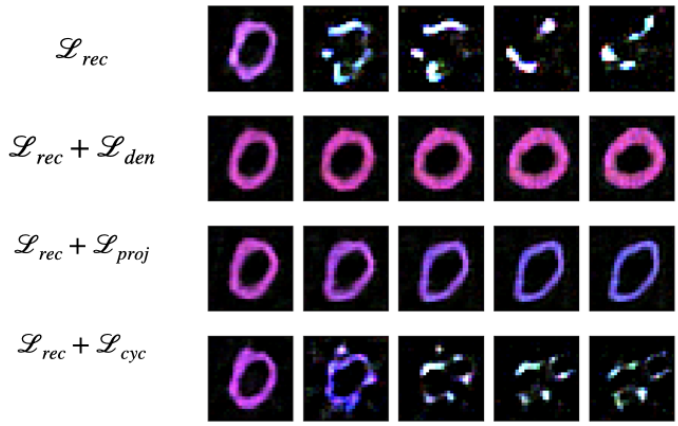


Figure 11: Undirected walks obtained via latent traversal on Colored MNIST.

D Extended Scaling Methodology

This section details the two procedures introduced in Sec. 6, memory bank refresh and subsampled projection lookup (Algorithm 2), and K-nearest-neighbor graph construction with chunked rollout training (Algorithm 3).

Algorithm 2 Memory bank refresh and subsampled projection lookup

Require: training pool $\mathcal{D}$, bank size n, subsample size s, refresh interval R_{refresh}, current step t

if $t \bmod R_{\text{refresh}} = 0$ **then**
  sample $\{x_1, \ldots, x_n\} \sim \mathcal{D}$, Bank $\leftarrow \{E(x_i)\}_{i=1}^{n}$ under no_grad
end if
$S_t \leftarrow$ uniform random subsample of Bank, $|S_t| = s$
for each perturbed code $\bar{z}_i$ in the current batch **do**
  $z_i^{\text{proj}} \leftarrow \arg\min_{z \in S_t} \|z - \bar{z}_i\|_2$
end for

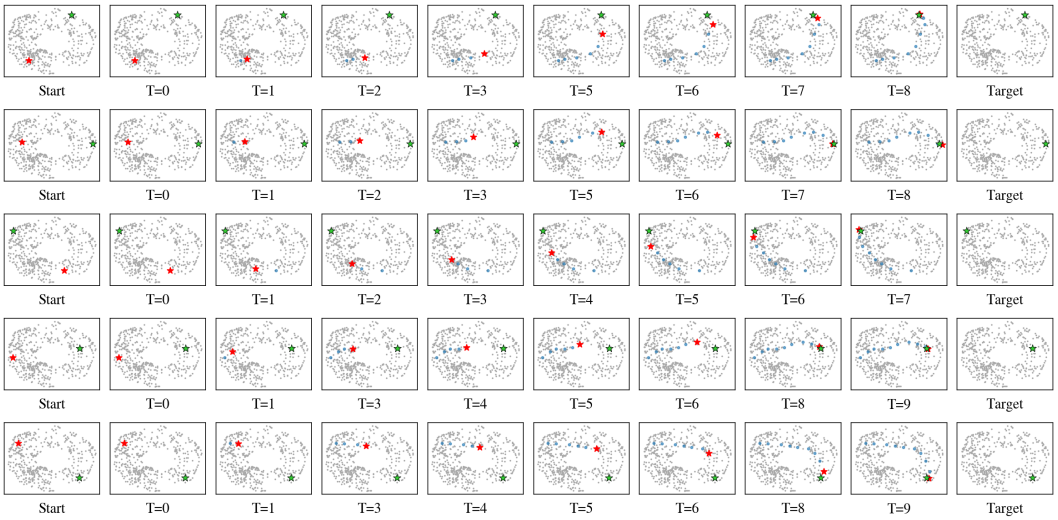


Figure 12: Targeted walk under the scaled mechanism on the synthetic dataset at the larger scale cell (N=16). The learned policy's trajectory (red) tracks the manifold's own geometry (gray) toward the target (green star).

