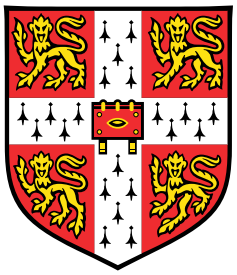
Improving Representation Learning through Generative Modeling

From Diffusion Models to Riemannian Geometry



Georgios Batzolis

Department of Applied Mathematics and Theoretical Physics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Churchill College October 2024

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Georgios Batzolis
October 2024

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures **xiii**

List of tables **xv**

1 Diffusion models encode the intrinsic dimension of data manifolds **1**
1.1 Introduction 1
1.2 Related Work 3
1.3 Proposed Method for Estimation of Intrinsic Dimension 5
1.4 Theoretical Analysis 7
1.5 Limitations 9
1.6 Experiments 10
1.6.1 Experiments on Euclidean datasets 10
1.6.2 Experiments on image datasets 12
1.7 Conclusions and further directions 14

References **17**

Appendix A Diffusion Models Encode the Intrinsic Dimension of Data Manifolds **21**
A.1 Extended background on diffusion models 21
A.2 Training details 23
A.2.1 Euclidean data 23
A.2.2 Image data 23
A.2.3 Auto-encoder 23
A.3 Benchmarking 24
A.4 Proofs 24
A.5 Details on the design of synthetic image manifolds 34
A.6 Additional Experimental Results 36
A.6.1 Euclidean Data 36
A.6.2 Synthetic Image Data 37

A.6.3 MNIST 37
A.7 Robustness analysis 39
A.7.1 Robustness to score approximation error 39
A.7.2 Robustness to non-uniform distribution on the manifold 39
A.7.3 Relaxing the strict manifold assumption 41

List of figures

1.1 The data manifold (in blue) and the neural approximation of the score field $\nabla_{\mathbf{x}} \ln p_{t_0}(\mathbf{x})$ obtained from a diffusion model. Near the manifold the score field is perpendicular to the manifold surface. 3

1.2 To estimate the manifold's dimension at point $\mathbf{x}_0$ (red dot), we sample K nearby points $\mathbf{x}_\varepsilon^{(i)}$ (blue dots) and use the trained diffusion model to evaluate the score function $s_\theta(\mathbf{x}_\varepsilon^{(i)}, \varepsilon)$ at these perturbed points. We assemble these vectors into a matrix and perform Singular Value Decomposition (SVD). The number of (almost) zero singular values reveals the manifold's dimension. . 3

1.3 Singular values for the scores of k-sphere for $k = 10, 50$. In both cases around k singular values almost vanish, clearly indicating the dimensionality of the manifold. Each line shows a score spectrum at different $\mathbf{x}_0^{(j)}$. 11

1.4 Auto-encoder reconstruction error on MNIST for different latent space dimensions. Vertical lines mark different estimations of intrinsic dimension. . 13

1.5 MNIST score spectra that yielded the highest estimated dimension for each digit 13

A.1 Caption without FN 26

A.2 Nine samples from Squares image manifolds of dimensions 10, 20 and 100 (from left to right). 35

A.3 Nine samples from Gaussian blob image manifolds of dimensions 10, 20 and 100 (from left to right). 35

A.4 Projection of the spaghetti line on the first three dimensions. 36

A.5 Score spectrum of the spaghetti line. The last singular value clearly vanishes indicating that the intrinsic dimensionality of the manifold is equal to one. . 36

A.6 Score spectrum for the union of k-spheres ($k_1 = 10, k_2 = 30$). The separated drops in the spectra clearly show that the data comes form the union of two manifolds of different dimensions. 36

A.7 The histogram of estimated dimensions for the union of k-spheres ($k_1 = 10, k_2 = 30$). The counts are taken over estimates $\hat{k}(\mathbf{x}_0^{(i)})$ at different points $\mathbf{x}_0^{(i)}$. 36
A.8 Score spectra and histogram of estimated dimension based on the score spectrum of the Squares image manifold of dimensions 10, 20 and 100. . . 37
A.9 Score spectra and histogram of estimated dimension based on the score spectrum of the Gaussian blobs image manifold of dimensions 10, 20 and 100. 37
A.10 MNIST score spectra for all digits 38
A.11 Score spectra for noise corrupted score model on 25-sphere. 40
A.12 Dimensionality estimates for uniform and non-uniform distributions on a 10-sphere. On the right, we present histograms showing how many points $\mathbf{x}_0^{(j)}$ result in a given $\hat{k}(\mathbf{x}_0^{(j)})$. Taking $\hat{k} = \max_j \hat{k}(\mathbf{x}_0^{(j)})$ allows for robust estimation for moderate values of the concentration parameter α. 42
A.13 Score spectra for score models on 25-sphere trained on noisy manifold data. 43

List of tables

1.1 Estimated intrinsic dimension for each MNIST digit 14
1.2 Comparison of dimensionality detection methods on various data manifolds. 14

A.1 DDPM Model Parameters 24
A.2 Dimensionality detection for non-uniform distribution. For our method the maximum over pointwise estimates $\hat{k}(\mathbf{x}_0)$ is considered. 39

Chapter 1

Diffusion models encode the intrinsic dimension of data manifolds

In this work, we provide a mathematical proof that diffusion models encode data manifolds by approximating their normal bundles. Based on this observation we propose a novel method for extracting the intrinsic dimension of the data manifold from a trained diffusion model. Our insights are based on the fact that a diffusion model approximates the score function i.e. the gradient of the log density of a noise-corrupted version of the target distribution for varying levels of corruption. We prove that as the level of corruption decreases, the score function points towards the manifold, as this direction becomes the direction of maximal likelihood increase. Therefore, at low noise levels, the diffusion model provides us with an approximation of the manifold's normal bundle, allowing for an estimation of the manifold's intrinsic dimension. To the best of our knowledge our method is the first estimator of intrinsic dimension based on diffusion models and it outperforms well established estimators in controlled experiments on both Euclidean and image data. The code is available at https://github.com/GBATZOLIS/ID-diff.

1.1 Introduction

Many modern real-world datasets contain a large number of variables, often exceeding the number of observations. This poses a major challenge in modelling them, due to the *curse of dimensionality*. Despite this complexity, due to the numerous relationships and symmetries among variables, even high-dimensional data often concentrates around a lower-dimensional manifold, a concept known as the *manifold hypothesis* [11]. The dimension of this manifold

is called *intrinsic dimension* (ID), while the high-dimensional space in which the data resides is known as the *ambient space*, with its dimensionality called the *ambient dimension*.

The manifold hypothesis has guided the development of modern high-dimensional data modelling techniques, such as Variational auto-encoders (VAEs) [24], Generative Adversarial Networks (GANs) [13] and M-flows [4].

The estimation of ID holds significant importance in the machine learning community due to its applicability in both theoretical and practical problems [6]. From a theoretical perspective, the ID is essential as it directly affects the convergence rates of fundamental statistical quantities [44]. The higher the ID, the more data is needed for a model to generalize well beyond the training set [6, 38], hence knowing the ID has numerous implications for the generalization and data efficiency of machine learning models [23, 25]. From practical point of view, ID is crucial for a wide range of dimensionality reduction methods [6]. Additionally, understanding the data's ID can help in fine-tuning the latent dimension of models such as GANs, VAEs or M-flows.

In recent years, diffusion models [39, 16] emerged as a new class of deep generative models capable of capturing complex high-dimensional distributions without relying on the notion of data manifold or prior knowledge of the data's ID. Our research reveals that diffusion models encode data manifolds via their normal bundle. Intriguingly, we find that while diffusion models do not *explicitly* rely on the ID, these models estimate it *implicitly*.

As discussed in [41, 16], diffusion models perform score matching [19] and, therefore, contain the information about the gradient of the log-density of the data distribution. We prove that near the data manifold, the gradient of the log-density is orthogonal to the manifold itself. This key observation serves as a tool for deducing the manifold's dimension.

In our study, we investigate three categories of ID estimators: traditional statistical methods (such as PCA and Nearest Neighbor based approaches), normalizing flow-based methods, and our innovative diffusion-based approach. We evaluate the performance of ID estimators on synthetic Euclidean and image datasets, where the dimension of the data manifold is known *a priori*. Moreover, we apply ID estimators to the MNIST dataset [28] (where the ID is unknown), and compare the estimated IDs with the reconstruction error of auto-encoders trained with different latent dimensions.

Our findings indicate that in datasets of high ID, methods that exploit the inductive biases of neural networks are the most effective. Our proposed method stands out by yielding the best results. This success is attributed to utilizing diffusion models, which offer enhanced training stability and avoid the architectural limitations associated with normalizing flows [2].

To summarize our contributions are as follows:

- We elucidate a geometric connection between diffusion models and data manifolds, by proving that a diffusion model encodes the data manifold by approximating its normal bundle.
- Based on this observation we propose a novel method for extracting the ID of the data manifold from a trained diffusion model.
- We perform an extensive evaluation of our novel method as well as several prominent existing methods for ID estimation on a wide range of datasets.

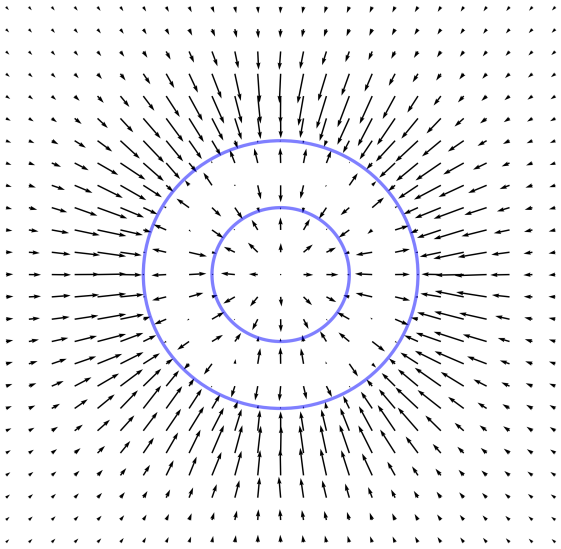


Fig. 1.1 The data manifold (in blue) and the neural approximation of the score field $\nabla_{\mathbf{x}} \ln p_{t_0}(\mathbf{x})$ obtained from a diffusion model. Near the manifold the score field is perpendicular to the manifold surface.

