Shaping Latent Geometry with Noise-Injected Hopfield Dynamics

Wooyul Jung WJDDNDUF5730@SNU.AC.KR
Youngseok Joo ROBINJOO1015@SNU.AC.KR
Interdisciplinary Program in Artificial Intelligence, Seoul National University

Dohyun Yu YDH0455@SNU.AC.KR
College of Liberal Studies, Seoul National University

Suhyung Choi S.CHOI@SNU.AC.KR
Byoung-Tak Zhang[*] BTZHANG@SNU.AC.KR
Interdisciplinary Program in Artificial Intelligence, Seoul National University

Editors: List of editors' names

Abstract

Latent representations that exhibit geometric structure are central to robust and generalizable learning. While such geometry often emerges incidentally, previous work has attempted to enforce it through regularization or architectural changes. In this work, we propose the Noise-Injected Hopfield Retrieval (NIHR) layer, a differentiable module that injects Gaussian noise into the update dynamics of Modern Hopfield Networks to actively shape latent space geometry. By controlling the number of retrieval iterations and the inverse temperature, NIHR enables a tunable transition between discrete attractors and smooth continuous manifolds. When integrated into autoencoding and classification pipelines, NIHR consistently improves robustness to corruptions, latent structure quality, and linear separability. Our results suggest that NIHR provides an effective mechanism for imposing meaningful geometric inductive biases in neural representations without auxiliary loss functions.

Keywords: Representation learning, Structured representations, Hopfield networks, Attractor dynamics, Robustness

1. Introduction

Representation learning seeks latent spaces whose geometric structure supports reasoning, robustness, and sample efficiency (Bengio et al., 2013). Empirical evidence suggests that such structure can emerge spontaneously. For example, class means tend to collapse in late stages of training (Papyan et al., 2020), and both human and neural representations exhibit characteristic power-law decay in their eigenspectra (Agrawal et al., 2022). These observations indicate that latent geometry often arises incidentally, motivating approaches that aim to shape it more explicitly.

A variety of methods aim to impose geometric structure on latent spaces. Local-sensitivity control includes contractive autoencoders (Rifai et al., 2011) and spectral or Lipschitz regularization for robustness (Jakubovitz and Giryes, 2018; Cisse et al., 2017). Other approaches introduce latent-space objectives or priors, such as disentanglement (Higgins et al., 2017), contrastive alignment (Chen et al., 2020), discrete codebooks (Van Den Oord

[*] Advisor

et al., 2017), and hyperspherical embeddings (Davidson et al., 2018). Associative memory models offer a complementary dynamical view. Modern Hopfield Networks (Hopfield, 1982; Krotov and Hopfield, 2020; Ramsauer et al., 2020) contract representations through attractor dynamics, providing robustness (Krotov and Hopfield, 2018), and recent work shows that even spurious attractors can aid generalization (Pham et al., 2025; Kalaj et al., 2024).

We build on this perspective by introducing noise-injected Hopfield dynamics as a mechanism for explicitly shaping latent geometry. Our Noise-Injected Hopfield Retrieval (NIHR) layer inserts stochastic Hopfield iterations between neural network layers, with iteration count and inverse temperature controlling the degree of contraction, ranging from discrete attractors to smoother continuous manifolds. Unlike generative latent dynamics methods such as score-based priors (Vahdat et al., 2021), energy-based regularizers (Pang et al., 2020), or latent diffusion (Rombach et al., 2022), NIHR is designed for discriminative pipelines and requires no auxiliary loss terms.

We evaluate NIHR on autoencoding and classification tasks using CIFAR-10, MNIST, and their corruption benchmarks (CIFAR-10-C (Hendrycks and Dietterich, 2019) and MNIST-C (Mu and Gilmer, 2019)). Across metrics including corruption accuracy, linear separability, and PCA organization, our results show that injecting Hopfield dynamics consistently shapes latent representations into more stable and robust structures.

