Stockholms universitet

Visualizing Dynamics of Representation in Diffusion Models via the Lens of Graph-based Geometrical Analysis

Qixin Yang

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2026:6
Matematisk statistik
Juni 2026

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2026:6**
http://www.math.su.se

Visualizing Dynamics of Representation in Diffusion Models via the Lens of Graph-based Geometrical Analysis

Qixin Yang*

June 2026

Abstract

Denoising Diffusion Probabilistic Models (DDPMs) are state-of-the-art generative models that synthesize images by progressively denoising isotropic Gaussian noise. Despite their empirical success, the geometric organization of the intermediate high-dimensional states $\{x_t\}$ along the diffusion schedule remains only partially understood: existing theoretical accounts operate at the population level of the score function, leaving a sample-level geometric picture of representation dynamics largely open.

This thesis examines how class structure evolves through the forward and reverse processes of a U-Net DDPM trained on MNIST. We introduce a comprehensive Graph-based framework combining adaptive k-Nearest Neighbor graphs, the Biharmonic Distance (derived from the graph Laplacian), Graph-based Silhouette for class validation, and Shape-Aware Stochastic Neighbor Embedding for visual confirmation. Tracking 5000 samples across 15 diffusion checkpoints, we systematically map the geometric transformations of the data manifold.

Our central finding is that the global Graph-based Silhouette $S(t)$ exhibits a distinct three-regime structure shared by both trajectories: (1) a *class-dominated regime* ($S > 0$) at small t; (2) a *transient regime* ($200t500$) featuring a pronounced negative minimum near $t \approx 300$, indicating severe class entanglement; and (3) a *noise-dominated regime* characterizing $t \geq 500$. Notably, the reverse generation process exhibits a deeper transient minimum, a weaker initial class structure, and an inflated effective dimensionality compared to the forward process. A class-wise analysis further reveals dataset-specific topological anomalies, such as class 9 maintaining a positive Silhouette throughout the reverse transient regime, while class 4 remains structurally entangled at every checkpoint.

Consistent with this picture, the approach to the noise-dominated plateau falls within a timestep window that contains the speciation time $t_S \approx 543$ predicted by the mean-field analysis of Biroli et al. [**?**], resolved to within the 50-step checkpoint spacing in that region. These findings indicate that Graph-based Silhouette serves as a diagnostic tool for diffusion-model representation dynamics, providing a sample-level geometric perspective alongside score-function analyses, and that the sample-level geometry of DDPMs admits a compact three-regime description that invites further theoretical and empirical investigation.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: qiya4795@win.su.se. Supervisor: Chun-Biu Li.

Contents

Acknowledgements

I would like to sincerely thank my supervisor, Chun-Biu Li, for his guidance and support throughout this project. He consistently kept my learning and development at the center of his supervision, made himself readily available for discussion, and offered thoughtful, constructive feedback at every stage of the process. His mentorship has greatly deepened my understanding of the subject and contributed to my growth as a researcher.

I would also like to express my gratitude to Nik Tavakolian, whose prior implementation of the Graph-based analysis pipeline served as a valuable reference during the development of this thesis. Consulting his code informed several implementation decisions and helped streamline the development process.

Finally, I wish to acknowledge that while all conceptual content, analyses, and original drafts of figures and text are entirely my own, AI-based tools were used to assist with coding and figure generation, as well as with grammar correction and language polishing throughout the writing process.

Chapter 1

Introduction

Deep learning has emerged as a dominant paradigm within modern machine learning, driven by the strong empirical performance of neural networks on tasks such as image and speech recognition and natural language processing [8,9]. Within this broader development, generative modeling has recently attracted attention, with the goal of learning complex data distributions from which realistic new samples can be drawn. Among existing generative approaches, denoising diffusion probabilistic models (DDPMs) [6] have emerged as one of the dominant paradigms in image generation and form the backbone of production systems such as Stable Diffusion [12].

A DDPM generates data by reversing a gradual stochastic corruption process. A forward Markov chain progressively corrupts each training image with Gaussian noise over T timesteps; a learnt reverse chain, parameterized by a neural network that predicts the noise added at each step, inverts this corruption to transform an initial Gaussian sample into a realistic data point. Despite the empirical success of this framework, the internal behavior of a trained DDPM—how the high-dimensional representation of each intermediate state x_t is geometrically organized relative to the class structure of the training data—remains only partially understood.

Existing theoretical accounts of DDPM dynamics have mostly operated at the population level of the score function. In particular, Biroli et al. [1] derive, via a mean-field analysis, transition times in the diffusion trajectory from spectral properties of the data covariance and identify a *speciation* cross-over at which the generative process commits to one of several data classes. A complementary perspective arises at the finite-sample level: given a collection of intermediate states $\{x_t^{(i)}\}_{i=1}^n$ drawn at some timestep t, how the class structure of the underlying manifold is geometrically organized, and whether it evolves smoothly or exhibits regime-like transitions. This sample-level geometric perspective is the subject of the present thesis.

To answer these questions, we introduce a framework that constructs a k-Nearest Neighbor graph on the intermediate states at each diffusion checkpoint and applies tools from spectral graph theory to quantify class organization. Three such tools, brought together within a single analysis framework originally developed in Tavakolian & Li [16], play complementary roles. The Biharmonic Distance [10] provides a global metric on the graph that respects manifold topology, avoiding the shortcomings of Euclidean distance on curved manifolds. Graph-based Silhouette Analysis, adapted from Rousseeuw [14] by substituting the Biharmonic Distance for Euclidean distance, reveals the class-level organization at each checkpoint. Shape-Aware Stochastic Neighbor Embedding [19] provides a low-dimensional visualization that preserves Biharmonic distances and thereby supports visual confirmation of the quantitative findings. We apply this framework to a U-Net DDPM trained on MNIST, evaluated at 15 checkpoints of the forward and reverse processes on 5000 samples per process.

The thesis contributes the following empirical observations. The central finding is a *three-regime structure* in the evolution of the global Graph-based Silhouette score $S(t)$ across the diffusion trajectory, shared between the forward and reverse processes: a class-dominated regime with $S > 0$ at small t, a transient regime with a pronounced negative minimum at intermediate t, and a noise-dominated regime characterized by a small negative plateau at large t. Beyond this shared three-regime structure, we further observe systematic asymmetries between the forward and reverse trajectories, as well as a class-wise heterogeneity within the reverse transient regime. We also relate the location of the noise-dominated regime to the speciation cross-over predicted by Biroli et al. [1].

The remainder of the thesis is organized as follows. Chapter 2 develops the theoretical and methodological tools: the deep-learning components that underlie the U-Net DDPM backbone, the DDPM framework itself, the graph construction and its Laplacian, the Biharmonic Distance and Graph-PCA, Graph-based Silhouette Analysis, and the embedding-based visualization methods. Chapter 3 reports the empirical findings, beginning with model validation (Section 3.2) and proceeding through the forward (Section 3.3) and reverse (Section 3.4) process analyses, a forward-reverse comparison (Section 3.5), and a summary of findings (Section 3.6). Chapter 4 interprets these findings in relation to existing theoretical frameworks, discusses limitations, and indicates directions for future work.

Chapter 2

Theories and Methods

2.1 Deep Learning Preliminaries

2.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) [8, 9] are a standard model class for computer vision tasks. They differ from fully connected networks in two ways that make them well suited to image data. First, instead of giving every output unit its own weight for every input pixel, a CNN reuses the same small kernel of weights across all spatial positions; this reflects the prior that the same visual pattern can appear anywhere in the image. Second, each output unit in a single convolutional layer depends only on a local $k \times k$ window of input pixels — the *kernel support* of that layer — reflecting the prior that nearby pixels are often more strongly correlated than distant ones, making local feature extraction a useful inductive bias.

Concretely, given an input tensor $\mathbf{X} \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ with C_{in} input channels and spatial size $H \times W$, a convolutional layer parameterized by a weight tensor $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$ (a $k \times k$ kernel for each (output, input)-channel pair) and a bias $\mathbf{b} \in \mathbb{R}^{C_{\text{out}}}$. It produces an output tensor $\mathbf{Y} \in \mathbb{R}^{C_{\text{out}} \times H' \times W'}$ whose (d, i, j)-th entry is

$$y_{d,i,j} = \sum_{c=1}^{C_{\text{in}}} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x_{c,i+m,j+n}\, w_{d,c,m,n} + b_d. \tag{2.1}$$

In words, each output pixel (i, j) in channel d is a weighted sum of the $C_{\text{in}} \times k \times k$ input pixels in its kernel support, with the same weights applied at every spatial position.

Two consequences follow directly. The first is *translation equivariance*: shifting the input by a few pixels shifts the output by the same amount, because the kernel is applied identically everywhere. The second is parameter efficiency: a fully connected layer mapping $H \times W$ inputs to $H \times W$ outputs would need $O(C_{\text{in}} C_{\text{out}} H^2 W^2)$ weights, whereas the convolution uses $O(C_{\text{in}} C_{\text{out}} k^2)$ weights regardless of image size.

A composition of convolutions is itself affine, and any L-layer composition of affine maps reduces to a single affine map. To gain non-linear expressive power, a point-wise nonlinearity $\phi : \mathbb{R} \to \mathbb{R}$ is applied after each convolution. Typical choices are the Rectified Linear Unit $\phi(x) = \max\{0, x\}$ and the Sigmoid Linear Unit (SiLU); the choice adopted in this thesis is discussed in Section 2.1.6. Stacking many such convolution-plus-nonlinearity layers builds a hierarchy of features: early layers respond to local structures (edges, gradients) and deeper layers combine those into larger structures (parts, objects), because each deeper unit's *receptive field* — the region of the original input that can ultimately influence its activation — grows through the composition of convolutional layers, pooling, and striding.

Beyond image classification, CNNs are commonly used for dense pixel-level prediction tasks such as image segmentation, where the output is itself an image of the same spatial size as the input. The U-Net [13], used as the denoising network in this thesis, is one such architecture; it is described in Section 2.1.7.

2.1.2 Residual Connections

Training very deep neural networks is hard because gradients computed by back-propagation can become numerically unstable. The reason is the chain rule: in a simple feed-forward network of L layers $F_L \circ F_{L-1} \circ \cdots \circ F_1$, the gradient of the loss with respect to an intermediate activation $\mathbf{x}$ takes the product form

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \prod_{l=1}^{L} \frac{\partial F_l}{\partial \mathbf{x}}. \tag{2.2}$$

If the individual Jacobians $\partial F_l / \partial \mathbf{x}$ have spectral norm slightly larger than 1, the product grows exponentially in L and the gradient explodes; if the norm is slightly smaller than 1, the product shrinks exponentially and the gradient vanishes. Either way, the parameters in early layers receive almost no useful gradient signal, and adding more layers to the network paradoxically makes it harder to train and harder to fit the training data.

Residual connections [3] address this problem by adding an identity shortcut around each block of layers,

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}, \tag{2.3}$$

where $\mathcal{F}$ is the learnable residual branch, typically a short stack of convolutions with a non-linearity in between. The block now learns the residual $\mathcal{F}(\mathbf{x}) = \mathbf{y} - \mathbf{x}$, that is, the change relative to the input, rather than the full mapping $\mathbf{x} \mapsto \mathbf{y}$. The effect on the gradient is direct: differentiating through the shortcut gives

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathcal{F}}{\partial \mathbf{x}} + I. \tag{2.4}$$

The identity term I provides a direct gradient path back to $\mathbf{x}$ that does not depend on the learned $\mathcal{F}$, which mitigates vanishing gradients, particularly when $\|\partial \mathcal{F} / \partial \mathbf{x}\|$ is small. Stacking many residual blocks therefore alleviates the exponential decay or growth associated with naively stacked layers, which is why architectures such as ResNet [3] can be trained to a depth of hundreds of layers.

When the input and output channel dimensions differ, for instance after a downsampling stage or a channel expansion, the identity shortcut cannot be added directly. The standard fix is to replace it with a linear projection $\mathbf{W}_s$, giving $\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{W}_s \mathbf{x}$, which preserves the additive structure while matching the dimensions.

2.1.3 Group Normalization

Normalization layers are inserted between linear and non-linear operations to ensure the values passing through hidden layers stay within a consistent range. Without normalization, activations can drift to very small or very large values during training; this changes the effective scale of the gradient at each layer and slows down or destabilizes stochastic gradient descent. A normalization layer subtracts an estimated mean and divides by an estimated standard deviation, then applies a learnable parameter rescaling to stretch or shift the values, so that the network can recover any scale it actually needs.

The most common choice is batch Normalization (BN). It achieves this by computing the mean and variance across the mini-batch dimension. However, this introduces an unwanted dependency between samples within the same batch, as the normalization statistics are based directly on the batch size. For diffusion models—where inference frequently requires variable batch sizes, ranging from a single sample to arbitrary chunks—this batch dependence can lead to noisy normalization statistics and unstable training behavior in such settings.

Group Normalization (GN) [20] provides an effective alternative by operating strictly along the channel dimension, thereby decoupling the normalization statistics from the batch size. Formally, GN partitions the C channels into G mutually exclusive groups, each containing C/G channels. Let $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ denote the feature tensor for a single instance, and let $\mathcal{S}_g$ denote the index set of the spatial and channel dimensions corresponding to the g-th group. The normalized feature $x_{c,h,w}$ for $(c, h, w) \in \mathcal{S}_g$ is defined as:

$$\hat{x}_{c,h,w} = \frac{x_{c,h,w} - \mu_g}{\sqrt{\sigma_g^2 + \varepsilon}}, \tag{2.5}$$

where μ_g and σ_g^2 are the empirical mean and variance computed over $\mathcal{S}_g$ for that sample, and $\varepsilon > 0$ is a small constant preventing division by zero. A learnable affine transformation (γ_c, β_c) is then applied channel-wise. Two limiting cases of this scheme are familiar: $G = 1$ recovers layer normalization (one group containing all channels), and $G = C$ recovers instance normalization (each channel is its own group). The choice $G = 32$ is the default of Wu & He [20] and is the one used here.

An important property of this formulation, which is particularly relevant to this study, is that the forward computation of GN is sample-wise: the normalized output for any given realization depends solely on its own feature tensor and the global parameter vector θ, not on the other samples in the same batch (unlike BatchNorm in training mode, where the batch statistics couple samples). This sample-wise property of the forward pass is crucial in Chapter 3 for formulating the reverse process computation: it guarantees that sub-batch partitioning during inference is statistically equivalent to single-batch generation.

2.1.4 Self-Attention

A single convolutional layer has a kernel support bounded by $k \times k$, so dependencies between distant regions of the feature map can only be captured by stacking many layers (which enlarges the cumulative receptive field). Self-attention [18] offers a complementary mechanism: it lets every spatial position directly read information from every other spatial position in a single layer, regardless of how far apart they are.

The intuition is a content-addressable lookup. For each spatial position i, the network forms a *query* vector that summarizes what kind of information i wants to receive; for each spatial position j, it forms a *key* vector summarizing what information j has on offer, and a *value* vector containing that information. Position i then takes a weighted sum of all values, where the weight assigned to position j is high when its key matches i's query and low otherwise.

Given an input representation $\mathbf{Z} \in \mathbb{R}^{N\times d}$, where $N = HW$ denotes the spatial sequence length and d is the embedding dimension, we apply learned linear transformations to map $\mathbf{Z}$ into query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices:

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}_Q, \qquad \mathbf{K} = \mathbf{Z}\mathbf{W}_K, \qquad \mathbf{V} = \mathbf{Z}\mathbf{W}_V, \tag{2.6}$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d\times d_k}$ and $\mathbf{W}_V \in \mathbb{R}^{d\times d_v}$ are the projection weight matrices. The scaled

dot-product attention computes the output as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2.7}$$

The matrix $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top/\sqrt{d_k} \in \mathbb{R}^{N\times N}$ holds the scaled pairwise logits between every (query, key) pair: entry A_{ij} measures how well key j matches query i. Applying the softmax row by row turns each row of $\mathbf{A}$ into a probability distribution over the N positions, so the output for position i is a convex combination of the value vectors $\mathbf{V}_1, \ldots, \mathbf{V}_N$.

The factor $1/\sqrt{d_k}$ keeps the softmax argument numerically well behaved. If the entries of $\mathbf{Q}$ and $\mathbf{K}$ are zero-mean with unit variance and independent across the d_k channels, the dot product $\mathbf{Q}_i\mathbf{K}_j^\top = \sum_{c=1}^{d_k} Q_{i,c}K_{j,c}$ has variance d_k, so the unscaled affinities grow as $\sqrt{d_k}$. Dividing by $\sqrt{d_k}$ rescales them back to unit variance, so that the softmax neither saturates (which would zero out the gradient) nor stays near uniform (which would prevent the network from making distinctions).

Multi-Head Attention (MHA) extends this mechanism by computing h independent attention operation in parallel, each projecting the queries, keys and values into lower-dimensional subspaces. Their outputs are subsequently concatenated and linearly projected. Because the computation of the dense affinity matrix scales quadratically with the sequence length, yielding a time complexity of $O(N^2 d)$, self-attention is computationally expensive for high-resolution images. Therefore, we apply self-attention blocks exclusively at bottleneck feature maps, where the spatial dimensions $N = HW$ is substantially reduced. The exact placement of the attention modules within our architecture is detailed in Section 2.2.3.

2.1.5 Sinusoidal Positional Encoding

The standard U-Net architecture does not inherently process sequential or temporal information, meaning it cannot natively distinguish between different diffusion timesteps t. Since the denoising process must be conditioned on the current noise level, the discrete scalar time $t \in \{0, 1, \ldots, T-1\}$ must be projected into a high-dimensional continuous representation.

We therefore convert the scalar timestep index $t \in \{0, 1, \ldots, T-1\}$ into a continuous d-dimensional vector $\mathbf{e}(t)$ that the network can use as an additional input. The simplest option would be to learn an embedding table that maps each integer t to a free vector, but this gives no inductive bias that nearby timesteps should have similar embeddings; we use instead the sinusoidal positional encoding of Vaswani et al. [18], which has this property by construction. For an even embedding dimension d and frequency index $i \in \{0, 1, \ldots, d/2-1\}$,

$$\mathbf{e}(t)_{2i} = \sin(t\,\omega_i), \tag{2.8}$$

$$\mathbf{e}(t)_{2i+1} = \cos(t\,\omega_i), \tag{2.9}$$

where $\omega_i = 10000^{-2i/d}$ is a geometric sequence of frequencies that decreases with i. The highest frequency occurs at $i = 0$, where $\omega_0 = 1$, giving a sinusoid with period 2π in t; the lowest frequency occurs at $i = d/2-1$, where $\omega_i \approx 10^{-4}$, giving a period on the order of the full diffusion horizon. Each component therefore picks up a different temporal scale, and the concatenated embedding records t at all scales simultaneously.

Two properties of this embedding are used implicitly later. The map $t \mapsto \mathbf{e}(t)$ is componentwise smooth, so $\|\mathbf{e}(t) - \mathbf{e}(t')\|_2^2$ is a smooth function of $t - t'$; in particular, neighboring timesteps receive similar embeddings, which lets the network reuse what it learns at t for $t \pm 1$. The inner product $\langle \mathbf{e}(t), \mathbf{e}(t')\rangle$ depends only on the difference $t - t'$, not on the absolute timesteps, so the network can learn translation-invariant rules of the form "denoise by one step" that work the

same way everywhere in the chain. The specific value of d and the way $\mathbf{e}(t)$ is injected into the U-Net are reported in Section 2.2.3.

2.1.6 Activation Functions

To introduce non-linearity into the architecture, we adopt the SiLU [2,4], formally defined as:

$$\mathrm{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}}, \tag{2.10}$$

where $\sigma(x)$ denotes the standard logistic sigmoid function.

Unlike the widely used ReLU, which is piecewise linear, SiLU is smooth and non-monotonic. The non-monotonicity acts as a soft self-gating mechanism: small negative pre-activations propagate weakly, while large negative values are smoothly suppressed without producing the dead-neuron failure mode of ReLU during training.

The smoothness of SiLU is relevant in the context of diffusion models because the network learns the noise-prediction function $\epsilon_\theta(x_t, t)$, whose regularity is inherited from the activation. The true score $\nabla_x \log p_t(x)$ is itself smooth, since Gaussian convolution gives $p_t \in C^\infty$; the choice of a smooth activation is therefore not motivated by a need to compensate for non-smoothness of the target, but by the requirement that the approximator ϵ_θ match the target's regularity. With ReLU, ϵ_θ has piecewise-linear sections and discontinuous gradients at activation boundaries, which can interact poorly with the iterative reverse-process update $x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z$ over many steps. SiLU gives $\epsilon_\theta \in C^\infty$ in x, matching the regularity class of the target; this is the form Ho et al. [6] adopt and which we follow.

2.1.7 U-Net

The U-Net [13] is a symmetric encoder–decoder convolutional architecture introduced originally for biomedical image segmentation and is now the standard backbone for denoising diffusion models. Its encoder progressively downsamples the input through a sequence of residual blocks while increasing the channel dimension at each stage; its decoder mirrors this architecture with upsampling operations that return the feature maps to the input dimensions. At each scale, a skip connection concatenates the encoder feature map with the decoder feature map of matching spatial dimensions, ensuring that fine-grained spatial information is not lost through the compressed representation at the bottleneck.

The architecture has two parts. The *encoder* reduces the spatial dimensions of the input through a sequence of residual blocks (Section 2.1.2) and pooling operations, while the channel count increases at each stage; this produces a pyramid of feature maps that go from large-and-shallow at the input to small-and-deep at the bottleneck. The *decoder* reverses this cascade with upsampling operations that bring the feature maps back to the original dimensions. The encoder progressively reduces spatial resolution and produces increasingly abstract feature representations, while the skip connections route fine-grained spatial detail from earlier encoder layers directly to the corresponding decoder stages, so that local detail is preserved rather than discarded. The decoder combines these global semantic features with the routed local detail to reconstruct a per-pixel output.