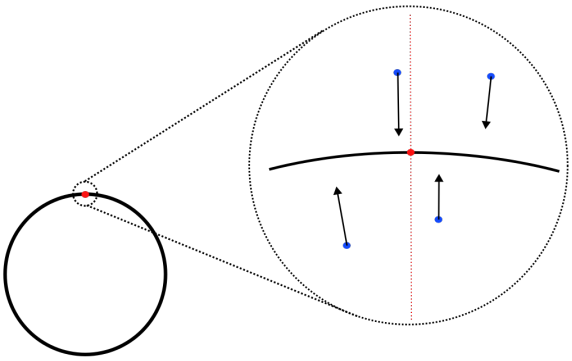


Fig. 1.2 To estimate the manifold's dimension at point $\mathbf{x}_0$ (red dot), we sample K nearby points $\mathbf{x}_\varepsilon^{(i)}$ (blue dots) and use the trained diffusion model to evaluate the score function $s_\theta(\mathbf{x}_\varepsilon^{(i)}, \varepsilon)$ at these perturbed points. We assemble these vectors into a matrix and perform Singular Value Decomposition (SVD). The number of (almost) zero singular values reveals the manifold's dimension.

1.2 Related Work

The relationship between diffusion models and manifold hypothesis has been explored in several recent works. In [37] author examines theoretical conditions under which diffusion models produce samples from the underlying data manifold. In [33] and [7] authors analyze

approximation and generalization abilities of diffusion models under manifold hypothesis. They establish that the data efficiency of training a diffusion model depends on the intrinsic dimension of the data manifold rather than the ambient dimension. This further motivates the importance of ID estimation.

The problem of estimating the intrinsic dimensionality has been widely studied. The two main lines of research are PCA based and nearest neighbour based approaches. In an early work [12] the authors suggest an approach based on using local Karhunen–Loève expansion. In following years many PCA based approaches have been developed. Most notably, in [31] the author suggests an intrinsic dimensionality estimator based on the probabilistic PCA (PPCA) framework [3]. In [10] a local PCA method has been suggested. In [36] authors suggested an estimator based on nearest neighbour information. In [30] authors introduce a maximum likelihood (MLE) procedure based on the distance to m nearest neighbours. Their method has been further improved in the work of [15]. The MLE method has been recently applied by [38] in the estimation of the intrinsic dimensionality of modern image datasets such as MNIST [28], CIFAR [26] and ImageNet [9]. Other works explored geometric approaches using fractal-based methods [5] or packing numbers [22]. We refer to [6] for a comprehensive survey of statistical approaches to ID estimation.

The aforementioned ID estimators do not leverage the inductive bias introduced by modern neural network architectures, which is a crucial reason for the success of modern deep leaning systems [14]. Therefore, their statistical efficiency may be insufficient to deal with datasets of high ID. This limitation has been observed in [6, 18] and is confirmed by our experimental findings.

The limitations of traditional statistical estimators have recently led to the development of deep learning-based intrinsic dimensionality (ID) estimators such as LIDL [42] and ID-NF [18]. These methods, which extract ID from trained normalizing flows, outperform their statistical counterparts by leveraging the inductive biases inherent in modern deep neural network architectures. Despite their advantages, they face challenges as they rely on normalizing flows which are invertible neural networks. Normalizing flows are subject to a trade-off between stability and expressivity, as discussed in [2],[20], [8],[27]. Expressive normalizing flow architectures often face stability problems during training or evaluation, risking the reliability of post-training ID estimation due to potential loss of numerical invertibility. On the other hand, Lipschitz constrained normalizing flow architectures, while more stable, tend to be less expressive, potentially limiting their capacity to accurately estimate the ID of complex, high-dimensional data manifolds.

In contrast, our proposed methodology, which utilizes diffusion models, effectively sidesteps these expressivity limitations as diffusion models do not suffer from similar stability

issues. Our method allows for more effective leveraging of the powerful inductive biases of modern deep neural network architectures, resulting in a more reliable and robust estimation of the ID of complex, high-dimensional data manifolds. This claim is further substantiated by our experimental findings, which show that our proposed diffusion-based estimation method accurates estimes the ID in challenging data manifolds where normalizing flow-based methods fail due to reduced expressivity.

1.3 Proposed Method for Estimation of Intrinsic Dimension

Algorithm 1 Estimate the Intrinsic Dimension at $\mathbf{x}_0$

Input: s_θ - trained diffusion model, $\mathbf{x}_0$ - data point, K - number of samples
Sample $\mathbf{x}_0 \sim p_0(\mathbf{x})$ from the data set
$d \leftarrow \dim(\mathbf{x}_0)$
$S \leftarrow$ empty matrix
for $i = 1, \ldots, K$ **do**
    Sample $\mathbf{x}_{t_0}^{(i)} \sim \mathcal{N}(\mathbf{x}_0, \sigma^2 I)$
    Append $s_\theta(\mathbf{x}_{t_0}^{(i)}, t_0)$ as a new row in S
end for
$(s_i)_{i=1}^d, (\mathbf{v}_i)_{i=1}^d, (\mathbf{w}_i)_{i=1}^d \leftarrow \mathrm{SVD}(S)$
$\hat{k}(\mathbf{x}_0) \leftarrow d - \arg\max_{i=1,\ldots,d-1}(s_i - s_{i+1})$
Output: $\hat{k}(\mathbf{x}_0)$

In [41] score-based [19] and diffusion-based [39, 16] generative models were unified into a single continuous-time score-based framework. The diffusion process is represented by a stochastic differential equation (SDE), which perturbs data distribution p_0 resulting in a series of progressively noise-corrupted distributions p_t. Diffusion models are trained to approximate the score function $\nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$ with a neural network $s_\theta(\mathbf{x}_t, t)$. Once the score function is approximated, the diffusion SDE can be reversed to generate samples from p_0. Additional details on training and sampling from diffusion models are described in Appendix A.1.

Consider a dataset $D = \{\mathbf{x}^{(i)}\}_{i=0}^N \sim p_0(\mathbf{x})$ which consists of N independent d-dimensional vectors $\mathbf{x}^{(i)} \in \mathbb{R}^d$ drawn from distribution $p_0(\mathbf{x})$. The distribution $p_0(\mathbf{x})$ is supported on a k-dimensional manifold $\mathscr{M}$, which is embedded in a space of ambient dimension d. Our goal is to infer the dimension k of the manifold $\mathscr{M}$ from D.

We perturb the data according to the variance exploding SDE $d\mathbf{x}_t = g(t)d\mathbf{w}_t$ [41] and train a neural network $s_\theta(\mathbf{x}_t, t)$ to approximate the score function of the noise perturbed target distribution, i.e. $\nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$ for a range of levels of perturbation indexed by diffusion time t.

We train the model using the weighted denoising score matching objective with likelihood weighting, see [40]. More details about the training of the diffusion model can be found in Appendix A.2.

Consider a datapoint $\mathbf{x}_0$ on a manifold $\mathscr{M}$ and its perturbation into the ambient space $\mathbf{x}_{t_0}$ obtained from the transition kernel $p_{t_0}(\mathbf{x}_{t_0}|\mathbf{x}_0) = \mathscr{N}(\mathbf{x}_{t_0}|\mathbf{x}_0, \sigma_{t_0}^2\mathbf{I})$[1] of the forward process at a small time t_0. As shown in the next section, at $\mathbf{x}_{t_0}$ the score vector $s_\theta(\mathbf{x}_{t_0}, t_0)$ will point towards its orthogonal projection onto $\mathscr{M}$, making it almost orthogonal to $T_{\mathbf{x}_0}\mathscr{M}$ (the tangent space at $\mathbf{x}_0$). This means that the projection of the score vector onto the normal space $\mathscr{N}_{\mathbf{x}_0}\mathscr{M}$ will be significantly larger than its projection onto the tangent space $T_{\mathbf{x}_0}\mathscr{M}$. Therefore, with enough samples, the spectrum obtained from the singular value decomposition of $S = [s_\theta(\mathbf{x}_{t_0}^{(1)}, t_0), ..., s_\theta(\mathbf{x}_t^{(K)}, t_0)]$ will have N_d large singular values and T_d very small singular values, where $N_d = \dim(\mathscr{N}_{\mathbf{x}_0}\mathscr{M})$ and $T_d = \dim(T_{\mathbf{x}_0}\mathscr{M})$. This will be the case because the projection of the score vector at every perturbed point considered is much larger on the normal space than on the tangent space. In our method, we sample $K = 4d$ diffused points at time $t_0 = \varepsilon$ and calculate the SVD of S. The number of vanishing singular values is the estimate of the intrinsic dimension $\hat{k}(\mathbf{x}_0)$.

The resulting spectrum shows a significant drop exactly or very close to the dimension of the normal space. The remaining non-zero, but much smaller singular values correspond to the tangential component of the score vector. This behaviour is expected as the score vector will unavoidably have a very small tangential component, for reasons explained in the following sections. The choice of the cut-off point $\hat{k}(\mathbf{x}_0)$ is usually very clear visually, but can be automated by choosing the point of largest drop in the spectrum:

$$\hat{k}(\mathbf{x}_0) = d - \underset{i=1,..,d-1}{\arg\max}\, s_i - s_{i+1}$$

When selecting $\mathbf{x}_0$, we ideally want a point with high score approximation quality, minimal tangential component, and low manifold curvature. However, since these factors are uncontrollable, we randomly choose multiple $\mathbf{x}_0^{(j)}$ values and plot a spectrum for each. For simple distributions, the score spectra look similar, with drops at accurate values. For more complex distributions, the drop location varies with $\mathbf{x}_0^{(j)}$ choice. We find that the maximum estimated $\hat{k}$ gives the best estimate. Theoretical understanding of the method supports this, as discussed in later sections.

[1] The transition kernels of the variance exploding SDE have this structure, where σ_t is an increasing function determined by $g(t)$ with $\sigma_t \xrightarrow[t\to 0]{} 0$.