Algorithm 3 K-NN graph construction and chunked rollout training step

Require: Bank (current refresh), neighbor count K, per-edge step cap $c_{\max}$, policy π_θ

if Bank was just refreshed **then**
  build $G = (\text{Bank}, E)$, connect each z_i to its K nearest neighbors in Bank
end if
sample (z_0, z_{target}), insert both into G via their K nearest bank points
$p_0, \dots, p_M \leftarrow$ Dijkstra shortest path from z_0 to z_{target} on G
$L \leftarrow 0$
for $i = 0, \dots, M-1$ **do**
  $z \leftarrow p_i$, $m_i \leftarrow \|p_{i+1} - p_i\|_2$
  for $c = 1, \dots, c_{\max}$ **do**
    $(d, p) \leftarrow \pi_\theta(z, z_{\text{target}}, \dots)$, $z \leftarrow E(D(z + p \cdot m_i \cdot d))$
    if $\|z - p_{i+1}\|_2 < \text{tol}$ **then**
      break
    end if
  end for
  $L \leftarrow L + \|z - p_{i+1}\|_2$, $z \leftarrow \text{detach}(z)$
end for
update θ by gradient of L

E Extended Scaling Results

This section expands Sec. 6.1's comparison with the synthetic dataset, the second scale cell on Colored MNIST, and additional qualitative examples.

Wall-clock time. At the smallest cell (synthetic) the unscaled mechanism is still cheaper in absolute terms, the fixed overhead of building a bank and warm-starting a policy not yet paying for itself at trivial scale, but this reverses as scale grows. By Colored MNIST's smaller cell the scaled mechanism is already 3.4× faster, and at the larger cell the unscaled mechanism does not run at all.

Colored MNIST specifically stresses the unscaled mechanism harder than the synthetic manifold does. The unscaled mechanism becomes infeasible on Colored MNIST already at the second scale cell tested, whereas the synthetic manifold's unscaled mechanism completed both cells we tried. This

is consistent with the cost analysis in Sec. 6, the unscaled mechanism's dominant cost term scales with n and the per-step activation cost of $E \circ D$, and a convolutional decoder over $3\times28\times28$ images has a substantially larger per-step activation cost than an MLP over a low-dimensional ambient vector, so the same $(N, T_{\max}, n)$ growth reaches the unscaled mechanism's infeasibility boundary sooner on Colored MNIST. We take this as the mechanism working as intended, not as a dataset-specific failure, the point of Sec. 6 is precisely that this boundary exists and is dataset-dependent.

Colored MNIST at the smaller scale cell. Figure 13 shows the scaled mechanism at Colored MNIST's smaller scale cell, for comparison with Figure 14's larger one. Qualitatively both remain coherent, smoothly-interpolating digit sequences, walk quality degrades somewhat as scale grows further, though trajectories stay recognizable throughout.

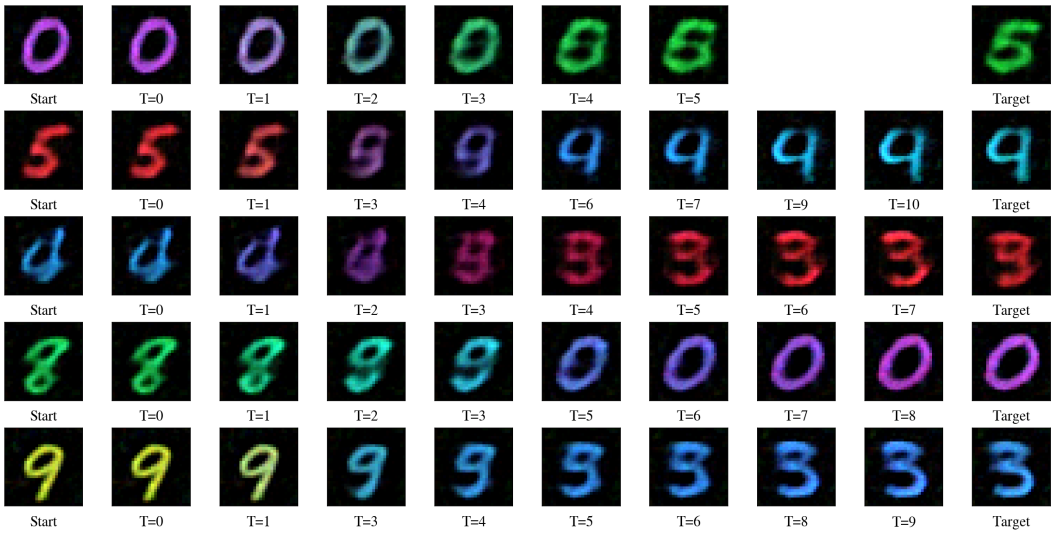


Figure 13: Targeted walks under the scaled mechanism at Colored MNIST's smaller scale cell (N=64), five held-out start/target pairs. Compare to Figure 14, showing the larger N=128 cell.