2. Method

2.1. Noise-Injected Hopfield Dynamics

We formulate a stochastic Hopfield update that introduces noise-driven attractor dynamics into latent representations. Given a set of unit-norm memory vectors $\{e_i\}_{i=1}^{P}$ with $\|e_i\|_2 = 1$ and memory matrix $C = [e_1, \ldots, e_P] \in \mathbb{R}^{D \times P}$, the query vector $z_t \in \mathbb{R}^D$ is updated as follows.

At each iteration t, Gaussian noise $\xi_t \sim \mathcal{N}(0, \sigma^2 I_D)$ is injected and the perturbed vector is normalized to unit norm:

$$\hat{z}_t = \frac{z_t + \xi_t}{\|z_t + \xi_t\|_2}.$$

The Hopfield update then performs soft retrieval over the memory with inverse temperature $\beta > 0$:

$$z_{t+1} = C \ \mathrm{softmax}\left(\beta\, C^\top \hat{z}_t\right). \tag{1}$$

This process corresponds to (noisy) gradient descent on the Modern Hopfield energy function (Ramsauer et al., 2020):

$$E_{\mathrm{MHN}}(z) = -\frac{1}{\beta} \log \sum_{i=1}^{P} \exp\left(\beta\, z^\top e_i\right). \tag{2}$$

When $\sigma = 0$, the energy decreases monotonically, yielding convergence to discrete attractors. For $\sigma > 0$, the expected energy decreases up to $O(\beta D \sigma^2)$, allowing stochastic transitions between nearby basins. This mechanism enables the retrieval dynamics to explore the attractor landscape while maintaining stability, thereby facilitating the emergence of continuous structure in representation space. Detailed analysis of geometry-induced continuous attractors is provided in Appendix C.

2.2. Hopfield Retrieval Layer

We implement the above dynamics as the *Noise-Injected Hopfield Retrieval* layer (NIHR), a general, differentiable module that can be placed between arbitrary layers in a neural network. NIHR performs T steps of the noise-injected Hopfield update, as defined in Equation 1:

$$z_{t+1} = \mathcal{H}_\sigma(z_t), \quad t = 0, \ldots, T-1.$$

Each step refines the representation through content-based retrieval over a learnable memory matrix, initialized randomly and updated via gradient descent. Noise perturbations injected at each step enable exploration of nearby attractors, encouraging smooth convergence toward robust patterns. This iterative refinement imposes local structure by guiding representations toward energy minima shaped by the Hopfield memory. NIHR is trained end-to-end with standard task losses and introduces no auxiliary objectives.

3. Experiments

3.1. Structural evolution under Autoencoder framework

We study NIHR as an intermediate layer that can be inserted between the encoder and decoder in an autoencoder framework. We investigate how such Hopfield retrieval influences the organization of latent structure, and how this in turn affects reconstruction robustness and representation quality. All models are trained on the MNIST dataset using binary cross-entropy loss for 1000 epochs with the Adam optimizer. We fix the latent dimension, memory size, latent noise level, and retrieval sharpness, and use consistent hyperparameters across all models (see Appendix E for details). NIHR is compared against three baselines: a plain autoencoder (AE), a single-head VQ-VAE (Van Den Oord et al., 2017), and a VAE (Kingma and Welling, 2013). All experiments are repeated with 5 random seeds, and mean ± standard deviation are reported.

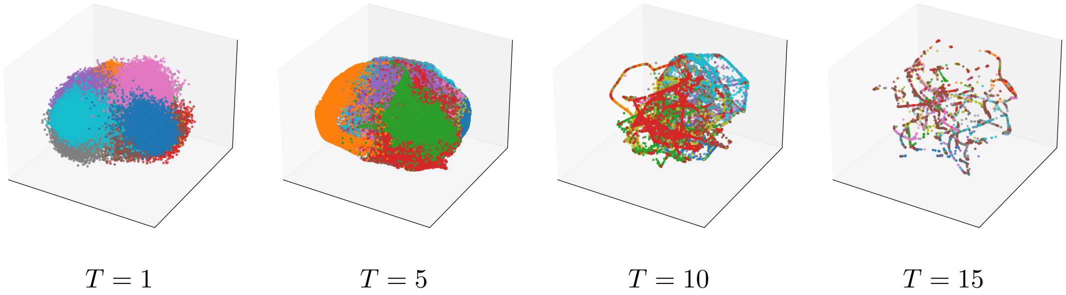


Figure 1: PCA projections of latents after different numbers of refinement iterations(T). Progressive clustering illustrates the transition from diffuse to structured latent organization.