1.4 Theoretical Analysis

Here, we provide a theoretical justification for our approach. We demonstrate that, given a collection of points $\mathbf{x}_i \in \mathbb{R}^d$ sufficiently close[2] to the manifold $\mathscr{M}$ with orthogonal projection $\pi(\mathbf{x}_i) \in \mathscr{M}$, the space spanned by the score vectors $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}_i)$ converges to the normal space at $\pi(\mathbf{x}_i)$ in the small t limit. To build intuition, consider a uniform data density on the manifold surface $\mathscr{M}$. The gradient along this density is zero, indicating that for $\mathbf{x}$ close to $\mathscr{M}$ tangential components of the score $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$ will also be approximately zero and the score will be mostly contained in the normal bundle $\mathscr{N}\mathscr{M}$. If the density is non-uniform on the manifold surface however, the score will have a tangential component. Fortunately, for sufficiently small t the change in log-density from moving orthogonally towards the manifold dominates the change from moving tangentially alongside the manifold. This results in the tangential component becoming negligible, and the score still being approximately contained in $\mathscr{N}\mathscr{M}$.

Specifically, we show in the following theorem that for any point $\mathbf{x}$ sufficiently close to the data manifold and $t \to 0$, the score $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$ points directly at the orthogonal projection $\pi(\mathbf{x})$[3].

Theorem 1.4.1. *Suppose that the support of the data distribution P_0 is contained in a compact embedded sub-manifold $\mathscr{M} \subseteq \mathbb{R}^d$ and let P_t be the distribution of samples from P_0 diffused for time t. Then, under mild assumptions (see Appendix A.4), for any point $\mathbf{x} \in \mathbb{R}^d$ sufficiently close to $\mathscr{M}$ with orthogonal projection on $\mathscr{M}$ given by $\pi(\mathbf{x})$, and $\mathbf{n} = (\pi(\mathbf{x}) - \mathbf{x}) / \|\pi(\mathbf{x}) - \mathbf{x}\|$, we have:*

$$S_{\cos}(\mathbf{n}, \nabla_{\boldsymbol{x}} \ln p_t(\mathbf{x})) \xrightarrow[t \to 0]{} 1$$

where $S_{\cos}$ denotes the cosine similarity, defined as $S_{\cos}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$. In other words, for sufficiently small t the score $\nabla_{\boldsymbol{x}} \ln p_t(\mathbf{x})$ points directly at the projection of $\mathbf{x}$ on the manifold.

This theorem leads to the conclusion that this score is contained within the normal space of the manifold, as we show with the following corollary:

Corollary 1.4.2. *The ratio of the projection of the score $\nabla_{\boldsymbol{x}} \ln p_t(\mathbf{x})$ on the tangent space of the data manifold $T_{\pi(\mathbf{x})}\mathscr{M}$ to the projection on the normal space $\mathscr{N}_{\pi(\mathbf{x})}\mathscr{M}$ approaches zero*

[2] Within the tubular neighbourhood of $\mathscr{M}$ to the manifold $\mathscr{M}$. See Appendix A.4.

[3] Every compact embedded sub-manifold $\mathscr{M}$ has a tubular neighbourhood, and every point $\mathbf{x}$ in the tubular neighbourhood of the manifold has a unique orthogonal projection $\pi(\mathbf{x})$ onto the manifold. See Appendix A.4 for more details.

as t approaches zero, i.e.

$$\frac{\|\mathbf{T}\nabla_{\boldsymbol{x}} \ln p_t(\mathbf{x})\|}{\|\mathbf{N}\nabla_x \ln p_t(\mathbf{x})\|} \to 0, \text{ as } t \to 0.$$

where $\mathbf{N}$ *and* $\mathbf{T}$ *are projection matrices on* $\mathcal{N}_{\pi(\mathbf{x})}\mathcal{M}$ *and* $T_{\pi(\mathbf{x})}\mathcal{M}$ *respectively. Therefore for sufficiently small* t *the score* $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$ *is (effectively) contained in the normal space* $\mathcal{N}_{\pi(\mathbf{x})}\mathcal{M}$.

Proof. The full proof of the theorem and corollary can be found in the Appendix A.4.

In practice we choose small $t > 0$, and for each chosen $\mathbf{x}_0 \in \mathcal{M}$ we sample points $\mathbf{x}_t^{(i)}$ around it from $p_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_0, \sigma_t^2\mathbf{I})$. With over 99% probability $\mathbf{x}_t^{(i)} \in B(\mathbf{x}_0, 3\sigma_t)$, and so as we decrease t most of our $\mathbf{x}_t$ will become very close to $\mathbf{x}_0$. For $B(\mathbf{x}_0, 3\sigma_t)$ sufficiently small, the effect of curvature of $\mathcal{M}$ becomes negligible inside $B(\mathbf{x}_0, 3\sigma_t)$, and so the normal spaces $\mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M}$ will all be approximately equal to $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$. Under this assumption, we can outline the practical implications of these theoretical results. Assuming $t > 0$ small and a trained score approximation $s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$, the score matrix $S = [s_\theta(\mathbf{x}_t^{(1)}, t), ..., s_\theta(\mathbf{x}_t^{(4d)}, t)]$ has the following properties:

1. The columns of S are approximately contained in the normal space $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$
2. The columns of S approximately span the normal space $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$
3. The singular values of S corresponding to singular vectors in the normal space are large relative to those corresponding to tangent singular vectors

The first point is a direct consequence of Corollary 1.4.2. For the second point, denote $n_{\mathbf{x}} := \frac{\pi(\mathbf{x})-\mathbf{x}}{\|\pi(\mathbf{x})-\mathbf{x}\|}$, and assume that t is sufficiently small such that $\mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M} \approx \mathcal{N}_{\mathbf{x}_0}\mathcal{M}$. Then locally, the vectors $\left\{n_{\mathbf{x}_t^{(1)}}, \ldots, n_{\mathbf{x}_t^{(K)}}\right\}$ are independent Gaussian perturbations from a linear subspace. With probability one this set contains $N_d = \dim(\mathcal{N}_{\mathbf{x}_0}\mathcal{M})$ linearly independent vectors spanning $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$, so by Theorem 1.4.1 the score vectors $\left\{\nabla_{\mathbf{x}} \ln p_t\left(\mathbf{x}_t^{(1)}\right), \ldots, \nabla_{\mathbf{x}} \ln p_t\left(\mathbf{x}_t^{(K)}\right)\right\}$ must therefore also span $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$. For the third point note that if the columns of S span $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$, then S has rank N_d, and its SVD yields singular values $s_i > 0$ for $i \leq N_d$ corresponding to singular vectors in $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$, and $s_j = 0$ for $j > N_d$. In practice we fix some small $t > 0$ and so small components of $T_{\mathbf{x}_0}\mathcal{M}$ are introduced to the score by factors such as non-uniform distribution of data samples on $\mathcal{M}$, therefore in applications the SVD of S yields small singular values $s_j > 0$ for $j > N_d$, however we still observe that $s_i \gg s_j$ where $i \leq N_d$.

To formalize the idea of "approximately spanning the normal space" and $\mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M} \approx \mathcal{N}_{\mathbf{x}_0}\mathcal{M}$, we use the concept of cosine similarity and the angle between subspaces. Cosine

similarity measures the cosine of the angle between two vectors, which helps quantify how closely two vectors (or subspaces) align.

Let $v_1, \ldots, v_k$ be vectors sampled from the normal space $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$. We define the cosine similarity between two vectors v_i and v_j as:

$$\cos(\theta_{ij}) = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|},$$

where θ_{ij} is the angle between v_i and v_j.

For subspaces $\mathcal{N}_{\mathbf{x}_0}\mathcal{M}$ and $\mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M}$, we consider the principal angles $\theta_1, \ldots, \theta_k$ between them. If the cosine of these angles is close to 1, the subspaces are well-aligned:

$$\cos(\theta_k) \approx 1 \quad \text{implies} \quad \mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M} \approx \mathcal{N}_{\mathbf{x}_0}\mathcal{M}.$$

Thus, by evaluating the cosine similarities and the angles between the subspaces, we can quantify the approximation quality of spanning the normal space. This formalization supports the intuitive claims by providing a measurable criterion for the alignment of normal spaces.

1.5 Limitations

In section 1.4, we established that given a perfect score approximation for sufficiently low t our method produces the correct estimation of the dimension. However, in practice, our method may encounter two types of errors: *approximation error* and *geometric error*. The approximation error arises as a result of having an imperfect score approximation $s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$.

Geometric error arises if the selected sampling time t isn't sufficiently small, potentially impacting our method's accuracy for two reasons. Firstly, it may result in an increased tangential component of the score vector. Secondly, if $\mathbf{x}_t^{(i)}$ lies too distant from $\mathcal{M}$, the manifold's curvature may create a difference between normal spaces $\mathcal{N}_{\pi(\mathbf{x}_t^{(i)})}\mathcal{M}$ across varying i.

We empirically assess our method's robustness to approximation error and find it robust to minor inaccuracies in score approximation. Additionally, we analyze our method's sensitivity to p_0 non-uniformity, which could induce a minor tangential score component for $t > 0$. We discover that using the maximum $\hat{k}(\mathbf{x}_0^{(j)})$ allows our method to accommodate varying levels of non-uniformity over the manifold surface, displaying superior robustness compared to other non-linear estimators without the need for reducing t. Details are in Appendix A.7

We note that Theorem 1.4.1 assumes that the data distribution's support is exclusively within a manifold. Therefore, we empirically investigate the method's applicability when data is concentrated around, but not entirely within, a manifold. We discover that for a k-sphere, our method remains reliable as long as the data is closely concentrated around the manifold. Details are available in Appendix A.7.

1.6 Experiments

We examine the effectiveness of our method on a multitude of manifold datasets, each embedded in a high-dimensional ambient space. The datasets fall into two categories: *Euclidean datasets* consisting of points from manifolds embedded in a high-dimensional Euclidean space and *image datasets* consisting of synthetic manifolds of images. In each case we know the intrinsic dimension of the manifold *a priori*. Additionally, we apply our method to the MNIST dataset, where the true intrinsic dimension is unknown. We assess its performance via comparison with the reconstruction error of auto-encoders with various latent dimensions. For each dataset we train a diffusion model, and then apply our method to estimate the intrinsic dimension of the data manifold. Details on hyperparameters and architectures used in our experiments can be found in Appendix A.2.

