Understanding Diffusion-based Representation Learning via Low-Dimensional Modeling

Anonymous Author(s)
Affiliation
Address
email

Abstract

This work addresses the critical question of why and when diffusion models, despite their generative design, are capable of learning high-quality representations in a self-supervised manner. We hypothesize that diffusion models excel in representation learning due to their ability to learn the low-dimensional distributions of image datasets via optimizing a noise-controlled denoising objective. Our empirical results support this hypothesis, indicating that variations in the representation learning performance of diffusion models across noise levels are closely linked to the quality of the corresponding posterior estimation. Grounded on this observation, we offer theoretical insights into the unimodal representation dynamics of diffusion models as noise scales vary, demonstrating how they effectively learn meaningful representations through the denoising process. We also highlight the impact of the inherent parameter-sharing mechanism in diffusion models, which accounts for their advantages over traditional denoising auto-encoders in representation learning.

1 Introduction

Diffusion models, a new family of likelihood-based generative models, have demonstrated superior performance among many generative tasks, including image generation [Alkhouri et al., 2024, Ho et al., 2020, Rombach et al., 2022, Zhang et al., 2024], video generation [Bar-Tal et al., 2024, Ho et al., 2022], speech and audio synthesis [Kong et al., 2020, 2021], semantic editing [Roich et al., 2022, Ruiz et al., 2023, Chen et al., 2024a] and solving inverse problem [Chung et al., 2022, Song et al., 2024, Li et al., 2024, Alkhouri et al., 2023]. At its core, diffusion models are learning a data distribution from training samples by imitating the non-equilibrium thermodynamic diffusion process [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021]. In the forward process, training samples are gradually combined with increasing Gaussian noise until the data structure is completely destroyed while in the backward process, a model is trained to restore the structure from the noised data [Hyvärinen and Dayan, 2005, Song et al., 2021].

In addition to their impressive generative capabilities, recent studies [Baranchuk et al., 2021, Xiang et al., 2023, Mukhopadhyay et al., 2023, Chen et al., 2024b, Tang et al., 2023] have highlighted the exceptional representation power of diffusion models, suggesting that they could serve as a unified foundation model for both generative and discriminative vision tasks. Specifically, recent evaluations across various applications, including classification [Xiang et al., 2023, Mukhopadhyay et al., 2023], semantic segmentation [Baranchuk et al., 2021], and image alignment [Tang et al., 2023], show that diffusion models are capable of learning high-quality representations, often matching or even surpassing the performance of previous state-of-the-art methods. However, it remains unclear whether the representation capabilities of diffusion models stem from the diffusion process or the denoising

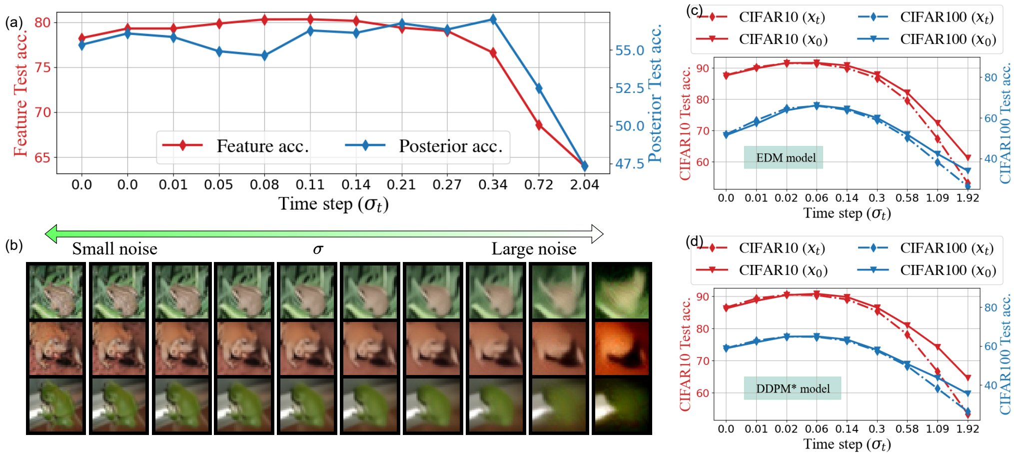


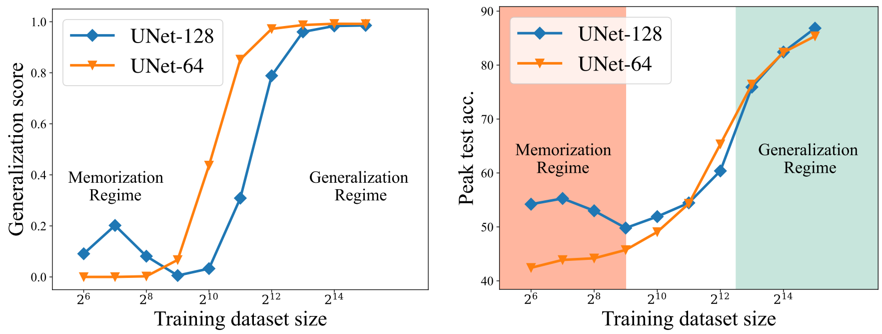
Figure 1: **Representation learning ability of a diffusion model at different time steps reflects the granularity in posterior estimation.** (a) Intermediate feature accuracy and posterior accuracy of the diffusion model exhibit a similar unimodal trend as noise level increases. (b) Posterior estimation for clean image inputs shows a transition from fine to coarse granularity with increasing noise levels. (c)-(d) Using clean image input $\boldsymbol{x}_0$ for feature extraction achieves comparable or superior representation learning performance compared to using noisy input $\boldsymbol{x}_t$.

mechanism [Fuest et al., 2024]. More fundamentally, given their generative design, *when and why diffusion models can learn high-quality representations in a self-supervised manner?*

This work aims to address this question through a comprehensive investigation, both empirically and theoretically, grounded in the formulation of denoising auto-encoders (DAEs) for learning diffusion models [Vincent et al., 2008, 2010, Vincent, 2011]. We hypothesize that diffusion models can learn high-quality representations without supervision due to their superior ability to approximate the low-dimensional distributions of image datasets, as supported by recent findings [Wang et al., 2024]. Although image dataset can be very high-dimensional, recent results [Pope et al., 2021, Stanczuk et al., 2022, Wang et al., 2024] demonstrate that the intrinsic dimension of these datasets are much lower than the ambient dimension, and it has shown that the number of samples to learn the underlying distribution using diffusion models scales with the intrinsic low-dimensionality. Therefore, by being trained to capture the underlying structure of data through a controlled process of noise injection and denoising, diffusion models effectively learn meaningful and compact features.

On the empirical side, we support our claim by reconciling several intriguing phenomena related to the quality of learned representations in diffusion models. Recent studies Zhang et al. [2023] reveal that diffusion models operate in two regimes: memorization and generalization, depending on training data size. In the memorization regime with limited samples, the model captures only the empirical distribution of training data without the ability to generate new samples. In contrast, in the generalization regime, diffusion models are able to learn the underlying distribution. Our experiments in Figure 2 confirm that high-quality representations are *only* learned in the generalization regime with sufficient samples due to its ability of learning the underlying distribution. More importantly, in the generalization regime, we show that the quality of hidden representations in diffusion models/DAEs follows a uni-modal curve (see Figure 1 and Figure 7): high-quality representations are learned at an intermediate step close to the clean image, whereas the representation quality degrades as it approaches either pure noise or the clean image.

Building on these empirical observations, we provide theoretical insights using a noisy mixture of low-rank Gaussian distributions. Our assumption captures the inherent low-dimensionality of the image data distribution [Pope et al., 2021, Gong et al., 2019, Stanczuk et al., 2022], where the data lies on a union of low-dimensional subspaces. We analyze the unimodal trend in representation performance by relating it to the Class-specific Signal-to-Noise Ratio (CSNR). Specifically, we consider the optimal posterior estimation function under our data assumption and show that the CSNR is determined by the interplay between data "denoising" and class confidence rate as the noise scale increases. Additionally, our study reveals an implicit weight-sharing mechanism inherent in diffusion models, which helps explain their strengths compared to traditional one-step DAEs, particularly in the small noise regions.



(a) Phase transition in generalization score (b) Phase transition in representation learning

Figure 2: **Better representations are learned in the generaliation regime.** We train EDM-based [Karras et al., 2022] diffusion models on the CIFAR-10 dataset using different training dataset sizes, ranging from 2^6 to 2^{15}. (a) The change in the generalization score [Zhang et al., 2023] as the dataset size increases, where regions with a generalization score close to 0 are labeled as the memorization regime, and those close to 1 are labeled as the generalization regime. (b) The peak representation learning accuracy achieved as a function of dataset size.

Contribution of this work. In summary, our findings can be highlighted as follows:

- **Linking posterior estimation ability of diffusion models to representation learning.** Our empirical results reveal that, much like the dynamics of diffusion representation learning, posterior estimation quality across noise levels follows a similar unimodal curve. This indicates that changes in representation quality are a direct reflection of changes in posterior estimation quality, prompting us to explore representation learning through the more fundamental lens of posterior recovery.
- **Theoretical analysis of the unimodal curve in the denoising process.** Building on the connection between posterior estimation and representation learning, we present the first theoretical framework for analyzing the unimodal evolution of representation quality. Using a mixture of low-rank Gaussian data model, we demonstrate that the unimodal curve arises from the interplay between denoising strength and class confidence as the noise level varies.
- **Weight sharing in the diffusion process.** Furthermore, we reveal that the diffusion process, by minimizing losses across all noise levels simultaneously, fosters an implicit parameter sharing mechanism within a diffusion model. This mechanism plays a crucial role for diffusion models to achieve superior and more consistent representation learning performances compared with traditional DAEs.

2 Representation Learning via diffusion models

In this section, we first review the fundamentals of diffusion models and outline the feature extraction method used in this work. Following this, we illustrate the connection between diffusion posterior estimation and representation learning, which serves as the foundation for the subsequent analysis in Section 3.

2.1 Preliminaries on denoising diffusion models

Diffusion models are a class of probabilistic generative models that aim to reverse a progressive noising process by mapping an underlying data distribution, p_{data}, to a Gaussian distribution.

The forward process. Starting from clean data $\boldsymbol{x}_0$, noise is gradually introduced according to a noise schedule determined by the time step t until the data becomes indistinguishable from pure Gaussian noise. Specifically, at any time step t, the noised data can be expressed as: $\boldsymbol{x}_t = s_t \boldsymbol{x}_0 + s_t \sigma_t \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ represents noise sampled from a Gaussian distribution, s_t and $s_t \sigma_t$ represent the scaling of the signal and noise, respectively.

The reverse process. Noise is gradually removed from $\boldsymbol{x}_1$ following the reverse-time SDE:

$$\mathrm{d}\boldsymbol{x}_t = \left(f(t)\boldsymbol{x}_t - g^2(t)\nabla \log p_t(\boldsymbol{x}_t)\right)\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t, \tag{1}$$

where $\{\bar{\boldsymbol{w}}_t\}_{t\in[0,1]}$ is the standard Wiener process running backward in time from $t = 1$ to $t = 0$ and the functions $f(t), g(t) : \mathbb{R} \to \mathbb{R}$ respectively denote the drift and diffusion coefficients. Notably, if both $\boldsymbol{x}_1$ and $\nabla \log p_t$ are known, the reverse process mirrors the forward process at each time step $t \geq 0$ [Anderson, 1982].