To illustrate the effect of iterative refinement, we project latent codes onto their top principal components after $T = 1, 5, 10, 15$ steps (Figure 1). The progression reveals a transition from diffuse to clustered organization, showing how refinement gradually sharpens latent structure. Additional experiments on reconstruction fidelity and representation quality (e.g., linear probing) are presented in Appendix D, which further demonstrate that NIHR provides explicit control over latent space geometry through its hyperparameters (β, P, σ), enabling a tunable trade-off between fidelity and robustness under corruption.

3.2. Robust Classification under Corruptions

We apply NIHR as a latent refinement layer within ResNet-18 to enhance robustness under distribution shifts. The insertion point is determined empirically: after the first residual block for MNIST and after the last residual block for CIFAR-10. Robustness is evaluated on the corruption benchmarks CIFAR-10-C (Hendrycks and Dietterich, 2019) and MNIST-C (Mu and Gilmer, 2019). CIFAR-10-C introduces a wide range of non-adversarial perturbations across five severity levels, whereas MNIST-C applies diverse corruption types without severity variation. All models are trained exclusively on the clean training sets of MNIST and CIFAR-10, without augmentation or corruption. Classification accuracies on the corrupted benchmarks are summarized in Table 1.

Table 1: Classification accuracy (%) on CIFAR-10-C and MNIST-C corruption benchmarks.

Dataset	Model	Clean (0)	s1	s2	s3	s4	s5	Overall
MNIST-C	ResNet	99.47 ± 0.05	–	–	–	–	–	72.13 ± 2.19
	ResNet+NIHR	**99.50 ± 0.02**	–	–	–	–	–	**80.56 ± 2.2**
CIFAR-C	ResNet	**84.78 ± 0.18**	**75.69 ± 0.68**	**70.39 ± 0.92**	**64.99** ± 1.03	58.51 ± 1.21	48.98 ± 1.04	**63.71 ± 0.96**
	ResNet+NIHR	81.81 ± 0.64	73.93 ± 1.25	69.28 ± 1.29	64.58 ± 1.38	**58.73 ± 1.38**	**51.00 ± 1.94**	63.50 ± 1.31
CIFAR-C (Gaussian)	ResNet	**84.78 ± 0.18**	70.10 ± 1.74	55.40 ± 2.93	41.91 ± 4.14	36.36 ± 4.63	31.93 ± 4.86	47.14 ± 3.53
	ResNet+NIHR	81.81 ± 0.64	**72.81 ± 2.30**	**63.83 ± 3.41**	**54.47 ± 4.21**	**49.75 ± 4.59**	**45.99 ± 4.75**	**57.37 ± 3.63**

Across CIFAR-10 and MNIST, NIHR consistently improves robustness to input corruptions. On MNIST, NIHR improves overall accuracy from 72.13% to 80.56% without sacrificing clean performance (from 99.47% to 99.50%). On CIFAR-10, NIHR improve performance on high-severity corruptions (e.g., +2.02% on s5). Notably, under CIFAR-10 Gaussian corruptions, NIHR substantially boosts accuracy across all severity levels, with up to +14.06% at the highest level (from 31.93% to 45.99%), resulting in a +10.23% gain in overall performance. These results demonstrate that NIHR enhances robustness to both semantic and low-level perturbations while preserving generalization on clean data.

We also analyze how NIHR transforms intermediate representations during classification; PCA visualizations in Appendix B show that attractor dynamics progressively organize features into more coherent clusters, reducing intra-class variance and sharpening boundaries between classes.