We compare our method against established approaches to intrinsic dimensionality estimation: the nearest neighbour based maximum likelihood estimator (MLE) [30], [15], Local PCA [10] and Probabilistic PCA (PPCA) [31] [3]. Additionally, we compare our method against ID-NF [18], which is the best performing method for extracting the ID from pretrained normalizing flows. The details about the implementation of the benchmarks are in the Appendix A.3.

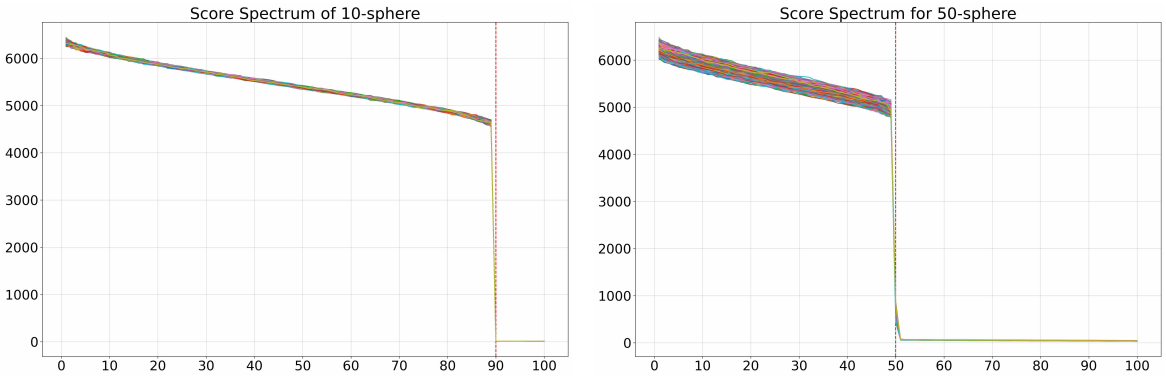
Our method consistently yields the best estimate or close to the best estimate among considered approaches. In the following subsections we present a detailed discussion of each experiment. The results are summarised in Table 1.2.

1.6.1 Experiments on Euclidean datasets

Embedded k-spheres: We examined our method on k dimensional spheres embedded in a $d = 100$ dimensional ambient space via a random isometric embedding[4]. We consider two cases $k_1 = 10$ and $k_2 = 50$. The spectra of resulting score matrices are presented in Figure

[4]To obtain a random isometric embedding we first generate the k sphere in a $k+1$ dimensional small ambient space. Then we sample a random $d \times (k+1)$ Gaussian matrix A. We perform a QR decomposition $A = QR$. Finally we use the $d \times (k+1)$ isometry matrix Q to embed the small $k+1$ dimensional space containing our manifold in the large d dimensional ambient space.

1.3. Our method gives estimates of $\hat{k}_1 = 11$ and $\hat{k}_2 = 51$, which are very close to the true intrinsic dimensionality of the manifolds.



(a) $k = 10$ (b) $k = 50$

Fig. 1.3 Singular values for the scores of k-sphere for $k = 10, 50$. In both cases around k singular values almost vanish, clearly indicating the dimensionality of the manifold. Each line shows a score spectrum at different $\mathbf{x}_0^{(j)}$.

Spaghetti line: The intrinsic dimensionality of k-spheres could be well approximated by a linear dimensionality detection methods such as [31]. This is because these manifolds are contained in a low dimensional *linear* subspace. In order to showcase the advantage of the *non-linear* nature of our method we consider a *spaghetti line* manifold. That is a curve $\tau \mapsto (\sin(\tau), \sin(2\tau), ..., \sin(100\tau))$ in a 100 dimensional ambient space, which is not contained in any low dimensional linear subspace (cf. Figure A.4 in Appendix A.6.1). As expected, the linear method [31] greatly overstates the intrinsic dimension with a result of $\hat{k}_{\text{PPCA}} = 98$. Yet, our approach utilizes the non-linear knowledge from the diffusion model to accurately predict an intrinsic dimensionality of one. The score spectrum is presented in Figure A.5 in Appendix A.6.1.

Union of k-spheres: Due to the local nature of our method, we are able to generate an estimate $\hat{k}(\mathbf{x}_0)$ of the intrinsic dimension around a given point $\mathbf{x}_0$. This allows us to apply our approach to a union of manifolds and identify the dimension of each component. We illustrate this feature with the following experiment. We embed two spheres of different radii and dimensions in a 100 dimensional ambient space. First sphere has dimension $k_1 = 10$ and radius $r_1 = 1$ and the second sphere has $k_2 = 30$ and radius $r_2 = 0.25$[5]. We apply our method to this data using multiple $\mathbf{x}_0^{(i)}$ randomly sampled from the dataset. We observe that our method produces a spectrum with two visible, separated drops. This indicates that the data comes from the union of manifolds of different dimensions. The resulting estimates are

[5] One can intuitively think of this manifold as a high-dimensional analog of a planet with a ring around it.

$\hat{k}_1 = 10$ and $\hat{k}_2 = 31$ depending on the chosen $\mathbf{x}_0^{(j)}$. The score spectra and the histogram of estimated dimensions are presented in Figures A.6 and A.7 in Appendix A.6.1.

1.6.2 Experiments on image datasets

Synthetic image manifolds: In this experiment, we investigated our method's ability to infer the dimension of synthetic image manifolds with known dimension. We crafted two synthetic image manifolds with controllable intrinsic dimension k: the "k squares images manifold" and the "k Gaussian blobs images manifold". The construction of these manifolds is detailed in Appendix A.5. We evaluated our method on k =10, 20, and 100 dimensional manifolds for both types, with the score spectra and histograms of estimated dimensions for numerous data points displayed in Figures A.8 and A.9 in Appendix A.6.2.

On the squares image manifold, our method, ID-NF and PPCA consistently yielded accurate dimension estimates. PPCA's success on this dataset was anticipated since the manifold resides within a k-dimensional linear subspace.

On the more complex Gaussian blobs image manifold, our method stood out as the sole technique to consistently deliver accurate dimension estimates. The accuracy of our method was not compromised by the manifold's increased complexity, unlike other methods. However, the estimation for the 100-dimensional manifold introduced some uncertainty, as indicated by a more leveled histogram and a less abrupt spectrum collapse (c.f. Figure A.9). This is attributed to the manifold's increased complexity and the inherent challenges in optimization, resulting in greater geometric and approximation errors.

MNIST: In our study, we additionally applied the proposed technique to estimate the intrinsic dimension of the well-known MNIST dataset - an image dataset with an as-of-yet-undetermined intrinsic dimension. Our findings suggest that there exists a variation in the intrinsic dimensions across different digits. For instance, the digit '1' yielded an estimated dimension of 66, whereas the digit '9' exhibited a significantly higher estimated dimension of 152. This discrepancy can be attributed to the increased geometric complexity inherent to the digit '9'. Figure 1.5 elucidates these observations by displaying the score spectra which yielded the maximum estimated dimensions for each digit. We present the estimated dimension for each digit in Table 1.1 and the complete set of spectra for each digit in the Appendix A.6.3.

We validate our estimates by comparing them with the reconstruction error of auto-encoders trained with different latent dimensions. As demonstrated in Figure 4, the ID estimate of our method is in close agreement with that of the ID-NF method, and both correlate with the point of diminishing returns on the reconstruction loss curve. This point marks a plateau in the effectiveness of additional latent dimensions to significantly reduce

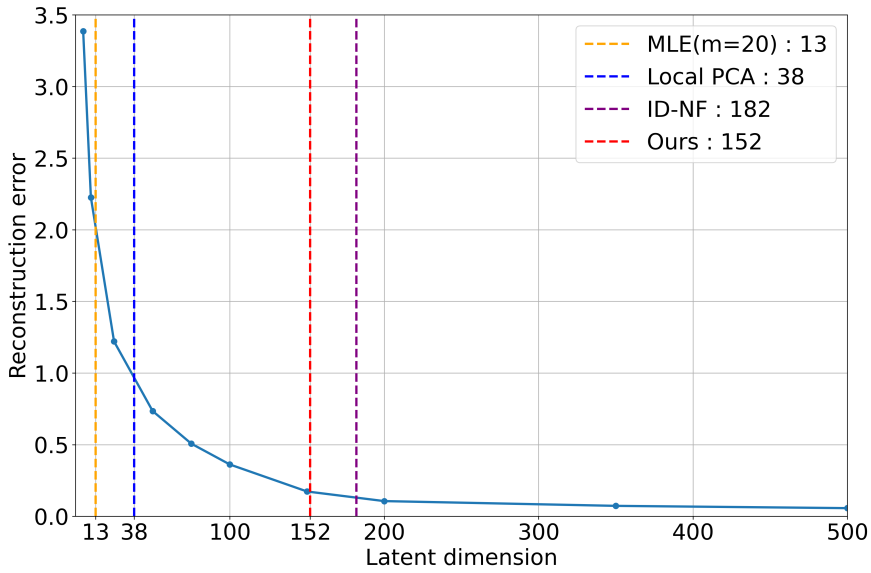


Fig. 1.4 Auto-encoder reconstruction error on MNIST for different latent space dimensions. Vertical lines mark different estimations of intrinsic dimension.

reconstruction error, further suggesting this point as the dataset's intrinsic dimension. In constrast, estimates produced by MLE and Local PCA are significantly lower, corresponding to regions of the curve where the reconstruction loss is still steeply decreasing. This suggests these methods underestimate the manifold dimension. These findings call for a careful interpretation of the intrinsic dimension estimates of popular machine learning datasets provided by [38], as they rely on the MLE method, which we have found to consistently underestimate manifold dimensions. On the other hand, PPCA notably overestimated the dimension, with $\hat{k}_{\text{PPCA}} = 706$.