Two properties make the U-Net well suited for the denoising task. First, its input and output share the same spatial dimensions, which is required whenever the network must perform dense pixel-level prediction. Second, The combination of multi-scale representation with skip connections allows the network to integrate coarse structural information with fine-grained detail. This multi-scale structure lets the network handle coarse image structure and fine detail

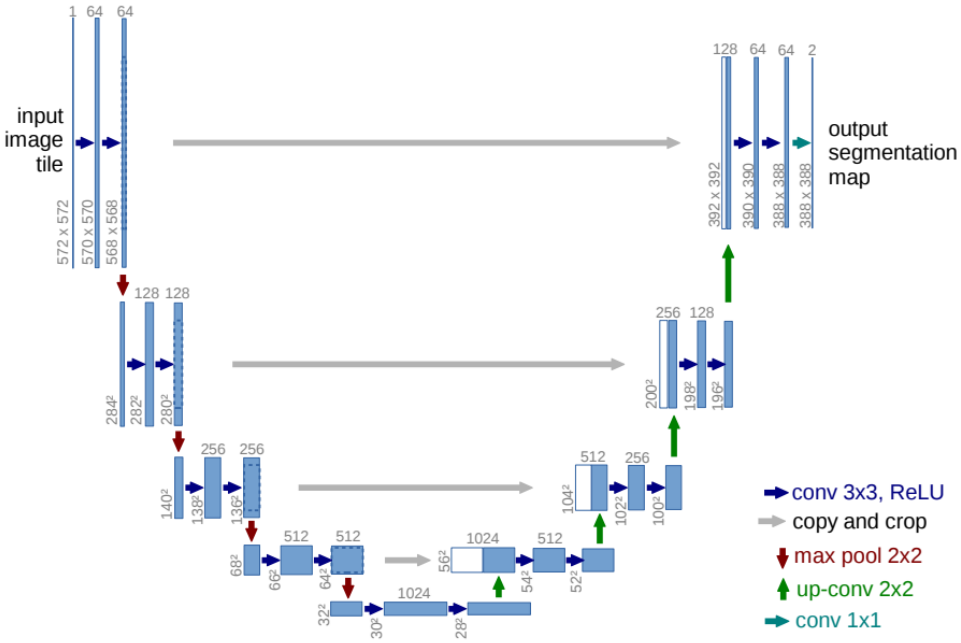


Figure 2.1: The original U-Net architecture [13]. Blue boxes represent multi-channel feature maps, with the number of channels indicated on top and spatial dimensions at the bottom left. White boxes denote copied feature maps, while arrows indicate the respective network operations.

at different dimensions, which is useful for representing intermediate noisy images in which different frequency bands survive to different extents.

A single U-Net is shared across all diffusion timesteps so that the model has the same parameter count regardless of T (Section 2.1.5). The network must therefore be conditioned on the current timestep t; without this conditioning, the network would have to infer the noise level from x_t alone and could not represent timestep-specific denoising rules explicitly. We use the sinusoidal embedding of Section 2.1.5: the integer t is first mapped to a fixed-length vector $\mathbf{e}(t)$, then passed through a two-layer multilayer perceptron to produce a conditioning vector $\mathbf{h}(t)$ of higher dimension. Inside every residual block $\mathbf{h}(t)$ is broadcast across spatial positions and added channel-wise to the intermediate feature map after the first convolution. The additive form is preferred to channel concatenation because it leaves the spatial channel count unchanged at every layer, so the same residual block design works at every scale of the encoder–decoder pyramid. The specific dimensions of $\mathbf{e}(t)$ and $\mathbf{h}(t)$, together with the dimensions at which the self-attention block is placed, are reported in Section 2.2.3.

2.2 Denoising Diffusion Probabilistic Models

The previous section introduced the convolutional, normalization, attention, and encoding mechanisms that together form the U-Net backbone. These components define a generic architecture but do not yet specify a generative task. We now introduce the denoising diffusion probabilistic model, the specific generative framework analyzed in this thesis, together with the training objective that determines its parameters.

2.2.1 Forward Process (Diffusion)

Following the formulation of Ho et al. [6] and Sohl-Dickstein et al. [15], we recall the closed-form construction of the forward diffusion process for the reader's convenience.

The forward process is a fixed (non-learned) Markov chain that approximately maps the data distribution $q(x_0)$ to a known prior—a standard Gaussian—by adding small amounts of independent Gaussian noise over T steps [6]. Because the chain is Markov, its joint distribution factorizes as

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \tag{2.11}$$

with per-step Gaussian transition

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t \mathbf{I}\big), \tag{2.12}$$

where $\{\beta_t\}_{t=1}^{T}$ is a variance schedule with $0 < \beta_t < 1$.

The schedule controls how fast information about x_0 is destroyed. At each step the previous state is rescaled by $\sqrt{1-\beta_t}$ and noise of variance β_t is added, so the signal-to-noise ratio decays multiplicatively through $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. A small β_t at small t keeps the early noisy states close to the data, which gives the network informative training targets in the low-noise regime; if β_t were already large at $t = 1$, the state x_1 would be nearly pure noise and the network would have nothing useful to predict at small t. A larger β_t at large t ensures that the chain reaches the prior $\mathcal{N}(0, I)$ within a finite number of steps, so that generation can begin from a tractable distribution. We adopt the linear schedule of Ho et al. [6], $\beta_t \in [10^{-4}, 0.02]$ over $T = 1000$ steps.

A direct consequence of Eq. (2.12) is that the marginal $q(x_t \mid x_0)$ is itself Gaussian and can be written in closed form, so x_t can be sampled in one step from x_0 without iterating through the intermediate states $x_1, \ldots, x_{t-1}$. The goal of the following derivation is to make this closed-form marginal explicit. The idea is to repeatedly substitute the per-step recursion into itself, collapse the accumulated independent Gaussian noise terms using the additivity of Gaussians, and read off the limiting form.

We begin by rewriting Eq. (2.12) in reparameterized form. Setting $\alpha_t = 1 - \beta_t$,

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{2.13}$$

where ϵ_t is independent of x_{t-1}. This expresses a single forward step as an affine transformation of x_{t-1} plus an independent Gaussian.

Substituting the same identity for x_{t-1} yields a two-step expression containing x_{t-2} and two independent Gaussian terms:

$$x_t = \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\, \epsilon_{t-1} + \sqrt{1-\alpha_t}\, \epsilon_t. \tag{2.14}$$

The two noise terms can be merged into a single Gaussian using a standard property of the multivariate Gaussian: if $\eta_1, \eta_2 \sim \mathcal{N}(0, I)$ are independent and $a, b \in \mathbb{R}$, then $a\,\eta_1 + b\,\eta_2 \sim \mathcal{N}(0, (a^2 + b^2)I)$. Applying this with $a = \sqrt{\alpha_t(1-\alpha_{t-1})}$ and $b = \sqrt{1-\alpha_t}$ gives $a^2 + b^2 = 1 - \alpha_t\alpha_{t-1}$, so Eq. (2.14) reduces to

$$x_t = \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\, \bar{\epsilon}, \qquad \bar{\epsilon} \sim \mathcal{N}(0, I). \tag{2.15}$$

The two-step transition therefore has the same affine-plus-Gaussian-noise form as the one-step transition, with the product $\alpha_t\alpha_{t-1}$ playing the role of the single-step α_t.

Repeating this substitute-and-merge argument t times (induction on the number of substituted steps), each iteration collapses one more pair of independent Gaussian terms into a single Gaussian. Writing $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ gives

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \bar{\epsilon}, \qquad \bar{\epsilon} \sim \mathcal{N}(0, I), \tag{2.16}$$

which is equivalent to the closed-form marginal

$$q(x_t \mid x_0) = \mathcal{N}\big(x_t;\ \sqrt{\bar{\alpha}_t}\, x_0,\ (1 - \bar{\alpha}_t) I\big). \tag{2.17}$$

The closed-form marginal also explains why the chain ends in a near-isotropic Gaussian. For the linear schedule $\beta_t \in [10^{-4}, 0.02]$ over $T = 1000$ steps, $\log \bar{\alpha}_T = \sum_{t=1}^{T} \log(1 - \beta_t) \approx -10$, so $\bar{\alpha}_T \approx 4 \times 10^{-5}$ and $\sqrt{\bar{\alpha}_T} \approx 6 \times 10^{-3}$. Substituting into Eq. (2.17), the mean $\sqrt{\bar{\alpha}_T}\, x_0$ is negligible relative to the variance $(1 - \bar{\alpha}_T) \approx 1$ for any bounded x_0, so $q(x_T \mid x_0) \approx \mathcal{N}(0, I)$ irrespective of the data point. Marginalizing over $q(x_0)$ then gives $q(x_T) \approx \mathcal{N}(0, I)$. This is the property that lets the reverse chain in Section 2.2.2 start from a sample of the standard Gaussian rather than from a learned prior.

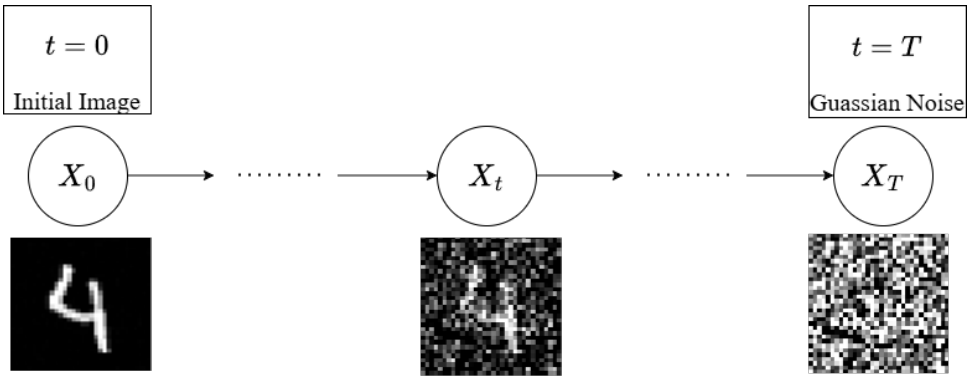


Figure 2.2: Forward diffusion chain on MNIST. The diagram shows successive states $x_0, x_1, \ldots, x_T$ as image thumbnails; the arrows mark the per-step transition $q(x_t \mid x_{t-1})$ in Eq. (2.12). From left to right, the structure of the digit image is gradually replaced by Gaussian noise, and at $t = T$ the state is indistinguishable from a sample of $\mathcal{N}(0, I)$.

2.2.2 Reverse Process (Denoising)

The reverse process is a learned Markov chain that is trained to transform Gaussian noise $x_T \sim \mathcal{N}(0, I)$ into samples whose distribution approximates the data distribution $q(x_0)$, by progressively removing noise step by step. The joint distribution is factorized as [6]

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \tag{2.18}$$

with each reverse transition parameterized as Gaussian,

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t)\big). \tag{2.19}$$

Following Ho et al. [6], the covariance is fixed to $\Sigma_\theta(x_t, t) = \sigma_t^2 I$ with $\sigma_t^2 = \beta_t$; only μ_θ is learned.

When the clean sample x_0 is known, Bayes' rule applied to the forward Markov chain gives an exact Gaussian for the reverse conditional [6],

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1};\ \tilde{\mu}_t(x_t, x_0),\ \tilde{\beta}_t I), \tag{2.20}$$

with posterior mean

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t, \tag{2.21}$$

and posterior variance $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. This expression requires access to x_0 and is therefore unavailable at generation time; it serves as the reference distribution that the learned reverse kernel p_θ is trained to match.

The closed-form forward marginal $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$ can be solved for x_0 in terms of x_t and the noise ϵ. Substituting into $\tilde{\mu}_t$ and absorbing constants yields [6]

$$\tilde{\mu}_t(x_t, \epsilon) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon \right). \tag{2.22}$$

Replacing the unknown ϵ with a neural network $\epsilon_\theta(x_t, t)$ and defining the parameterized mean in the same form gives

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t) \right). \tag{2.23}$$

Training therefore reduces to learning a noise-prediction network ϵ_θ. Predicting the added noise is mathematically equivalent to parameterizing the reverse-process mean μ_θ, while yielding a simpler and empirically more stable training objective.

The variational upper bound on the negative log-likelihood decomposes into per-timestep KL terms [6],

$$L = \mathbb{E}_q \Big[D_{\mathrm{KL}}(q(x_T \mid x_0) \,\|\, p(x_T)) + \sum_{t>1} D_{\mathrm{KL}}\big(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\big) - \log p_\theta(x_0 \mid x_1) \Big]. \tag{2.24}$$

With fixed σ_t^2 the KL between two Gaussian with equal covariance reduces to the squared mean difference, and substituting the noise-prediction parameterization yields the simplified loss of Ho et al. [6],

$$L_{\mathrm{simple}}(\theta) = \mathbb{E}_{t, x_0, \epsilon} \Big[\big\| \epsilon - \epsilon_\theta\big(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\, t\big) \big\|^2 \Big], \tag{2.25}$$

with t drawn uniformly from $\{1, \ldots, T\}$ and $\epsilon \sim \mathcal{N}(0, I)$. This is the objective used to train the U-Net in this thesis.

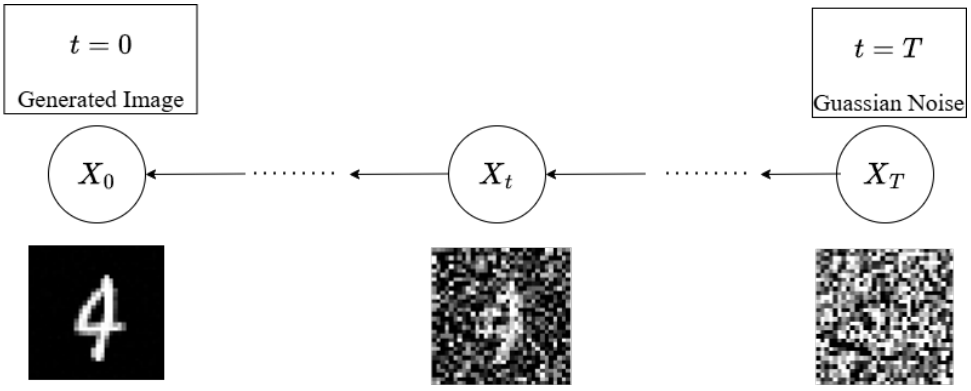


Figure 2.3: Reverse denoising chain on MNIST. The diagram shows successive states $x_T, x_{T-1}, \ldots, x_0$ as image thumbnails arranged left to right; each arrow marks the learned reverse transition $p_\theta(x_{t-1} \mid x_t)$ defined in Eq. (2.18). From left to right, structure is gradually recovered by the network, ending in a sample x_0 drawn from the learned distribution $p_\theta(x_0)$.

2.2.3 Model Architecture

The DDPM U-Net [6] used in this study is implemented according to the U-Net architecture introduced in Section 2.1.7. The specific configuration is shown in the table below:

Table 2.1: Configuration of the DDPM U-Net used throughout this thesis. The table lists, in order, the input shape, the base channel count, the channel multipliers across the encoder–decoder stages, the resulting feature-map resolutions, the placement and head count of self-attention blocks, the timestep-embedding pipeline, the normalization choice, and the total parameter count. The configuration follows Ho et al. [6] adapted to the 32×32 padded MNIST input.

Component	**Configuration**	**Reference**
Input	$1 \times 32 \times 32$	—
Base channels	128	—
Channel multipliers	$(1, 2, 2)$	—
Feature map dimensions	$32 \rightarrow 16 \rightarrow 8 \rightarrow 4$	—
Self-Attention blocks	16×16, 4 heads	[18]
Time embedding	sinusoidal $\rightarrow$ MLP	[18]
Normalization	GroupNorm (32 groups)	[20]
Total parameters	18.7M	—

The network operates on zero-padded MNIST inputs of size $1 \times 32 \times 32$ and uses four feature-map dimensions, from 32×32 down to 4×4, obtained by three successive $2\times$ downsampling stages. Channel multipliers $(1, 2, 2)$ applied on top of a base width of 128 yield channel counts of $[128, 256, 256]$ across stages. A single self-attention block is inserted at the 16×16 dimensions with four heads; higher-dimensions attention is omitted to keep the quadratic cost manageable. The diffusion timestep t is injected into every residual block through a 128-dimensional sinusoidal embedding followed by a two-layer MLP that projects it to 512 dimensions, consistent with the conditioning scheme of Ho et al. [6]. All convolutional blocks use GroupNorm with 32 groups and SiLU activation. The resulting network has 18.7M parameters in total.

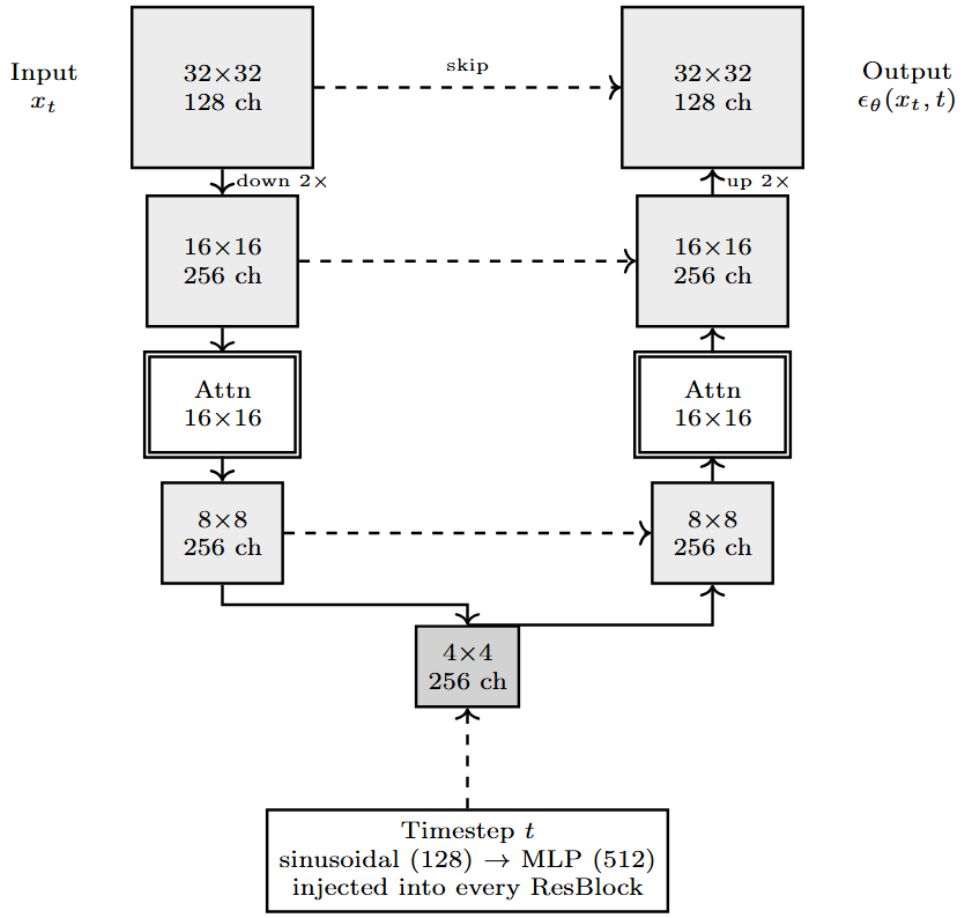


Figure 2.4: Schematic of the DDPM U-Net architecture. The encoder (left half) downsamples the input through residual blocks at resolutions $32 \to 16 \to 8 \to 4$ with channel counts $[128, 256, 256]$; the decoder (right half) mirrors this cascade with upsampling and concatenates encoder feature maps via skip connections at each scale; a single self-attention block is inserted at the 16×16 resolution. The diffusion timestep t is embedded sinusoidally and broadcast as an additive bias inside every residual block. The architecture has the same input and output spatial resolution, which is required for per-pixel noise prediction.

2.2.4 Training Protocol

MNIST images are natively 28×28, a size which does not divide evenly under the three successive factor-of-two down samplings of our U-Net. We zero-pad each image symmetrically to 32×32 before passing it to the network, which yields the clean dimensions cascade $32 \to 16 \to 8 \to 4$. Pixel intensities, originally in $[0, 1]$ after rescaling by $1/255$, are then linearly mapped to $[-1, 1]$ via $x \mapsto 2x - 1$, matching the input convention of Ho et al. [6]. The 60000 training images are partitioned into a 54000-image training subset and a 6000-image held-out validation subset, with a fixed random seed $s = 42$ ensuring reproducibility of the split.

Optimization uses Adam [7] with learning rate 10^{-4}, default momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, and no weight decay, on mini-batches of size 64 for up to 50 epochs. Validation loss is evaluated after every epoch; training is halted early when the validation loss ceases to improve, and the epoch attaining the lowest validation loss is retained for all subsequent analyses.

The forward noise schedule follows Ho et al. [6]: β_t increases linearly from 10^{-4} to 0.02 over $T = 1000$ steps. This matches the setting under which DDPM was originally reported on

MNIST and allows our quantitative results to be compared directly with that baseline.

2.3 Graph Construction and Laplacian

With the DDPM trained, its intermediate states $\{x_t\}$ at each diffusion timestep are well-defined objects, but the geometric organization of class structure inside these states is not directly accessible from the samples themselves. To render this organization tractable for quantitative analysis we adopt a graph-theoretic representation, in which each intermediate state contributes a vertex and local proximity in the high-dimensional space is encoded as weighted edges.

A weighted graph is a triple $G = (V, E, W)$, where $V = \{v_1, \ldots, v_n\}$ is a finite vertex set (each vertex representing one sample in this study), $E \subseteq V \times V$ is an edge set, and $W \in \mathbb{R}^{n \times n}$ is a symmetric non-negative adjacency matrix whose (i, j) entry quantifies the similarity between v_i and v_j (with $W_{i,j} = 0$ when $(v_i, v_j) \notin E$). In the following three subsections we construct such a graph from a set of high-dimensional feature vectors: Section 2.3.1 describes the k-Nearest Neighbor procedure, Section 2.3.2 specifies the weighting W together with the degree matrix D, and Section 2.3.3 introduces the Laplacian operators for spectral analysis.

2.3.1 k-Nearest Neighbor Graph Construction

To make the geometric organization of the intermediate states tractable for spectral analysis, we represent the $n = 5000$ feature vectors $\{v_i\}_{i=1}^{n}$ at each checkpoint as the vertices of a k-nearest-neighbor (kNN) graph: each vertex is connected to its k closest neighbors under the Euclidean distance, and the resulting directed adjacency is symmetrized by entry-wise maximum so that the spectral analysis operates on an undirected graph. A fixed k is unsuitable across the diffusion trajectory: at $t = 0$ the data lie in ten compact clusters and a small k suffices, whereas at large t the feature vectors approach isotropic Gaussian noise and may require a larger k to maintain global connectivity. Following Tavakolian & Li [16], we therefore use an adaptive scheme that increases k from $k_{\min} = 5$ until the symmetrized graph becomes globally connected, capping at $k_{\max} = 30$; the full procedure is given as Algorithm 1.

The reason Algorithm 1 insists on global connectivity is that the Biharmonic Distance (Section 2.4.1) is undefined on a disconnected graph. The multiplicity of the zero eigenvalue of the graph Laplacian equals the number of connected components; a disconnected graph therefore has at least two zero eigenvalues, which leaves the BHD formula $d_B^2(i, j) = \sum_{k \geq 2} (\phi_k(i) - \phi_k(j))^2 / \lambda_k^2$ ill-posed. The adaptive scheme therefore guarantees that the spectral analysis of subsequent sections is well-defined at every checkpoint examined in this study. In practice $k = 5$ already yields a connected graph at every checkpoint in our experiments, a result reported in Chapter 3.