4. Conclusion

We introduced the Noise-Injected Hopfield Retrieval (NIHR) layer, which shapes latent representations through stochastic Hopfield retrieval dynamics. Unlike methods that rely on auxiliary losses or architectural changes, NIHR imposes geometric structure directly by injecting controlled noise into the update process. This yields representations that are both smooth and clustered, providing a tunable inductive bias that improves robustness to corruptions and enhances linear separability. Overall, NIHR offers a simple and effective way to equip discriminative models with explicitly structured latent spaces.

Acknowledgments

This work was partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/15%, RS-2022-II220953-PICA/15%), NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/15%, RS-2025-25425421/10%), KEIT (RS-2024-00423940/10%) grant funded by the Korean government.

References

Kumar K Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. α-req: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. *Advances in Neural Information Processing Systems*, 35:17626–17638, 2022.

Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Information storage in neural networks with low levels of activity. *Physical Review A*, 35(5):2293, 1987.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR, 2017.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

John J Hopfield, David I Feinstein, and Richard G Palmer. 'unlearning'has a stabilizing effect in collective memories. *Nature*, 304(5922):158–159, 1983.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 514–529, 2018.

Silvio Kalaj, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, Enrico M Malatesta, and Matteo Negri. Random features hopfield networks generalize retrieval to previously unseen examples. *arXiv preprint arXiv:2407.05658*, 2024.

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

Dmitry Krotov and John Hopfield. Dense associative memory is robust to adversarial inputs. *Neural computation*, 30(12):3151–3167, 2018.

Dmitry Krotov and John Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33: 21994–22008, 2020.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Bao Pham, Gabriel Raya, Matteo Negri, Mohammed J Zaki, Luca Ambrogioni, and Dmitry Krotov. Memorization to generalization: Emergence of diffusion models from associative memory. *arXiv preprint arXiv:2505.21777*, 2025.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

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 international conference on machine learning*, pages 833–840, 2011.

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*, pages 10684–10695, 2022.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

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

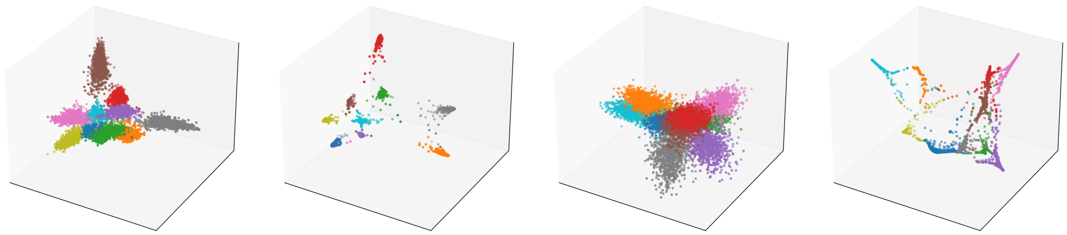
Appendix A. Related Work

Shaping Latent Representations Many methods explicitly influence latent-space geometry. Some regulate local sensitivity, such as contractive autoencoders that penalize encoder Jacobians (Rifai et al., 2011) or spectral and Lipschitz regularization for robustness (Jakubovitz and Giryes, 2018; Cisse et al., 2017). Others impose structural priors: disentanglement objectives like β-VAE (Higgins et al., 2017), contrastive alignment–uniformity formulations (Chen et al., 2020), or priors such as discrete codebooks (Van Den Oord et al., 2017) and hyperspherical latents (Davidson et al., 2018). These approaches provide explicit geometric control but typically rely on auxiliary losses or specified priors.