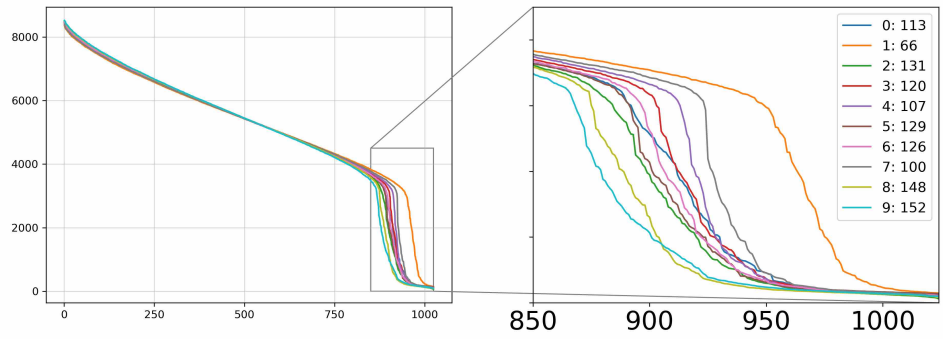


Fig. 1.5 MNIST score spectra that yielded the highest estimated dimension for each digit

0	1	2	3	4	5	6	7	8	9
113	66	131	120	107	129	126	100	148	152

Table 1.1 Estimated intrinsic dimension for each MNIST digit

	Ground Truth	Ours	ID-NF	MLE (m=5)	MLE (m=20)	Local PCA	PPCA
Euclidean Data Manifolds							
10-sphere	10	11	11	9.61	9.46	11	11
50-sphere	50	51	51	35.52	34.04	51	51
Spaghetti line	1	1	1	1.01	1.00	32	98
Image Manifolds							
Squares							
$k = 10$	10	11	9.7	8.48	8.17	10	10
$k = 20$	20	22	19.5	14.96	14.36	20	20
$k = 100$	100	100	94.2	37.69	34.42	78	99
Gaussian blobs							
$k = 10$	10	12	9.8	8.88	8.67	10	136
$k = 20$	20	21	17.8	16.34	15.75	20	264
$k = 100$	100	98	56.3	39.66	35.31	18	985
MNIST	N/A	152	182	14.12	13.27	38	706

Table 1.2 Comparison of dimensionality detection methods on various data manifolds.

1.7 Conclusions and further directions

In this work, we proved theoretically and confirmed experimentally that diffusion models can infer the intrinsic dimension from the data. We introduced an approach that estimates the intrinsic dimension of the data manifold from a pre-trained diffusion model. This approach capitalizes on the observation that, the diffusion model evaluated at sufficiently small diffusion time approximates the normal bundle of the data manifold. Our work offers a twofold contribution: it highlights that diffusion model detects the lower dimensional structure of data and provides a rigorous method for intrinsic dimension estimation.

We conducted a rigorous comparison of three types of ID estimators: traditional statistical methods, normalizing flow-based techniques, and our diffusion-based approach. Our findings consistently show that for high-ID datasets, methods leveraging neural networks' inductive biases are superior. Notably, our diffusion-based method emerges as the most effective, owing to its enhanced training stability and freedom from the architectural constraints of normalizing flows.

Furthermore, our research introduces new estimates for the MNIST's dimensionality, demonstrating strong alignment with the predictions of an auto-encoder trained across a range of latent dimensions.

Our work opens new paths for understanding and estimating intrinsic data dimension, with potential implications across the field of machine learning. Future research should explore this method's applicability to other data types and its potential across various domains.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

JS aknowledges support from the Cantab Capital Institute for the Mathematics of Information and Aviva. GB acknowledges support from GSK. TD acknowledges support from the EPSRC programme grant in 'The Mathematics of Deep Learning', under the project EP/L015684/1. CBS acknowledges support from the Philip Leverhulme Prize, the Royal Society Wolfson Fellowship, the EPSRC advanced career fellowship EP/V029428/1, the EPSRC programme grant EP/V026259/1, and the EPSRC grants EP/S026045/1 and EP/T003553/1, EP/N014588/1, EP/T017961/1, the Wellcome Innovator Awards 215733/Z/19/Z and 221633/Z/20/Z, the European Union Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 777826 NoMADS, the Cantab Capital Institute for the Mathematics of Information and the Alan Turing Institute. This research was supported by the NIHR Cambridge Biomedical Research Centre (NIHR203312). The views expressed are those of the author(s) and not necessarily those of the NIHR or the Department of Health and Social Care.

References

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

[2] Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R., and Jacobsen, J.-H. (2021). Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR.

[3] Bishop, C. M. and Tipping, M. E. (2001). Probabilistic principal component analysis. PPCA.

[4] Brehmer, J. and Cranmer, K. (2020). Flows for simultaneous manifold learning and density estimation. *Advances in Neural Information Processing Systems*, 33:442–453.

[5] Camastra, F. and Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407.

[6] Campadelli, P., Casiraghi, E., Ceruti, C., and Rozza, A. (2015). Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015:1–21.

[7] Chen, M., Huang, K., Zhao, T., and Wang, M. (2023). Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data.

[8] Cornish, R., Caterini, A., Deligiannidis, G., and Doucet, A. (2020). Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR.

[9] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[10] Fan, M., Gu, N., Qiao, H., and Zhang, B. (2010). Intrinsic dimension estimation of data by principal component analysis.

[11] Fefferman, C., Mitter, S., and Narayanan, H. (2013). Testing the manifold hypothesis.

[12] Fukunaga, K. and Olsen, D. (1971). An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20(2):176–183.

[13] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[14] Goyal, A. and Bengio, Y. (2022). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068.

[15] Haro, G., Randall, G., and Sapiro, G. (2008). Translated poisson mixture model for stratification learning. *Int. J. Comput. Vis.*, 80(3):358–374.

[16] Ho, J., Jain, A., and Abbeel, P. (2020a). Denoising diffusion probabilistic models.

[17] Ho, J., Jain, A., and Abbeel, P. (2020b). Denoising diffusion probabilistic models.

[18] Horvat, C. and Pfister, J.-P. (2022). Intrinsic dimensionality estimation using normalizing flows. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 12225–12236. Curran Associates, Inc.

[19] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

[20] Jaini, P., Kobyzev, I., Yu, Y., and Brubaker, M. (2020). Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR.

[21] Johnsson, K. (2016). intrinsicdimension: Intrinsic dimension estimation.

[22] Kégl, B. (2002). Intrinsic dimension estimation using packing numbers. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

[23] Kim, J., Shin, J., Rinaldo, A., and Wasserman, L. (2019). Uniform convergence rate of the kernel density estimator adaptive to intrinsic volume dimension. In *International Conference on Machine Learning*, pages 3398–3407. PMLR.

[24] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014*.

[25] Kpotufe, S. (2011). k-nn regression adapts to local intrinsic dimension. *Advances in neural information processing systems*, 24.

[26] Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. *University of Toronto*.

[27] Laszkiewicz, M., Lederer, J., and Fischer, A. (2021). Copula-based normalizing flows. *arXiv preprint arXiv:2107.07352*.

[28] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

[29] Lee, J. (2019). *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing.

[30] Levina, E. and Bickel, P. (2004). Maximum likelihood estimation of intrinsic dimension. In Saul, L., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

[31] Minka, T. (2000). Automatic choice of dimensionality for pca. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

[32] Nicolaescu, L. (2011). *An Invitation to Morse Theory*. Universitext. Springer New York.

[33] Oko, K., Akiyama, S., and Suzuki, T. (2023). Diffusion models are minimax optimal distribution estimators.

[34] Palais, R. S. and Terng, C. (1988). *Critical Point Theory and Submanifold Geometry*. Critical Point Theory and Submanifold Geometry. Springer.

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[36] Pettis, K. W., Bailey, T. A., Jain, A. K., and Dubes, R. C. (1979). An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(1):25–37.

[37] Pidstrigach, J. (2022). Score-based generative models detect manifolds.

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

[39] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics.

[40] Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021). Maximum likelihood training of score-based diffusion models.

[41] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

[42] Tempczyk, P., Michaluk, R., Łukasz Garncarek, Spurek, P., Tabor, J., and Goliński, A. (2022). Lidl: Local intrinsic dimension estimation using approximate likelihood.

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

[44] Weed, J. and Bach, F. (2019). Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance.

Appendix A

Diffusion Models Encode the Intrinsic Dimension of Data Manifolds

A.1 Extended background on diffusion models

Setup: In [41] score-based [19] and diffusion-based [39, 16] generative models have been unified into a single continuous-time score-based framework where the diffusion is driven by a stochastic differential equation. This framework relies on Anderson's Theorem [1], which states that under certain Lipschitz conditions on the drift coefficient $f : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ and on the diffusion coefficient $G : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ and an integrability condition on the target distribution $p_0(\mathbf{x}_0)$ a forward diffusion process governed by the following SDE:

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + G(\mathbf{x}_t, t)d\mathbf{w}_t \tag{A.1}$$

has a reverse diffusion process governed by the following SDE:

$$d\mathbf{x}_t = [f(\mathbf{x}_t, t) - G(\mathbf{x}_t, t)G(\mathbf{x}_t, t)^T \nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)]dt + G(\mathbf{x}_t, t)d\bar{\mathbf{w}}_t, \tag{A.2}$$

where $\bar{\mathbf{w}}_t$ is a standard Wiener process in reverse time.

The forward diffusion process transforms the *target distribution* $p_0(\mathbf{x}_0)$ to a *diffused distribution* $p_T(\mathbf{x}_T)$ after diffusion time T. By appropriately selecting the drift and the diffusion coefficients of the forward SDE, we can make sure that after sufficiently long time T, the diffused distribution $p_T(\mathbf{x}_T)$ approximates a simple distribution, such as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We refer to this simple distribution as the *prior distribution*, denoted by π. The reverse diffusion process transforms the diffused distribution $p_T(\mathbf{x}_T)$ to the data distribution $p_0(\mathbf{x}_0)$ and the prior distribution π to a distribution p^{SDE}. The distribution p^{SDE} is close to $p_0(\mathbf{x}_0)$ if the

diffused distribution $p_T(\mathbf{x}_T)$ is close to the prior distribution π. We get samples from p^{SDE} by sampling from π and simulating the reverse SDE from time T to time 0.

Sampling: To get samples by simulating the reverse SDE, we need access to the time-dependent *score function* $\nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$. In practice, we approximate the time-dependent score function with a neural network $s_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$ and simulate the reverse SDE presented in equation A.3 to map the prior distribution π to p_θ^{SDE}.

$$d\mathbf{x}_t = [f(\mathbf{x}_t, t) - G(\mathbf{x}_t, t)G(\mathbf{x}_t, t)^T s_\theta(\mathbf{x}_t, t)]dt + G(\mathbf{x}_t, t)d\bar{\mathbf{w}}_t, \tag{A.3}$$

If the prior distribution is close to the diffused distribution and the approximated score function is close to the ground truth score function, the modeled distribution p_θ^{SDE} is provably close to the target distribution $p_0(\mathbf{x}_0)$. This statement is formalised in the language of distributional distances in the work of [40].