Score approximation and denoising auto-encoders (DAEs). However, the score function $\nabla \log p_t$ is typically unknown, as it depends on the underlying data distribution p_{data}. To address this, a neural network $s_{\boldsymbol{\theta}}$ is trained to estimate the score at various time steps [Ho et al., 2020, Song et al., 2021]. Given the relationship between the score function and the posterior mean $\mathbb{E}[\hat{\boldsymbol{x}_0}|\boldsymbol{x}_t]$ [Vincent, 2011, Wang et al., 2024]:

$$s_t\,\mathbb{E}\left[\hat{\boldsymbol{x}_0}|\boldsymbol{x}_t\right] = \boldsymbol{x}_t + s_t^2\sigma_t^2\nabla\log p_t(\boldsymbol{x}_t) \approx \boldsymbol{x}_t + s_t^2\sigma_t^2 s_{\boldsymbol{\theta}}(\boldsymbol{x}_t), \tag{2}$$

prior works [Chen et al., 2024b, Xiang et al., 2023, Kadkhodaie et al., 2023] have also proposed an alternative DAE-based training objective that directly estimates the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$:

$$\min_{\boldsymbol{\theta}}\ \ell(\boldsymbol{\theta}) := \frac{1}{2N}\sum_{i=1}^{N}\int_0^1 \lambda_t\,\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_n)}\left[\left\|\boldsymbol{x}_{\boldsymbol{\theta}}(s_t\boldsymbol{x}_0^{(i)} + s_t\sigma_t\boldsymbol{\epsilon}, t) - \boldsymbol{x}_0^{(i)}\right\|^2\right]\mathrm{d}t, \tag{3}$$

where $\boldsymbol{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_0, t)$ denotes the posterior estimating network, N represents the size of the training dataset, and λ_t denotes the weighting for each noise level. To simplify the analysis, we assume throughout the paper that $s_t = 1$ and λ_t remain constant across all noise levels, with the noise level denoted as σ_t.

We note that if we remove the integration in (3) and fix t, the loss simplifies to the traditional single-level DAE loss [Vincent et al., 2008], where the DAE is trained at a single noise level. Previous work [Chen et al., 2024b] has decomposed the training objective of diffusion models into the denoising process (through the denoising loss) and the diffusion process (integrating the loss across all noise levels in (3)). To comprehensively investigate the distinct roles of these two processes in representation learning, we consider both diffusion models and individual DAEs in our experiments where the individual DAEs serve as a control group, allowing us to isolate and analyze the effects of the denoising process alone.

2.2 Extracting representations from diffusion model

In this work, we adopt the following feature extraction setups to leverage diffusion models for representation learning:

Use clean images as network inputs. First, we use the clean image $\boldsymbol{x}_0$ as input to the network in contrast to conventional approaches that use the noisy image $\boldsymbol{x}_t$ [Xiang et al., 2023, Baranchuk et al., 2021, Tang et al., 2023]. This setup aligns with the goal of representation learning, where additive noise is not necessary (e.g., similar to training a classifier with data augmentations while using non-augmented data during inference). As demonstrated in Figure 1(c)-(d), this approach preserves the overall unimodal representation dynamic while achieving better performance at higher noise levels. As such, throughout the remainder of this paper, we use the clean data $\boldsymbol{x}_0$ as input to the diffusion model, i.e., we always consider $\boldsymbol{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_0, t)$ where t serves solely as an indicator of the noise level for diffusion model to adopt during feature extraction.

Layer selection for representations. Second, we extract features only from the bottleneck layer of the U-Net architecture [Ronneberger et al., 2015],[1] following the protocols used in [Kwon et al., 2022, Park et al., 2023].[2] Unlike prior methods [Xiang et al., 2023, Baranchuk et al., 2021], we do not conduct a grid search for the optimal layer, as our focus is on understanding the process rather than achieving state-of-the-art results.

2.3 Relationship Between Learned Representations & Posterior Estimation

Relationship among posterior estimation, distribution recovery, and representation learning. Since directly studying representation ability is challenging, in Section 3 we approach the problem through its strong correlation with posterior mean estimation, $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$. As we will argue, diffusion representation quality is closely linked with the semantic information encoded in the posterior estimation. Additionally, empirical validations can be found in Figure 1.

- *Posterior estimation and distribution recovery.* Diffusion models are trained to learn the underlying data distribution by reconstructing the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ for a given input $\boldsymbol{x}_t$ at the specified noise level. Therefore, the quality of posterior estimation $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ reflects the degree to which the underlying distribution is captured [Choi et al., 2022].

[1] In other words, the layer with the smallest feature resolution.

[2] After feature extraction, we apply a global average pooling to the features. For instance, given a feature map of dimension $256 \times 4 \times 4$, we pool the last two dimensions, resulting in a 256-dimensional vector.

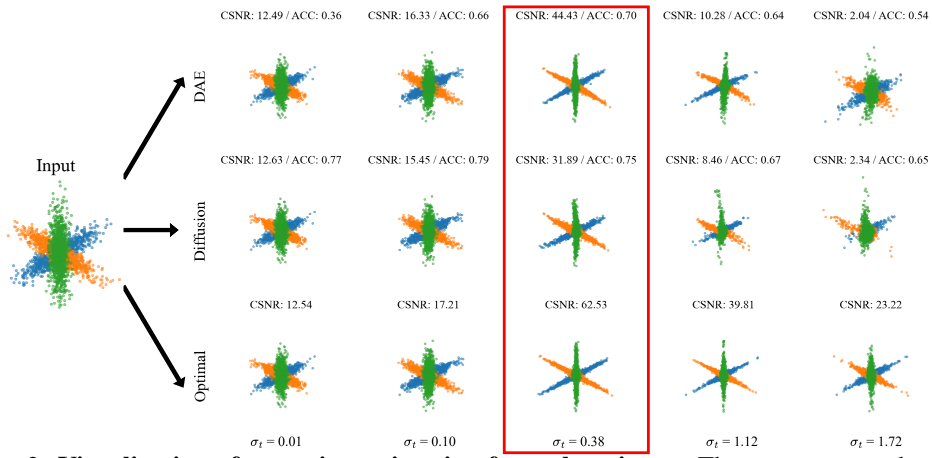


Figure 3: **Visualization of posterior estimation for a clean input.** The **same** `MoLRG` data is fed into the models; each row represents a different denoising model, and each column corresponds to a different time step with noise scale (σ_t). The red box indicates the best posterior estimation and feature probing accuracy.

- *Representation learning through distribution approximation.* On the other hand, achieving high-quality distribution approximation results in more meaningful and informative representations in unsupervised learning. This is supported by Figure 2, where the findings, inspired by recent works [Zhang et al., 2023], demonstrate that diffusion models transition from memorizing the training data distribution to accurately approximating the underlying data distribution as the amount of training data increases. Consequently, better approximation of the underlying data distribution improves the quality of representation learning.

Given this relationship, we use posterior estimation as a proxy for representation quality throughout our analysis. Additionally, since diffusion models tend to memorize the training data instead of learning underlying data distribution when the training dataset is small [Zhang et al., 2023], we focus on the case where sufficient training data is available throughout our analysis in Section 3.

Unimodal curve of representation quality. Previous studies [Xiang et al., 2023, Baranchuk et al., 2021, Tang et al., 2023] have empirically shown that the representation dynamics of diffusion models follow a unimodal curve as the noise scale increases, across various tasks such as classification, segmentation, and image correspondence. Our findings corroborate this observation, as demonstrated in Figure 1(a), where the representation quality consistently exhibits a unimodal trend, regardless of the specific network architecture or dataset used (see Figure 1(c)-(d)). In the following analysis, we argue that this unimodal behavior arises from subtle differences between the requirements of representation learning and the generative nature of diffusion models.

High-fidelity image generation demands that diffusion models capture every aspect of the data distribution—from coarse structures to fine details. In contrast, representation learning, particularly for high-level tasks such as classification [Allen-Zhu and Li, 2022], prefers an abstract representation, where finer image details may even act as 'noise' that hinders performance. As shown in Figure 1(b), as the noise level increases, the predicted posteriors for clean input $\boldsymbol{x}_0$ transition from 'fine' to 'coarse' [Wang and Vastola, 2023], gradually removing fine-grained details. For the classification task in the plot, the best performance is achieved when the posterior estimation retains the essential information while discarding some class-irrelevant details. These findings indicate a trade-off between generative quality and representation performance [Chen et al., 2024b], prompting us to attribute variations in feature quality across noise levels to differences in posterior prediction.

3 Theoretical Understanding Through Low-Dimensional Models

In this section, we theoretically examine the representation learning capabilities of diffusion models across varying noise levels by evaluating the quality of posterior estimation, $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ for low-dimensional distributions.

3.1 Assumptions of Low-Dimensional Data Distribution

Although real-world image datasets are high-dimensional in terms of pixel count and data volume, extensive empirical studies Gong et al. [2019], Pope et al. [2021], Stanczuk et al. [2022] suggest that their intrinsic dimensionality is considerably lower. Moreover, state-of-the-art large-scale diffusion models [Peebles and Xie, 2023, Podell et al., 2023] commonly employ auto-encoders [Kingma, 2013] to map images to a low-dimensional latent space [Rombach et al., 2022] for better training efficiency. Consequently, image datasets often reside on a union of low-dimensional manifolds.

In light of this, many recent studies of diffusion models have been focused on approximating low-dimensional distributions [Wang et al., 2024]. Moreover, as union of low-dimensional manifolds can be locally approximated by a union of linear subspaces, it motivates us to model the underlying data distribution as a mixture of low-rank Gaussians (`MoLRG`). The data points generated by `MoLRG` lie on a union of subspaces. Within each subspace, the data follows a Gaussian distribution with a low-rank covariance matrix that represents the subspace basis. Formally, we introduce a noisy version of the `MoLRG` distribution as follows:

Assumption 1 (K-Subspace Noisy `MoLRG` Distribution)**.** *For any sample $\boldsymbol{x}_0$ drawn from the noisy `MoLRG` distribution with K subspaces, the following holds:*

$$\boldsymbol{x}_0 = \boldsymbol{U}_k \boldsymbol{a} + \delta \boldsymbol{U}_k^{\perp} \boldsymbol{e}, \ \textit{with probability } \pi_k \geq 0, \ k \in [K]. \tag{4}$$

Here, $\sum_{k=1}^{K} \pi_k = 1$, $\boldsymbol{U}_k \in \mathcal{O}^{n \times d_k}$ denotes an orthonormal basis for the k-th subspace, d_k is the subspace dimension with $d_k \ll n$, and the coefficient $\boldsymbol{a} \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_k})$ is drawn from a standard normal distribution. For the noise, we assume $\boldsymbol{e} \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n-d_k})$ with magnitude controlled by the scalar $\delta < 1$. Additionally, $\boldsymbol{U}_k^{\perp} \in \mathcal{O}^{n \times (n-d_k)}$ is the orthogonal compliment of $\boldsymbol{U}_k$.