Associative Memory and Dynamical Perspectives Hopfield networks shape representations through attractor dynamics: classical models (Hopfield, 1982) retrieve stored patterns via energy minimization, leading to inherent robustness (Krotov and Hopfield, 2018). Modern continuous formulations reinterpret these dynamics through attention-like update rules (Krotov and Hopfield, 2020; Ramsauer et al., 2020), facilitating their integration into deep architectures. While spurious attractors were traditionally viewed as undesirable (Hopfield et al., 1983; Amit et al., 1987), recent work shows that they can, under certain conditions, contribute positively to generalization (Kalaj et al., 2024; Pham et al., 2025). This dynamical perspective also underlies related approaches such as score-based priors (Vahdat et al., 2021), energy-based models (Pang et al., 2020), and latent diffusion systems (Rombach et al., 2022), which similarly impose structure through iterative latent-space dynamics.

Appendix B. Structural Evolution under Classification

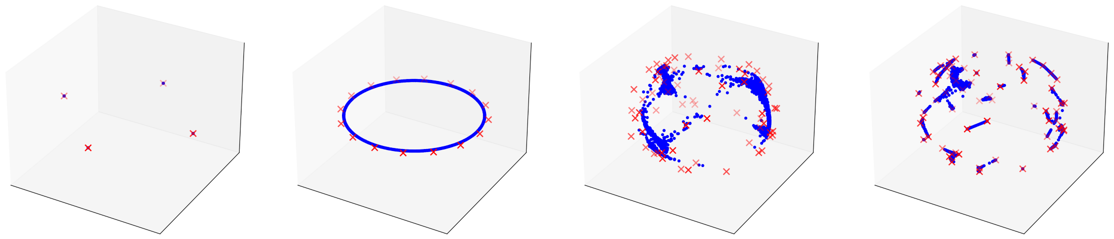
We examined how intermediate representations on the training data evolve in ResNet-18 when NIHR is inserted between layers. Compared to the baseline, NIHR produces noticeably tighter and more structured class clusters, reflecting the contraction toward attractor states introduced by its retrieval dynamics. These visualizations indicate that NIHR stabilizes latent geometry during training and enhances class separation on both MNIST and CIFAR-10, complementing the robustness gains reported in the main text.



MNIST w/o NIHR MNIST w/ NIHR CIFAR-10 w/o NIHR CIFAR-10 w/ NIHR

Figure 2: PCA visualizations of intermediate training representations with and without NIHR. NIHR induces tighter intra-class clusters and clearer inter-class separation on both MNIST and CIFAR-10, reflecting the attractor-driven contraction of the latent space.

Appendix C. Geometry-Induced Continuous Attractors



(a) Ring: $\beta = 10$, $P = 4$ (b) Ring: $\beta = 10$, $P = 16$ (c) Sphere: $\beta = 10$, $P = 64$ (d) Sphere: $\beta = 100$, $P = 64$

Figure 3: 3D PCA of 5,000 queries after 10 noise-injected Hopfield updates ($\sigma = 0.05$, $D = 64$). Blue dots: queries, Red crosses: memory vectors. Varying β and P controls the transition from discrete to continuous attractor dynamics.

To better understand the conditions under which continuous attractors emerge in Noise-Injected Hopfield Dynamics, we analyze retrieval behavior under controlled geometric memory configurations. Specifically, we examine how the interplay between memory geometry and inverse temperature β shapes the attractor landscape and influences its stability. We fix the embedding dimension to $D = 64$, and evaluate retrieval over $T = 10$ noise-injected updates ($\sigma = 0.05$) using $N = 5000$ random queries with $\beta = 10$.

Two structured configurations are considered: (1) evenly spaced points on a 2D ring, and (2) uniformly sampled points on the 3D unit sphere, both embedded into D dimensions by zero-padding. With few patterns and high β, retrieval yields discrete attractors. For instance, on the ring with $P = 4$ and $\beta = 10$, trajectories collapse into isolated basins (Fig. 3(a)). As P increases, attractors merge into continuous manifolds: with $P = 16$ on the ring, queries diffuse smoothly (Fig. 3(b)). On the sphere with $P = 64$ and $\beta = 10$, diffusion extends across the surface (Fig. 3(c)), while increasing β to 100 restores sharp convergence (Fig. 3(d)).