Training: A neural network $s_\theta(\mathbf{x}_t, t)$ can be trained to approximate the score function $\nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$ by minimizing the weighted score matching objective

$$\mathscr{L}_{SM}(\theta, \lambda(\cdot)) := \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ \mathbf{x}_t \sim p_t(\mathbf{x}_t)}} [\lambda(t) \, \| \nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t) \|_2^2] \tag{A.4}$$

where $\lambda : [0, T] \to \mathbb{R}_+$ is a positive weighting function.

However, the above quantity cannot be optimized directly since we don't have access to the ground truth score $\nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t)$. Therefore in practice, a different objective has to be used [19, 43, 41]. In [41], the weighted denoising score-matching objective is used, which is defined as

$$\mathscr{L}_{DSM}(\theta, \lambda(\cdot)) := \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ \mathbf{x}_0 \sim p_0(\mathbf{x}_0) \\ \mathbf{x}_t \sim p_t(\mathbf{x}_t | \mathbf{x}_0)}} [\lambda(t) \, \| \nabla_{\mathbf{x}_t} \ln p_t(\mathbf{x}_t | \mathbf{x}_0) - s_\theta(\mathbf{x}_t, t) \|_2^2] \tag{A.5}$$

The difference between DSM and SM is the replacement of the ground truth score which we do not know by the score of the perturbation kernel which we know analytically for many choices of forward SDEs. The choice of the weighted DSM objective is justified because the weighted DSM objective is equal to the SM objective up to a constant that does not depend on the parameters of the model θ. The reader can refer to [43] for the proof.

A.2 Training details

We trained the score model using the weighted denoising score matching objective [41], presented in eq. A.5. We used the likelihood weighting function, i.e. $\lambda(t) = g(t)^2$, where $g(t)$ is the diffusion coefficient of the forward SDE.

A.2.1 Euclidean data

For all of our experiments on Euclidean data, we used a fully connected network with 5 hidden layers and 2048 nodes in each hidden layer to approximate the score function. The input and output dimension is the same as the ambient dimension. For the optimisation of the model, we used the Adam algorithm with a learning rate of $2e{-}5$ and exponential moving average (EMA) on the weights of the model with a decay rate of 0.9999. Moreover, we chose the variance exploding SDE [41] as the forward process with $\sigma_{min} = 0.01$ and $\sigma_{max} = 4$.

A.2.2 Image data

For all our of our experiments on image data, we used the DDPM architecture [17] with variance exploding SDE [41] and hyperparameters indicated in Table A.1.

A.2.3 Auto-encoder

The encoder and decoder encoder architectures are based on the DDPM U-Net [16], which we call half-U nets.

For the encoder we used the downsampling part of the U-Net and removed the upsampling part and the skip connections. The downscaled tensor is flattened and mapped to the latent dimension with an additional linear layer.

For the decoder we start by linearly transforming the latent vector and reshaping it into a tensor of appropriate dimension. Then we used the upsampling part of the DDPM U-Net.

We used the Adam optimizer and EMA rate 0.999. We used learning rate scheduler reducing the loss on plateau starting form 10^{-4} and stopping at 10^{-5}. We trained the auto-encoder for each latent dimension for 36h on NVIDIA A-100 GPU. At the end we used checkpoints which minimized the validation loss to evaluate the reconstruction error.

All other hyperparameters are included in Table A.1.

Hyper-parameter	**MNIST**	**Synthetic Image data**
Number of filters	128	128
Channel multipliers	(1, 2, 2, 4)	(1, 2, 2, 2)
Dropout	0.1	0.1
EMA rate	0.999	0.999
Normalization	GroupNorm	GroupNorm
Nonlinearity	Swish	Swish
Number of residual blocks	4	4
Attention resolution	16	16
Convolution size	3	3
σ_{min}	0.009	0.01
σ_{max}	50	50
Learning rate	Scheduler(10^{-4}, 10^{-5})	$2 \cdot 10^{-4}$

Table A.1 DDPM Model Parameters

A.3 Benchmarking

We compared our method against well established approaches to intrinsic dimensionality estimation: the MLE estimator [30], [15], Local PCA [10] and Probabilistic PCA [31] [3]. For MLE estimator and local PCA we used the implementation provided in the R package INTRINSICDIMENSION [21]. The MLE estimator has an important hyperparameter m - the number of nearest neighbour distances that should be used for the dimension estimation. We used values $m = 5$ and 20 since these are extremal values considered in [38]. For PPCA we used the SCIKIT-LEARN implementation [35]. The code for reproducing the benchmarking experiments is included in our codebase.

For the ID-NF method [18], we used the official implementation available at https://github.com/chrvt/ID-NF. For the Euclidean data, we utilized the "vector data" folder, which employs block neural autoregressive flows to learn the normalizing flows, from which the intrinsic dimension is extracted using the ID-NF method. For the synthetic image data and MNIST, we used the "images" folder, which uses rational quadratic splines to train the normalizing flows.

A.4 Proofs

Here we provide full proofs for the statements in Section 1.4. First, we show that for any point $\mathbf{x}$ sufficiently close to the data manifold and sufficiently small t the score $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$ points directly at the manifold. We demonstrate this by showing that projection of the score

in any direction $v \perp \mathbf{n}$ vanishes in proportion to the projection on $\mathbf{n} = \frac{(\pi(\mathbf{x})-\mathbf{x})}{\|\pi(\mathbf{x})-\mathbf{x}\|}$ as $t \to 0$. Then Theorem 1.4.1 and Corollary 1.4.2 will follow easily from this result.

Theorem A.4.1. *Suppose that the the support of the data distribution P_0 is contained in a compact embedded sub-manifold $\mathscr{M} \subseteq \mathbb{R}^d$ and let P_t be the distribution of samples from P_0 diffused for time t. Then, under mild assumptions, for any point $\mathbf{x} \in \mathbb{R}^d$ sufficiently close to $\mathscr{M}$, with orthogonal projection on $\mathscr{M}$, given by $\pi(\mathbf{x})$. Let $\mathbf{n}$ be a unit vector pointing from $\mathbf{x}$ to $\pi(\mathbf{x})$, then we have that for any unit vector v orthogonal to $\mathbf{n}$:*

$$\frac{v^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})} \to 0, as\ t \to 0.$$

Assumptions

1. The distribution P_0 has a smooth density p_0 wrt the volume measure on the manifold.
2. The density p_0 is bounded away from zero on the manifold.

Illustrative simple case

We first present an illustrative proof of a simple case of $\mathscr{M}$ being a linear subspace with $k = 1$ and $d = 2$. This case gives all of the essential ideas behind the general proof without much of the technicality. For the more interested reader, we then provide a proof of the result for a general manifold, using tools such as the notion of tubular neighbourhoods and some results from Morse theory.

Without the loss of generality assume that $\mathscr{M} = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 = 0\}$ the line given by the x_1-axis. Pick a point $\mathbf{x} \in \mathbb{R}^2$. The score at point $\mathbf{x}$ is given by

$$\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}) = \frac{1}{\sigma_t^2 p_t(\mathbf{x})} \int_{\mathscr{M}} (\mathbf{y} - \mathbf{x}) \mathscr{N}(\mathbf{y}|\mathbf{x}, \sigma_t^2 \mathbf{I}) p_0(\mathbf{y}) d\mathbf{y}.$$

Notice that $\mathscr{N}((y_1, y_1)|(x_1, x_2), \sigma_t^2 \mathbf{I})$[1] is a bivariate normal distribution and its restriction to $\mathscr{M}$ is equal to $\mathscr{N}((y_1, 0)|(x_1, x_2), \sigma_t^2 \mathbf{I}) = \mathscr{N}(0|x_2, \sigma_t^2) \mathscr{N}(y_1|x_1, \sigma_t^2)$. Therefore

$$\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}) = \frac{\mathscr{N}(0|x_2, \sigma_t^2)}{\sigma_t^2 p_t(\mathbf{x})} \int_{\mathscr{M}} (\mathbf{y} - \mathbf{x}) \mathscr{N}(y_1|x_1, \sigma_t^2) p_0(\mathbf{y}) d\mathbf{y}.$$

This means that the score is the weighted average of vectors pointing from $\mathbf{x}$ to $\mathbf{y}$ over all choices of points $\mathbf{y}$ on the manifold, with weights given by $w(\mathbf{y}; \sigma_t) := \mathscr{N}(y_1|x_1, \sigma_t^2) p_0(\mathbf{y})$

[1] In component-wise notation $\mathbf{y} = (y_1, y_2)$ and $\mathbf{x} = (x_1, x_2)$.

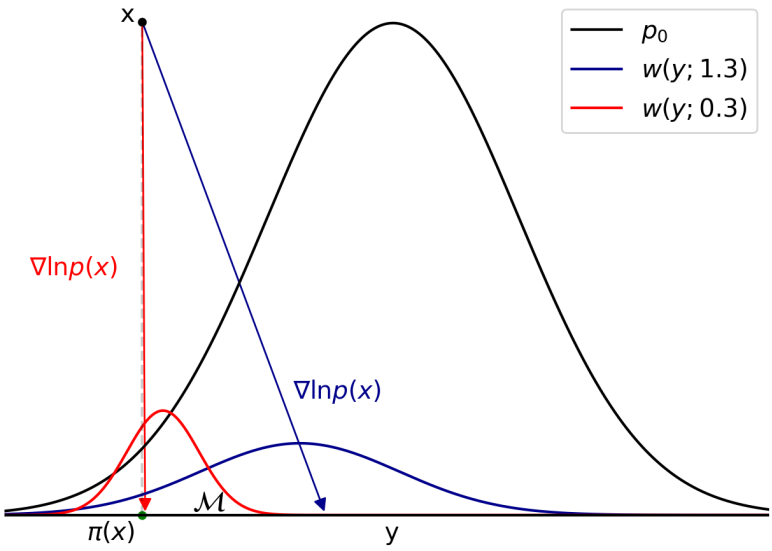


Fig. A.1 The score is the weighted average of vectors pointing from $\mathbf{x}$ to $\mathbf{y}$ with weights given by $w(\mathbf{y};\sigma_t)$. As σ_t decreases weights $w(\mathbf{y};\sigma_t)$ concentrate around $\pi(\mathbf{x})$ and the influence of points $\mathbf{y}$ away from projection $\pi(\mathbf{x})$ becomes insignificant. Therefore, the direction of the score tends to a align with $\pi(\mathbf{x})-\mathbf{x}$. (Norm of the score vectors on the figure was scaled for better visibility. The direction is preserved.)