For simplicity of analysis, we let $d_1 = \cdots = d_K = d$, and we assume that the basis $\{\boldsymbol{U}_k\}$ are orthogonal to each other with $\boldsymbol{U}_k^T \boldsymbol{U}_l = \mathbf{0}$ for all $k \neq l$. Additionally, we assume all mixing weights $\{\pi_k\}$ are equal with $\pi_1 = \cdots = \pi_K = 1/K$, and we define $\boldsymbol{U}_{\perp} = \bigcap_{k=1}^{K} \boldsymbol{U}_k^{\perp} \in \mathcal{O}^{n \times (n-Kd)}$ to be the noise space that is the orthogonal complement to all basis $\{\boldsymbol{U}_k\}_{k=1}^{K}$.

We note that the noise term $\delta \boldsymbol{U}_k^{\perp} \boldsymbol{e}_i$ captures perturbations unrelated to the k-th subspace via the orthogonal complement $\boldsymbol{U}_k^{\perp}$, thereby aligning the model more closely with real-world scenarios. These perturbations can be interpreted as attributes irrelevant to the subspace, such as the background in an image of a bird or the color/texture of a car. The extra noise term may not be relevant for representation learning, but it plays an importance role for diffusion model to generate high-fidelity samples. Additionally, for the noisy `MoLRG` distribution, ground truth posterior mean $\mathbb{E}[\hat{\boldsymbol{x}}_0 | \boldsymbol{x}_t]$ is:

Proposition 1. *For a K-class `MoLRG` data distribution, for each time $t > 0$, it holds that*

$$\boldsymbol{x}_{\boldsymbol{\theta}}^{\star}(\boldsymbol{x}_t, t) := \mathbb{E}[\hat{\boldsymbol{x}}_0 | \boldsymbol{x}_t] = \sum_{k=1}^{K} w_k^{\star}(\boldsymbol{x}_t) \left(\frac{1}{1+\sigma_t^2} \boldsymbol{U}_k \boldsymbol{U}_k^T + \frac{\delta^2}{\delta^2 + \sigma_t^2} \boldsymbol{U}_k^{\perp} \boldsymbol{U}_k^{\perp T} \right) \boldsymbol{x}_t \tag{5}$$

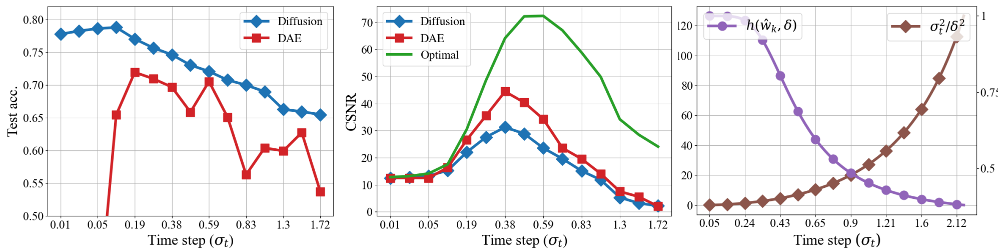
$$\textit{where } w_k^{\star}(\boldsymbol{x}_t) := \frac{\exp\left(g_k(\boldsymbol{x}_t, t)\right)}{\sum_{k=1}^{K} \exp\left(g_k(\boldsymbol{x}_t, t)\right)}, \tag{6}$$

$$\textit{and } g_k(\boldsymbol{x}) = \frac{1}{2\sigma_t^2(1+\sigma_t^2)} \|\boldsymbol{U}_k^T \boldsymbol{x}\|^2 + \frac{\delta^2}{2\sigma_t^2(\delta^2+\sigma_t^2)} \|\boldsymbol{U}_k^{\perp T} \boldsymbol{x}\|^2. \tag{7}$$

Remark. In the above proposition, we present the ground truth posterior estimation function that a diffusion model can achieve by minimizing the training objective defined in (3). We denote this optimal model $\boldsymbol{x}_{\boldsymbol{\theta}}^{\star}$. Given the established relationship between posterior estimation and representation learning on clean inputs $\boldsymbol{x}_0$, we can now analyze the representation learning dynamics under this optimal setting by evaluating $\boldsymbol{x}_{\boldsymbol{\theta}}^{\star}(\boldsymbol{x}_0, t)$ at different time step t.

3.2 Main Theoretical Results

As we discussed in Section 2.3, based upon the strong correlation between representation quality and the posterior mean estimation, we analyze $\boldsymbol{x}_{\boldsymbol{\theta}}^{\star}(\boldsymbol{x}_0, t)$ across different time step $t \in [0, 1]$. Here, we use $\boldsymbol{x}_0$ as the input instead of $\boldsymbol{x}_t$ according to our discussion in Section 2.2.



(a) Probing accuracy of diffusion models and DAEs

(b) CSNR of diffusion model and DAEs

(c) Interplay between denoising rate (σ_t^2/δ^2) and class confidence ($h(\hat{w}_k, \delta)$)

Figure 4: **Dynamics of feature probing accuracy,** CSNR**, and denoising/class confidence rate with increasing noise levels.** Panels (a) and (b) show the feature probing accuracy and CSNR trends using the same `MoLRG` data as in Figure 3, both exhibiting a unimodal pattern. The interplay between the "denoising rate" and the class confidence rate for the approximate optimal solution $f^\star$ is illustrated in panel (c).

Given $\boldsymbol{x}_0 \sim$ `MoLRG` and without loss of generality, let k represent the true class to which x_0 belongs. We quantify the accuracy of posterior mean estimation by introducing a measure of Class-specific Signal-to-Noise Ratio (CSNR) as follows:

$$\mathrm{CSNR}(t, \boldsymbol{x}_{\boldsymbol{\theta}}^\star) := \frac{\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2]}{\mathbb{E}_{\boldsymbol{x}_0}[\sum_{l \neq k} \|\boldsymbol{U}_l \boldsymbol{U}_l^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2]} \tag{8}$$

We know that successful prediction of the class for $\boldsymbol{x}_0$ occurs when the class-specific signal $\|\boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|$ dominates over the noise term $\|\boldsymbol{U}_k^\perp \boldsymbol{U}_k^{\perp T} \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|$. On the other hand, because

$$\|\boldsymbol{U}_k^\perp \boldsymbol{U}_k^{\perp T} \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2 = \sum_{l \neq k} \|\boldsymbol{U}_l \boldsymbol{U}_l^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2 + \|\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2$$

and $\boldsymbol{U}_\perp$ does not affect classification due to its presence in every data point, it leads to our definition of CSNR in (8) which measures the ratio between the true class signal and irrelevant noise from other classes.

Therefore, intuitively, a higher CSNR indicates a better recovery of the underlying low-dimensional data subspace, and thus the predicted posterior is more likely to be assigned to the correct class. This is supported by Figure 4(a)-(b) which shows that both $\mathrm{CSNR}(t)$ and classification accuracy using the learned representation follow similar unimodal curves.

To simplify the calculation of (8), which involves the expectation over the softmax term $w_k^\star$, we approximate $\boldsymbol{x}_{\boldsymbol{\theta}}^\star$ as follows:

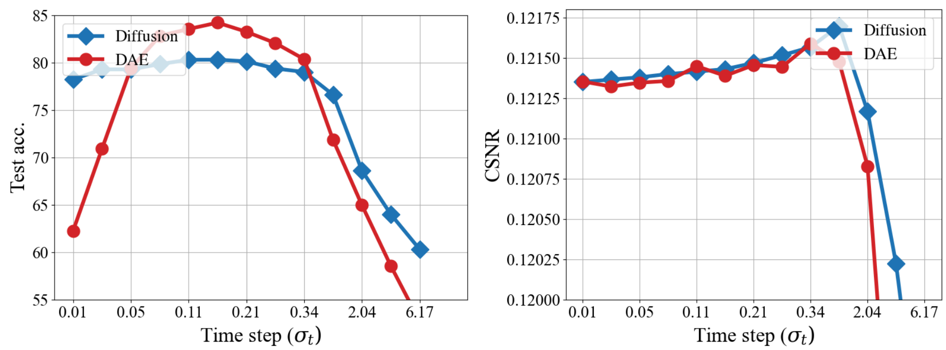
$$\begin{gathered} f^\star(\boldsymbol{x}, t) = \sum_{k=1}^K \hat{w}_k \left(\frac{1}{1+\sigma_t^2} \boldsymbol{U}_k \boldsymbol{U}_k^T + \frac{\delta^2}{\delta^2 + \sigma_t^2} \boldsymbol{U}_k^\perp \boldsymbol{U}_k^{\perp T} \right) \boldsymbol{x}, \\ \text{where } \hat{w}_k := \frac{\exp\left(\mathbb{E}_{\boldsymbol{x}_0}[g_k(\boldsymbol{x}_0, t)]\right)}{\sum_{k=1}^K \exp\left(\mathbb{E}_{\boldsymbol{x}_0}[g_k(\boldsymbol{x}_0, t)]\right)}. \end{gathered} \tag{9}$$

In other words, we use $\hat{w}_k$ in (9) to approximate $w_k^\star(\boldsymbol{x}_0)$ in (6) by taking expectation inside the softmax with respect to $\boldsymbol{x}_0$. This allows us to treat $\hat{w}_k$ as a constant when calculating CSNR, making the analysis more tractable while maintaining $\mathbb{E}[\|\boldsymbol{U}_l \boldsymbol{U}_l^T \boldsymbol{x}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_0, t)\|^2] \approx \mathbb{E}[\|\boldsymbol{U}_l \boldsymbol{U}_l^T f^\star(\boldsymbol{x}, t)(\boldsymbol{x}_0, t)\|^2]$ for all $l \in [K]$. We verify the tightness of this approximation at Appendix A.3 (Figure 9). Now, we are ready to state our main theorem as follows.

Theorem 1. *Let data $\boldsymbol{x}_0$ be any arbitrary data point drawn from the `MoLRG` distribution defined in Assumption 1 and let k denote the true class $\boldsymbol{x}_0$ belongs to. Then* CSNR *introduced in* (8) *depends on the noise level σ_t in the following form:*

$$\mathrm{CSNR}(t, f^\star) = \frac{1}{(K-1)\delta^2} \cdot \left(\frac{1 + \frac{\sigma_t^2}{\delta^2} h(\hat{w}_k, \delta)}{1 + \frac{\sigma_t^2}{\delta^2} h(\hat{w}_l, \delta)} \right)^2 \tag{10}$$

where $h(w, \delta) := (1-\delta^2)w + \delta^2$. Since δ is fixed, $h(w, \delta)$ is a monotonically increasing function with respect to w. Note that here δ represents the magnitude of the fixed intrinsic noise in the data where σ_t denotes the level of additive Gaussian noise introduced during the diffusion training process.



(a) Probing Accuracy of diffusion models and DAEs

(b) CSNR of diffusion models and DAEs

Figure 5: **Dynamics of feature probing accuracy and** CSNR **on CIFAR10.** Panels (a) and (b) show the feature probing accuracy and CSNR trends computed using the CIFAR10 test dataset, both exhibiting a unimodal pattern.

Remark. Intuitively, the unimodal curve of CSNR reflects how the additive noise level σ_t in the diffusion process helps counteract the intrinsic data noise δ. The noise ratio (σ_t/δ) can be interpreted as the “denoising" rate, where a larger ratio indicates more data noise being canceled out and vice versa. Meanwhile, $h(\hat{w}_k, \delta)$ represents the class confidence rate, with lower values meaning less class-specific information is captured by the model. With σ_t increases from 0 to ∞, the “denoising rate" rises accordingly, while the class confidence rate decreases monotonically. Thus, from Theorem 1, we can derive the rationale behind the unimodal behavior of CSNR.