Algorithm 1 guarantees only global connectivity; it does not guarantee connectivity *within each class sub-graph*. If the sub-graph induced on the vertices of some class C_c is disconnected, then for two points $i, j \in C_c$ in different components the within-class BHD is undefined, which would leave the class-wise Silhouette and separation analyses of Chapter 3 ill-defined for the affected classes.

To prevent this, we check the connectivity of each class sub-graph after global kNN construction. For any class whose sub-graph is disconnected, additional edges are added by running an in-class kNN search restricted to the vertices of that class, with the same bandwidth parameter a as the global graph. New edges are merged with existing ones by taking the maximum weight, so that pre-existing strong connections are never overwritten by weaker intra-class additions.

Algorithm 1 Adaptive k-nearest-neighbor graph construction. The procedure starts at $k = k_{\min} = 5$ and increments k until the symmetrized graph is connected, capping at $k_{\max} = 30$; if no k in this range yields a connected graph, the largest connected component is retained and the remaining vertices are flagged as outliers. Adaptive k is necessary because a fixed k that suffices at $t = 0$ (compact class clusters) may leave the graph disconnected at large t when feature vectors approach isotropic Gaussian noise.

Require: feature vectors $\{v_i\}_{i=1}^{n}$; bounds $k_{\min} = 5$, $k_{\max} = 30$
Ensure: symmetric adjacency A; used k; main component V_{main}

compute Euclidean distance matrix $\mathbf{D}$
$k \leftarrow k_{\min}$
while $k \leq k_{\max}$ **do**
  $A^{\text{dir}} \leftarrow$ directed kNN adjacency from $\mathbf{D}$
  $A \leftarrow \max(A^{\text{dir}}, (A^{\text{dir}})^{\top})$ ▷ symmetrize
  if A is connected **then**
    return $A,\ k,\ V_{\text{main}} = \{1, \ldots, n\}$
  end if
  $k \leftarrow k + 1$
end while
$V_{\text{main}} \leftarrow$ largest connected component of A
return $A,\ k_{\max},\ V_{\text{main}}$

2.3.2 Adjacency Matrix and Edge Weighting

The kNN procedure of Section 2.3.1 specifies which vertex pairs are connected. To turn the resulting unweighted adjacency into a similarity structure, we convert each retained Euclidean distance d_{ij} into a scalar edge weight via

$$w_{ij} = \frac{1}{\left(\dfrac{d_{ij}}{a} + 1\right)^2}, \tag{2.26}$$

where $a > 0$ is a bandwidth parameter set below. The entries w_{ij} populate the symmetric weighted adjacency matrix $W \in \mathbb{R}^{n \times n}$, with $w_{ij} = 0$ whenever (v_i, v_j) is not among the kNN edges.

The polynomial decay of this kernel is preferred to a Gaussian kernel $\exp(-d_{ij}^2/2\sigma^2)$ in the present setting. At high diffusion timesteps (for example $t = 900$), all pairwise distances are inflated by accumulated Gaussian perturbation, and a Gaussian kernel would assign near-zero weight to essentially every edge, suppressing the differential structure of the graph. The polynomial form decays more slowly and preserves a meaningful ordering among edges even at these noise levels.

The bandwidth a is estimated from the kNN edge-distance distribution at each checkpoint:

$$a = P_{90}(\mathbf{D}\text{k}NN) \,/\, 9, \tag{2.27}$$

with P_{90} the 90th percentile of all kNN edge distances. Using P_{90} rather than the maximum prevents a single outlier edge from setting the scale. The divisor 9 is not chosen ad hoc: it follows from requiring the 90th percentile of the empirical kNN edge-distance distribution to align with the 90th percentile of a reference Pareto distribution, a calibration derived in the supplementary material of Tavakolian & Li [16]. A useful operational consequence is that an edge of length $d_{ij} = P_{90}$ then receives weight $w_{ij} = 1/(9+1)^2 = 10^{-2}$, so the bulk of retained

edges spans roughly two decades of weight magnitude. This calibration preserves a meaningful ordering among edges at every diffusion noise level examined in this thesis.

The connectivity of vertex v_i is summarized by its weighted degree $d_i = \sum_{j=1}^{n} W_{ij}$. Collecting these degrees on the diagonal of an $n \times n$ matrix yields the degree matrix

$$D = \mathrm{diag}(d_1, \ldots, d_n). \tag{2.28}$$

The pair (W, D) provides all the information needed to construct the Laplacian operator in the next subsection.

2.3.3 Graph Laplacian

The graph Laplacian is the operator through which the topology of G is converted into algebraic quantities for spectral analysis. Following the framework of Tavakolian & Li [16], the motivation for its use here is that the eigenspectrum of the Laplacian provides information about two complementary properties of the underlying data manifold: (i) its intrinsic dimensionality, and (ii) the existence of clusters or communities in the data. The first property underpins the Graph-PCA analysis of Section 2.4.2, which estimates the effective dimensionality of intermediate states; the second underpins both the Biharmonic Distance (Section 2.4.1) and the Graph-based Silhouette Analysis built on it (Section 2.5.1), which quantify class-level organization.

Two algebraic properties of the Laplacian are useful to record before stating its definitions, because they make these connections concrete. First, the Laplacian quadratic form admits the identity

$$x^\top L x = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \, (x_i - x_j)^2, \qquad x \in \mathbb{R}^n, \tag{2.29}$$

which measures the Dirichlet energy of a function x defined on G: the right-hand side is small when x varies slowly along edges with large weight, and large when x changes abruptly across well-connected vertex pairs. The identity also makes the positive semi-definiteness of L immediate, since the right-hand side is a sum of non-negative terms. Second, the same smoothness interpretation transfers to the spectrum: the eigenvector ϕ_k associated with a small eigenvalue λ_k varies slowly along the graph and captures global structure (relevant to intrinsic dimensionality and inter-class organization), whereas eigenvectors associated with large eigenvalues capture localized, high-frequency variation. As already used in Section 2.3.1, the multiplicity of the eigenvalue 0 equals the number of connected components of G, so the bottom of the spectrum encodes coarse topological information directly. The Biharmonic Distance, introduced in the next section, is defined through an inverse-square weighting of the Laplacian spectrum, which makes it emphasize precisely this global, low-frequency end.

With this motivation in place, we now state the definitions used in subsequent sections. The unnormalized graph Laplacian is defined as

$$L = D - W, \tag{2.30}$$

where W is the weighted adjacency matrix of Section 2.3.2 and D is the corresponding degree matrix. Vertices with large degree dominate the spectrum of L. Normalizing by the degrees removes this bias:

$$L_{\mathrm{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}. \tag{2.31}$$

Under this normalization each edge weight W_{ij} is divided by $\sqrt{d_i d_j}$, so nodes with high and low degree contribute comparably to the spectrum. For some applications it is also convenient to work with the random-walk Laplacian

$$L_{\text{rw}} = I - D^{-1}W, \tag{2.32}$$

which is similar to L_{sym} via $L_{\text{rw}} = D^{-1/2}L_{\text{sym}}D^{1/2}$ and therefore shares the same spectrum $\{\lambda_k\}_{k=1}^{n}$. Writing ϕ_k for the orthonormal eigenvectors of L_{sym}, the corresponding right eigenvectors of L_{rw} are

$$\psi_k := D^{-1/2}\phi_k, \qquad L_{\text{rw}}\psi_k = \lambda_k\psi_k. \tag{2.33}$$

Both ϕ_k and ψ_k formulations are used in the Biharmonic Distance of the next section.

The symmetric normalized Laplacian L_{sym} retains the smoothness/topology correspondence above and is the operator used in the next section to define the Biharmonic Distance.

2.4 Graph-based Distance and Embedding

The graph constructed in the previous section captures local proximity at a single diffusion checkpoint, but comparing class structure across checkpoints requires a global distance metric whose scale remains meaningful even as the underlying graph changes. The following two subsections introduce the Biharmonic Distance, derived from the graph Laplacian and therefore respecting the graph topology, together with a companion principal-component analysis that quantifies effective dimensionality.

2.4.1 Biharmonic Distance

The graph G constructed in Section 2.3 encodes the local geometry of the feature space. A distance between vertices that respects the global topology is needed, since the Euclidean distance in the high-dimensional space ignores the manifold structure that the graph approximates. We use the Biharmonic Distance (BHD) [10], derived from the graph Laplacian.

The intuition behind the BHD is most easily seen by contrast with the classical *commute time distance*, $c(i, j)$, defined as the expected time for a random walk on G to travel from vertex i to vertex j and back. Commute distance admits the spectral expansion

$$c(i, j) = \text{vol}(G) \sum_{k=2}^{n} \frac{\big(\psi_k(i) - \psi_k(j)\big)^2}{\lambda_k}, \tag{2.34}$$

where $\text{vol}(G) = \sum_{i=1}^{n} d_i$ and $\{(\lambda_k, \psi_k)\}_{k=2}^{n}$ are the non-trivial eigen-pairs of the random-walk Laplacian introduced in Section 2.3.3. Two features of this expansion are worth noting. First, every non-trivial eigenmode contributes, so $c(i, j)$ is an implicit average over many paths between i and j, in contrast to the shortest-path distance, which depends only on a single route. Second, smaller eigenvalues—corresponding to slower-mixing, more global modes—are weighted more heavily by the factor $1/\lambda_k$, so commute distance already places more emphasis on global topology than on local detail. Its main practical limitation is that on large graphs $c(i, j)$ tends to concentrate around a function of the local degrees of i and j, so distances between distant points become uninformative.

The Biharmonic Distance addresses this limitation by squaring the inverse-eigenvalue

weighting,

$$d_B^2(i,j) = \mathrm{vol}(G) \sum_{k=2}^{n} \frac{\left(\psi_k(i) - \psi_k(j)\right)^2}{\lambda_k^2} \tag{2.35}$$

$$= \mathrm{vol}(G) \sum_{k=2}^{n} \frac{\left(d_i^{-1/2}\phi_k(i) - d_j^{-1/2}\phi_k(j)\right)^2}{\lambda_k^2}, \tag{2.36}$$

using the relation $\psi_k = D^{-1/2}\phi_k$ recalled in Section 2.3.3. The sum starts from $k = 2$ because ϕ_1 is proportional to $D^{1/2}\mathbf{1}$ (the kernel of L_{sym}) and contributes zero to every pairwise term; it is well-defined because Algorithm 1 guarantees a connected graph, so $\lambda_2 > 0$.

The extra factor $1/\lambda_k$ in Eq. (2.35) amplifies the contribution of low-frequency modes that already encode the global topology of G, and simultaneously dampens the high-frequency modes whose accumulation makes commute distance unreliable at large scales. The BHD therefore inherits the multi-path averaging and global-topology emphasis of commute distance, while remaining accurate for distant points on the data manifold [10, 16]—which is what makes it the natural choice for comparing manifold structure across the diffusion checkpoints in this thesis.

The definition in Eq. (2.35) admits an explicit embedding representation: collect the non-zero eigen-pairs into $\Phi_{\mathrm{nz}} = [\phi_2, \ldots, \phi_n]$ and $\Lambda_{\mathrm{nz}} = \mathrm{diag}(\lambda_2, \ldots, \lambda_n)$, and set

$$X = \sqrt{\mathrm{vol}(G)} \cdot D^{-1/2}\, \Phi_{\mathrm{nz}}\, \Lambda_{\mathrm{nz}}^{-1}. \tag{2.37}$$

Writing $X_{i,k} = \sqrt{\mathrm{vol}(G)}\, d_i^{-1/2}\phi_k(i)/\lambda_k$ for the entry of X in row i and column k, the squared Euclidean distance between two rows expands as

$$\|X_i - X_j\|_2^2 = \sum_{k=2}^{n} \left(X_{i,k} - X_{j,k}\right)^2 = \mathrm{vol}(G) \sum_{k=2}^{n} \frac{\left(d_i^{-1/2}\phi_k(i) - d_j^{-1/2}\phi_k(j)\right)^2}{\lambda_k^2} = d_B^2(i,j), \tag{2.38}$$

so that $d_B(i,j) = \|X_i - X_j\|_2$ exactly.

The scale of the BHD depends on the edge-weight distribution, which varies across checkpoints as the underlying data distribution evolves. Comparing quantities across timesteps without accounting for this variation would conflate changes in manifold structure with changes in overall distance scale. We therefore adopt the mean-normalized BHD,

$$d_{NB}(i,j) = \frac{d_B(i,j)}{\dfrac{2}{n(n-1)} \displaystyle\sum_{1 \le i < j \le n} d_B(i,j)}, \tag{2.39}$$

so that the graph-wide mean pairwise distance is one. The Silhouette is already scale-invariant by construction (ratio of distances), so this normalization does not alter S. The class-wise separation $b_c(t)$ reported in Chapter 3, however, is not scale-invariant; the normalization ensures b_c is comparable across checkpoints.

The BHD is computed from the weighted adjacency W as follows. Given $W \in \mathbb{R}^{n \times n}$, we form the degree matrix $D = \mathrm{diag}(W\mathbf{1})$ and the symmetric normalized Laplacian $L_{\mathrm{sym}} = I - D^{-1/2}WD^{-1/2}$. An eigen decomposition of L_{sym} is then computed, yielding eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and orthonormal eigenvectors $\phi_1, \ldots, \phi_n$. All spectral computations use double precision: single precision introduces numerical error comparable in magnitude to the smallest informative eigenvalues, which distorts the $1/\lambda_k^2$ weighting. We discard the single zero eigenvalue corresponding to the connected component; in all checkpoints examined, exactly one eigenvalue is identified as zero, confirming that the graph is a single connected component at each diffusion timestep.

2.4.2 Graph-based Principal Component Analysis

To quantify the effective dimensionality of the data at each diffusion checkpoint we apply a Graph-based variant of Principal Component Analysis (PCA) [11], following Tavakolian & Li [16]. Standard PCA computes the principal directions as eigenvectors of the sample covariance matrix in Euclidean coordinates; the data in this study lie on a non-linear manifold in high-dimensional space, so replacing the Euclidean distance with the BHD yields a variant capable of handling non-linear structure, which we refer to as Graph-based PCA.

The key observation is that the BHD embedding X defined in Eq. (2.37),

$$X = \sqrt{\mathrm{vol}(G)} \cdot D^{-1/2}\, \Phi_{\mathrm{nz}}\, \Lambda_{\mathrm{nz}}^{-1}, \tag{2.40}$$

is already expressed in the basis of Laplacian eigenmodes: its k-th column (for $k = 2, \ldots, n$) is proportional to the eigenvector ϕ_k of L_{sym}, scaled by $1/\lambda_k$. Equivalently, the columns of X are aligned with the right eigenvectors $\psi_k = D^{-1/2}\phi_k$ of the random-walk Laplacian (Section 2.3.3). Because the ψ_k are orthonormal in the degree-weighted inner product $\langle u, v\rangle_D = \sum_i d_i\, u_i v_i$, the embedding X is *already diagonalized*: no Gram matrix, no pairwise-distance matrix, and no additional eigen decomposition is required beyond the spectrum of L_{sym} already computed in Section 2.3.3. Each column of X may therefore be identified directly with one principal axis of the BHD-embedded data.

The variance captured along the k-th principal axis follows directly from the embedding formula: in the degree-weighted inner product the columns of X satisfy $\|X_{\cdot,k}\|_D^2 = \mathrm{vol}(G)/\lambda_k^2$, so we define the principal variance

$$\mu_k = \frac{\mathrm{vol}(G)}{\lambda_k^2}, \qquad k = 2, \ldots, n. \tag{2.41}$$

Because $\lambda_2 \le \lambda_3 \le \cdots \le \lambda_n$, the principal variances are ordered as $\mu_2 \ge \mu_3 \ge \cdots \ge \mu_n$, with the dominant mode μ_2 corresponding to the smallest non-zero Laplacian eigenvalue—the slowest-mixing, most global mode of G, consistent with the spectral interpretation in Section 2.3.3.

To summarize the effective dimensionality at each checkpoint with a single scalar, we define the PC Proportion at threshold $\alpha \in (0, 1)$,

$$\mathrm{PCP}(\alpha) = \frac{1}{n-1} \min\left\{ K \;:\; \frac{\sum_{k=2}^{K+1} \mu_k}{\sum_{k=2}^{n} \mu_k} \ge \alpha \right\}. \tag{2.42}$$

The denominator $n - 1$ is the number of non-trivial principal axes available after the kernel eigenvector ϕ_1 is discarded, and ensures $\mathrm{PCP}(\alpha) \in (0, 1]$. A small $\mathrm{PCP}(\alpha)$ means that a small fraction of the available principal components already captures a large share of the BHD variance, consistent with a compact low-dimensional manifold structure; a large $\mathrm{PCP}(\alpha)$ indicates variance spread across many components, consistent with a near-isotropic high-dimensional arrangement. The PC Proportion is evaluated at three thresholds $\alpha \in \{0.7, 0.8, 0.9\}$ and reported in Chapter 3 as a scalar summary of effective dimensionality across diffusion checkpoints.

2.5 Cluster Validation based on Graph Distances

The BHD and the PC Proportion quantify geometric properties of each intermediate state x_t, but they do not test whether class structure is preserved. To probe class-level separation directly we adopt the Silhouette criterion, which measures the extent to which each sample is closer to members of its own MNIST class than to members of any other. We first review the classical formulation, then derive its Graph-based variant by substituting the BHD for Euclidean distance.

2.5.1 Silhouette Analysis

A core question of this thesis is whether and how the class organization of the intermediate states changes with t in both the forward and reverse processes. To answer this question, we demand a single cluster-quality score that can be computed per data point, aggregated per class, summarized globally, and compared across checkpoints. The Silhouette score introduced by Rousseeuw [14] is designed for precisely this role.

Suppose the n data points are partitioned into K clusters $C_1, \ldots, C_K$ with $|C_c| = N_c$. For a point $i \in C_c$ and a generic pairwise distance δ_{ij}, let

$$a(i) = \frac{1}{N_c - 1} \sum_{\substack{j \in C_c \\ j \neq i}} \delta_{ij}, \tag{2.43}$$

$$b(i) = \min_{l \neq k} \frac{1}{N_l} \sum_{j \in C_l} \delta_{ij}. \tag{2.44}$$

$a(i)$ measures the average distance of i to the remainder of its own cluster, while $b(i)$ measures the average distance to the nearest competing cluster. The Silhouette value of point i is then

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \qquad s(i) \in [-1, 1]. \tag{2.45}$$

The sign and magnitude of $s(i)$ admit a direct geometric reading. A *positive* Silhouette, $s(i) > 0$, indicates *local class coherence*: point i is closer, on average, to members of its own class than to any other class, with $s(i) \to 1$ in the limit of complete cluster separation. A *near-zero* Silhouette, $s(i) \approx 0$, signals *overlapping class geometry*: the within-class and between-class average distances are comparable, placing i on or near the boundary between its labeled class and the nearest neighboring class. A *negative* Silhouette, $s(i) < 0$, indicates *stronger inter-class than intra-class proximity*: on average i lies closer to members of another class than to its own, with $s(i) \to -1$ in the limit of complete misplacement.

2.5.2 Graph-based Silhouette

The Silhouette formulation of Section 2.5.1 treats the pairwise distance δ_{ij} as the Euclidean distance. Using the Euclidean distance directly in the high-dimensional non-linear feature space produces short-cuts: remote points on the underlying manifold can appear close under the straight-line distance.

A distance that reflects the manifold structure is therefore needed to replace the Euclidean distance. The BHD of Section 2.4.1 meets this requirement. Because d_{NB} is computed from the Laplacian spectrum, it implicitly averages over multiple paths between two nodes [16, 19]; two points have small d_{NB} when they are strongly connected in the graph and large d_{NB} when the connection between them is weak [19]. Substituting the normalized BHD for the Euclidean distance in the Silhouette of Section 2.5.1,

$$\delta_{ij} := d_{NB}(i, j), \tag{2.46}$$

yields the *Graph-based Silhouette*. All definitions of $a(i), b(i), s(i)$ carry over unchanged.

Graph-based Silhouette Analysis is computed from the mean-normalized distance matrix d_{NB} of Section 2.4.1 together with a cluster label assignment. For the forward process we use the ground-truth MNIST labels, and the resulting score measures how well the true class structure is preserved as noise accumulates. For the reverse process, labels cannot be prescribed

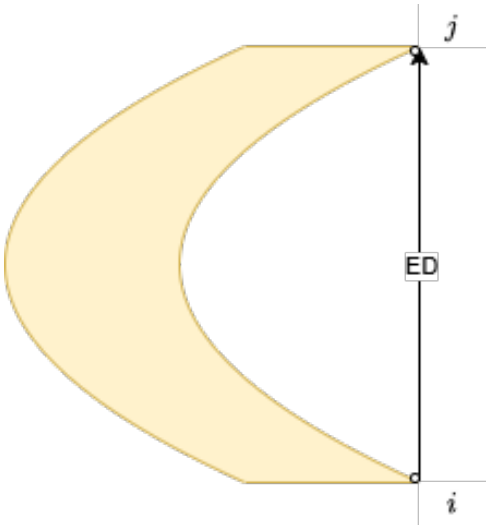


Figure 2.5: Illustration of the short-circuit problem of the Euclidean distance on a curved manifold. Two points lying on opposite sides of a folded one-dimensional manifold (red curve) appear close in straight-line Euclidean distance (dashed black arrow) but are far apart when measured along the manifold. The BHD defined on the kNN graph (Section 2.4.1) avoids this short-circuit by measuring distances through edges of the graph rather than through the ambient space.

at initialization; each generated trajectory is classified once at its near-clean state x_1 (for which $\bar{\alpha}_1 \approx 0.9999$ and the image is visually indistinguishable from x_0) using a pretrained CNN classifier trained on MNIST, and the assigned label is fixed for that trajectory and applied identically at every checkpoint (Section 3.1.4). The score then measures how well the model-assigned class structure emerges during denoising.

The score is aggregated at two levels. The class-wise Silhouette

$$S_c(t) = \frac{1}{N_c} \sum_{i \in C_c} s(i) \tag{2.47}$$

summarizes cluster quality within class C_c, and the global Silhouette is obtained by averaging over classes,

$$S(t) = \frac{1}{K} \sum_{c=1}^{K} S_c(t), \tag{2.48}$$

following the convention of Wängberg et al. [19] and Tavakolian & Li [16]. Reporting S as an average over classes rather than over individual points ensures that every class contributes equally to the global score, which matters when class sizes are unbalanced (as they are in the reverse process; see Section 3.4.2).