To assess stability, we compute the Jacobian $\mathbf{J} = \frac{\partial z_{t+1}}{\partial z_t}$ at convergence. Eigenvalues capture local geometry: orthogonal directions contract ($|\lambda| < 1$), while tangent modes often remain marginal ($|\lambda| \approx 1$), enabling diffusion. Occasionally, $|\lambda| > 1$ signals local instability. Spectral analysis confirms the discrete-to-continuous transition. For a ring with $P = 4$, eigenvalues vanish ($\lambda_1 \approx 0$), showing rigid basins. At $P = 8$, a soft mode emerges ($\lambda_1 = 0.637 \pm 0.320$), and at $P = 16$, marginal stability appears ($\lambda_1 = 1.004 \pm 0.001$). On the sphere, multiple continuous modes emerge: for $P = 64$, $\lambda_1 = 0.749 \pm 0.171$, $\lambda_2 = 0.438 \pm 0.081$, indicating drift along at least two directions. For $P = 1024$, both eigenvalues grow toward unity, suggesting increasingly extended trajectories. By contrast, increasing β at $P = 64$ compresses the spectrum ($\lambda_1 = 0.331 \pm 0.071$), suppressing drift.

Overall, continuous attractors emerge in Noise-Injected Hopfield Dynamics through the interplay between memory geometry and thermodynamic precision. When memory patterns are aligned with an underlying manifold, retrieval dynamics exhibit smooth transitions and marginal stability. In contrast, high inverse temperature β suppresses drift and enforces convergence to discrete attractors.

Appendix D. Robustness under Autoencoding Framework

This section investigates the robustness of different autoencoding models under input corruption, focusing on their ability to maintain reconstruction fidelity and representation quality when presented with noisy inputs. We conduct controlled experiments on MNIST with additive Gaussian noise of varying standard deviation σ_I to simulate input degradation. Two metrics are used: the structural similarity index (SSIM) for reconstruction quality, and linear probing accuracy for assessing the discriminative power of learned representations. For linear probing, encoders trained on clean data are frozen. Latent codes are extracted from corrupted inputs, and a linear classifier is trained to predict class labels from these codes.

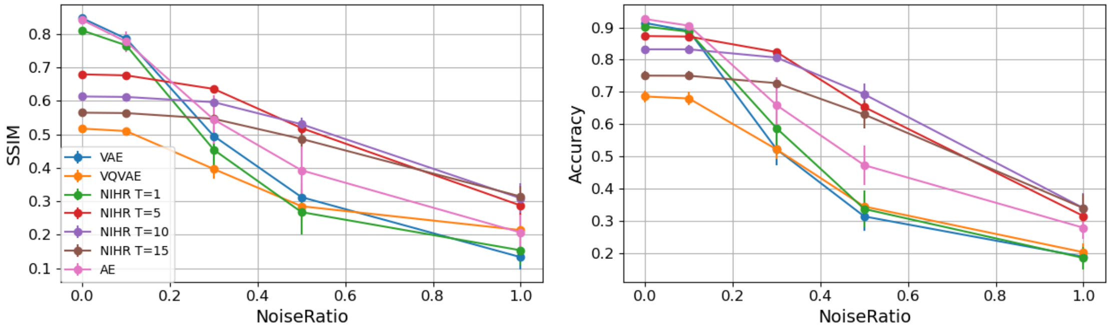
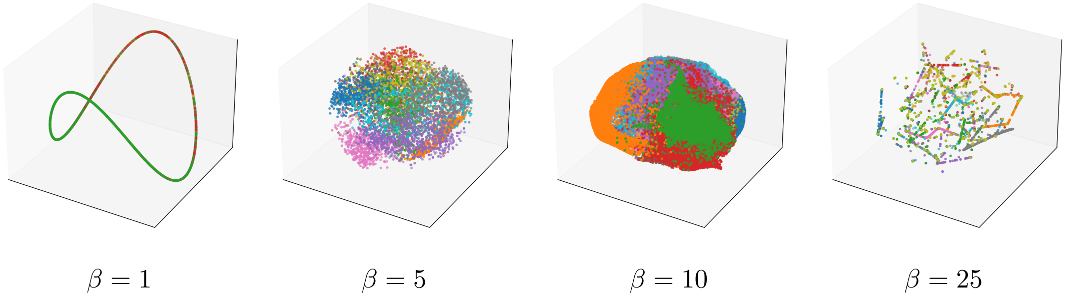