- **The unimodal curve of** CSNR**.** The unimodal curve is decided by the interplay between the “denoising rate" and the class confidence rate as noise increases. As observed in Figure 4(c), the “denoising rate" (σ_t^2/δ^2) increases monotonically with σ_t while the class confidence rate $h(\hat{w}_k, \delta)$ monotonically declines. Initially, as σ_t increases, the class confidence rate remains relatively stable due to its flat slope (as seen in Figure 4(c)), and an increasing “denoising rate" enhances the CSNR, resulting in improved posterior estimation. However, as indicated by (7), when σ_t becomes too large, $h(\hat{w}_k, \delta)$ approaches $h(\hat{w}_l, \delta)$, leading to a drop in CSNR, which limits the model’s ability to project $\boldsymbol{x}_0$ onto the correct signal space and ultimately impairs posterior estimation. This interpretation is validated by the visualization in Figure 3. In the plot, each class is represented by a colored straight line, while deviations from these lines correspond to the δ-related noise term. Initially, increasing the noise scale effectively cancels out the δ-related data noise, resulting in a cleaner posterior estimation and improved probing accuracy. However, as the noise continues to increase, the class confidence rate drops, leading to an overlap between classes, which ultimately degrades the feature quality and probing performance.

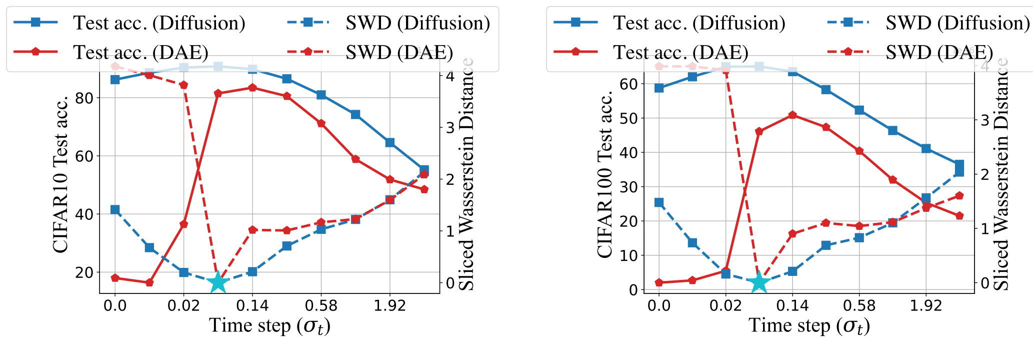
Back to our real-world analogy, the proportion of data associated with δ represents class-irrelevant attributes or finer image details. The unimodal representation learning dynamic thus captures a “fine-to-coarse" shift [Choi et al., 2022, Wang and Vastola, 2023], where these details are progressively stripped away. During this process, peak representation performance is achieved at a balance point where class-irrelevant attributes are eliminated, while class-essential information is preserved.

3.3 Empirical Validation

In this subsection, we conduct experiments on both synthetic and real datasets to validate our theory on the representation learning dynamics.

We use two datasets: a 3-class `MoLRG` dataset, where each subspace has dimension $d = 1$ and ambient dimension $n = 10$, with noise scale $\delta = 0.2$, and the standard CIFAR10 dataset [Krizhevsky et al., 2009]. We consider two training settings: (a) a DDPM-based diffusion training configuration and (b) a vanilla DAE training configuration, where separate DAEs are trained for different noise levels. Here, the separate DAEs serve as a control group, enabling us to isolate the effects of the denoising process, as discussed in Section 2.1. We leave further training details in Appendix A.2.

After training, we extract intermediate features and posterior predictions from both diffusion models and DAEs, followed by linear probing on the features and computation of empirical CSNR for the posterior estimations. The results for the two datasets are presented in Figure 4 and Figure 5, respectively. As shown in the plots, both feature probing accuracy and the empirical CSNR exhibit a



(a) CIFAR10

(b) CIFAR100

Figure 6: **Comparison of representation learning performance and feature similarity between diffusion model and individual DAEs.** We train DDPM-based diffusion models and individual DAEs on the CIFAR10 and CIFAR100 datasets. After training, we plotted their representation learning performance and feature similarity against the best features (indicated by $\star$) as the noise level increases.

matching unimodal curve, consistent across training configurations and datasets, thus supporting our theoretical results.

4 Additional Experiments

In Section 3, we analyzed diffusion representation dynamics with a focus on the denoising process, assuming sufficient training data for learning the underlying distribution. In this section, we explore the impact of the diffusion process (Section 4.1) and data complexity (Section 4.2) in shaping diffusion models' representation learning dynamics.

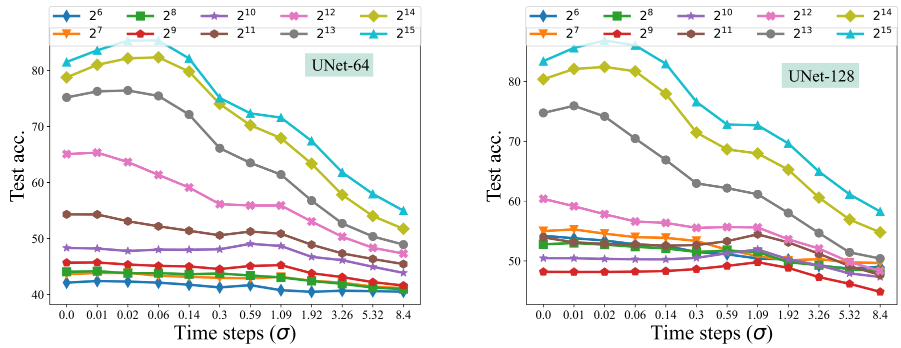
4.1 Weight sharing in diffusion models helps representation learning

In this subsection, we demonstrate how the inherent weight-sharing mechanism in diffusion models, stemming from their loss design, enhances representation learning performances compared with traditional DAEs.

Previously, in Section 3, we analyzed the optimal posterior function by treating each noise level independently. However, the training objective for diffusion models in (3) involves minimizing the loss across all noise levels simultaneously, which results in interactions and parameter sharing among denoising subcomponents at different noise levels. We hypothesize that these interactions and parameter sharing create greater feature similarity across noise scales, effectively functioning as an implicit “ensemble" mechanism that enhances the performance of diffusion models compared to individual DAEs [Chen et al., 2024b], which accounts for the significant performance gap between DAEs and diffusion models, as shown in Figure 4(a) and Figure 5(a).

To test this hypothesis, we trained 10 individual DAEs, each at a different noise level, as well as a single DDPM-based diffusion model on CIFAR10 and CIFAR100 datasets. We then conducted linear probing on the features extracted from both setups. To evaluate feature similarity, we calculated the sliced Wasserstein distance (SWD) [Doan et al., 2024] between features for both diffusion and DAE models at various noise levels and their corresponding features at $\sigma_t = 0.06$, which achieves near-optimal accuracy for all scenarios.

As shown in Figure 6, diffusion models consistently outperform individual DAEs, particularly at lower noise levels, where the performance gap is most pronounced. In these low-noise regions, due to the almost negligible additive noise, individual DAEs are more likely to be trained as identity functions, leading to trivial representations. In contrast, the parameter sharing in diffusion models alleviates this issue significantly. The SWD curve demonstrates an inverse correlation with the test accuracy curve, indicating that features closer to their optimal state possess stronger representational capacity. Furthermore, the plot shows that diffusion model features across different noise levels remain significantly closer to their optimal features at $\sigma_t = 0.06$, while DAE features show less similarity. These results strongly support our hypothesis.



(a) UNet-64 accuracy trend

(b) UNet-128 accuracy trend

Figure 7: **The influence of data complexity in diffusion-based representation learning.** With the same model trained in Figure 2, we plot the representation learning dynamics for each trained model as a function of changing noise levels.

The concept of this “sharing mechanism" is also supprted by previous empirical studies on DAEs, which have shown that sequential training over multiple noise scales enhances representation quality [Chandra and Sharma, 2014, Geras and Sutton, 2014, Zhang and Zhang, 2018]. In this work, we conduct an ablation study to explore methods for improving DAE performance at lower noise levels, finding that training with multiple noise scales provided the most promising results. Further details can be found in Appendix A.3 (Table 1).

4.2 The influence of data complexity in diffusion representation learning

So far, our analyses are based on the assumption that the training dataset contains sufficient samples for the diffusion model to learn the underlying distribution. Interestingly, if this assumption is violated by training the model on insufficient data, the unimodal representation learning dynamic disappears and the probing accuracy also drops severely.

As illustrated in Figure 7, we train 2 different UNets following the EDM [Karras et al., 2022] configuration with training dataset size ranging from 2^5 to 2^{15}. The unimodal curve emerges only when the dataset size exceeds 2^{12}, whereas smaller datasets produce flat curves.

The underlying reason for this observation is that, when training data is limited, diffusion models memorize all individual data points rather than learn the true underlying data structure [Wang et al., 2024]. In this scenario, the model memorizes an empirical distribution that lacks meaningful low-dimensional structures and thus deviates from the setting in our theory, leading to the loss of the unimodal representation dynamic. To confirm this, we calculated the generalization score, which measures the percentage of generated data that does not belong to the training dataset, as defined in [Zhang et al., 2023]. As shown in Figure 2, representation learning only achieves strong accuracy and displays the unimodal dynamic when the generalization score approaches 1, aligning with our theoretical assumptions.

5 Conclusion

In this work, we establish a link between distribution recovery, posterior estimation, and representation learning, providing the first theoretical study of diffusion-based representation learning dynamics across varying noise scales. Using a low-dimensional mixture of low-rank Gaussians, we show that the unimodal representation learning dynamic arises from the interplay between data denoising and class specification. Additionally, our analysis highlights the inherent weight-sharing mechanism in diffusion models, demonstrating its benefits for peak representation performance as well as its limitations in optimizing high-noise regions due to increased complexity. Experiments on both synthetic and real datasets validate our findings.

References

K. Abstreiter, S. Mittal, S. Bauer, B. Schölkopf, and A. Mehrjou. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021.

I. Alkhouri, S. Liang, R. Wang, Q. Qu, and S. Ravishankar. Diffusion-based adversarial purification for robust deep mri reconstruction. *arXiv preprint arXiv:2309.05794*, 2023.

I. Alkhouri, S. Liang, R. Wang, Q. Qu, and S. Ravishankar. Diffusion-based adversarial purification for robust deep mri reconstruction. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12841–12845. IEEE, 2024.

Z. Allen-Zhu and Y. Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 977–988. IEEE, 2022.

B. D. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, Y. Li, T. Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

BohaoZou. Denoisingdiffusionprobabilitymodel-ddpm-. `https://github.com/zoubohao/DenoisingDiffusionProbabilityModel-ddpm-`, 2022.

B. Chandra and R. K. Sharma. Adaptive noise schedule for denoising autoencoder. In *International Conference on Neural Information Processing*, 2014.