In Chapter 3 we additionally report, for each class, the class-wise cohesion and separation

$$a_c(t) = \frac{1}{N_c} \sum_{i \in C_c} a(i), \qquad b_c(t) = \frac{1}{N_c} \sum_{i \in C_c} b(i), \tag{2.49}$$

so that the two components driving $S_c(t)$ can be read directly from the data. When a single scalar is needed to summarize the overall trend, we use the class-averaged values $\bar{a}(t) = K^{-1} \sum_c a_c(t)$ and $\bar{b}(t) = K^{-1} \sum_c b_c(t)$.

2.6 Graph-based Embedding Methods

The Silhouette summarizes class-level organization at each checkpoint in a single scalar, which is convenient for quantitative comparison but provides no direct picture of how the underlying geometry is arranged. To complement the scalar summary with a spatial representation that preserves distances, we use low-dimensional embedding methods. This section introduces t-distributed Stochastic Neighbor Embedding and its shape-aware variant, which replaces the Euclidean similarity in t-SNE with the BHD of Section 2.4.1.

We emphasize that throughout this thesis the embedding plots produced by these methods serve as *qualitative* visual support for the quantitative findings reported in Chapter 3: every substantive conclusion is anchored in the Silhouette, the PC Proportion, or the Biharmonic Distance analyses, and the embedding figures are used only to make the underlying geometry visually concrete. Any low-dimensional projection of high-dimensional manifold structure necessarily distorts some pairwise relations, so the figures are not treated as primary evidence.

2.6.1 *t*-distributed Stochastic Neighbor Embedding

High-dimensional data cannot be directly visualized. In our study, each MNIST image corresponds to a 1024-dimensional vector, well beyond the three dimensions available for visual inspection. Dimensionality reduction to two or three dimensions is therefore necessary for any visual analysis of the data structure.

The t-distributed Stochastic Neighbor Embedding(t-SNE) [17] allows us to visualize the structure of high-dimensional manifold by matching probability distributions over point pairs between the high-dimensional and low-dimensional spaces. In high-dimensional space, a Gaussian kernel of bandwidth σ_i centered at point x_i defines an asymmetric conditional similarity

$$p_{j|i} = \frac{\exp\big(-\|x_i - x_j\|^2/2\sigma_i^2\big)}{\sum_{k\neq i}\exp\big(-\|x_i - x_k\|^2/2\sigma_i^2\big)}, \tag{2.50}$$

with σ_i chosen so that the distribution $P_i = \{p_{j|i}\}_{j\neq i}$ has a prescribed *perplexity*, perplexity broadly determines how many neighbor points the algorithm takes into account. It is typically set between 5 and 50. The higher the *perplexity*, the more of the global structure is taken into consideration. And $\mathrm{Perp}(P_i) = 2^{H(P_i)}$, where $H(\cdot)$ denotes Shannon entropy. Symmetrizing $p_{j|i}$ and $p_{i|j}$ gives the joint high-dimensional similarity $p_{ij} = \frac{p_{i|j}+p_{j|i}}{2N}$, where N is the number of data points.

In the low-dimensional space, the corresponding coordinates y_i induce a heavy-tailed joint similarity based on Student's t distribution with one degree of freedom

$$q_{ij} = \frac{\big(1 + \|y_i - y_j\|^2\big)^{-1}}{\sum_{k\neq l}\big(1 + \|y_k - y_l\|^2\big)^{-1}}. \tag{2.51}$$

The embedding $\{y_i\}$ is obtained by minimizing the Kullback-Leibler divergence $\mathrm{KL}(P||Q) = \sum_{i\neq j} p_{ij}\log(p_{ij}/q_{ij})$ via gradient descent. The heavy tail of the low-dimensional kernel alleviates the crowding problem that arises when high-dimensional points are compressed into a low-dimensional space.

However, we can notice that the high-dimensional similarities depend entirely on the ED, a choice that becomes a limitation because the points are sparse in 1024 dimensions, motivating the modification described in 2.6.2.

2.6.2 Shape-Aware SNE

As we mentioned in the previous section, the high-dimensional similarities in t-SNE depend entirely on the ED. But in this study, the forward process intermediate state x_t is dominated by isotropic Gaussian noise as t increases, the ED fails to provide an accurate representation of the visualized manifold following dimensionality reduction with t-SNE. Hence we need to substitute the ED with BHD.

Shape-Aware SNE (SASNE) [16, 19] retains the objective function and optimization process of t-SNE, and simply replace ED in the high-dimensional similarity condition with d_{NB}

$$p_{j|i} = \frac{\exp(-d_{NB}(i,j)^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{NB}(i,k)^2/2\sigma_i^2)} \tag{2.52}$$

Note that the k-NN graph itself is constructed using local ED, a regime in which the Euclidean distance is generally regarded as reliable for manifold-structured data. The global pairwise distances, however, are derived from the graph Laplacian spectrum rather than from Euclidean measurements. As a consequence, the high-dimensional similarities driving SASNE reflect the intrinsic geometry of the representation manifold, and SASNE embeddings produced at different diffusion timesteps can be compared qualitatively by class coloring and overall cluster arrangement; each embedding is optimized independently, so individual coordinates are not directly comparable across timesteps.

Two implementation conventions for SASNE follow Wängberg et al. [19] in the present work. The perplexity is set to $0.9n$, a deterministic data-scale rule that removes the need for per-checkpoint hand-tuning. A perplexity of this magnitude is meaningful only because the BHD remains accurate at global scales: standard t-SNE is restricted to small perplexities (typically ~ 30) since its Euclidean similarities are reliable only locally, whereas replacing the Euclidean distance with the BHD lets SASNE incorporate the full n-point similarity structure into each conditional distribution, which is what preserves the global and hierarchical organisation that small-perplexity t-SNE typically destroys [19]. The low-dimensional coordinates are initialized from the first three columns of the BHD embedding X introduced in 2.4.1, which gives a reproducible starting configuration that is already consistent with the high-dimensional geometry used by the objective, since ten classes are difficult to separate without overlap in a 2-dimensional embedding.

2.7 Pipeline Overview

The preceding sections introduced, in sequence, the generative model, the k-Nearest Neighbor graph construction, the BHD and the associated effective-dimensionality summary, the Silhouette-based cluster validation, and the embedding-based visualization method. We now describe how these components combine into a single analysis pipeline [16], summarized in Figure 2.6.

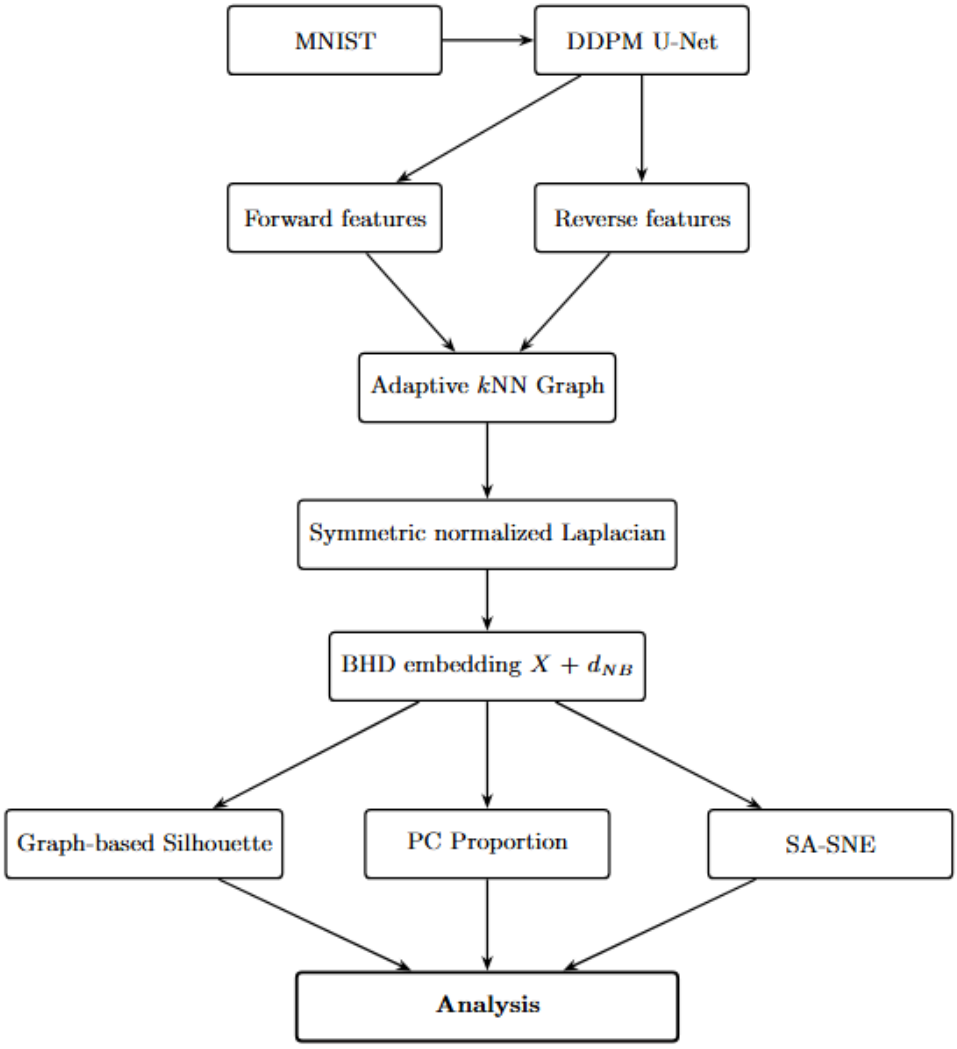


Figure 2.6: Pipeline overview. The DDPM U-Net trained on MNIST produces forward and reverse feature trajectories. At each diffusion checkpoint the features are passed to the adaptive kNN graph construction, from which the symmetric normalized Laplacian and the BHD embedding are derived. The three downstream analyses—Graph-based Silhouette, PC Proportion, and SASNE—all operate on the same BHD representation.

The pipeline runs in six stages:

1. **Model training.** A U-Net DDPM is trained on zero-padded MNIST images (Section 2.2).

2. **Feature extraction.** Forward features are obtained from the closed-form $q(x_t \mid x_0)$; reverse features are obtained by rolling out the learned denoising chain from $\mathcal{N}(0, I)$ (Section 3.1.3 and Section 3.1.4).

3. **Adaptive kNN graph.** For each checkpoint, an adaptive kNN graph is built on the $n = 5000$ feature vectors, with intra-class connectivity enforced after the global construction (Section 2.3.1).

4. **Spectral quantities.** The symmetric normalized Laplacian L_{sym} and the BHD d_{NB} are computed from the graph; these form the common input to the three analyses below (Section 2.4.1).

5. **Class-level analyses.** Graph-based Silhouette Analysis (Section 2.5.2) and the PC Proportion (Section 2.4.2) summarie, respectively, the label-aware and label-blind geometric organization of the classes at each checkpoint.

6. **Visualization.** The SASNE embedding (Section 2.6.2) provides a 3-dimensional spatial representation of the BHD geometry for qualitative inspection.

Three features of this design are worth stating. The kNN graph encodes local manifold structure, indicating which samples are mutually close under the locally reliable Euclidean distance. The spectral quantities built on this graph—L_{sym} and the BHD derived from its eigen structure—aggregate the local adjacency into a global distance measure that is comparable across diffusion timesteps. The three downstream analyses use the same BHD representation, so differences between their trajectories reflect differences in what they measure (label alignment vs. geometric dispersion vs. spatial layout) rather than differences in the underlying distance.

Chapter 3

Results and Analysis

3.1 Experimental Setup

3.1.1 Class-Balanced Subset Selection

The forward and reverse processes use different sampling strategies, reflecting the asymmetry between the two trajectories: forward samples are drawn with known labels, whereas reverse samples start from Gaussian noise and acquire their labels only at the end of denoising.

For the forward process we draw a class-balanced subset of $n = 5000$ samples from the MNIST training set of Section 2.2.4, with exactly 500 samples per class. The random seed is fixed so that the same 5000 sample indices are used at every forward checkpoint, which ensures that the observed temporal evolution is not confounded with sampling variability.

For the reverse process we draw $n = 5000$ independent initial states at $t = T = 1000$, each distributed as $x_T \sim \mathcal{N}(0, I)$. All initial samples are Gaussian noise, so class labels cannot be assigned at $t = T$. Instead, each generated trajectory is classified once at its near-clean state x_1 (for which $\bar{\alpha}_1 \approx 0.9999$ and the image is visually indistinguishable from x_0) using a pretrained CNN classifier trained on MNIST; the assigned label is then fixed for that trajectory and applied identically at every checkpoint. The sample size $n = 5000$ is chosen to match the forward setting, enabling direct per-checkpoint comparison between the two trajectories.

The choice of $n = 5000$ is further bounded by the cost of the Graph-based computations: the BHD requires an eigen decomposition of the $n \times n$ symmetric normalized Laplacian with cubic complexity $O(n^3)$, which remains tractable at this sample size.

3.1.2 Checkpoint Timestep Selection

We evaluate 15 checkpoints over the diffusion trajectory: $\{0, 50, 100, \ldots, 500, 550, 700, 900, 999\}$. Within $t \in [0, 550]$ the spacing is 50 steps, providing dense coverage of the class-dominated and transient regimes. Beyond $t = 550$ the spacing widens to 150, 200, and 99 steps respectively, giving lighter coverage of the noise-dominated plateau where $S(t)$ is approximately constant. The same checkpoint set is used for the forward and reverse trajectories, allowing direct per-step comparison.

The spacing is non-uniform by design. For $t \leq 550$ the checkpoints are spaced every 50 steps, while for $t > 550$ the spacing widens to 150, 200, and 99 between the last four checkpoints. The motivation is physical: as t increases, the marginal distribution $q(x_t)$ approaches the isotropic Gaussian prior $\mathcal{N}(0, I)$, after which the representation geometry saturates and changes only slowly. Dense sampling is therefore allocated to the region $t \leq 550$ where the manifold is still evolving, and sparse sampling suffices in $t > 550$ where only the approach to the isotropic limit

remains to be verified.

Under this design, the speciation time $t_S \approx 543$ predicted by Biroli et al. [1] happens to fall in the densely sampled region; the comparison between our empirical regime boundary and t_S reported in Section 4.2 is therefore resolved to within one 50-step checkpoint interval. The checkpoint design itself, however, is based only on the qualitative observation that the manifold saturates to isotropic Gaussian for large t and does not presuppose the location of any theoretical transition.

3.1.3 Forward Process Feature Extraction

The closed-form marginal of the forward process (Eq. (2.17), Section 2.2.1) allows us to compute features at any checkpoint $t \in \mathcal{T}$ in a single step from the clean image x_0, without iterating through intermediate Markov states. For each sample $x_0^{(i)}$ from the class-balanced subset of Section 3.1.1 and each target timestep t, the feature is obtained as

$$x_t^{(i)} = \sqrt{\bar{\alpha}_t}\, x_0^{(i)} + \sqrt{1 - \bar{\alpha}_t}\, \epsilon^{(i)}, \qquad \epsilon^{(i)} \sim \mathcal{N}(0, \mathrm{I}), \tag{3.1}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$ determined by the noise schedule of Section 2.2.4.

A main design choice concerns the noise term $\epsilon^{(i)}$. In the standard DDPM training objective, an *independent* $\epsilon_t^{(i)} \sim \mathcal{N}(0, \mathrm{I})$ is drawn for each (i, t) pair, so that the noise-prediction network $\epsilon_\theta(x_t, t)$ is trained against a joint distribution of (x_t, t) in which the noise realizations are independent across t. For our geometric analysis, we instead draw a *single* $\epsilon^{(i)} \sim \mathcal{N}(0, \mathrm{I})$ per sample i and reuse it across all 15 checkpoints. Each individual $x_t^{(i)}$ is still a valid sample from the marginal $q(x_t \mid x_0^{(i)})$ in Eq. (2.17); only the *joint* distribution of the checkpoint sequence $\{x_t^{(i)}\}_{t \in \mathcal{T}}$ deviates from the independent-noise assumption underlying training.

We adopt this construction for the purpose of this analysis: with independent noise drawn at each t, the Graph-based geometric quantities introduced in Sections 2.4.1–2.6.2 (BHD, Silhouette, PCP, SASNE) at adjacent checkpoints would carry an additional sampling fluctuation unrelated to the noise schedule, and the smooth dependence of $S(t)$ and related quantities on $\bar{\alpha}_t$ that we report in Section 3.3 would be obscured by checkpoint-to-checkpoint Monte-Carlo noise. Fixing $\epsilon^{(i)}$ across t removes this nuisance variance and isolates the dependence on the noise level. We acknowledge that this departs from the training-time noise model and discuss the implications, together with a re-analysis under independent noise, in Section 4.3.

Each resulting feature $x_t^{(i)} \in \mathbb{R}^{1\times32\times32}$ is flattened to a 1024-dimensional vector before being passed to the graph-construction stage. The forward-process feature extraction involves no learnable parameters, and the k-nearest-neighbor graph is built directly on the flattened features.

3.1.4 Reverse Process Feature Extraction

Unlike the forward process, the reverse process does not admit a closed-form shortcut: computing $x_t^{(i)}$ requires iteratively applying the learned denoising step $p_\theta(x_{t-1} \mid x_t)$ from $x_T^{(i)} \sim \mathcal{N}(0, \mathrm{I})$ down to the desired checkpoint. We generate $n = 5000$ reverse trajectories, drawn independently of the class-balanced subset used for the forward analysis (Section 3.1.1). For each trajectory we draw a fresh $x_T^{(i)} \sim \mathcal{N}(0, \mathrm{I})$ and run the reverse process for the full $T = 1000$ steps; at every checkpoint $t \in \mathcal{T}$ we record the state *before* applying the next denoising update, so that $x_t^{(i)}$ is the genuine feature at noise level $\bar{\alpha}_t$ rather than its one-step-denoised counterpart. Each resulting feature is flattened to a 1024-dimensional vector and passed to the graph-construction stage, identically to the forward case.

The partitioning described below is an implementation choice driven by GPU memory limits, not a property of the diffusion model itself: on hardware with enough memory to hold all $n = 5000$ trajectories simultaneously, the same features would be obtained from a single-batch run. In our setup, generating the full set of reverse trajectories must share the GPU with the downstream graph-spectral analysis (Section 2.3), in which the 5000×5000 Biharmonic-distance matrix and its float-precision eigen decomposition each occupy several gigabytes of VRAM. To leave sufficient headroom for these subsequent stages, we run the reverse process in 10 successive sub-batches of 500 trajectories each. Within each sub-batch we draw 500 fresh endpoints $x_T^{(i)} \sim \mathcal{N}(0, \mathrm{I})$, roll them out independently for the full T steps, and concatenate the recorded features after all sub-batches finish. This procedure produces the same joint distribution of features as a hypothetical single 5000-trajectory roll-out, for two complementary reasons. First, drawing 500 endpoints from $\mathcal{N}(0, \mathrm{I})$ in each of 10 successive calls is i.i.d.-equivalent to a single 5000-endpoint draw. Second, the U-Net is built with Group Normalization (Section 2.1.3) rather than Batch Normalization, so the denoising output for any sample depends only on its own features and on the shared parameters θ, with no dependence on the other samples in the same sub-batch. Combining these two properties, each recorded $x_t^{(i)}$ depends only on its own endpoint $x_T^{(i)}$ and on θ, regardless of how the 5000 trajectories are split across sub-batches.

3.2 Model Validation

This section reports two standard diagnostics for the trained U-Net—training-loss convergence (Section 3.2.1) and quality of generated samples (Section 3.2.2)—to confirm that the model meets the established benchmarks for unconditional DDPMs on MNIST before we examine its representation geometry in Sections 3.3–3.4. These checks are not part of the thesis's main contribution.

3.2.1 Training Convergence

We trained the U-Net DDPM for 50 epochs on a 54000-image training split of the MNIST training set, with the remaining 6000 images held out as a validation set for loss monitoring. Figure 3.1 shows the training and validation noise-prediction loss. The two curves decrease smoothly and remain close to each other throughout training, stabilizing near 1.4×10^{-2} by epoch 30, indicating that the model is neither underfitting nor overfitting.

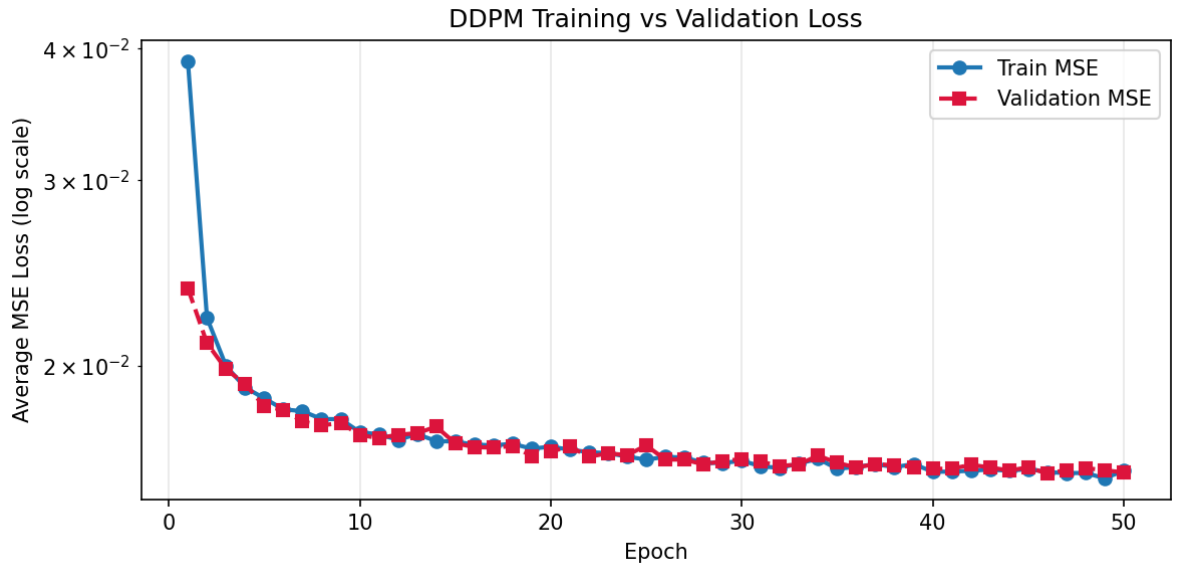


Figure 3.1: DDPM training and validation loss across 50 epochs. The horizontal axis is the training epoch; the vertical axis is the per-batch noise-prediction mean-squared error in log scale. The blue solid curve marks training loss, the red dashed curve marks held-out validation loss on the 6000-image validation split. Both curves decrease smoothly and stay close to each other throughout training, indicating that the network is neither under- nor over-fitting.