Figure 4: Evaluation under Gaussian input corruption. Left: SSIM scores for reconstruction fidelity. Right: Linear probing accuracy for representation quality. Models include AE, VAE, VQ-VAE, and NIHR with $T \in \{1, 5, 10, 15\}$.

On nearly clean inputs ($\sigma_I \leq 0.1$), NIHR with $T = 1$ yields the highest SSIM, closely matching AE behavior. Under increasing noise, NIHR with $T = 5$–10 consistently outperforms all baselines in both SSIM and linear accuracy. These results indicate a trade-off controlled by the refinement step count T: small T preserves fine-grained details, while larger T contracts latent codes toward more robust prototypes. NIHR thus enables a tunable trade-off between fidelity and robustness, with $T = 5$ providing a favorable operating point across corruption levels.

D.1. Ablation on Hyperparameters (β, P, σ)

Table 2 and Figure 5 together illustrate how the key hyperparameters—temperature β, memory size P, and injected noise σ—shape NIHR performance under Gaussian corruption. For the inverse temperature β, PCA visualizations reveal distinct latent geometries. At β=1, latents collapse into a ring-shaped attractor with poor discriminability, leading to steep SSIM and accuracy drops. Increasing to β=5 spreads the latents more evenly, similar to AE, preserving fidelity on clean data but only moderate robustness. At β=10, well-separated basins emerge, giving the best balance of fidelity and robustness (strong SSIM at σ_I=0.3, 0.5 and highest probe accuracy up to σ_I=0.5). At β=25, latents collapse into discrete prototypes, which improves robustness under heavy noise but reduces clean-image fidelity. Varying the memory size P shows a complementary effect. Larger memories



$\beta = 1$ $\beta = 5$ $\beta = 10$ $\beta = 25$

Figure 5: PCA projections of refined latents for different inverse temperatures ($\beta \in \{1, 5, 10, 25\}$) after $T = 10$ iterations.

Table 2: SSIM and linear probe accuracy (%) under Gaussian corruption, varying inverse temperature (β), memory size (P), and injected latent noise (σ). Mean over 5 seeds; best in each column in bold.

Setting / Model	SSIM					Linear Probe Accuracy (%)				
	σ_I=0.0	σ_I=0.1	σ_I=0.3	σ_I=0.5	σ_I=1.0	σ_I=0.0	σ_I=0.1	σ_I=0.3	σ_I=0.5	σ_I=1.0
Inverse temperature β										
β=1	0.4598	0.4586	0.4439	0.4014	0.2769	49.71	49.73	49.00	44.81	26.72
β=5	**0.7376**	**0.7301**	0.6274	0.4454	0.2338	**88.52**	**87.97**	77.77	55.38	27.87
β=10	0.6795	0.6764	**0.6358**	**0.5189**	0.2872	87.32	87.17	**82.36**	**65.30**	31.49
β=25	0.5707	0.5695	0.5530	0.5046	**0.3447**	75.28	75.24	72.93	64.72	**38.11**
Memory size P										
P=64	0.5952	0.5935	0.5800	**0.5365**	**0.3566**	82.19	82.23	81.07	**74.41**	**43.64**
P=256	0.6795	0.6764	0.6358	0.5189	0.2872	87.32	87.17	**82.36**	65.30	31.49
P=1024	**0.7748**	**0.7657**	**0.6510**	0.4494	0.2010	**92.48**	**92.12**	81.86	58.62	26.54
Injected noise σ										
σ=0.0	**0.7943**	0.7457	0.4843	0.3208	0.1468	90.64	88.82	55.38	32.85	17.71
σ=0.01	0.7795	**0.7607**	0.5887	0.3960	0.2163	**91.93**	**91.09**	70.69	42.31	24.47
σ=0.05	0.6795	0.6764	**0.6358**	0.5189	0.2872	87.32	87.17	82.36	65.30	31.49
σ=0.1	0.6478	0.6457	0.6307	**0.5802**	**0.3890**	87.66	87.49	**85.01**	**76.24**	**44.16**
Baselines										
AE	0.8419	0.7774	0.5480	0.3974	0.2105	92.74	90.72	65.79	46.46	27.27
VAE	0.8475	0.7773	0.4631	0.2787	0.1162	91.24	88.12	49.39	29.69	17.69
VQ-VAE	0.5171	0.5100	0.3963	0.2851	0.2138	68.58	67.97	52.17	34.46	20.27