S. Chen, H. Zhang, M. Guo, Y. Lu, P. Wang, and Q. Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing. *arXiv preprint arXiv:2409.02374*, 2024a.

X. Chen, Z. Liu, S. Xie, and K. He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024b.

J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

H. Chung, B. Sim, D. Ryu, and J. C. Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

A.-D. Doan, B. L. Nguyen, S. Gupta, I. Reid, M. Wagner, and T.-J. Chin. Assessing domain gap for continual domain adaptation in object detection. *Computer Vision and Image Understanding*, 238: 103885, 2024.

M. Fuest, P. Ma, M. Gui, J. S. Fischer, V. T. Hu, and B. Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024.

K. J. Geras and C. Sutton. Scheduled denoising autoencoders. *arXiv preprint arXiv:1406.3269*, 2014.

S. Gong, V. N. Boddeti, and A. K. Jain. On the intrinsic dimensionality of image representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3987–3996, 2019.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

D. A. Hudson, D. Zoran, M. Malinowski, A. K. Lampinen, A. Jaegle, J. L. McClelland, L. Matthey, F. Hill, and A. Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23115–23127, 2024.

A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Z. Kadkhodaie, F. Guth, E. P. Simoncelli, and S. Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

D. P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

J. Kong, J. Kim, and J. Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. DIFFWAVE: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

D. Kunin, J. Bloom, A. Goeva, and C. Seed. Loss landscapes of regularized linear autoencoders. In *International conference on machine learning*, pages 3560–3569. PMLR, 2019.

M. Kwon, J. Jeong, and Y. Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

M. Lewis. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

D. Li, H. Ling, A. Kar, D. Acuna, S. W. Kim, K. Kreis, A. Torralba, and S. Fidler. Dreamteacher: Pretraining image backbones with deep generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16698–16708, 2023.

X. Li, S. M. Kwon, I. R. Alkhouri, S. Ravishanka, and Q. Qu. Decoupled data consistency with diffusion purification for image restoration. *arXiv preprint arXiv:2403.06054*, 2024.

S. Mukhopadhyay, M. Gwilliam, V. Agarwal, N. Padmanabhan, A. Swaminathan, S. Hegde, T. Zhou, and A. Shrivastava. Diffusion models beat gans on image classification. *arXiv preprint arXiv:2307.08702*, 2023.

M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.

Y.-H. Park, M. Kwon, J. Jo, and Y. Uh. Unsupervised discovery of semantic latent directions in diffusion models. *arXiv preprint arXiv:2302.12469*, 2023.

W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

P. Pope, C. Zhu, A. Abdelkader, M. Goldblum, and T. Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10629, 2022.

A. Pretorius, S. Kroon, and H. Kamper. Learning dynamics of linear denoising autoencoders. In *International Conference on Machine Learning*, pages 4141–4150. PMLR, 2018.

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

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

Y. Shi, C. Xue, J. H. Liew, J. Pan, H. Yan, W. Zhang, V. Y. Tan, and S. Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

B. Song, S. M. Kwon, Z. Zhang, X. Hu, Q. Qu, and L. Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

J. Stanczuk, G. Batzolis, T. Deveney, and C.-B. Schönlieb. Your diffusion model secretly knows the dimension of the data manifold. *arXiv preprint arXiv:2212.12611*, 2022.

H. Steck. Autoencoders that don't overfit towards the identity. In *Neural Information Processing Systems*, 2020.

tanelp. tiny-diffusion. `https://github.com/tanelp/tiny-diffusion`, 2022.

L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, 2008.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, 2010. URL `https://api.semanticscholar.org/CorpusID:17804904`.

B. Wang and J. J. Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later. *arXiv preprint arXiv:2303.02490*, 2023.

P. Wang, H. Zhang, Z. Zhang, S. Chen, Y. Ma, and Q. Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

Y. Wang, Y. Schiff, A. Gokaslan, W. Pan, F. Wang, C. De Sa, and V. Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. In *International Conference on Machine Learning*, pages 36336–36354. PMLR, 2023.

W. Xiang, H. Yang, D. Huang, and Y. Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15802–15812, 2023.

X. Yang and X. Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18938–18949, 2023.

H. Zhang, J. Zhou, Y. Lu, M. Guo, P. Wang, L. Shen, and Q. Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2023.

H. Zhang, Y. Lu, I. Alkhouri, S. Ravishankar, D. Song, and Q. Qu. Improving training efficiency of diffusion models via multi-stage framework and tailored multi-decoder architectures. In *Conference on Computer Vision and Pattern Recognition 2024*, 2024. URL `https://openreview.net/forum?id=YtptmpZQOg`.

Q. Zhang and L. Zhang. Convolutional adaptive denoising autoencoders for hierarchical feature extraction. *Frontiers of Computer Science*, 12:1140 – 1148, 2018.

A Appendix / supplemental material

The Appendis is organized as follows: in Appendix A.1, we discuss related works; in Appendix A.2, we present the detailed experimental setups for the empirical results in the paper; in Appendix A.3, we provide complementary experiments. Lastly, in Appendix A.4, we provide proof details for Section 3.

A.1 Related Works

Denoising auto-encoders. Denoising autoencoders (DAEs) are trained to reconstruct corrupted images to extract semantically meaningful information, which can be applied to various vision [Vincent et al., 2008, 2010] and language downstream tasks [Lewis, 2019]. Related to our analysis of the weight-sharing mechanism, several studies have shown that training with a noise scheduler can enhance downstream performance [Chandra and Sharma, 2014, Geras and Sutton, 2014, Zhang and Zhang, 2018]. On the theoretical side, prior works have studied the learning dynamics [Pretorius et al., 2018, Steck, 2020] and optimization landscape [Kunin et al., 2019] through the simplified linear DAE models.

Diffusion-based representation learning. Diffusion-based representation learning Fuest et al. [2024] has demonstrated significant success in various downstream tasks, including image classification [Xiang et al., 2023, Mukhopadhyay et al., 2023], segmentation [Baranchuk et al., 2021], correspondence [Tang et al., 2023], and image editing [Shi et al., 2024]. To further enhance the utility of diffusion features, knowledge distillation methods [Yang and Wang, 2023, Li et al., 2023] have been proposed, aiming to bypass the computationally expensive grid search for the optimal t in feature extraction and improving downstream performance. Beyond directly using intermediate features from pre-trained diffusion models, research efforts has also explored novel loss functions [Abstreiter et al., 2021, Wang et al., 2023] and network modifications [Hudson et al., 2024, Preechakul et al., 2022] to develop more unified generative and representation learning capabilities within diffusion models. Unlike the aforementioned efforts, our work focuses more on understanding the representation learning capabilities of diffusion models.

A.2 Experimental Details

In this section, we provide technical details for all the experiments in the main body of the paper.

Experimental details for Figure 1 (a)-(b). We utilize a minimal implementation of the original DDPM model from an online public repository [BohaoZou, 2022], consisting of a 12-layer UNet (including input/output embedding layers), and train it on the CIFAR10 dataset with $T = 1000$ time steps for 200 epochs with an AdamW optimizer and learning rate 1×10^{-4}. Features are extracted as 512-dimensional vectors from the output of the 7th layer (i.e., the bottleneck layer) at time steps [1, 5, 10, 20, 30, 40, 60, 80, 100, 200, 400, 500, 600], each corresponding to a specific σ_t ranging from 0.01 to 6.17. Linear probing is applied to the extracted features, as in [Xiang et al., 2023], to plot the feature probing accuracy curve in Figure 1(a). For the posterior estimation ($\boldsymbol{x}_\theta(\boldsymbol{x}_0, t)$) probing accuracy curve, also shown in Figure 1(a), we use a two-layer MLP probe with ReLU activation. The estimated posterior at these time steps is visualized in Figure 1(b).

Experimental details for Figure 1 (c)-(d). We train diffusion models based on the unified framework proposed by Karras et al. [2022]. Specifically, we use the DDPM+ network, and use EDM configuration for Figure 1 (c) while taking VP configuration Figure 1 (d). Karras et al. [2022] has shown equivalence between VP configuration and the traditional DDPM setting, thus we call the models in Figure 1 (d) as DDPM* models. For each of EDM and VP configuration, we train two models on CIFAR10 and CIFAR100, respectively. After training, we conduct linear probe on CIFAR10 and CIFAR100. At a specific noise level $\sigma(t)$, we either use clean image $\boldsymbol{x}_0$ or noisy image $\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{n}$ as input to the EDM or the DDPM* models for extracting features after the '8x8_block3' layer. Here, $\boldsymbol{n}$ represents random noise and $\boldsymbol{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma(t)^2\mathbf{I}\right)$. We train a logistic regression on features in the train split and report the classification accuracy on the test split of the dataset. We perform the linear probe for each of the following noise levels: [0.002, 0.008, 0.023, 0.060, 0.140, 0.296, 0.585, 1.088, 1.923, 3.257].

Experimental details for Figure 3 and Figure 4. For the MoLRG experiments, we train a 3-layer MLP with ReLU activation and a hidden dimension of 128, following the setup provided in an open-source repository [tanelp, 2022]. The MLP is trained for 200 epochs using DDPM scheduling with $T = 500$, employing the Adam optimizer with a learning rate of 1×10^{-3}. For feature extraction, we use the activations of the second layer of the MLP (dimension 128) as intermediate features for linear probing. For CSNR computation, we follow the definition in Equation (8) since we have access to the ground-truth basis for the MoLRG data. In Figure 3, we visualize the posterior estimations at time steps [1, 20, 80, 200, 260] by projecting them onto the union of $\boldsymbol{U}_1, \boldsymbol{U}_2$, and $\boldsymbol{U}_3$ (a 3D space), then further projecting onto the 2D plane along the (1, 1, 1) direction. The subtitles of each visualization show the corresponding probing accuracy and CSNR calculated as explained above. For Figure 4(a)(b), we plot the accuracy and CSNR at time steps [1, 5, 10, 20, 40, 60, 80, 100, 120, 140, 160, 180, 220, 240, 260]. We perform linear probing using the features extracted from the training set and test on five different MoLRG datasets generated with five different random seeds, reporting the average accuracy.

Experimental details for Figure 5. We use the same experimental settings as in Figure 1(a)(b). Additionally, we train individual DAEs for each different time step. The accuracy curves in Figure 5(a) are plotted identically as in Figure 1(a). The CSNR metric in Figure 5(b) is calculated from the definition Equation (8), with the basis $\boldsymbol{U}_k$ for each CIFAR10 class estimated as the first five right singular vectors of the data from the k-th class.

Experimental details for Figure 6. We train individual DAEs using the DDPM++ network and VP configuration outlined in Karras et al. [2022] at the following noise scales: $[0.002, 0.008, 0.023, 0.06, 0.14, 0.296, 0.585, 1.088, 1.923, 3.257]$. Each model is trained for 500 epochs using the Adam optimizer [Kingma, 2014] with a fixed learning rate of 1×10^{-4}. For the diffusion models, we reuse the model from Figure 1(d). The sliced Wasserstein distance is computed according to the implementation described in Doan et al. [2024].

Experimental details for Figure 7. We use the DDPM++ network and VP configuration to train diffusion models[Karras et al., 2022] on the CIFAR10 dataset, using two network configurations: UNet-64 and UNet-128, by varying the embedding dimension of the UNet. Training dataset sizes range exponentially from 2^6 to 2^{15}. For each dataset size, both UNet-64 and UNet-128 are trained on the same subset of the training data. All models are trained with a duration of 50K images following the EDM training setup. After training, we calculate the generalization score as described in Zhang et al. [2023], using 10K generated images and the full training subset to compute the score.