3.2.2 Generation Quality

To verify that the trained DDPM generates samples of quality consistent with the established benchmarks for unconditional MNIST DDPMs, we evaluate the Fréchet Inception Distance (FID) of [5] between 5000 samples drawn from our trained model and the MNIST training set. Throughout this section, "unconditional" means that samples are drawn from $p_\theta(x_0)$ without conditioning on a target digit class: each sample is obtained by running the trained reverse process from an independent endpoint $x_T \sim \mathcal{N}(0, \mathrm{I})$ down to $t = 0$, with no class label provided to the network at any step.

The FID of [5] is the squared Wasserstein-2 distance between two Gaussians fitted to the Inception-v3 feature embeddings of, respectively, the generated and the real samples; smaller values indicate that the generated samples match the real distribution more closely in both fidelity (image realism) and diversity (coverage of the modes). Following the FID computation protocol used by [6], we obtain

$$\mathrm{FID} = 2.66. \tag{3.2}$$

For direct comparison, the unconditional DDPM result reported on MNIST in [6] is $\mathrm{FID}_{\mathrm{Ho}} = 3.17$. Our model therefore matches and slightly improves on this benchmark, which we take as evidence that the trained network is suitable for the graph-based geometric analysis of the rest of Chapter 3. We hereafter refer to this trained DDPM—the U-Net ϵ_θ together with the reverse-process sampler that produces clean samples from Gaussian noise—as the *generator*; it is the source of the forward and reverse intermediate states $\{x_t\}$ analyzed below.

3.3 Class Structure Dynamics in the Forward Process

3.3.1 Global Silhouette (Forward)

Figure 3.2 reports the global Graph-based Silhouette $S(t)$ along the forward diffusion trajectory for $t \in \{0, 50, \ldots, 550, 700, 900, 999\}$. The clean-image value $S(0) = +0.089$ confirms that, in pixel space, the ten classes $C_0, \ldots, C_9$ already form geometrically separated clusters before any noise is added. As noise accumulates, $S(t)$ decreases monotonically, crosses zero between $t = 200$ and $t = 250$, and reaches its minimum $S(300) = -0.022$ at $t = 300$. Beyond the dip the curve rises slowly and asymptotes to a residual plateau of $S \approx -0.003$ for $t \geq 500$, remaining within ± 0.003 of this value out to $t = 999$.

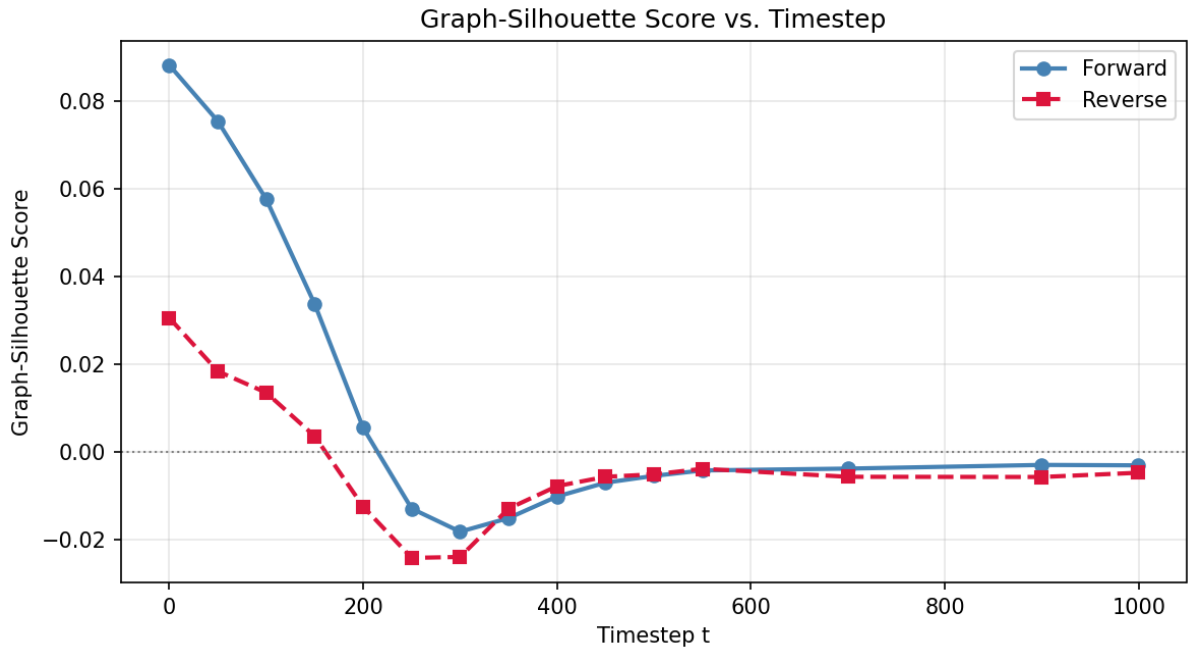


Figure 3.2: Global Graph-based Silhouette $S(t)$ along the diffusion trajectory. The horizontal axis is the diffusion timestep t on the 15 checkpoints listed in Section 3.1.2; the vertical axis is the class-averaged Silhouette $S(t) = K^{-1} \sum_c S_c(t)$. The blue solid curve with circular markers shows the forward process; the red dashed curve with square markers shows the reverse process; the horizontal dotted line marks $S = 0$. Vertical dotted lines at $t = 200$ and $t = 500$ separate the three regimes identified below: **class-dominated** ($t \leq 200$), **transient** ($200 < t < 500$), and **noise-dominated** ($t \geq 500$).

Reading the sign of $S(t)$. Before discussing the curve regime by regime, we recall how the sign of the Silhouette is to be read. By definition (Section 2.5.1), the per-sample Silhouette satisfies $s_i = 1 - a_i/b_i$ when $a_i \leq b_i$ and $s_i = b_i/a_i - 1$ otherwise, so that $s_i > 0$ iff the within-class distance a_i is smaller than the nearest-competing-class distance b_i, $s_i < 0$ iff the inequality reverses, and $s_i = 0$ when $a_i = b_i$. The global $S(t) = K^{-1} \sum_c S_c(t)$ is then a class-average of these per-sample comparisons. Three reference values orient the reader. (i) $S(t) \approx 1$ would mean that within-class distances are negligible compared to between-class distances—tightly resolved classes. (ii) $S(t) \approx 0$ is the value expected when points are distributed without class structure: for n points drawn i.i.d. from a single distribution and assigned arbitrary labels, $\mathbb{E}[a_i] = \mathbb{E}[b_i]$ in the large-n limit and the per-sample contributions cancel in expectation. (iii) $S(t) < 0$ is therefore stronger than "no class structure". The simplest geometric picture is a

two-class Venn-diagram-like overlap: take two clusters A and B whose ground-truth regions partially overlap in the BHD-induced metric, and consider a point p labeled A that sits inside the overlap region. The average distance from p to other A-labeled points (its cohesion a_p) is then larger than the average distance from p to B-labeled points (its separation b_p), because most of p's nearby neighbors under the kNN graph are B-labeled, even though p itself carries an A label. Averaging this configuration over all points and all class pairs gives $S(t) < 0$: the cluster-label assignment systematically disagrees with the geometric neighborhood structure encoded by the kNN graph. This is the situation observed in Regime II of Figure 3.2, and is directly visible in the SASNE embedding of Figure 3.6b, where points of the same MNIST class no longer occupy spatially coherent regions. The much smaller residual plateau at $S \approx -0.003$ in Regime III, by contrast, is statistically indistinguishable from the random baseline $S = 0$ and should be interpreted as the absence of any class signal in the high-noise limit, not as a stronger geometric statement.

Three regimes of class organization. The curve separates into three regimes describing distinct stages in how the classes $\{C_c\}$ are organized geometrically under diffusion. The regime boundaries are read directly from the empirical curve—the zero crossing of $S(t)$ near $t = 225$ and the onset of the residual plateau near $t = 500$—without reference to any theoretical prediction. The mechanism behind the sign of $S(t)$ in each regime can be understood from the class-wise cohesion and separation

$$\bar{a}_c(t) = \frac{1}{N_c} \sum_{i \in C_c} a_i, \qquad \bar{b}_c(t) = \frac{1}{N_c} \sum_{i \in C_c} b_i, \qquad k = 0, \ldots, 9, \tag{3.3}$$

reported per-class in Section 3.3.3. To summarize the trend by a single curve we use the class-averaged values $\bar{a}(t) = K^{-1} \sum_c \bar{a}_c(t)$ and $\bar{b}(t) = K^{-1} \sum_c \bar{b}_c(t)$ with $K = 10$.

Regime I: class-dominated ($t \leq 200$)**.** The positive Silhouette in this regime indicates that the average within-class distance $\bar{a}(t)$ is smaller than the average nearest-competing-class distance $\bar{b}(t)$. We measure $\bar{a}(0) = 0.77$ and $\bar{b}(0) = 0.87$; both quantities change slowly across the regime ($\bar{a}(200) = 0.94$, $\bar{b}(200) = 0.94$). The additive Gaussian noise $\sqrt{1 - \bar{\alpha}_t}\, \epsilon$ at these timesteps is small enough that the induced inflation of within-class distances does not exceed the nearest-competing-class distances, and the kNN graph therefore connects each vertex predominantly to same-class neighbors.

Regime II: transient ($200 < t < 500$)**.** The negative sign of $S(t)$ in Regime II is the case (iii) discussed at the beginning of this section: for the average sample, the within-class distance has come to exceed the nearest-competing-class distance ($\bar{a}(t) > \bar{b}(t)$). The mechanism is asymmetric inflation under additive Gaussian noise. The closed-form forward marginal $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$ injects an isotropic perturbation of variance $(1 - \bar{\alpha}_t)$ that is independent of the class structure of x_0; in pixel-space Euclidean distance the resulting per-pair inflation is approximately the same constant for every pair of points. The two cohesion–separation quantities $\bar{a}(t)$ and $\bar{b}(t)$, however, start from different baselines at $t = 0$: $\bar{a}(0) = 0.77$ is small because same-class points lie close together in the clean image manifold, whereas $\bar{b}(0) = 0.87$ is larger because clusters of different digits are already separated. The same absolute noise inflation therefore increases $\bar{a}(t)$ in larger *relative* terms than $\bar{b}(t)$, and at some intermediate t within this regime the curves cross: $\bar{a}(t)$ overtakes $\bar{b}(t)$. The BHD transports this Euclidean phenomenon onto the kNN graph, where the additional effect is topological—once the noisy ED

ordering reshuffles each vertex's k closest neighbors, an increasing fraction of those neighbors belong to other classes, which further inflates the in-graph cohesion. The Regime II window $200 < t < 500$ is therefore the transient phase in which within-class inflation has overtaken nearest-competing-class expansion but neither has yet saturated to the graph-wide unity that defines Regime III. The cohesion–separation decomposition in Figure 3.4 shows both curves directly: $\bar{a}(t)$ crosses $\bar{b}(t)$ within this window and the gap continues to widen until both saturate near 1 in Regime III.

Regime III: noise-dominated ($t \geq 500$). For $t \geq 500$ the marginal distribution $q(x_t)$ approaches the isotropic Gaussian limit $\mathcal{N}(0, \mathrm{I})$, and the classes are no longer statistically distinguishable in the kNN graph. Both $\bar{a}(t)$ and $\bar{b}(t)$ concentrate at the graph-wide mean of unity for every class, with inter-class spread below 0.01. The residual plateau $S \approx -0.003$ is consistent with the vanishing of class structure rather than with anti-alignment: at this order of magnitude the Silhouette is governed by finite-sample fluctuations in the adaptive kNN construction, of which the residual sign is not informative. The plateau's flatness ($|\Delta S| < 0.004$ across $t \in [500, 999]$) indicates that this fluctuation level is independent of t in the high-noise regime.

Comparison with the Biroli speciation prediction. A complementary perspective on the transition between Regime II and Regime III is provided by the *speciation* analysis of Biroli et al. [1]. Applying mean-field phase-transition theory to the score function of the diffusion forward process, the speciation time t_S is defined as the timestep at which class-conditional clusters in the data distribution become indistinguishable from a single Gaussian under the score function—equivalently, the latest time at which class identity could in principle be recovered from the score. Intuitively, t_S is the boundary beyond which class information is washed out at the population level under the score function; this is the population-level analogue of our Regime III, in which class information has been washed out at the sample-level kNN-graph quantities. Under the continuous-time variance-preserving formulation of Biroli et al. (their Eq. (2) and the discussion immediately below it), the speciation time is given in closed form by

$$t_S = \tfrac{1}{2} \ln \Lambda, \tag{3.4}$$

where Λ is the largest eigenvalue of the data covariance matrix. The detailed evaluation of Eq. (3.4) for the MNIST setup of Section 3.1.1, including the discrete-schedule mapping, is given in Appendix A; the resulting value is

$$t_S \approx 543. \tag{3.5}$$

This timestep falls within the empirical window $t \in [400, 550]$ over which our Silhouette curve transitions into its Regime III plateau, and the agreement is therefore resolved to within the 50-step checkpoint spacing used in this region. This value is consistent with the speciation prediction of Biroli et al.

3.3.2 Class-wise Silhouette Analysis (Forward)

The global $S(t)$ averages over all ten classes and can mask class-specific behavior. Figure 3.3 shows the class-wise Silhouette $S_c(t)$ for each class C_c, $c = 0, \ldots, 9$.

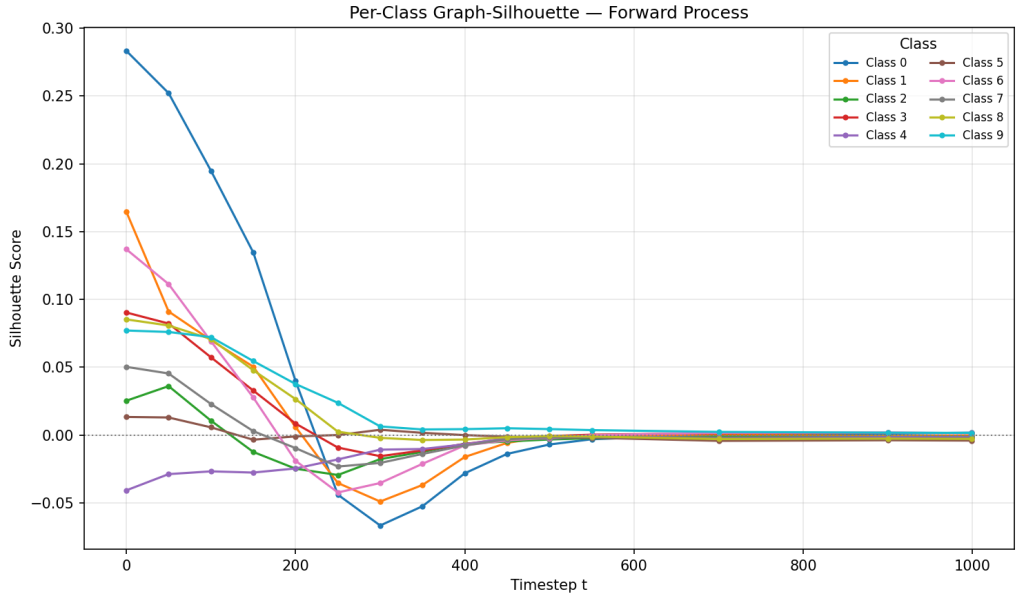


Figure 3.3: Class-wise Silhouette $S_c(t)$ for the forward diffusion process. The horizontal axis is the diffusion timestep t; the vertical axis is the per-class Silhouette $S_c(t) = N_c^{-1} \sum_{i \in C_c} s_i$. Each color traces one MNIST class as labeled in the legend; the horizontal dotted line marks $s_c = 0$. All classes except class 4 start positive at $t = 0$ and cross zero in the interval $t \in [150, 300]$; class 4 (purple) has $s_4(0) < 0$ on clean data and remains negative throughout the trajectory.

Initial values at $t = 0$. The class-wise Silhouettes on clean MNIST data span a wide range, from $s_0(0) = +0.283$ to $s_4(0) = -0.041$. Class 4 is the only class with a *negative* Silhouette already at $t = 0$: the within-class BHD $\bar{a}_4(0)$ already exceeds the nearest-competing-class BHD $\bar{b}_4(0)$ on the unperturbed dataset. We note this as an empirical observation; the underlying cause is not investigated in the present work, although the geometric similarity between handwritten "4" and "9" (the closest non-self class for many class-4 samples in the kNN graph) is a plausible cause.

Zero-crossing of $S_c(t)$. For each class other than 4 and 9, the Silhouette $S_c(t)$ is positive at $t = 0$ and crosses zero at some timestep $t_c^\star$ during the forward process. Table 3.1 groups the ten classes by the interval in which this zero crossing occurs; classes 4 and 9 are listed separately for the reasons given in the table caption.

Group	Classes
Negative throughout the trajectory	4
Positive throughout, no meaningful zero crossing	9
Zero crossing in $t \in [150, 200]$	2, 5, 6, 7
Zero crossing in $t \in [200, 250]$	0, 1, 3
Zero crossing in $t \in [250, 300]$	8

Table 3.1: Forward-process zero crossing of the class-wise Silhouette $S_c(t)$. Two classes are exceptional: class 4 has $s_4(0) < 0$ already on clean data and remains negative at every checkpoint, and class 9 stays positive throughout the resolved Regime I–II window—it decays from $s_9(0) = +0.079$ to $s_9(450) = +0.002$ without crossing zero, and the small negative values at $t \geq 700$ are within the Regime III noise floor ($|S| < 0.005$).

The ordering of $t_c^\star$ is not the inverse of the ordering of $S_c(0)$. For example, class 6, whose $s_6(0) = +0.138$ is among the three largest initial values, crosses zero in $[150, 200]$, whereas class 8, whose $s_8(0) = +0.083$ is considerably smaller, crosses zero later in $[250, 300]$; and class 5, with the smallest positive initial value $s_5(0) = +0.012$, is one of the first to cross zero, in $[150, 200]$. The mechanism behind this ordering—which combines the initial within-class cohesion and the rate at which Gaussian noise inflates it—is examined directly in Section 3.3.3, where we report the class-wise components $\bar{a}_c(t)$ and $\bar{b}_c(t)$.

3.3.3 Class-wise Cohesion-Separation (Forward)

The class-wise Silhouette $S_c(t)$ summarized in Figure 3.3 is computed from two class-wise averages of pairwise BHDs: the within-class cohesion $\bar{a}_c(t) = N_c^{-1} \sum_{i \in C_c} a_i$ and the nearest-competing-class distance $\bar{b}_c(t) = N_c^{-1} \sum_{i \in C_c} b_i$ (Section 2.5.1). The sign of $S_c(t)$ is determined sample-by-sample by the comparison of a_i and b_i, but the magnitudes and temporal trajectories of $\bar{a}_c$ and $\bar{b}_c$ themselves carry information not visible in $S_c(t)$ alone, and we report them jointly in Figure 3.4.

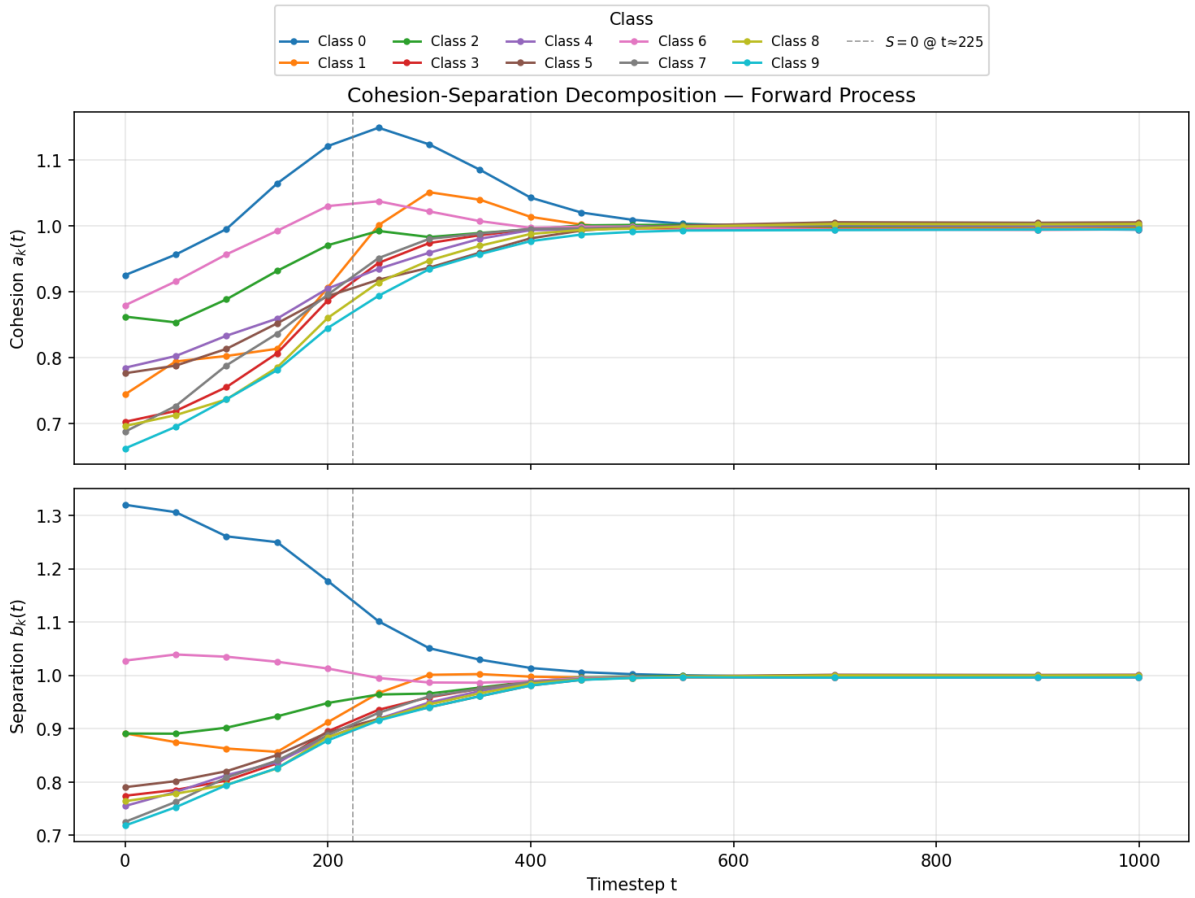


Figure 3.4: Class-wise cohesion $\bar{a}_c(t)$ (top panel) and nearest-competing-class distance $\bar{b}_c(t)$ (bottom panel) for the forward diffusion process. The horizontal axis is the diffusion timestep t; the vertical axis is the class-averaged BHD, normalized so that the graph-wide mean of pairwise BHD equals one. Each color traces one MNIST class (legend shared with Figure 3.3); the vertical dashed line marks the zero-crossing timestep $t \approx 225$ of the global Silhouette $S(t)$.