Memory reduction holds at both scales tested, not only the largest. The $\sim10\times$ memory reduction highlighted in the main text is the more dramatic of the two Colored-MNIST cells only because the unscaled mechanism could not be measured at the larger cell at all. At every cell where a direct comparison exists (both synthetic cells, Colored MNIST's first cell), the scaled mechanism's peak memory is lower, consistent with replacing an n-dependent batch search with a fixed-size bank.

Scope. The comparison above covers two scale cells per dataset, chosen so that the unscaled mechanism is still attemptable at the smaller cell and infeasible or nearly so at the larger one, enough to demonstrate the crossover the mechanism is designed around, not an exhaustive scaling curve. Further scaling to substantially larger cells, where only the scaled mechanism can run at all, is a natural next step.

F Extended Discussion

What we do and do not claim. A reader may reasonably prefer interventional editing, where one names the attribute to change. That requires labels, auxiliary variables Khemakhem et al. (2020), or heterogeneous environments Zhang et al. (2024), and is provably out of reach without them Hyvärinen & Pajunen (1999); Locatello et al. (2019), we do not claim to have bypassed it. Our trajectories are uncontrolled, we can say the image changes and stayed valid, not that a particular factor was intervened upon. The contribution is that this weaker property is not automatic, is achievable without supervision, and is prerequisite for label-free editing method built on a latent space.

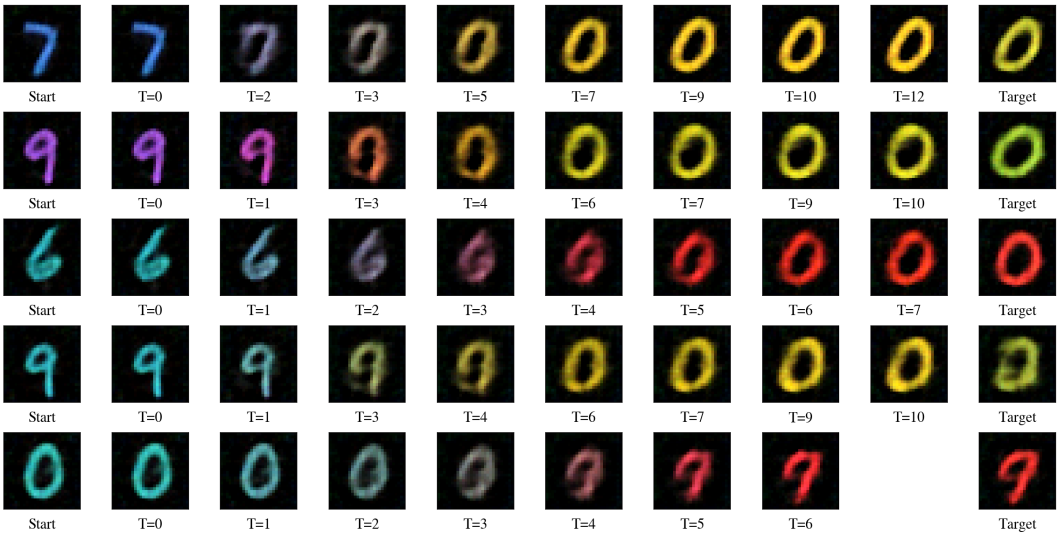


Figure 14: Targeted walks under the scaled mechanism on Colored MNIST at the larger scale cell ($N{=}128$), five held-out start/target pairs. Digit shape and color both interpolate smoothly toward the target.

Relation to Pearl's Hierarchy. It is tempting to call these traversals counterfactual, since they modify an observed sample. The more accurate statement is that they sit outside the hierarchy rather than on its third level Bareinboim et al. (2022), no intervention is specified, so no counterfactual query is being answered. What the projection provides is closure of the observational support under latent perturbation.

Limitations and Future Work. The batch nearest neighbor is a coarse estimate of $\Pi_{\mathcal{M}}$ and degrades when the batch covers $\mathcal{M}$ sparsely, nothing selects a semantically meaningful direction, and our results are on small-scale data only.