A.3 Additional Experiments

Additional representation learning experiments on DDPM. Apart from EDM and DDPM* models pre-trained using the framework proposed by Karras et al. [2022], we also experiment with the features extracted by classic DDPM models [Ho et al., 2020] to make sure the observations do not depend on the specific training framework. We use the same groups of noise levels and also test using clean or noisy images as input to extract features at the bottleneck layer, and then conduct the linear probe. The DDPM models we use are trained on the Flowers-102 [Nilsback and Zisserman, 2008] and the CIFAR10 dataset accordingly. Different from the framework proposed by Karras et al. [2022], the input to the classic DDPM model is the same as the input to the UNet inside. Therefore, we calculate the scaling factor $\sqrt{\bar{\alpha}_t} = 1/\sqrt{\sigma^2(t)+1}$, and use $\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$ as the clean image input. Besides, for noisy input, we set $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0 + \boldsymbol{n})$, with $\boldsymbol{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma(t)^2\mathbf{I}\right)$. The linear probe results are presented in Figure 8, where we consistently see an unimodal curve, as well as compatible or even superior representation learning performance of clean input $\boldsymbol{x}_0$.

Validation of $f^\star$ approximation in Section 3. In Section 3, we approximate the optimal posterior estimation function $\boldsymbol{x}_{\boldsymbol{\theta}}$ using $f^\star$ by taking the expectation inside the softmax with respect to $\boldsymbol{x}_0$. To validate this approximation, we compare the CSNR calculated from $\boldsymbol{x}_{\boldsymbol{\theta}}$ and from $f^\star$ using (8) and (9), respectively. We use a fixed dataset size of 2400 and set the default parameters to $n = 50$, $d = 5$, $K = 3$, and $\delta = 0.1$ to generate MoLRG data. We then vary one parameter at a time while keeping the others constant, and present the computed CSNR in Figure 9. As shown, the approximated CSNR score consistently aligns with the actual score.

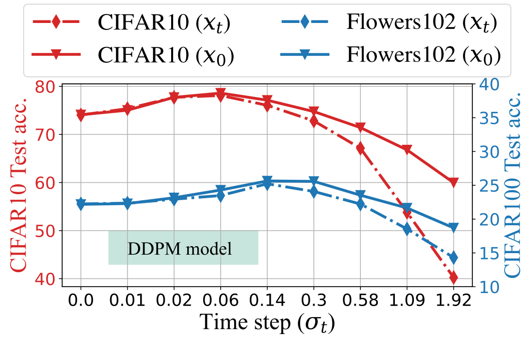


Figure 8: **Performance comparison: clean vs. noisy inputs.** We use pre-trained DDPM model on the Flowers-102 [Nilsback and Zisserman, 2008] and CIFAR10 dataset. The feature probing accuracy is plotted to compare the performance when using clean versus noisy inputs.

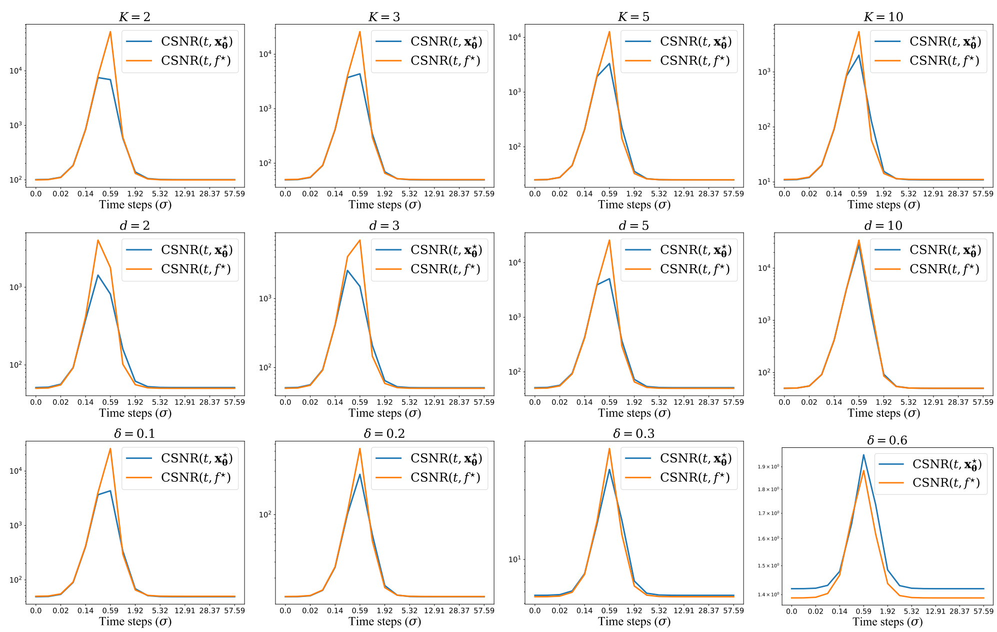


Figure 9: **Comparison between** CSNR **calculated using the optimal model** $\boldsymbol{x}_{\boldsymbol{\theta}}^\star$ **and the** CSNR **calculated with our approximation in Theorem 1.** We generate `MoLRG` data and calculate CSNR using both the corresponding optimal posterior function $\boldsymbol{x}_{\boldsymbol{\theta}}^\star$ and our approximation $f^\star$ from Theorem 1. Default parameters are set as $n = 50$, $d = 5$, $K = 3$, and $\delta = 0.1$. In each row, we vary one parameter while keeping the others fixed, comparing the actual and approximated CSNR.

Mitigating the performance gap between DAE and diffusion models. Throughout the empirical results presented in this paper, we consistently observe a performance gap between individual DAEs and diffusion models, especially in low-noise regions. Here, we use a DAE trained on the CIFAR-10 dataset with a single noise level $\sigma = 0.002$, using the NCSN++ architecture [Karras et al., 2022]. In the default setting, the DAE achieves a test accuracy of 32.3. We then explore three methods to improve the test performance: (a) adding dropout, as noise regularization and dropout have been effective in preventing autoencoders from learning identity functions [Steck, 2020]; (b) adopting EDM-based preconditioning during training, including input/output scaling, loss weighting, etc.; and (c) multi-level noise training, in which the DAE is trained simultaneously on three noise levels $[0.002, 0.012, 0.102]$. Each modification is applied independently, and the results are reported in Table 1. As shown, dropout helps improve performance, but even with a dropout rate of 0.95, the improvement is minor. EDM-based preconditioning achieves moderate improvement, while multi-level noise training yields the most promising results, demonstrating the benefit of incorporating the diffusion process in DAE training.

Table 1: **Improve DAE representation performance at low noise region.** A vanilla DAE trained on the CIFAR-10 dataset with a single noise level of $\sigma = 0.002$ serves as the baseline. We evaluate the performance improvement of dropout regularization, EDM-based preconditioning, and multi-level noise training ($\sigma = \{0.002, 0.012, 0.102\}$). Each technique is applied independently to assess its contribution to performance enhancement.

Modifications	Test acc.
Vanilla DAE	32.3
+Dropout (0.5)	35.3
+Dropout (0.9)	36.4
+Dropout (0.95)	38.1
+EDM preconditioning	49.2
+Multi-level noise training	**58.6**

A.4 Proofs

A.4.1 Proof of Proposition 1

Proof. We follow the same proof steps as in [Wang et al., 2024] Lemma 1 with a change of variable. Let $\boldsymbol{c}_k = \begin{bmatrix}\boldsymbol{a}_k \\ \boldsymbol{e}_k\end{bmatrix}$ and $\widetilde{\boldsymbol{U}_k} = \begin{bmatrix}\boldsymbol{U}_k & \delta \boldsymbol{U}_k^{\perp}\end{bmatrix}$, we first compute

$$
\begin{aligned}
& p_t(\boldsymbol{x}|Y=k) \\
&= \int p_t\left(\boldsymbol{x}|Y=k, \boldsymbol{c}_k\right)\mathcal{N}(\boldsymbol{c}_k; \boldsymbol{0}, \boldsymbol{I}_{d+D})\,\mathrm{d}\boldsymbol{c}_k \\
&= \int p_t(\boldsymbol{x}|\boldsymbol{x}_0 = \widetilde{\boldsymbol{U}_k}\boldsymbol{c}_k)\mathcal{N}\left(\boldsymbol{c}_k; \boldsymbol{0}, \boldsymbol{I}_{d+D}\right)\mathrm{d}\boldsymbol{c}_k \\
&= \int \mathcal{N}(\boldsymbol{x}; s_t\widetilde{\boldsymbol{U}_k}\boldsymbol{c}_k, \gamma_t^2\boldsymbol{I}_n)\mathcal{N}\left(\boldsymbol{c}_k; \boldsymbol{0}, \boldsymbol{I}_{d+D}\right)\mathrm{d}\boldsymbol{c}_k \\
&= \frac{1}{(2\pi)^{n/2}(2\pi)^{(d+D)/2}\gamma_t^n}\int \exp\left(-\frac{1}{2\gamma_t^2}\|\boldsymbol{x} - s_t\widetilde{\boldsymbol{U}_k}\boldsymbol{c}_k\|^2\right)\exp\left(-\frac{1}{2}\|\boldsymbol{c}_k\|^2\right)\mathrm{d}\boldsymbol{c}_k \\
&= \frac{1}{(2\pi)^{n/2}(2\pi)^{(d+D)/2}\gamma_t^n}\int \exp\left(-\frac{1}{2\gamma_t^2}\left(\boldsymbol{x}^T\boldsymbol{x} - 2s_t\boldsymbol{x}^T\widetilde{\boldsymbol{U}_k}\boldsymbol{c}_k + s_t^2\boldsymbol{c}_k^T\widetilde{\boldsymbol{U}_k}^T\widetilde{\boldsymbol{U}_k}\boldsymbol{c}_k + \gamma_t^2\boldsymbol{c}_k^T\boldsymbol{c}_k\right)\right)\mathrm{d}\boldsymbol{c}_k \\
&= \frac{1}{(2\pi)^{n/2}\gamma_t^n}\left(\frac{s_t^2+\gamma_t^2}{\gamma_t^2}\right)^{-d/2}\left(\frac{s_t^2\delta^2+\gamma_t^2}{\gamma_t^2}\right)^{-D/2}\exp\left(-\frac{1}{2\gamma_t^2}\boldsymbol{x}^T\left(\boldsymbol{I}_n - \frac{s_t^2}{s_t^2+\gamma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T - \frac{s_t^2\delta^2}{s_t^2\delta^2+\gamma_t^2}\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}\right) \\
&\int \frac{1}{(2\pi)^{d/2}}\left(\frac{\gamma_t^2}{s_t^2+\gamma_t^2}\right)^{-d/2}\exp\left(-\frac{s_t^2+\gamma_t^2}{2\gamma_t^2}\left\|\boldsymbol{a}_k - \frac{s_t}{s_t^2+\gamma_t^2}\boldsymbol{U}_k^T\boldsymbol{x}\right\|^2\right)\mathrm{d}\boldsymbol{a}_k \\
&\int \frac{1}{(2\pi)^{D/2}}\left(\frac{\gamma_t^2}{s_t^2\delta^2+\gamma_t^2}\right)^{-D/2}\exp\left(-\frac{s_t^2\delta^2+\gamma_t^2}{2\gamma_t^2}\left\|\boldsymbol{e}_k - \frac{s_t\delta}{s_t^2\delta^2+\gamma_t^2}\boldsymbol{U}_k^{\perp T}\boldsymbol{x}\right\|^2\right)\mathrm{d}\boldsymbol{e}_k \\
&= \frac{1}{(2\pi)^{n/2}}\frac{1}{(s_t^2+\gamma_t^2)^{d/2}(s_t^2\delta^2+\gamma_t^2)^{D/2}}\exp\left(-\frac{1}{2\gamma_t^2}\boldsymbol{x}^T\left(\boldsymbol{I}_n - \frac{s_t^2}{s_t^2+\gamma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T - \frac{s_t^2\delta^2}{s_t^2\delta^2+\gamma_t^2}\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}\right) \\
&= \frac{1}{(2\pi)^{n/2}\det^{1/2}(s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T + s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T} + \gamma_t^2\boldsymbol{I}_n)} \\
&\quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T\left(s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T + s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T} + \gamma_t^2\boldsymbol{I}_n\right)^{-1}\boldsymbol{x}\right) \\
&= \mathcal{N}(\boldsymbol{x}; \boldsymbol{0}, s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T + s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T} + \gamma_t^2\boldsymbol{I}_n),
\end{aligned}
$$