(see Figure A.1 for visual explanation). For small σ_t these weights concentrate around $\pi(\mathbf{x}) = (x_1, 0)$ the projection of $\mathbf{x}$ on $\mathscr{M}$, and vanishing far away from it. Consider a ratio of the tangential part to the normal part of the score:

$$\frac{v^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})} = \frac{\int_{\mathscr{M}} v^T(\mathbf{y}-\mathbf{x}) \mathscr{N}(y_1|x_1,\sigma_t^2) p_0(\mathbf{y}) d\mathbf{y}}{\int_{\mathscr{M}} \mathbf{n}^T(\mathbf{y}-\mathbf{x}) \mathscr{N}(y_1|x_1,\sigma_t^2) p_0(\mathbf{y}) d\mathbf{y}} \xrightarrow[\sigma_t \to 0]{} \frac{\int_{\mathscr{M}} v^T(\mathbf{y}-\mathbf{x}) \delta_{x_1}(y_1) p_0(\mathbf{y}) d\mathbf{y}}{\int_{\mathscr{M}} \mathbf{n}^T(\mathbf{y}-\mathbf{x}) \delta_{x_1}(y_1) p_0(\mathbf{y}) d\mathbf{y}}$$
$$= \frac{v^T((x_1,0)-\mathbf{x})}{\mathbf{n}^T((x_1,0)-\mathbf{x})} = \frac{(1,0)^T(0,-x_2)}{(0,1)^T(0,-x_2)} = 0$$

where v and $\mathbf{n}$ are unit vectors in tangential and normal directions respectively. This implies,

$$\mathrm{S}_{\cos}(\mathbf{n}, \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})) = \frac{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\|\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})\|} = \frac{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\sqrt{(\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2 + (v^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2}}$$
$$= \frac{1}{\sqrt{1 + \left(\frac{v^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}\right)^2}} \xrightarrow[t \to 0]{} 1.$$

This establishes the theorem for the simple case. The corollary follows immediately since in the simple case we have $\mathbf{T} = v^T$ and $\mathbf{N} = \mathbf{n}^T$. □

Deriving the formula for the density of P_t

Let $\mathscr{M}$ be a compact k-dimensional manifold embedded in $\mathbb{R}^d$. Let $A \subseteq \mathbb{R}^d$. We define the measure P_0 on $\mathbb{R}^d$ as

$$P_0(A) := \int_{A\cap M} p_0(\mathbf{y})d\mathbf{y} \tag{A.6}$$

where $d\mathbf{y}$ is the volume form[2] on M and p_0 is a smooth function on $\mathscr{M}$ [3] such that $\int_{\mathscr{M}} p_0(\mathbf{y})d\mathbf{y} = 1$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a P_0-measurable function. By approximating f with simple functions (linear combination of indicator functions) we conclude that:

$$\int_A f dP_0 = \int_{A\cap M} f(\mathbf{y})p_0(\mathbf{y})d\mathbf{y} \tag{A.7}$$

Consider a measure P_t as a convolution of P_0 with a normal distribution on $\mathbb{R}^d$. For any measurable $A \subseteq \mathbb{R}^d$ we have

$$\begin{aligned}(P_0 * \mathscr{N}_{0,t})(A) &:= \int_{\mathbb{R}^d}\int_{A-\mathbf{y}} d\mathscr{N}_{0,t}(\mathbf{x})dP_0(\mathbf{y}) = \int_{\mathbb{R}^d}\int_{A-\mathbf{y}} \mathscr{N}(\mathbf{x}|0,\sigma_t^2\mathbf{I})d\mathbf{x}dP_0(\mathbf{y}) \\ &= \int_{\mathbb{R}^d}\int_{A} \mathscr{N}(\mathbf{x}-\mathbf{y}|0,\sigma_t^2\mathbf{I})d\mathbf{x}dP_0(\mathbf{y}) = \int_{\mathbb{R}^d}\int_{A} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})d\mathbf{x}dP_0(\mathbf{y}) \\ &= \int_{A}\int_{\mathbb{R}^d} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})dP_0(\mathbf{y})d\mathbf{x} \overset{\text{(A.7)}}{=} \int_{A}\int_{\mathscr{M}} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})p_0(\mathbf{y})d\mathbf{y}d\mathbf{x}\end{aligned}$$

where $d\mathbf{y}$ is a volume form on M and $d\mathbf{x}$ is a volume form on $\mathbb{R}^d$. Therefore the measure P_t has a density on $\mathbb{R}^d$ given by:

$$p_t(\mathbf{x}) = \int_{\mathscr{M}} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})p_0(\mathbf{y})d\mathbf{y}. \tag{A.8}$$

Note the $\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})$ here. Typically one would write this as $\mathscr{N}(\mathbf{x}|\mathbf{y},\sigma_t^2\mathbf{I})$ and think of (A.8) as the quantity of probability mass at point $\mathbf{x}$ after diffusing for time t with initial distribution $p_0(\mathbf{y})$. We instead write it this way as it will be more intuitive to think of (A.8) as the average probability mass that intersects the manifold after diffusing from a delta distribution at $\mathbf{x}$ (where the average is taken over $p_0(\mathbf{y})$). These are of course equivalent as $\mathscr{N}(\mathbf{x}|\mathbf{y},\sigma_t^2\mathbf{I})$ is symmetric is $\mathbf{x}$ and $\mathbf{y}$.

[2] Integrating over A using the volume form of $\mathscr{M}$ can be thought of as taking an appropriately re-scaled Lebesgue integral over A. That is $P_0(A) = \int_{A\cap\mathscr{M}} p_0(\mathbf{y})d\mathbf{y} = \int_A \int_{\mathbb{R}^d} \delta(\mathbf{s}-\mathbf{y})\hat{p}_0(\mathbf{s})d\mathbf{s}d\mathbf{y}$. Where the latter are Lebesgue integrals and $\hat{p}_0(\mathbf{s}) = p_0(\mathbf{s})$ for $\mathbf{s} \in \mathscr{M}$ and zero otherwise.

[3] i.e. for any chart $\phi : \mathbb{R}^k \supseteq U \to M$ the composition $p_0 \circ \phi : \mathbb{R}^k \supseteq U \to \mathbb{R}$ is smooth.

Tubular Neighbourhoods

First we need to ensure that the point $\mathbf{x}$ has a unique projection on $\mathscr{M}$. This is always true for an $\mathbf{x}$ sufficiently close to $\mathscr{M}$. We can formalize this with the notion of *tubular neighbourhood* - a tube around $\mathscr{M}$ such that every point $\mathbf{x}$ inside can be uniquely represented as a sum of the point on the manifold and a vector from the normal bundle i.e. $\mathbf{x} = \mathbf{y} + \mathbf{v}$ where $\mathbf{y} \in \mathscr{M}$ and $\mathbf{v} \in \mathscr{N}_{\mathbf{y}}\mathscr{M}$. Formally:

Definition A.4.2. Endpoint Map
The endpoint map $Y : \mathscr{N}\mathscr{M} \to \mathbb{R}^d$ is defined by $Y(\mathbf{y}, \mathbf{v}) = \mathbf{y} + \mathbf{v}$ for $\mathbf{y} \in \mathscr{M}$ and $\mathbf{v} \in \mathscr{N}_{\mathbf{y}}\mathscr{M}$.

Definition A.4.3. Tubular Neighbourhood
A (uniform) tubular neighbourhood of $\mathscr{M}$ is a neighbourhood U_R of $\mathscr{M}$ in $\mathbb{R}^d$ that is a diffeomorphic image under the endpoint map of an open subset $V_R \subseteq \mathscr{N}\mathscr{M}$ of the form:

$$V_R = \{(\mathbf{y}, \mathbf{v}) \in \mathscr{N}\mathscr{M} : \|\mathbf{v}\|_2 < R\}$$

Since Y restricted to V_R is a diffeomorphism[4], it follows that every point $\mathbf{x} = (\mathbf{y}, \mathbf{v})$ in the tubular neighbourhood has a unique orthogonal projection on $\mathscr{M}$ given by $\mathbf{y}$. We will denote this projection as $\pi(\mathbf{x})$.

Conveniently, it turns out that every compact embedded submanifold of $\mathbb{R}^d$ has a tubular neighborhood.

Theorem A.4.4. *Tubular Neighborhood Theorem [Theorem 5.25][29]*
Every compact embedded submanifold of $\mathbb{R}^d$ has a uniform tubular neighborhood.

Preliminary lemmas and Morse theory

In this section we will establish that for every $\mathbf{x}$ in the tubular neighbourhood of $\mathscr{M}$ there exists an open neighbourhood E of $\pi(\mathbf{x})$ such that:

1. $\mathbf{n}^T(\mathbf{y} - \mathbf{x}) > \|\pi(\mathbf{x}) - \mathbf{x}\|_2 - \varepsilon$ on E .
2. $|v^T(\mathbf{y} - \mathbf{x})| < \varepsilon$ on E.
3. The mass of a Gaussian centred at $\mathbf{x}$ is concentrated in E,

$$\frac{\int_{\mathscr{M}\setminus E} \mathscr{N}(\mathbf{y}|\mathbf{x}, \sigma_t^2\mathbf{I})d\mathbf{y}}{\int_E \mathscr{N}(\mathbf{y}|\mathbf{x}, \sigma_t^2\mathbf{I})d\mathbf{y}} \to 0 \text{ as } t \to 0.$$

[4] so in particular it is a bijection

We begin by defining an E which satisfies the first two conditions.