(P=1024) provide higher capacity, producing the best SSIM and accuracy on clean inputs, but degrade quickly as corruption grows. Smaller memories (P=64) reduce peak performance yet yield more stable accuracy under noise. Injected noise σ during training plays a similar role. Low noise (σ=0.0) favors fidelity, while higher noise (σ=0.05, 0.1) lowers clean scores but significantly boosts robustness once $\sigma_I \geq 0.5$, consistent with smoothing of the latent energy landscape.

Overall, these results reveal a consistent trade-off: large P and low σ emphasize reconstruction fidelity, while small P and high σ enhance robustness. The inverse temperature β acts as a knob that directly shapes latent geometry, with intermediate values such as β=10 offering the best compromise.

Appendix E. Architecture and Training Details

Environment. All models are trained for 1000 epochs using the Adam optimizer with a learning rate of 10^{-4} and binary cross-entropy loss. All experiments are conducted on a single NVIDIA A100 GPU with a batch size of 100. The implementation uses PyTorch 2.7.1 with CUDA 12.4 (cu126 build).

Table 3: Fixed hyperparameters used across experiments.

Hyperparameter	Autoencoding (MNIST)	Classification (CIFAR-10-C)	Classification (MNIST-C)
Latent dimension (d)	10	512	64
Memory size (P)	256	1024	1024
Latent noise std (σ)	0.05	0.01	0.01
Retrieval sharpness (β)	10	10	5
Number of iterations (T)	5	5	1
NIHR insertion point	After Encoder	After `layer4`	After `layer1`
Number of seeds	0,1,2,3,4	0,1,2,3,4	0,1,2,3,4

Encoder/Decoder Architecture. The convolutional encoder and upsampling-based decoder used for MNIST are summarized in Table 4. Here, F denotes the base channel width. The latent dimension is fixed to d=10 for all models (AE, VAE, VQ-VAE, and NIHR). For VQ-VAE, we set the codebook (memory) size to P=256 with single head. For both VQ-VAE and VAE, the additional regularization terms (commitment loss for VQ-VAE, KL divergence for VAE) are weighted with coefficient 1.

Table 4: Encoder and decoder architecture.

Encoder	**Output**
Input: $1 \times 28 \times 28$	–
Conv2d(1→F), stride=2 + ReLU	$F \times 14 \times 14$
Conv2d(F→2F), stride=2 + ReLU	$2F \times 7 \times 7$
Conv2d(2F→4F), stride=2 + ReLU	$4F \times 4 \times 4$
Flatten + Linear($4F \cdot 4 \cdot 4 \to d$)	$\mathbb{R}^d$
Decoder	**Output**
Linear($d \to 4F \cdot 4 \cdot 4$), reshape	$4F \times 4 \times 4$
Upsample to 7×7 + Conv(4F→2F) + ReLU	$2F \times 7 \times 7$
Upsample×2 + Conv(2F→F) + ReLU	$F \times 14 \times 14$
Upsample×2 + Conv(F→1) + Sigmoid	$1 \times 28 \times 28$