where we repeatedly apply the pdf of multi-variate Gaussian and the second last equality uses $\det(s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T + s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T} + \gamma_t^2\boldsymbol{I}_n) = (s_t^2+\gamma_t^2)^d(s_t^2\delta^2+\gamma_t^2)^D$ and $(s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T + s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T} + \gamma_t^2\boldsymbol{I}_n)^{-1} = \left(\boldsymbol{I}_n - s_t^2/(s_t^2+\gamma_t^2)\boldsymbol{U}_k\boldsymbol{U}_k^T - s_t^2\delta^2/(s_t^2\delta^2+\gamma_t^2)\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\right)/\gamma_t^2$ because of the Woodbury matrix inversion lemma. Hence, with $\mathbb{P}\left(Y=k\right)=\pi_k$ for each $k\in[K]$, we have

$$p_t(\boldsymbol{x})=\sum_{k=1}^{K}p_t(\boldsymbol{x}|Y=k)\mathbb{P}(Y=k)=\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n).$$

Now we can compute the score function

$$\begin{aligned}\nabla\log p_t(\boldsymbol{x})=\frac{\nabla p_t(\boldsymbol{x})}{p_t(\boldsymbol{x})}&=\frac{\begin{array}{r}\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)\\ \left(-\frac{1}{\gamma_t^2}\boldsymbol{x}+\frac{s_t^2}{\gamma_t^2(s_t^2+\gamma_t^2)}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x}+\frac{s_t^2\delta^2}{\gamma_t^2(s_t^2\delta^2+\gamma_t^2)}\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\boldsymbol{x}\right)\end{array}}{\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)}\\&=-\frac{1}{\gamma_t^2}\left(\boldsymbol{x}-\frac{\begin{array}{r}\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)\\ \left(\frac{s_t^2}{s_t^2+\gamma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x})+\frac{s_t^2\delta^2}{s_t^2\delta^2+\gamma_t^2}\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\boldsymbol{x})\right)\end{array}}{\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)}\right).\end{aligned}$$

According to Tweedie's formula, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}_t\right]&=\frac{\boldsymbol{x}_t+\gamma_t^2\nabla\log p_t(\boldsymbol{x}_t)}{s_t}\\&=\frac{s_t}{s_t^2+\gamma_t^2}\frac{\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x}}{\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)}\\&\quad+\frac{s_t\delta^2}{s_t^2\delta^2+\gamma_t^2}\frac{\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\boldsymbol{x}}{\mathcal{N}(\boldsymbol{x};\boldsymbol{0},s_t^2\boldsymbol{U}_k\boldsymbol{U}_k^T+s_t^2\delta^2\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}+\gamma_t^2\boldsymbol{I}_n)}\\&=\frac{s_t}{s_t^2+\gamma_t^2}\frac{\sum_{k=1}^{K}\pi_k\exp\left(\phi_t\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2\right)\exp\left(\psi_t\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x}_t}{\sum_{k=1}^{K}\pi_k\exp\left(\phi_t\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2\right)\exp\left(\psi_t\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)}\\&\quad+\frac{s_t\delta^2}{s_t^2\delta^2+\gamma_t^2}\frac{\sum_{k=1}^{K}\pi_k\exp\left(\phi_t\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2\right)\exp\left(\psi_t\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t}{\sum_{k=1}^{K}\pi_k\exp\left(\phi_t\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2\right)\exp\left(\psi_t\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)},\end{aligned}$$

with $\phi_t=s_t^2/(2\gamma_t^2(s_t^2+\gamma_t^2))$ and $\psi_t=s_t^2\delta^2/(2\gamma_t^2(s_t^2\delta^2+\gamma_t^2))$. The final equality uses the pdf of multi-variant Gaussian and the matrix inversion lemma discussed earlier.

Now since π_k is consistent for all k and $s_t=1$, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}_t\right]&=\sum_{k=1}^{K}w_k^{\star}(\boldsymbol{x}_t)\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k^{\perp}\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}_t\\ \text{where }w_k^{\star}(\boldsymbol{x}_t)&:=\frac{\exp\left(\frac{1}{2\sigma_t^2(1+\sigma_t^2)}\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2+\frac{\delta^2}{2\sigma_t^2(\delta^2+\sigma_t^2)}\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)}{\sum_{k=1}^{K}\exp\left(\frac{1}{2\sigma_t^2(1+\sigma_t^2)}\|\boldsymbol{U}_k^T\boldsymbol{x}_t\|^2+\frac{\delta^2}{2\sigma_t^2(\delta^2+\sigma_t^2)}\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_t\|^2\right)}.\end{aligned}$$

$\square$

A.4.2 Proof of Theorem 1

Proof. Following Equation (8) and Lemma 1, we can write

$$
\begin{aligned}
\mathrm{CSNR}(t, f^\star) &= \frac{\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t)\|^2]}{\mathbb{E}_{\boldsymbol{x}_0}[\sum_{l\neq k}\|\boldsymbol{U}_l\boldsymbol{U}_l^T \boldsymbol{f}^\star(\boldsymbol{x}_0,t)\|^2]} = \frac{\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t)\|^2]}{\sum_{l\neq k}\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_l\boldsymbol{U}_l^T \boldsymbol{f}^\star(\boldsymbol{x}_0,t)\|^2]} \\
&= \frac{\left(\frac{\hat{w}_k}{1+\sigma_t^2} + \frac{(K-1)\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)^2 d}{(K-1)\left(\frac{\hat{w}_l}{1+\sigma_t^2} + \frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_l)}{\delta^2+\sigma_t^2}\right)^2\delta^2 d} \\
&= \frac{1}{(K-1)\delta^2}\cdot\left(\frac{\hat{w}_k\delta^2+\hat{w}_k\sigma_t^2+(K-1)\delta^2\hat{w}_l+(K-1)\delta^2\hat{w}_l\sigma_t^2}{\hat{w}_l\delta^2+\hat{w}_l\sigma_t^2+\delta^2\hat{w}_k+(K-2)\delta^2\hat{w}_l+\delta^2\hat{w}_k\sigma_t^2+(K-2)\delta^2\hat{w}_l\sigma_t^2}\right)^2 \\
&= \frac{1}{(K-1)\delta^2}\cdot\left(\frac{\delta^2+\sigma_t^2\left(\hat{w}_k+(K-1)\delta^2\hat{w}_l\right)}{\delta^2+\sigma_t^2\left(\hat{w}_l+\delta^2\hat{w}_k+(K-2)\delta^2\hat{w}_l\right)}\right)^2 \\
&= \frac{1}{(K-1)\delta^2}\cdot\left(\frac{1+\frac{\sigma_t^2}{\delta^2}\left((1-\delta^2)\hat{w}_k+\delta^2(\hat{w}_k+(K-1)\hat{w}_l)\right)}{1+\frac{\sigma_t^2}{\delta^2}\left((1-\delta^2)\hat{w}_l+\delta^2(\hat{w}_l+\hat{w}_k+(K-2)\hat{w}_l)\right)}\right)^2 \\
&= \frac{1}{(K-1)\delta^2}\cdot\left(\frac{1+\frac{\sigma_t^2}{\delta^2}\left((1-\delta^2)\hat{w}_k+\delta^2\right)}{1+\frac{\sigma_t^2}{\delta^2}\left((1-\delta^2)\hat{w}_l+\delta^2\right)}\right)^2 \\
&= \frac{1}{(K-1)\delta^2}\cdot\left(\frac{1+\frac{\sigma_t^2}{\delta^2}h(\hat{w}_k,\delta)}{1+\frac{\sigma_t^2}{\delta^2}h(\hat{w}_l,\delta)}\right)^2
\end{aligned}
$$

where $h(w,\delta) := (1-\delta^2)w+\delta^2$. □

Lemma 1. *With the set up of a K-class `MoLRG` data distribution as defined in (4), consider the following the function:*

$$
f^\star(\boldsymbol{x},t) = \sum_{k=1}^{K}\hat{w}_k(\boldsymbol{x})\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T + \frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k^\perp\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}, \tag{11}
$$

$$
\textit{where } \hat{w}_k(\boldsymbol{x}) := \frac{\exp\left(\mathbb{E}_{\boldsymbol{x}}[g_k(\boldsymbol{x},t)]\right)}{\sum_{k=1}^{K}\exp\left(\mathbb{E}_{\boldsymbol{x}}[g_k(\boldsymbol{x},t)]\right)}, \tag{12}
$$

$$
\textit{and } g_k(\boldsymbol{x}) = \frac{1}{2\sigma_t^2(1+\sigma_t^2)}\|\boldsymbol{U}_k^T\boldsymbol{x}\|^2 + \frac{\delta^2}{2\sigma_t^2(\delta^2+\sigma_t^2)}\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}\|^2. \tag{13}
$$