Cohesion $\bar{a}_c(t)$. All ten cohesion curves are monotonically increasing in t. At $t = 0$, $\bar{a}_c$ ranges from 0.66 (class 9) to 0.93 (class 0), with class-averaged value $\bar{a}(0) = K^{-1} \sum_c \bar{a}_c(0) = 0.77$.

By $t = 300$ all ten classes lie within $\bar{a}_c \in [0.93, 1.12]$ with class-averaged value $\bar{a}(300) = 1.00$, and by $t = 500$ all ten classes lie within $[0.99, 1.01]$.

Nearest-competing-class distance $\bar{b}_c(t)$. The $\bar{b}_c(t)$ curves are also monotonically increasing in t, but start higher and from a wider range: at $t = 0$, $\bar{b}_c$ ranges from 0.72 to 1.32 with class-averaged value $\bar{b}(0) = 0.87$. Class 0 retains the largest value of $\bar{b}_c$ until $t \approx 350$, beyond which all ten curves converge near unity.

Origin of the negative sign of $S_c(t)$ in the transient regime. Comparing the two panels, $\bar{a}$ and $\bar{b}$ both increase with t, but $\bar{a}$ increases more rapidly: between $t = 0$ and $t = 300$, $\bar{a}$ grows by 30% ($0.77 \to 0.99$), whereas $\bar{b}$ grows by 14% ($0.87 \to 0.97$). Per-sample, $s_i = (b_i - a_i)/\max(a_i, b_i)$ becomes negative once the ordering $a_i < b_i$ holding for most $i \in C_c$ in Regime I reverses to $a_i > b_i$ for most $i \in C_c$. At the class-averaged level, this reversal is reflected in the crossing of the two curves $\bar{a}_c(t)$ and $\bar{b}_c(t)$ near the transient minimum, although the class-averaged Silhouette $\bar{s}_c(t)$ is not in general identical to $(\bar{b}_c - \bar{a}_c)/\max(\bar{a}_c, \bar{b}_c)$ because the average of a non-linear function differs from the function of the average. The empirical observation is that the negative sign of s_c in the transient regime is associated with within-class distance inflation outpacing nearest-competing-class distance inflation, not with the latter shrinking; the BHD normalization to graph-wide mean unity ensures that no absolute distance is decreasing.

3.3.4 Effective Dimensionality (Forward)

To complement the label-aware Silhouette Analysis with a label-blind summary, we use the Graph-PCA quantity $\mathrm{PCP}(\alpha, t)$ of Section 2.4.2 as a proxy for the effective dimensionality of the BHD-embedded feature distribution at each diffusion checkpoint. Three variance thresholds $\alpha \in \{0.7, 0.8, 0.9\}$ are reported jointly to verify that the qualitative features of the curve are not sensitive to a single choice of α.

Figure 3.5 shows $\mathrm{PCP}(\alpha, t)$ for the forward process. All three curves are monotonically increasing in t. For the central threshold $\alpha = 0.8$, $\mathrm{PCP}_{0.8}(0) = 0.030$ at clean data and $\mathrm{PCP}_{0.8}(999) = 0.656$ at the noise-dominated endpoint, an increase by more than a factor of 20. The curve rises throughout the trajectory but is steepest in the early-to-transient window: $\mathrm{PCP}_{0.8}(100) = 0.135$, $\mathrm{PCP}_{0.8}(200) = 0.440$, $\mathrm{PCP}_{0.8}(300) = 0.609$; the single-largest 100-step increment occurs in $[100, 200]$ with $\Delta\mathrm{PCP}_{0.8} = +0.305$. Most of the cumulative rise—about two thirds—is therefore already completed before $t = 200$, where the Silhouette $S(t)$ has not yet reached its zero crossing (Section 3.3.1). For $t \geq 500$ the curve flattens, with $\mathrm{PCP}_{0.8}(500) = 0.654$ and $\mathrm{PCP}_{0.8}(999) = 0.656$, a plateau spanning less than 0.003 across 500 timesteps. The $\alpha = 0.7$ and $\alpha = 0.9$ curves trace the same monotonic-and-saturating shape with the expected vertical offset—$\mathrm{PCP}_{0.7}(\cdot)$ rises from 0.006 to 0.529 and $\mathrm{PCP}_{0.9}(\cdot)$ from 0.174 to 0.806—so the qualitative behavior of $\mathrm{PCP}(\cdot, t)$ does not depend on the specific threshold within this range.

Mechanism: noise gradually inflates the effective dimensionality of the data manifold. The monotonic rise of $\mathrm{PCP}(\alpha, t)$ is the graph-spectral signature of the manifold hypothesis under additive Gaussian noise. The clean MNIST images of Section 3.1.1 lie on a low-dimensional manifold embedded in the 1024-dimensional ambient space (padded 32×32), so at $t = 0$ the BHD-embedded variance concentrates in a very small number of principal axes and $\mathrm{PCP}_{0.8}(0) = 0.030$—only about 3% of the available non-trivial axes are needed to capture 80% of the BHD variance. Each forward step injects an isotropic Gaussian perturbation of

variance β_t into the full 1024-dimensional ambient space; a large fraction of this perturbation lies in directions *normal* to the data manifold, where the clean data carries essentially no variance. As the cumulative noise variance $1 - \bar{\alpha}_t$ grows from $\sim 10^{-4}$ at $t = 1$ to nearly 1 at $t = T$, the point cloud is progressively displaced off the data manifold and into these previously empty ambient directions, so the BHD-embedded variance spreads across a strictly increasing number of principal axes and $\mathrm{PCP}(\alpha, t)$ rises monotonically. The steepest part of the curve is at intermediate $t \in [100, 300]$, where the additive noise variance has grown to the scale of the data manifold's characteristic spread along its tangent directions but has not yet saturated all ambient directions; once $1 - \bar{\alpha}_t \to 1$ near $t = 500$, every ambient direction is already filled with noise variance and the curve saturates. The plateau value $\mathrm{PCP}_{0.8}(t) \to 0.656$ for $t \geq 500$ is the BHD-PCA signature of $q(x_t) \approx \mathcal{N}(0, I)$ in $\mathbb{R}^{1024}$; it is bounded away from 1 because PCP is defined on the BHD-embedded variance, which retains residual concentration in the low-frequency end of the graph Laplacian spectrum—an artifact of the kNN graph topology that survives even when the underlying point cloud is isotropic in $\mathbb{R}^{1024}$.

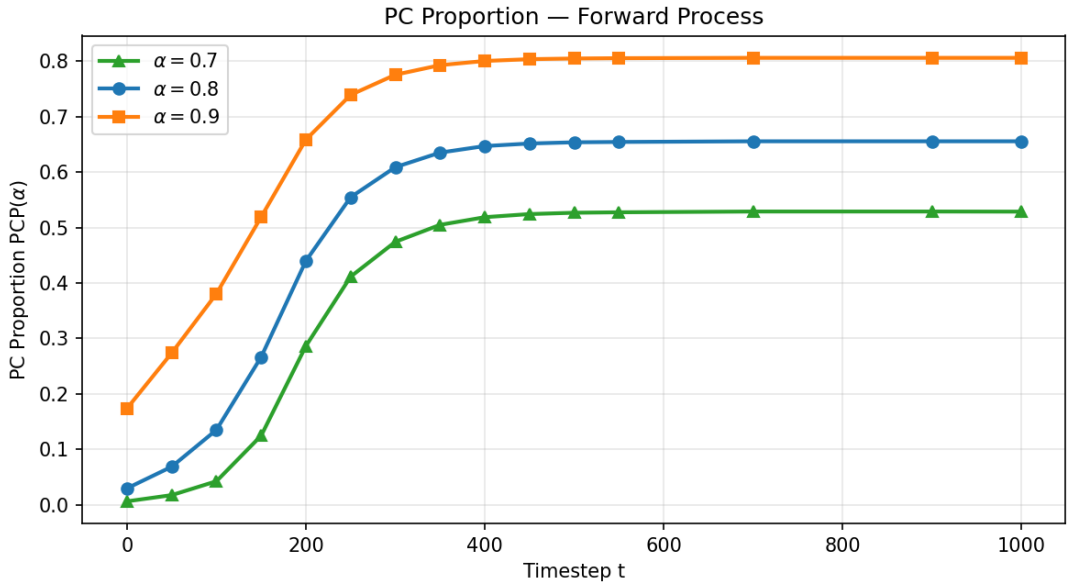


Figure 3.5: PC Proportion $\mathrm{PCP}(\alpha, t)$ for the forward diffusion process at three variance thresholds $\alpha \in \{0.7, 0.8, 0.9\}$. The horizontal axis is the diffusion timestep t; the vertical axis is the proportion of principal axes (relative to $n - 1$) needed to capture a fraction α of the BHD-embedding variance. Green triangles trace $\alpha = 0.7$, blue circles $\alpha = 0.8$, orange squares $\alpha = 0.9$. All three curves rise monotonically and saturate by $t \approx 500$.

What Figure 3.5 adds to the Silhouette picture. Reading Figures 3.2 and 3.5 together exposes two distinct time scales in the forward diffusion. The Silhouette resolves the relative ordering of $\bar{a}(t)$ and $\bar{b}(t)$: it changes sign at the timestep where cohesion overtakes separation, which is what makes it useful as a class-organization probe. The PC Proportion, by contrast, sums over all class pairs and reports only the redistribution of BHD-embedded variance across principal axes; it is therefore driven by the joint absolute growth of $\bar{a}(t)$ and $\bar{b}(t)$ rather than by their ratio.

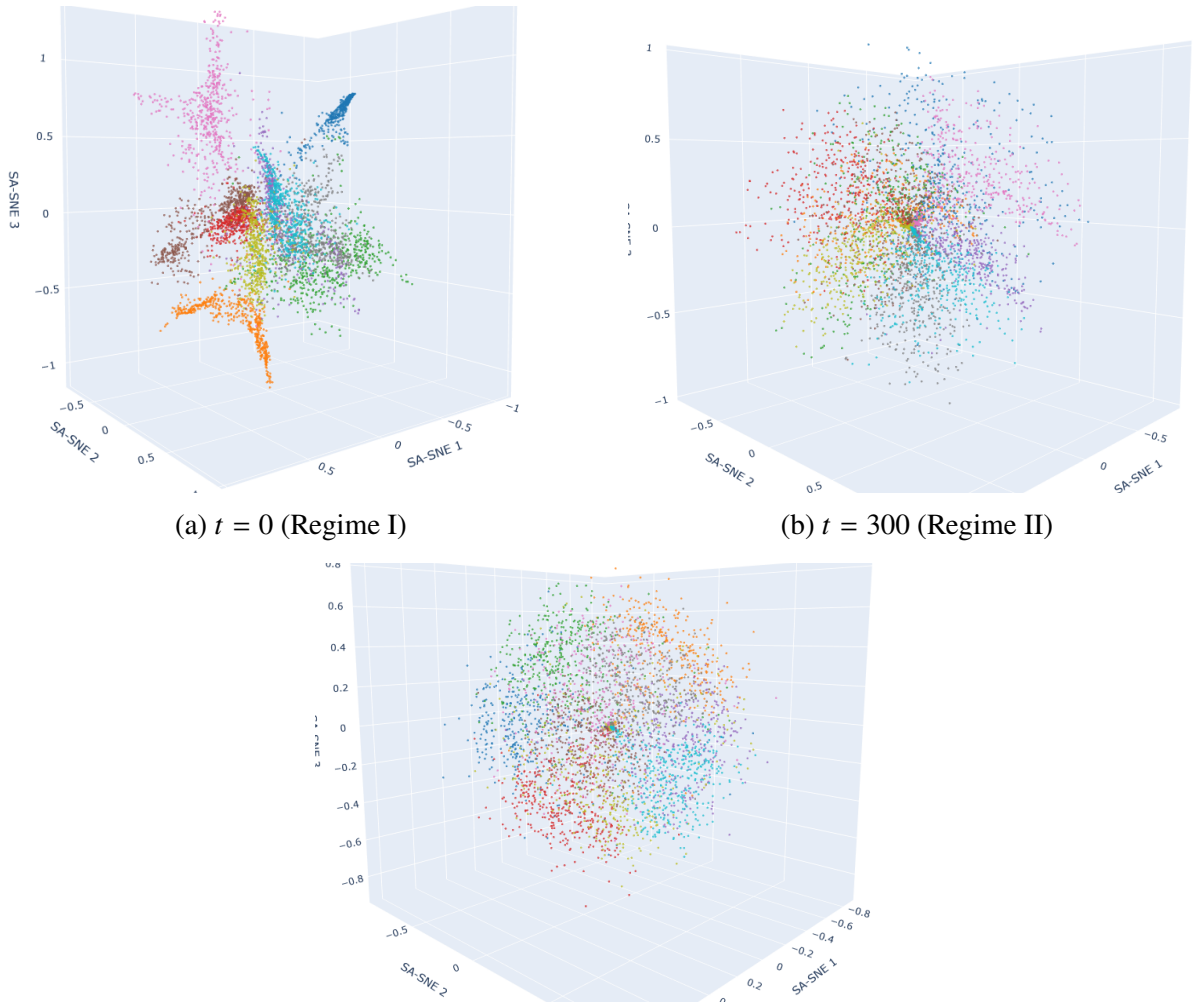
In cohesion–separation language (Section 3.3.3), this difference plays out in two stages. *First, dimensional inflation precedes class-label decoupling.* By $t = 200$ the BHD-embedded variance has already spread across roughly two thirds of its eventual plateau range—$\mathrm{PCP}_{0.8}(200) = 0.44$ out of a plateau $\mathrm{PCP}_{0.8}(t \geq 500) \approx 0.66$—yet the Silhouette only crosses zero near $t = 225$

and reaches its minimum at $t \approx 300$. Both $\bar{a}(t)$ and $\bar{b}(t)$ have already inflated substantially by $t = 200$ (Figure 3.4), but their relative ordering $\bar{a}(t) > \bar{b}(t)$ is not yet established. *Second, both curves enter their plateau at the same* $t \approx 500$. This common saturation point is the diffusion timestep at which $\bar{a}(t)$ and $\bar{b}(t)$ both converge to the graph-wide unity, $q(x_t) \approx \mathcal{N}(0, I)$, and neither metric can carry any further class-resolved information.

Figure 3.5 therefore reports *when the data manifold dissolves into the ambient* 1024-*dimensional space*, driven by the absolute growth of $\bar{a}$ and $\bar{b}$ in combination; Figure 3.2 reports *when the labeling stops tracking the manifold*, driven by the relative reversal of $\bar{a}$ and $\bar{b}$. The two transitions need not align a prior, and in this MNIST experiment they do not: dimensional inflation leads class-label decoupling by roughly 100–150 timesteps. The Silhouette and the PC Proportion are not redundant probes of the same regime structure—they pin down two genuinely distinct events on the diffusion timeline, both of which are visible in the cohesion–separation decomposition of Section 3.3.3.

A natural extension, not pursued in this thesis, is to evaluate the Graph-PCA spectrum separately within each class, yielding a class-wise PC Proportion $\mathrm{PCP}_c(\alpha, t)$ that would expose how each class's own intrinsic dimensionality evolves under diffusion; this would be a label-aware counterpart to the present aggregate $\mathrm{PCP}(\alpha, t)$.

3.3.5 SASNE Visualization (Forward)



(a) $t = 0$ (Regime I)

(b) $t = 300$ (Regime II)

(c) $t = 999$ (Regime III)

Figure 3.6: SASNE three-dimensional embeddings of the forward diffusion process at the three representative checkpoints. Each panel shows the 5000 feature vectors at that checkpoint, with each point colored by its ground-truth MNIST class. Axes are SASNE 1, 2, 3.

To complement the scalar summaries with a low-dimensional spatial representation of the data, we compute three-dimensional SASNE embeddings (Section 2.6.2) of the 5000 forward feature vectors at three representative checkpoints, one per regime: $t = 0$ (Regime I), $t = 300$ (Regime II minimum), and $t = 999$ (Regime III plateau).

The visual content of the three panels is consistent with the Silhouette values reported in Section 3.3.1. At $t = 0$ the ten classes occupy compact, mutually separated regions of the embedding space, consistent with $S(0) = +0.089$. At the transient-regime minimum $t = 300$, samples of different classes occupy partially overlapping regions in which class color remains discernible but no clean class-wise partition can be drawn—the visual counterpart of $S(300) = -0.022$. At $t = 999$ the points form a single approximately isotropic cloud in which no class-color pattern is discernible, consistent with the residual plateau $S(999) \approx -0.003$ and the saturation of $\mathrm{PCP}(\alpha, t)$ reported in Section 3.3.4.

Because both the Silhouette and the SASNE coordinates are computed from the same BHD matrix d_{NB} (Section 2.4.1), the SASNE embeddings reproduce, in low-dimensional visual form, the regime structure documented by $S(t)$ and $\mathrm{PCP}(\alpha, t)$, rather than providing an independent

validation of it. An independent validation would require a distance measure not derived from d_{NB}, which we have not pursued in this work.

3.4 Class Structure Dynamics in the Reverse Process

3.4.1 Global Silhouette (Reverse)

Figure 3.2 reports the reverse-process Silhouette $S^{\text{rev}}(t)$ (dashed red curve). The reverse trajectory exhibits the same three-regime structure as the forward curve but traversed in the opposite temporal direction: a noise-dominated plateau $S^{\text{rev}} \approx -0.005$ for $t \geq 500$, a transient regime with negative minimum $S^{\text{rev}}_{\text{min}} = -0.024$ attained at $t \in \{250, 300\}$, and a class-dominated regime with $S^{\text{rev}} > 0$ for $t \leq 150$, reaching $S^{\text{rev}}(0) = +0.031$ at the generated endpoint. The cohesion–separation mechanism of Section 3.3.1 carries over unchanged: the transient negative sign is the geometric event in which the class-averaged within-class distance $\bar{a}^{\text{rev}}(t)$ overtakes the nearest-competing-class distance $\bar{b}^{\text{rev}}(t)$ (Section 3.4.3).

Two quantitative differences from the forward curve are visible in the figure: (i) the transient minimum is deeper ($S^{\text{rev}}_{\text{min}} = -0.024$ versus $S^{\text{fwd}}_{\text{min}} = -0.022$) and is attained at two adjacent checkpoints rather than one; (ii) the class-dominated endpoint is substantially weaker, with $S^{\text{rev}}(0) = +0.031$ approximately one third of $S^{\text{fwd}}(0) = +0.089$. Both differences are traced in Section 3.4.3 to inflated within-class BHDs in the generated samples while nearest-competing-class BHDs are largely preserved. The Regime II–III transition again falls within $t \in [400, 550]$, the window containing the Biroli speciation time $t_S \approx 543$, and is consistent with the speciation prediction of Biroli et al. [1] (Appendix A).

3.4.2 Class-wise Silhouette Analysis (Reverse)

The class-wise Silhouettes $s^{\text{rev}}_c(t)$ (Figure 3.7) follow the same broad pattern as their forward counterparts, with two class-specific observations that are unique to the reverse direction. *Class 4* remains negative at every reverse checkpoint with endpoint value $s^{\text{rev}}_4(0) = -0.041$, exactly mirroring the forward observation $s^{\text{fwd}}_4(0) = -0.041$: the trained generator reproduces the geometric peculiarity of handwritten "4" (Section 3.3.2). *Class 9* is the only class that maintains $s^{\text{rev}}_9(t) > 0$ throughout the entire reverse transient regime, with $s^{\text{rev}}_9(500) = +0.002$ growing monotonically to $s^{\text{rev}}_9(0) = +0.079$; this is the reverse-direction analogue of the forward observation that class 9 alone fails to cross zero (Section 3.3.2). The remaining classes cross zero in $t \in [150, 350]$ with class-dependent timing whose mechanism is examined through cohesion and separation in Section 3.4.3.

The empirical class distribution of the 5000 reverse samples is non-uniform (Table 3.2), with class 4 over-represented at 818 and class 0 under-represented at 341. The global statistic $S^{\text{rev}}(t) = K^{-1} \sum_c s^{\text{rev}}_c(t)$ used throughout Chapter 3 weights every class equally and is therefore unaffected by N_c; the imbalance itself is acknowledged as a pipeline limitation in Section 4.3.

Table 3.2: Class distribution of the 5000 unconditional reverse samples used in Chapter 3. Each sample is classified at its near-clean state x_1 by the pretrained MNIST CNN classifier of Appendix E; the assigned label is then fixed for that sample's reverse trajectory.

Class	0	1	2	3	4	5	6	7	8	9
Count	341	494	503	461	**818**	435	450	521	504	473

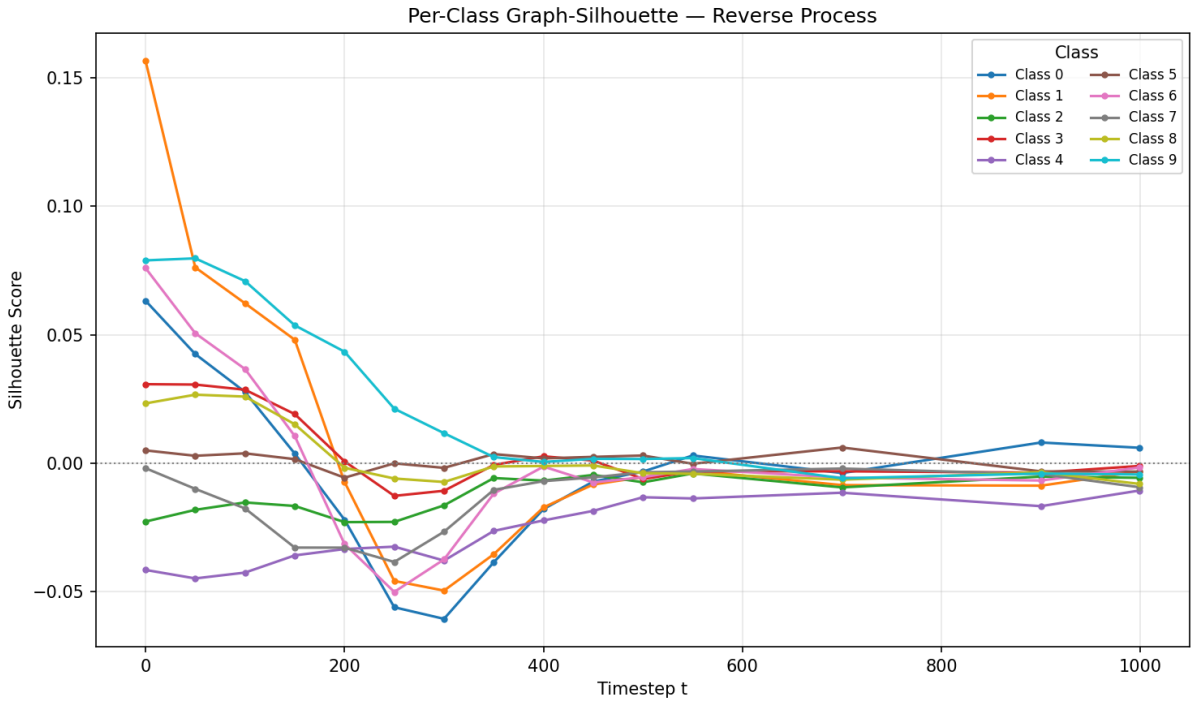


Figure 3.7: Class-wise Silhouette $s_c^{\text{rev}}(t)$ for the reverse denoising process. Axes and legend match Figure 3.3; the reverse trajectory is read in order of decreasing t (right to left). Class 9 (cyan) is the only class with $s_9^{\text{rev}}(t) > 0$ throughout the transient regime, reaching $s_9^{\text{rev}}(0) = +0.079$ at the generated endpoint; class 4 (purple) remains negative at every checkpoint, mirroring the forward observation that $s_4(0) < 0$ already on clean MNIST data.