I.e., we consider a simplified version of the expected posterior mean as in (5) *by taking expectation of $g_k(\boldsymbol{x})$ prior to the softmax operation. Under this setting, for any clean $\boldsymbol{x}_0$ from class k (i.e., $\boldsymbol{x}_0 = \boldsymbol{U}_k\boldsymbol{a}_i + b\boldsymbol{U}_k^\perp\boldsymbol{e}_i$), we have:*

$$
\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t)\|^2] = \left(\frac{\hat{w}_k}{1+\sigma_t^2} + \frac{(K-1)\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)^2 d \tag{14}
$$

$$
\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_l\boldsymbol{U}_l^T f^\star(\boldsymbol{x}_0,t)\|^2] = \left(\frac{\hat{w}_l}{1+\sigma_t^2} + \frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_l)}{\delta^2+\sigma_t^2}\right)^2 \delta^2 d \tag{15}
$$

$$
\mathbb{E}_{\boldsymbol{x}_0}[\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T f^\star(\boldsymbol{x}_0,t)\|^2] = \frac{\delta^6(n-kd)}{(\delta^2+\sigma_t^2)^2} \tag{16}
$$

$$
\begin{aligned}
\mathbb{E}[\|f^\star(\boldsymbol{x}_0,t)\|^2] &= \underbrace{\left(\frac{\hat{w}_k}{1+\sigma_t^2} + \frac{(K-1)\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)^2 d}_{\mathbb{E}\,\|\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t)\|^2]} \\
&\quad + \underbrace{(K-1)\left(\frac{\hat{w}_l}{1+\sigma_t^2} + \frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_l)}{\delta^2+\sigma_t^2}\right)^2 \delta^2 d}_{\mathbb{E}[\sum_{l\neq k}^K \boldsymbol{U}_l\boldsymbol{U}_l^T f^\star(\boldsymbol{x}_0,t)\|^2]} + \underbrace{\frac{\delta^6(n-Kd)}{(\delta^2+\sigma_t^2)^2}}_{\mathbb{E}[\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T f^\star(\boldsymbol{x}_0,t)\|^2]}
\end{aligned} \tag{17}
$$

and

$$
\begin{aligned}
\hat{w}_k := \hat{w}_k(\boldsymbol{x}_0) &= \frac{\exp\left(\frac{d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^4 D}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right)}{\exp\left(\frac{d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^4 D}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right) + (K-1)\exp\left(\frac{\delta^2 d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^2 d+\delta^4(D-d)}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right)}, \\
\hat{w}_l := \hat{w}_l(\boldsymbol{x}_0) &= \frac{\exp\left(\frac{\delta^2 d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^2 d+\delta^4(D-d)}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right)}{\exp\left(\frac{d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^4 D}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right) + (K-1)\exp\left(\frac{\delta^2 d}{2\sigma_t^2(1+\sigma_t^2)} + \frac{\delta^2 d+\delta^4(D-d)}{2\sigma_t^2(\delta^2+\sigma_t^2)}\right)}
\end{aligned} \tag{18}
$$

for all class index $l \neq k$.

Proof. Throughout the proof, we use the following notation for slices of vectors.

$\boldsymbol{e}_i[a:b]$ Slices of vector $\boldsymbol{e}_i$ from ath entry to bth entry.

We begin with the softmax terms. Since each class has its unique disjoint subspace, it suffices to consider $g_k(\boldsymbol{x}_0,t)$ and $g_l(\boldsymbol{x}_0,t)$ for any $l \neq k$. Let $a_t = \frac{1}{2\sigma_t^2(1+\sigma_t^2)}$ and $c_t = \frac{\delta^2}{2\sigma_t^2(\delta^2+\sigma_t^2)}$, we have:

$$
\begin{aligned}
\mathbb{E}[g_k(\boldsymbol{x}_0,t)] &= \mathbb{E}[a_t\|\boldsymbol{U}_k^T\boldsymbol{x}_0\|^2 + c_t\|\boldsymbol{U}_k^{\perp T}\boldsymbol{x}_0\|^2] \\
&= \mathbb{E}[a_t\|\boldsymbol{U}_k^T(\boldsymbol{U}_k\boldsymbol{a}_i + b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\|^2] + \mathbb{E}[c_t\|\boldsymbol{U}_k^{\perp T}(\boldsymbol{U}_k\boldsymbol{a}_i + b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\|^2] \\
&= \mathbb{E}[a_t\|\boldsymbol{a}_i\|^2] + \mathbb{E}[c_t\|b\boldsymbol{e}_i\|^2] \\
&= a_t d + c_t\delta^2 D
\end{aligned}
$$

where the last equality follows from $\boldsymbol{a}_i \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)$ and $\boldsymbol{e}_i \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{0},\boldsymbol{I}_D)$.

Without loss of generality, assume the $j = k+1$, we have:

$$
\begin{aligned}
\mathbb{E}[g_l(\boldsymbol{x}_0,t)] &= \mathbb{E}[a_t\|\boldsymbol{U}_l^T\boldsymbol{x}_0\|^2 + c_t\|\boldsymbol{U}_l^{\perp T}\boldsymbol{x}_0\|^2] \\
&= \mathbb{E}[a_t\|\boldsymbol{U}_l^T(\boldsymbol{U}_k\boldsymbol{a}_i + b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\|^2] + \mathbb{E}[c_t\|\boldsymbol{U}_l^{\perp T}(\boldsymbol{U}_k\boldsymbol{a}_i + b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\|^2] \\
&= \mathbb{E}[a_t\|b\boldsymbol{e}_i[1:d]\|^2] + \mathbb{E}\left[c_t\left\|\begin{bmatrix}\boldsymbol{a}_i \\ \boldsymbol{0}\in\mathbb{R}^{D-d}\end{bmatrix} + b\begin{bmatrix}\boldsymbol{0}\in\mathbb{R}^d \\ \boldsymbol{e}_i[d:D]]\end{bmatrix}\right\|^2\right] \\
&= a_t\delta^2 d + c_t(d + \delta^2(D-d))
\end{aligned}
$$

Plug a_t and b_t back with the exponentials, we get $\hat{w}_k$ and $\hat{w}_l$.

Now we prove (14):

$$\begin{aligned}
\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t) &= \hat{w}_k\boldsymbol{U}_k\boldsymbol{U}_k^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k^\perp\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}_0\\
&\quad+\sum_{l\neq k}\hat{w}_l\boldsymbol{U}_k\boldsymbol{U}_k^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_l^\perp\boldsymbol{U}_l^{\perp T}\right)\boldsymbol{x}_0\\
&=\hat{w}_k\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x}_0\right)+\sum_{l\neq k}\hat{w}_l\left(\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{x}_0\right)\\
&=\left(\frac{\hat{w}_k}{1+\sigma_t^2}+\frac{(K-1)\,\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)\boldsymbol{U}_k\boldsymbol{U}_k^T(\boldsymbol{U}_k\boldsymbol{a}_i+b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\\
&=\left(\frac{\hat{w}_k}{1+\sigma_t^2}+\frac{(K-1)\,\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)\boldsymbol{U}_k\boldsymbol{a}_i
\end{aligned}$$

Since $\boldsymbol{U}_k\in\mathcal{O}^{n\times d}$:

$$\mathbb{E}[\|\boldsymbol{U}_k\boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0,t)\|^2]=\left(\frac{\hat{w}_k}{1+\sigma_t^2}+\frac{(K-1)\,\delta^2\hat{w}_l}{\delta^2+\sigma_t^2}\right)^2 d$$

and similarly for (15):

$$\begin{aligned}
\boldsymbol{U}_l\boldsymbol{U}_l^T f^\star(\boldsymbol{x}_0,t) &= \hat{w}_k\boldsymbol{U}_l\boldsymbol{U}_l^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k^\perp\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}_0\\
&\quad+\hat{w}_l\boldsymbol{U}_l\boldsymbol{U}_l^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_l^\perp\boldsymbol{U}_l^{\perp T}\right)\boldsymbol{x}_0\\
&\quad+\sum_{j\neq k,l}\hat{w}_j\boldsymbol{U}_l\boldsymbol{U}_l^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_j\boldsymbol{U}_j^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_j^\perp\boldsymbol{U}_j^{\perp T}\right)\boldsymbol{x}_0\\
&=\hat{w}_k\left(\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T\boldsymbol{x}_0\right)+\hat{w}_l\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T\boldsymbol{x}_0\right)+\sum_{j\neq k,l}\hat{w}_j\left(\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T\boldsymbol{x}_0\right)\\
&=\left(\frac{\hat{w}_l}{1+\sigma_t^2}+\frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_j)}{\delta^2+\sigma_t^2}\right)\boldsymbol{U}_l\boldsymbol{U}_l^T(\boldsymbol{U}_k\boldsymbol{a}_i+b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\\
&=\left(\frac{\hat{w}_l}{1+\sigma_t^2}+\frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_l)}{\delta^2+\sigma_t^2}\right)b\boldsymbol{U}_l\boldsymbol{e}_i[1:d]
\end{aligned}$$

where the third equality follows since $\hat{w}_j=\hat{w}_l$ for all $j\neq k,l$. Further, we have:

$$\mathbb{E}[\|\boldsymbol{U}_l\boldsymbol{U}_l^T f^\star(\boldsymbol{x}_0,t)\|^2]=\left(\frac{\hat{w}_l}{1+\sigma_t^2}+\frac{\delta^2(\hat{w}_k+(K-2)\hat{w}_l)}{\delta^2+\sigma_t^2}\right)^2\delta^2 d$$

Next, we consider (16):

$$\begin{aligned}
\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T f^\star(\boldsymbol{x}_0,t) &= \hat{w}_k\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_k\boldsymbol{U}_k^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_k^\perp\boldsymbol{U}_k^{\perp T}\right)\boldsymbol{x}_0\\
&\quad+\sum_{l\neq k}\hat{w}_l\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\left(\frac{1}{1+\sigma_t^2}\boldsymbol{U}_l\boldsymbol{U}_l^T+\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_l^\perp\boldsymbol{U}_l^{\perp T}\right)\boldsymbol{x}_0\\
&=\hat{w}_k\left(\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{x}_0\right)+\sum_{l\neq k}\hat{w}_l\left(\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{x}_0\right)\\
&=\frac{\delta^2}{\delta^2+\sigma_t^2}\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T(\boldsymbol{U}_k\boldsymbol{a}_i+b\boldsymbol{U}_k^\perp\boldsymbol{e}_i)\\
&=\frac{\delta^3}{\delta^2+\sigma_t^2}\boldsymbol{U}_\perp\boldsymbol{e}_i[(K-1)d:D]
\end{aligned}$$

Hence:

$$\mathbb{E}[\|\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T f^\star(\boldsymbol{x}_0, t)\|^2] = \frac{\delta^6(n - Kd)}{(\delta^2 + \sigma_t^2)^2}$$

Lastly, we prove (17). Given that the subspaces of all classes and the complement space are both orthonormal and mutually orthogonal, we can write:

$$\mathbb{E}[\|f^\star(\boldsymbol{x}_0, t)\|^2] = \mathbb{E}[\|\boldsymbol{U}_k \boldsymbol{U}_k^T f^\star(\boldsymbol{x}_0, t)\|^2] + \mathbb{E}[\sum_{l \neq k} \|\boldsymbol{U}_l \boldsymbol{U}_l^T f^\star(\boldsymbol{x}_0, t)\|^2] + \mathbb{E}[\|\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T f^\star(\boldsymbol{x}_0, t)\|^2]$$

Combine terms, we get:

$$\begin{aligned}\mathbb{E}[\|f^\star(\boldsymbol{x}_0, t)\|^2] = &\left(\frac{\hat{w}_k}{1 + \sigma_t^2} + \frac{(K-1)\delta^2 \hat{w}_l}{\delta^2 + \sigma_t^2} \right)^2 d \\ &+ (K-1) \left(\frac{\hat{w}_l}{1 + \sigma_t^2} + \frac{\delta^2(\hat{w}_k + (K-2)\hat{w}_l)}{\delta^2 + \sigma_t^2} \right)^2 \delta^2 d + \frac{\delta^6(n - Kd)}{(\delta^2 + \sigma_t^2)^2}.\end{aligned}$$

□