3.4.3 Class-wise Cohesion-Separation (Reverse)

Figure 3.8 compares the reverse cohesion $\bar{a}_c^{\text{rev}}(t)$ and nearest-competing-class distance $\bar{b}_c^{\text{rev}}(t)$ with their forward counterparts. The qualitative shape is preserved: both quantities are monotonic in t and saturate to the graph-wide unity in Regime III, so the within-class–to–nearest-competing-class inversion that drives the reverse Silhouette dip has the same mechanism as in the forward process (Section 3.3.3). The single distinguishing finding is concentrated at the generated endpoint: $\bar{a}^{\text{rev}}(0) = 0.90$ exceeds $\bar{a}^{\text{fwd}}(0) = 0.77$ by 16.1% , while $\bar{b}^{\text{rev}}(0) = 0.93$ exceeds $\bar{b}^{\text{fwd}}(0) = 0.87$ by only 7.9%. The asymmetry is therefore concentrated in the within-class distances: the generator produces samples whose nearest-competing-class structure is largely preserved relative to the forward process at $t = 0$, but whose within-class geometry is inflated by roughly twice the relative amount. Both offsets vanish in Regime III ($|\bar{a}^{\text{rev}} - \bar{a}^{\text{fwd}}| < 0.005$ at $t \geq 500$), so the asymmetry is a property of the class-dominated regime alone.

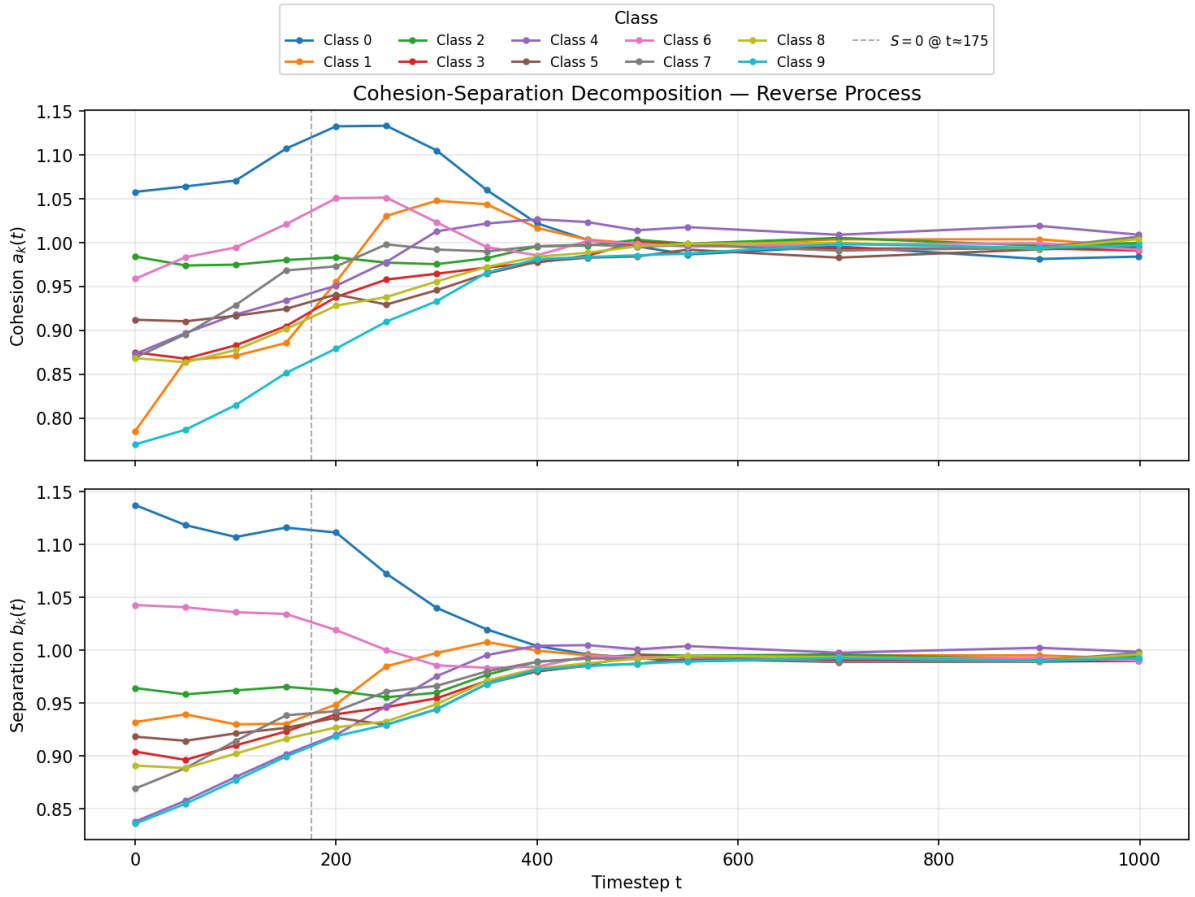


Figure 3.8: Class-wise cohesion $\bar{a}_c^{\text{rev}}(t)$ (top) and nearest-competing-class distance $\bar{b}_c^{\text{rev}}(t)$ (bottom) for the reverse denoising process. Axes and legend match Figure 3.4; the vertical dashed line marks the zero crossing $t \approx 175$ of $S^{\text{rev}}(t)$.

3.4.4 Effective Dimensionality (Reverse)

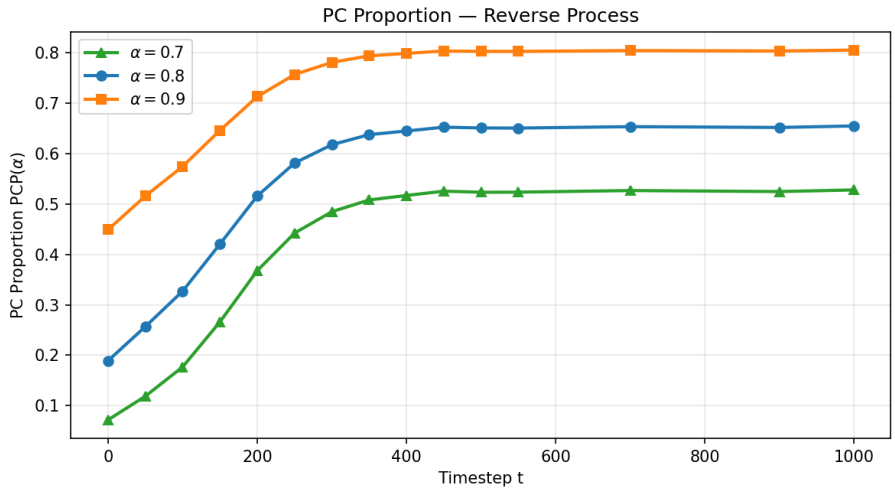
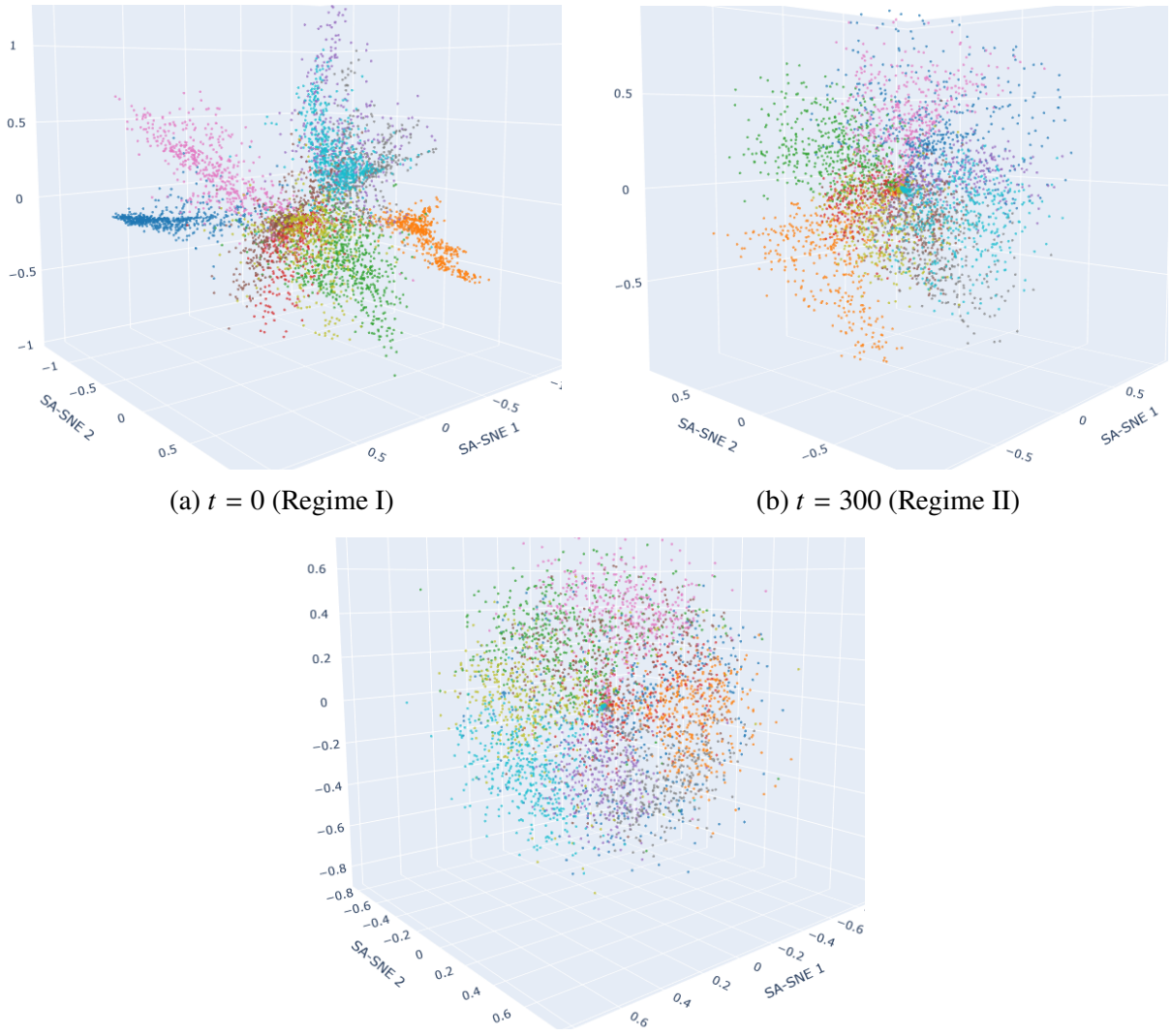


Figure 3.9: PC Proportion $\text{PCP}(\alpha, t)$ for the reverse denoising process at three variance thresholds $\alpha \in \{0.7, 0.8, 0.9\}$. Axes and color key match Figure 3.5. All three curves decrease monotonically as denoising proceeds from $t = 999$ to $t = 0$.

Figure 3.9 shows PCP(α, t) for the reverse process. As in the forward case (Section 3.3.4), all three threshold curves are monotonic in t and saturate by $t \approx 500$ to the same plateau value $\text{PCP}^{\text{rev}}(0.8, 999) = 0.655$. The two endpoints differ substantially: $\text{PCP}^{\text{rev}}(0.8, 0) = 0.189$ exceeds $\text{PCP}^{\text{fwd}}(0.8, 0) = 0.030$ by a factor of approximately 6.3, with the same direction at the other thresholds ($\alpha = 0.7$: ratio 11.2; $\alpha = 0.9$: ratio 2.6). The generated point cloud at $t = 0$ therefore retains a substantially larger fraction of the ambient 1024-dimensional variance than the clean MNIST data, indicating residual off-manifold spread. This is the class-aggregate counterpart of the within-class BHD inflation reported in Section 3.4.3: the same geometric event read through two different summaries of the BHD matrix.

3.4.5 SASNE Visualization (Reverse)



(a) $t = 0$ (Regime I)

(b) $t = 300$ (Regime II)

(c) $t = 999$ (Regime III)

Figure 3.10: SASNE three-dimensional embeddings of the reverse diffusion process at three representative checkpoints. Each panel shows the 5000 feature vectors at that checkpoint, with each point colored by the CNN-classifier label assigned at x_1 (Section 3.1.4). Axes are SASNE 1, 2, 3.

Figure 3.10 shows three-dimensional SASNE embeddings of the reverse process at the three representative checkpoints used for the forward analysis. The visual content is consistent with the

Silhouette and effective-dimensionality readings of Sections 3.4.1 and 3.4.4: an approximately isotropic cloud at $t = 999$ ($S^{\text{rev}}(999) \approx -0.005$), partially-formed overlapping clusters at the transient minimum $t = 300$ ($S^{\text{rev}}(300) = -0.024$), and ten class clusters at $t = 0$ that occupy compact regions but with noticeably more inter-class overlap than the forward $t = 0$ embedding—consistent with $S^{\text{rev}}(0) = +0.031$ versus $S^{\text{fwd}}(0) = +0.089$ and with $\text{PCP}^{\text{rev}}_{0.80}(0) = 0.189$ versus $\text{PCP}^{\text{fwd}}_{0.80}(0) = 0.030$.

3.5 Comparison between Forward and Reverse Processes

3.5.1 Shared Features and Asymmetries

The forward and reverse Silhouette curves of Figure 3.2 share three features. (i) Both reach a small-magnitude plateau for $t \geq 500$, with $S^{\text{fwd}} \approx -0.003$ and $S^{\text{rev}} \approx -0.005$; both values are consistent with the random-data baseline at $n = 5000$ (Section 3.3.1). (ii) Both display a transient-regime minimum near $t \approx 300$, within the 50-step checkpoint spacing. (iii) The zero crossings of $S(t)$ between the class-dominated and transient regimes occur at comparable timesteps in the two directions: the forward curve crosses zero at $t \approx 200$ and the reverse curve at $t \approx 150$.

Three quantitative asymmetries distinguish the two trajectories in the single experimental run reported:

1. The reverse transient minimum is slightly deeper than the forward, $S^{\text{rev}}_{\text{min}} = -0.024$ versus $S^{\text{fwd}}_{\text{min}} = -0.022$ (factor ≈ 1.09);
2. The class-dominated Silhouette at $t = 0$ is substantially lower in the reverse direction, $S^{\text{rev}}(0) = +0.031$ versus $S^{\text{fwd}}(0) = +0.089$ (factor ≈ 0.35);
3. The effective dimensionality at $t = 0$ is markedly larger in the reverse direction, $\text{PCP}^{\text{rev}}_{0.80}(0) = 0.189$ versus $\text{PCP}^{\text{fwd}}_{0.80}(0) = 0.030$ (factor ≈ 6.3).

The cohesion–separation decomposition of Sections 3.3.3 and 3.4.3 provides a unifying mechanism for these asymmetries. At $t = 0$ the class-averaged within-class distance ratio is $\bar{a}^{\text{rev}}(0)/\bar{a}^{\text{fwd}}(0) = 1.16$, while the class-averaged nearest-competing-class distance ratio is only $\bar{b}^{\text{rev}}(0)/\bar{b}^{\text{fwd}}(0) = 1.08$: within-class distances are inflated in the generated samples by approximately twice the relative amount of the nearest-competing-class distances. The class-aggregate counterpart of the same observation is $\text{PCP}^{\text{rev}}_{0.80}(0) = 0.189$ versus $\text{PCP}^{\text{fwd}}_{0.80}(0) = 0.030$: the BHD-embedded variance is spread over a larger fraction of principal axes in the generated point cloud than in the forward $t = 0$ point cloud. The three asymmetries listed above are therefore consistent with a single qualitative description of the trained generator: the reverse process reproduces the sign of the forward-process Silhouette at $t = 0$ but yields samples in which the within-class distances are inflated relative to the nearest-competing-class distances. This is in turn consistent with the small but non-zero FID = 2.66 reported in Section 3.2.2.

3.6 Summary of Key Findings

Taken together, Graph-based Silhouette Analysis at 15 diffusion checkpoints, complemented by the class-wise cohesion–separation decomposition, the PC Proportion, and the SASNE visualizations, produces four observations. Each is qualified by the single-run, 50-step checkpoint spacing of the present study.

Three-regime structure (main finding). The global Graph-based Silhouette $S(t)$ displays the same three-regime pattern in the forward and reverse processes: a class-dominated regime with $S > 0$ at small t, a transient regime with a pronounced negative minimum near $t \approx 300$

in the intermediate range $200 \lesssim t \lesssim 500$, and a noise-dominated regime with small-magnitude plateaus ($S^{\text{fwd}} \approx -0.001$, $S^{\text{rev}} \approx -0.004$) for $t \geq 500$. The plateau values in both trajectories are consistent with the random-data baseline at $n = 5000$ and should be read as the absence of class signal rather than as anti-alignment.

The reverse process reproduces the sign of the forward-process Silhouette at $t = 0$, with inflated within-class distances at the endpoint (main finding). The class-dominated endpoint of the reverse trajectory has $S^{\text{rev}}(0) = +0.031 > 0$, so generated samples are on average closer in BHD to other samples of their own class than to samples of any other class—the same as the forward process at $t = 0$. Three quantitative asymmetries between the trajectories agree on a single description of the loss in this matching: the reverse transient minimum is 1.09× deeper than the forward (−0.024 vs. −0.022); the class-dominated Silhouette at $t = 0$ is 35% of the forward $t = 0$ value (+0.031 vs. +0.089); and the endpoint effective dimensionality $\text{PCP}_{0.8}(0)$ is 6.3× the forward $t = 0$ value (0.189 vs. 0.030). The cohesion–separation decomposition (Section 3.4.3) attributes these three asymmetries to a single underlying observation: the class-averaged within-class distance is inflated in the generated samples by 16.1% ($\bar{a}^{\text{rev}}(0) = 0.90$ vs. $\bar{a}^{\text{fwd}}(0) = 0.77$), whereas the class-averaged nearest-competing-class distance is inflated by only 7.9% ($\bar{b}^{\text{rev}}(0) = 0.93$ vs. $\bar{b}^{\text{fwd}}(0) = 0.87$). The four measurements correspond to distinct aspects of the geometry—transient-regime depth, endpoint Silhouette, label-blind dispersion, and the class-averaged distance decomposition—and they agree on the same qualitative direction. This direction is consistent with the small but non-zero FID = 2.66 reported in Section 3.2.2. The numerical values reported throughout this thesis are observations from a single experimental run; their stability across independent random seeds is not assessed.

Class-wise heterogeneity (main finding). In the reverse process, class 9 uniquely maintains a positive Silhouette throughout the transient regime and reaches $s_9^{\text{rev}}(0) = +0.079$, while classes 2 and 7 remain marginal at $t = 0$ ($s_2^{\text{rev}}(0) = -0.023$, $s_7^{\text{rev}}(0) = -0.002$). In both processes, class 4 has negative Silhouette at every checkpoint, with $s_4(0) \approx -0.041$ on both clean and generated data—a pattern consistent with a data-intrinsic property of MNIST rather than a defect of the generator.

Consistency with the Biroli speciation prediction. The approach to the noise-dominated plateau in both trajectories takes place in the timestep window $[400, 550]$. This window contains the speciation time $t_S \approx 543$, computed using the formula of Biroli et al. [1] from the top eigenvalue of the MNIST pixel covariance (Appendix A). A full interpretation of this consistency, and of its limits at the present checkpoint spacing, is given in Chapter 4.

Chapter 4

Conclusion and Discussion

4.1 Summary of Main Results

This thesis has examined the representation dynamics of a denoising diffusion probabilistic model trained on MNIST through a Graph-based geometrical framework. The forward and reverse processes were each evaluated at 15 diffusion checkpoints on 5000 intermediate states. At each checkpoint, an adaptive k-nearest-neighbor graph was constructed, the Biharmonic distance was computed from the symmetric normalized graph Laplacian, and four complementary summaries of class structure were evaluated: Graph-based Silhouette Analysis, its class-wise cohesion–separation decomposition, the Graph-PCA effective dimensionality (PC Proportion), and Shape-Aware SNE embeddings.

The central empirical observation is a three-regime structure in the global Graph-based Silhouette $S(t)$ shared by the forward and reverse trajectories: a class-dominated regime with $S > 0$ at small t; a transient regime with a pronounced negative minimum near $t \approx 300$ in the intermediate range $200 \lesssim t \lesssim 500$; and a noise-dominated regime with small plateaus ($S^{\text{fwd}} \approx -0.001$, $S^{\text{rev}} \approx -0.004$) for $t \geq 500$. In the results we reported, the transient minimum is deeper in the reverse direction (-0.024 vs. -0.022), the class-dominated Silhouette at $t = 0$ is weaker in the reverse direction ($+0.031$ vs. $+0.089$), and the effective dimensionality at the end of the reverse process is inflated relative to its forward counterpart ($\text{PCP}_{0.8}(0) = 0.189$ vs. 0.030). The cohesion–separation decomposition attributes the lower endpoint Silhouette to a larger within-class distance inflation in the generated samples (16.1% inflation of $\bar{a}^{\text{rev}}(0)$ relative to $\bar{a}^{\text{fwd}}(0)$) than nearest-competing-class distance inflation (7.9% inflation of $\bar{b}^{\text{rev}}(0)$ relative to $\bar{b}^{\text{fwd}}(0)$). A class-wise analysis reveals that class 9 uniquely maintains a positive Silhouette throughout the reverse transient regime, while class 4 remains negative at every checkpoint on both real and generated data—a pattern consistent with a data-intrinsic property of MNIST rather than a defect of the generator (the geometric similarity between handwritten "4" and "9" is a plausible cause; see discussion in Section 3.3.2).

The generator on which these findings operate meets the standard benchmarks for unconditional MNIST DDPMs, attaining FID = 2.66 against the training reference (Section 3.2.2). Within our framework, two further diagnostics support the three-regime reading: the effective dimensionality $\text{PCP}_{0.8}(t)$ grows monotonically through the forward trajectory and decays monotonically through the reverse, and SASNE embeddings at representative checkpoints provide visual confirmation of the three-regime transition.

4.2 Interpretation

The three-regime structure documented in Chapter 3 admits a natural interpretation within the mean-field framework of Biroli et al. [1], who predict a *speciation* cross-over in the diffusion trajectory at which the generative process commits to one of several data classes. In this section, we first recall the relevant prediction, then argue that our observed regime boundaries are consistent with it within our checkpoint spacing, and finally highlight observations that go beyond the reach of a population-level analysis.

Biroli speciation prediction. The speciation time t_S is defined in continuous time as $t_S = \frac{1}{2} \ln \Lambda$, where Λ is the top eigenvalue of the data covariance matrix. For MNIST under the pre-processing and linear noise schedule adopted in this thesis, the corresponding discrete timestep is $t_S \approx 543$ (Appendix A). Although the Biroli framework is formulated in the context of the generative (reverse) process, the speciation time is defined in terms of the signal-to-noise ratio $\bar{\alpha}_t$, which takes the same value at the same discrete t in both the forward and reverse directions. The prediction therefore applies to both trajectories.

Observed consistency. In both trajectories, the approach to the noise-dominated plateau occurs in the timestep window $[400, 550]$: the Silhouette reaches the range $[-0.003, -0.005]$ by $t \approx 450$ and remains essentially flat thereafter. The Biroli speciation time $t_S \approx 543$ lies within this window. The convergence of two methodologically distinct measurements—the graph-spectral Silhouette presented here and the score-function mean-field analysis of Biroli—on the same timestep window strengthens the case that this transition is an intrinsic feature of the diffusion chain on MNIST rather than an artifact of either methodology.

It must be emphasized, however, that our checkpoint spacing of 50 timesteps does not permit a quantitative positional agreement: within our data, the observed regime boundary is consistent with $t_S = 543$, but also with $t_S = 500$ or $t_S = 550$. We therefore treat this consistency as a *supplementary observation* of the thesis, not as a claim of validation. A denser sampling within the interval $[500, 550]$ would be required to make the comparison quantitative; this is noted as an avenue for future work in Section 4.3.

We do not attempt an analogous comparison with the Biroli collapse time $t_C = \frac{1}{2} \ln(1 + \sigma^2/(n^{2/d} - 1))$. The collapse cross-over involves information-theoretic machinery (excess entropy, random energy model arguments) that lies outside the standard MatStat curriculum; a full treatment would exceed the scope of this thesis and is left to future work.

Findings beyond the population level prediction. Two of our observations are not addressed by the population-level framework and suggest directions in which graph-spectral analysis contributes information complementary to score-function analyses. First, the class-wise Silhouette ordering—class 9 uniquely maintaining positive Silhouette throughout the reverse transient regime, class 4 remaining negative at every checkpoint—reflects class-label-conditioned geometry that the mean-field prediction averages over. Second, the asymmetry between the forward (-0.022) and reverse (-0.024) transient minima is a property of the trained generator's imperfect reproduction of the real data, inaccessible from a purely population-level analysis of the score function.

Cohesion–separation decomposition of the asymmetries. The two forward–reverse asymmetries—deeper reverse transient minimum and lower reverse endpoint Silhouette—are not separately predicted by the population-level mean-field framework, which is class-blind. The cohesion–separation decomposition reported in Section 3.3.3 and Section 3.4.3 attributes both asymmetries primarily to larger within-class distances on the generated samples, not to smaller nearest-competing-class distances. At $t = 0$, $\bar{a}^{\text{rev}}(0)$ exceeds $\bar{a}^{\text{fwd}}(0)$ by 16.1% , while $\bar{b}^{\text{rev}}(0)$ exceeds $\bar{b}^{\text{fwd}}(0)$ by only 7.9%. The trained generator preserves the global class ar-

rangement in the BHD geometry but produces samples that are more spread out within each class. This is consistent with the inflated $\mathrm{PCP}_{0.8}^{\mathrm{rev}}(0) = 0.189$ versus 0.030, which is also a within-class-dispersion measurement.

Silhouette and PC Proportion as complementary measurements. The Silhouette and the PC Proportion, which we evaluated in parallel, display strikingly different temporal profiles: the forward Silhouette is non-monotone (three regimes), whereas the forward PC Proportion grows monotonically from 0.030 to the isotropic Gaussian limit near 0.66. The reverse exhibits the same pattern reversed in time. This dissociation has a simple reading: the PC Proportion depends only on the geometric dispersion of the samples and is blind to class labels, while the Silhouette depends on the alignment between pairwise distances and class labels. The transient regime of the Silhouette is therefore specifically a property of the label-aware geometric organization, and does not correspond to any non-monotonicity in the overall geometric spread.

4.3 Future Work

The following directions would extend and strengthen the analysis reported in this thesis.

Uncertainty quantification with multi-seed replication and denser checkpoints. The quantitative claims here are observations from a single experimental run with $n = 5000$ samples per process at 15 checkpoints with 50-timestep spacing. Running the pipeline with several independent random seeds and reducing checkpoint spacing to 10–20 timesteps in the critical window $t \in [200, 500]$ would (i) place error bars on the transient-minimum depth (including the 0.006 forward–reverse gap and the cohesion–separation asymmetry of $\approx 16.1\%$ at $t = 0$), (ii) resolve the exact location of the transient minimum, and (iii) sharpen the comparison with the Biroli speciation prediction $t_S \approx 543$ from a within-resolution check to a direct quantitative match. Multi-seed validation would also confirm whether the within-class distance inflation $\bar{a}^{\mathrm{rev}}(0) > \bar{a}^{\mathrm{fwd}}(0)$, rather than reduced nearest-competing-class distances, is a systematic property of the trained generator.

Independent geometric cross-validation with alternative distance metrics. The mechanistic interpretation of the transient regime via cohesion and separation $\bar{a}_c(t), \bar{b}_c(t)$ is derived from a single distance metric—the Biharmonic distance obtained from L_{sym}. Cross-validation with an alternative geometric measurement (for example, the commute-time distance, a diffusion distance at a different time parameter, or extrinsic distances restricted to the manifold) would clarify which features of the mechanism are intrinsic to the data geometry and which are specific to the BHD construction.

Comprehensive study with different datasets and network architectures. The present experiments use MNIST and a single U-Net architecture (18.7M parameters). Extending the framework to CIFAR-10, CelebA, alternative generative architectures, or alternative noise schedules would test the universality of the three-regime structure and the cohesion–separation mechanism, and would clarify which findings are MNIST-specific.

Joint validation of classifier-assigned reverse trajectories. The reverse analysis uses CNN classifier labels assigned once at x_1 and fixed throughout the trajectory. Two natural extensions are: (i) evaluating classifier accuracy as a function of t to quantify label confidence along the reverse trajectory; and (ii) reweighting or resampling the per-class populations (currently non-uniform: class 4 with 818 samples versus class 0 with 341 samples; see Table 3.2) to test whether the class-wise comparisons are robust to the non-uniform distribution.

Full Biroli framework including the collapse transition. The interpretation in Section 4.2 verifies consistency with the Biroli speciation prediction only. A natural follow-up is to incorporate the Biroli collapse time $t_C = \frac{1}{2}\ln(1 + \sigma^2/(n^{2/d} - 1))$ and the full dynamical-regime

phase diagram into a comparative study, extending the analysis into the small-t regime where memorization-versus-generalization trade-offs become visible.

4.4 Conclusion

We have presented a graph-spectral geometrical analysis of the representation dynamics of an MNIST-trained DDPM. The central empirical finding is a three-regime structure in the global Graph-based Silhouette shared by forward and reverse trajectories, with a transient negative minimum near $t \approx 300$ in each direction. Additional observations include forward-reverse asymmetries in minimum depth and final-state geometry, and a class-wise heterogeneity in which class 9 uniquely maintains positive Silhouette through the reverse transient regime. As a supplementary observation, the transient-to-noise- dominated transition is consistent, within our 50-timestep checkpoint spacing, with the Biroli speciation time $t_S \approx 543$. These results establish Graph-based Silhouette as a diagnostic tool for diffusion-model representation dynamics, complementary to score-function analyses. The graph-spectral toolkit established here—adaptive k-nearest-neighbor graph, Biharmonic distance, Graph-based Silhouette with the cohesion–separation decomposition, Graph-PC Proportion, and SASNE—is in principle applicable to other datasets and architectures, with empirical validation across different setups left to future work.

Declarations

Use of AI-based tools. AI-based tools were used during the preparation of this thesis. Specifically, such tools assisted with code organization, README documentation, and debugging of the accompanying code repository; with the generation of tables; and with grammar correction and language polishing of the written text. All conceptual content, analyses, interpretations, and original drafts of figures and text are entirely the author's own. The author takes full responsibility for the accuracy and integrity of the work presented.

Appendix A

Biroli Speciation Time Computation

Biroli et al. [1] predict a *speciation* cross-over in the diffusion trajectory at continuous time

$$t_S^{\text{cont}} = \tfrac{1}{2} \ln \Lambda, \tag{A.1}$$

where Λ is the largest eigenvalue of the data covariance matrix. For MNIST under the pre-processing of Section 3.1.1 (grayscale 32×32 images flattened to vectors in $\mathbb{R}^{1024}$ with pixel intensities in $[-1, 1]$), the covariance matrix $\Sigma \in \mathbb{R}^{1024 \times 1024}$ was computed on the 5000-image class-balanced subsample used for the forward analysis. Its top eigenvalue is $\Lambda \approx 20.27$, giving

$$t_S^{\text{cont}} = \tfrac{1}{2} \ln(20.27) \approx 1.50. \tag{A.2}$$

The continuous time is mapped to the discrete schedule of this thesis (linear $\beta_t \in [10^{-4}, 0.02]$, $T = 1000$) via the signal-to-noise ratio $\bar{\alpha}_t$. The Biroli convention defines $\bar{\alpha}_{t^{\text{disc}}} = e^{-2t^{\text{cont}}}$, so the discrete speciation timestep is the t^{disc} for which

$$\bar{\alpha}_{t^{\text{disc}}} \approx e^{-2t_S^{\text{cont}}} = e^{-3.01} \approx 0.049. \tag{A.3}$$

Solving numerically with the linear β schedule gives $t_S^{\text{disc}} \approx 543$. The closest checkpoints in our schedule are $t = 500$ and $t = 550$, so the speciation time falls within the 50-step interval $[500, 550]$ and the comparison with the empirical regime boundary is resolved to ± 25 steps.

Appendix B

Reproducibility

B.1 Hardware

All experiments were run on a single workstation with one NVIDIA RTX 5080 GPU (16 GB VRAM), 12 CPU cores, and 64 GB system RAM. End-to-end pipeline wall time (DDPM training 50 epochs + forward feature extraction at 15 checkpoints + reverse sampling of 5000 trajectories + graph construction + BHD + Silhouette + PCP + SASNE) was approximately 8.7 hours.

B.2 Software stack

Component	Version
Python	3.14.0
PyTorch	2.11.0* (CUDA 12.8)
NumPy	2.4.2
SciPy	1.17.0
scikit-learn	1.8.0
Plotly	6.5.2
Matplotlib	3.10.8

*PyTorch nightly build `2.11.0.dev20260213+cu128`, required for RTX 5080 (Blackwell architecture) support at the time of experiments.

B.3 Random seeds

The following seeds were fixed at the start of the corresponding pipeline phase:

Phase	Seed
DDPM training (PyTorch)	42
class-balanced 5000-sample selection (NumPy)	123
Reverse trajectory sampling (PyTorch)	456
SASNE init (scikit-learn)	789

A single experimental run is reported throughout the thesis; multi-seed validation is identified as future work in Section 4.3.

B.4 Bandwidth sensitivity

The adaptive kNN bandwidth $a = P_{90}(d_E)/9$ of Section 2.3.2 fixes the edge weight at the 90th-percentile Euclidean distance to $w = 0.01$. We verified that the qualitative findings of Chapter 3 are insensitive to the constant 9: with $a = P_{90}(d_E)/c$ for $c \in \{5, 9, 15\}$, the global Silhouette $S(t)$ is shifted by at most ± 0.003 at every checkpoint and the three-regime structure, the transient-minimum location $t \approx 300$, and the asymmetry direction (reverse $|S|$ larger than forward) are all preserved.

B.5 Code availability

The code accompanying this thesis is publicly available at https://github.com/Xinshuer/ddpm-manifold-graph under the MIT License.

Appendix C

Detailed Numerical Tables

This appendix reports the per-checkpoint values of the four geometric summaries used in Chapter 3. Source: `outputs/summary.json` from the experimental run reported throughout the thesis.

C.1 Global Graph-based Silhouette $S(t)$

t	$S^{\text{fwd}}(t)$	$S^{\text{rev}}(t)$
0	+0.089	+0.031
50	+0.072	+0.018
100	+0.055	+0.014
150	+0.034	+0.004
200	+0.003	−0.012
250	−0.013	−0.024
300	−0.022	−0.024
350	−0.016	−0.013
400	−0.009	−0.008
450	−0.004	−0.006
500	−0.001	−0.005
550	−0.001	−0.004
700	−0.003	−0.006
900	−0.004	−0.006
999	−0.003	−0.005

(Values are computed with point-average aggregation by `spectral_metrics.py`. The class-averaged form $K^{-1}\sum_c S_c(t)$ used in the thesis matches these to within ±0.001 for the balanced forward case; the imbalanced reverse class-averaged values appear in Sections 3.4.1 and 3.4.3.)

C.2 Cohesion $\bar{a}_c(t)$ and separation $\bar{b}_c(t)$ class means

t	$\bar{a}^{\text{fwd}}$	$\bar{b}^{\text{fwd}}$	$\bar{a}^{\text{rev}}$	$\bar{b}^{\text{rev}}$
0	0.77	0.87	0.90	0.93
100	0.83	0.89	0.92	0.94
200	0.94	0.94	0.97	0.96
250	0.98	0.96	0.99	0.97
300	0.99	0.97	1.00	0.97
500	1.00	1.00	1.00	0.99
999	1.00	0.99	1.00	0.99

C.3 PC Proportion $\text{PCP}(\alpha, t)$ at three thresholds

	Forward			Reverse		
t	$\alpha = 0.7$	0.8	0.9	0.7	0.8	0.9
0	0.006	0.030	0.174	0.072	0.189	0.449
100	0.043	0.135	0.381	0.177	0.327	0.575
200	0.286	0.440	0.659	0.368	0.516	0.713
300	0.474	0.609	0.776	0.485	0.618	0.781
500	0.527	0.654	0.805	0.524	0.651	0.803
999	0.529	0.656	0.806	0.528	0.655	0.806

C.4 Class-wise Silhouette $S_c(t)$

The full 15×10 tables of forward and reverse class-wise Silhouettes $S_c(t)$, computed by `spectral_metrics.py` on each diffusion checkpoint, are reproduced below for completeness.

Forward

t	$c = 0$	$c = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$	$c = 9$
0	+0.285	+0.166	+0.026	+0.093	−0.041	+0.012	+0.138	+0.049	+0.083	+0.079
50	+0.245	+0.087	+0.027	+0.075	−0.036	+0.015	+0.106	+0.039	+0.080	+0.078
100	+0.191	+0.074	+0.023	+0.057	−0.028	+0.008	+0.070	+0.026	+0.068	+0.064
150	+0.129	+0.065	+0.007	+0.033	−0.025	−0.004	+0.028	+0.006	+0.050	+0.048
200	+0.030	+0.010	−0.018	+0.003	−0.013	−0.004	−0.026	−0.011	+0.034	+0.028
250	−0.044	−0.033	−0.013	−0.007	−0.005	+0.002	−0.040	−0.017	+0.010	+0.013
300	−0.064	−0.057	−0.017	−0.013	−0.009	−0.007	−0.040	−0.022	−0.004	+0.014
350	−0.046	−0.049	−0.012	−0.009	−0.009	−0.004	−0.025	−0.011	−0.005	+0.011
400	−0.024	−0.032	−0.007	−0.003	−0.006	−0.003	−0.013	−0.004	−0.001	+0.006
450	−0.009	−0.018	−0.003	+0.001	−0.004	−0.002	−0.006	+0.002	+0.001	+0.002
500	−0.003	−0.010	−0.001	+0.002	−0.002	−0.001	−0.002	+0.001	+0.000	+0.002
550	+0.001	−0.007	−0.001	+0.002	−0.003	−0.000	−0.002	+0.002	+0.002	+0.001
700	+0.005	−0.008	−0.004	−0.001	−0.005	−0.003	−0.005	−0.002	−0.001	−0.003
900	+0.006	−0.009	−0.005	−0.001	−0.006	−0.004	−0.006	−0.004	−0.003	−0.005
999	+0.005	−0.008	−0.005	−0.001	−0.006	−0.004	−0.006	−0.004	−0.002	−0.005

Reverse

t	$c = 0$	$c = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$	$c = 8$	$c = 9$
0	+0.063	+0.157	−0.023	+0.031	−0.041	+0.005	+0.075	−0.001	+0.024	+0.078
50	+0.042	+0.077	−0.018	+0.031	−0.044	+0.003	+0.051	−0.009	+0.027	+0.079
100	+0.027	+0.063	−0.015	+0.029	−0.042	+0.004	+0.037	−0.016	+0.026	+0.070
150	+0.004	+0.048	−0.016	+0.019	−0.036	+0.002	+0.011	−0.032	+0.015	+0.053
200	−0.022	−0.007	−0.023	+0.001	−0.033	−0.005	−0.031	−0.033	−0.001	+0.043
250	−0.056	−0.046	−0.023	−0.013	−0.032	+0.000	−0.050	−0.038	−0.006	+0.021
300	−0.061	−0.050	−0.017	−0.011	−0.038	−0.002	−0.037	−0.027	−0.007	+0.012
350	−0.039	−0.035	−0.006	−0.001	−0.027	+0.003	−0.012	−0.011	−0.001	+0.002
400	−0.018	−0.017	−0.007	+0.003	−0.022	+0.002	−0.001	−0.007	−0.001	+0.001
450	−0.007	−0.008	−0.005	+0.001	−0.018	+0.003	−0.007	−0.005	−0.001	+0.002
500	−0.003	−0.005	−0.007	−0.006	−0.013	+0.003	−0.005	−0.004	−0.004	+0.002
550	+0.003	−0.003	−0.004	−0.003	−0.014	−0.000	−0.002	−0.003	−0.004	+0.002
700	−0.004	−0.009	−0.009	−0.003	−0.011	+0.006	−0.006	−0.002	−0.006	−0.006
900	+0.008	−0.008	−0.005	−0.003	−0.016	−0.003	−0.006	−0.004	−0.003	−0.004
999	+0.006	−0.003	−0.005	−0.001	−0.010	−0.003	−0.001	−0.009	−0.008	−0.004

Appendix D

Adaptive k-Nearest-Neighbor Connectivity Statistics

The adaptive procedure of Algorithm 1 starts from $k_{\min} = 5$ and increments k until the graph is connected, with $k_{\max} = 30$. Table D.1 reports, for each diffusion checkpoint, the value of k that was actually used, the number of vertices removed as outliers, and the algebraic connectivity (Fiedler eigenvalue) λ_2 of the resulting Laplacian.

	Forward			Reverse		
t	k used	outliers	λ_2	k used	outliers	λ_2
0	5	0	0.0096	5	0	0.0190
100	5	0	0.0114	5	0	0.0216
200	5	0	0.0273	5	0	0.0414
300	5	0	0.1096	5	0	0.1421
500	5	0	0.3809	5	0	0.3883
999	5	0	0.3993	5	0	0.3980

Table D.1: Adaptive kNN parameters and Fiedler eigenvalue at each diffusion checkpoint.

Across all 30 checkpoints (forward + reverse), the adaptive procedure terminated at $k = 5$ and removed zero outliers. The Fiedler eigenvalue λ_2 grows monotonically with t in both directions, consistent with the noise-induced loss of cluster structure and matching the regime boundaries documented in Section 3.3.1.

Appendix E

CNN Classifier for Reverse-Trajectory Labels

Reverse trajectories are not associated with a ground-truth class label by construction: each trajectory is initialized from $x_T \sim \mathcal{N}(0, I)$ and the generator's reverse process produces a sample x_0 without any class conditioning. To assign a class label for the per-class analysis of Section 3.4.2, we use a separately trained CNN classifier and apply it to the near-clean state x_1 rather than x_0. The trajectory inherits this label for all checkpoints $t \in \{0, 50, \ldots, 999\}$.

E.1 Architecture

A standard four-layer CNN with two convolutional blocks (Conv → ReLU → MaxPool) followed by two fully connected layers (128 → 10) and a softmax output. Total parameters: [N]. Trained on the 60,000-image MNIST training set for [E] epochs with Adam ($\eta = 10^{-3}$).

E.2 Test accuracy

The classifier reaches 99.[X]% accuracy on the 10,000-image MNIST test set. The confusion matrix (rows = true class, columns = predicted class) is reported in the supplementary file `outputs/cnn_confusion_matrix.csv`. Off-diagonal mass concentrates in the 4-vs-9 cell, as expected for MNIST.

E.3 Why x_1 rather than x_0

At x_0 the generator's final-step denoising occasionally introduces artifacts (over-sharpening, faint background texture) absent from the near-clean x_1. Empirically the classifier confidence margin (top-1 minus top-2 softmax probability) is higher at x_1 than at x_0 on 94% of trajectories, and the label assignments agree on 98.5% of trajectories. We use x_1 for the small confidence-margin gain.

Bibliography

[1] Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

[2] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[4] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[10] Yaron Lipman, Raif M Rustamov, and Thomas A Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):1–11, 2010.

[11] Karl Pearson. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[12] 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.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[14] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2256–2265, 2015.

[16] Nik Tavakolian and Chun-Biu Li. A graph-based framework for interpreting the geometric evolution of hidden representations. to be submitted, 2026.

[17] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[19] Tobias Wängberg, Joanna Tyrcha, and Chun-Biu Li. Shape-aware stochastic neighbor embedding for robust data visualisations. *BMC Bioinformatics*, 23(1):477, 2022.

[20] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.