Lemma A.4.5. *Choose E contained in a ball of radius $0 < \varepsilon < \|\mathbf{x} - \pi(\mathbf{x})\|$ around $\pi(\mathbf{x})$. Let $\mathbf{y} \in E$, and let $\mathbf{v}_\varepsilon := \mathbf{y} - \pi(\mathbf{x})$. Then*

1. $\mathbf{n}^T(\mathbf{y}-\mathbf{x}) > \|\pi(\mathbf{x}) - \mathbf{x}\|_2 - \varepsilon$ *on E .*
2. $|\nu^T(\mathbf{y}-\mathbf{x})| < \varepsilon$ *on E.*

Proof. By direct computation

$$\begin{aligned}\mathbf{n}^T(\mathbf{y}-\mathbf{x}) &= \mathbf{n}^T((\mathbf{x}-\pi(\mathbf{x})) + \mathbf{v}_\varepsilon) \\ &= \mathbf{n}^T(\mathbf{x}-\pi(\mathbf{x})) + \mathbf{n}^T\mathbf{v}_\varepsilon \\ &= \|\mathbf{x}-\pi(\mathbf{x})\|_2 + \mathbf{n}^T\mathbf{v}_\varepsilon \\ &\geq \|\mathbf{x}-\pi(\mathbf{x})\| - \|\mathbf{v}_\varepsilon\|_2 .\end{aligned}$$

We have that $\|\mathbf{v}_\varepsilon\| < \varepsilon$, hence for all $\mathbf{y}$ in E we have that $\mathbf{n}^T(\mathbf{y}-\mathbf{x}) \geq \|\mathbf{x}-\pi(\mathbf{x})\| - \varepsilon > 0$. For the second inequality

$$\begin{aligned}|\nu^T(\mathbf{y}-\mathbf{x})| &\leq |\nu^T((\mathbf{x}-\pi(\mathbf{x}))| + |\nu^T\mathbf{v}_\varepsilon| \\ &\leq \|\mathbf{v}_\varepsilon\|_2 \\ &\leq \varepsilon.\end{aligned}$$

□

Now to find E which also satisfies the last condition we proceed by recalling some elementary definitions and results of Morse theory.

Theorem A.4.6. *Morse lemma [Corollary 1.17][32]*
If $\mathbf{y}_0$ is a non-degenerate critical point of index γ of a smooth function $f : \mathscr{M} \to \mathbb{R}$, then there exist a chart $\phi = (\phi_i)_{i=1}^k$ in a neighbourhood U of $\mathbf{y}_0$ such that $\phi(\mathbf{y}_0) = 0$, and in this chart we have the equality:

$$f(\mathbf{y}) = f(\mathbf{y}_0) - \sum_{i=1}^{\gamma} \phi_i(\mathbf{y})^2 + \sum_{i=\gamma+1}^{k} \phi_i(\mathbf{y})^2$$

Let $f_\mathbf{x}(\mathbf{y}) : \mathscr{M} \to \mathbb{R}$ denote the squared distance function from $\mathbf{x}$ given by $f_\mathbf{x}(\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2^2$. We will establish that if $\mathbf{x}$ is in a tubular neighbourhood, then its projection $\pi(\mathbf{x})$ is a non-degenerate critical point of $f_\mathbf{x}$ of index zero.

Definition A.4.7. Focal point
A point $\mathbf{x} = Y(\mathbf{y}, \mathbf{v})$ in the image of the endpoint map Y, is called a non-focal point of $\mathscr{M}$ with respect to $\mathbf{y}$ if $dY(\mathbf{y}, \mathbf{v})$ is an isomorphism. Otherwise it is called a focal point.

Theorem A.4.8. *Critical points and focal points [34]*
Let $\mathscr{M}$ be an embedded submanifold of $\mathbb{R}^d$, $\mathbf{y} \in M$, $\mathbf{v} \in \mathscr{N}_{\mathbf{y}}\mathscr{M}$, and $\mathbf{x} = Y(\mathbf{y}, \mathbf{v}) = \mathbf{y} + \mathbf{v}$. Then

1. *$\mathbf{y}$ is a critical point of $f_{\mathbf{x}}$.*
2. *$\mathbf{y}$ is a non-degenerate critical point of $f_{\mathbf{x}}$ if and only if $\mathbf{x}$ is a non-focal point.*
3. *Index of f_a at $\mathbf{y}$ is equal to the number of focal points of $\mathscr{M}$ with respect to $\mathbf{y}$ on the line segment joining $\mathbf{y}$ to $\mathbf{x}$.*

Because the restriction of the endpoint map Y to the tubular neighbourhood is a diffeomorphism, the differential dY is an isomorphism for every point in the tubular neighbourhood. Therefore there are no focal points of $\mathscr{M}$ in the tubular neighbourhood. Hence, it follows directly from the above theorem, that if $\mathbf{x}$ is in the tubular neighbourhood, then the projection $\pi(\mathbf{x})$ is a non-degenerate critical point of $f_{\mathbf{x}}$ of index zero. Now we are ready to prove the following lemma.

Lemma A.4.9. *There exists a connected open neighbourhood E of $\pi(\mathbf{x})$ satisfying conditions of lemma A.4.5 and such that,*

$$\frac{\int_{\mathscr{M}\setminus E} \mathscr{N}(\mathbf{y}|\mathbf{x}, \sigma_t^2\mathbf{I})d\mathbf{y}}{\int_E \mathscr{N}(\mathbf{y}|\mathbf{x}, \sigma_t^2\mathbf{I})d\mathbf{y}} \to 0 \text{ as } t \to 0. \tag{A.9}$$

Proof. Fix $\varepsilon > 0$. Then conditions of lemma A.4.5 are satisfied inside $B(\pi(\mathbf{x}), \varepsilon)$. We have demonstrated that $\pi(\mathbf{x})$ fulfills the criteria stipulated by the Morse lemma. Consequently, we can pick $\tilde{U}$ as the neighborhood and ϕ as the coordinate system that the Morse lemma provides. Now let $U = \tilde{U} \cap B(\pi(\mathbf{x}), \varepsilon)$. Since $\mathscr{M}$ is compact and U is open, there exists $m = \min_{\mathscr{M}\setminus U} f_{\mathbf{x}}(\mathbf{y})$ and by uniqueness of projection $f_{\mathbf{x}}(\pi(\mathbf{x})) < m$. Let $r = \sqrt{m - (f_{\mathbf{x}}(\pi(\mathbf{x})))/2}$ and let $E = \phi^{-1}(B(0, r))$. For all $\mathbf{y} \in E$ we have:

$$f_{\mathbf{x}}(\mathbf{y}) = f_{\mathbf{x}}(\pi(\mathbf{x})) + \|\phi(\mathbf{y})\|^2 < f_{\mathbf{x}}(\pi(\mathbf{x})) + r^2 < m.$$

Notice that for every $\mathbf{y} \in U \setminus E$ we have $f_{\mathbf{x}}(\mathbf{y}) \geq f_{\mathbf{x}}(\pi(\mathbf{x})) + r^2$. Therefore we have established that

$$\forall_{\mathbf{y}\in E} \forall_{\tilde{\mathbf{y}}\in \mathscr{M}\setminus E} f_{\mathbf{x}}(\mathbf{y}) < f_{\mathbf{x}}(\tilde{\mathbf{y}}). \tag{A.10}$$

Computing directly, we have

$$\frac{\int_{\mathscr{M}\setminus E}\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})d\mathbf{y}}{\int_E\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})d\mathbf{y}}=\frac{\int_{\mathscr{M}\setminus E}\exp\{-f_{\mathbf{x}}(\mathbf{y})/2\sigma_t^2\}d\mathbf{y}}{\int_E\exp\{-f_{\mathbf{x}}(\mathbf{y})/2\sigma_t^2\}d\mathbf{y}} \tag{A.11}$$

By the mean value theorem there exists $\mathbf{y}^*\in E$ and $\tilde{\mathbf{y}}^*\in\mathscr{M}\setminus E$ such that

$$\int_E\exp\{-f_{\mathbf{x}}(\mathbf{y})/2\sigma_t^2\}d\mathbf{y}=\mathrm{Vol}(E)\exp\{-f_{\mathbf{x}}(\mathbf{y}^*)/2\sigma_t^2\}$$
$$\int_{\mathscr{M}\setminus E}\exp\{-f_{\mathbf{x}}(\mathbf{y})/2\sigma_t^2\}d\mathbf{y}=\mathrm{Vol}(\mathscr{M}\setminus E)\exp\{-f_{\mathbf{x}}(\tilde{\mathbf{y}}^*)/2\sigma_t^2\}$$

We can use this to evaluate (A.11) to give,

$$\begin{aligned}\frac{\int_{\mathscr{M}\setminus E}\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})d\mathbf{y}}{\int_E\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})d\mathbf{y}}&=\frac{\mathrm{Vol}(\mathscr{M}\setminus E)\exp\left\{-f_{\mathbf{x}}(\tilde{\mathbf{y}}^*)/2\sigma_t^2\right\}}{\mathrm{Vol}(E)\exp\left\{-f_{\mathbf{x}}(\mathbf{y}^*)/2\sigma_t^2\right\}}\\&=\frac{\mathrm{Vol}(\mathscr{M}\setminus E)}{\mathrm{Vol}(E)}\exp\left\{-\frac{f_{\mathbf{x}}(\tilde{\mathbf{y}}^*)-f_{\mathbf{x}}(\mathbf{y}^*)}{2\sigma_t^2}\right\}\end{aligned}$$

Since by (A.10) $f_{\mathbf{x}}(\tilde{\mathbf{y}}^*)-f_{\mathbf{x}}(\mathbf{y}^*)>0$ the above goes to zero as σ_t goes to zero. Moreover, since $E\subseteq U\subseteq B(\pi(\mathbf{x}),\varepsilon)$, the conditions of lemma A.4.5 are also satisfied. □

Proof of Theorem A.4.1

Fix $\varepsilon>0$. Assume that $\mathbf{x}$ is in a tubular neighbourhood of $\mathscr{M}$, so that the projection $\pi(\mathbf{x})$ exists. To simplify the expressions that appear multiple times, we define the following functions:

$$g(\mathbf{y},\mathbf{x},t)=\nu^T(\mathbf{y}-\mathbf{x})\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})p_0(\mathbf{y}).$$

$$h(\mathbf{y},\mathbf{x},t)=\mathbf{n}^T(\mathbf{y}-\mathbf{x})\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})p_0(\mathbf{y}),$$

Then, we have

$$\frac{\nu^T\nabla_{\mathbf{x}}\ln p_t(\mathbf{x})}{\mathbf{n}^T\nabla_{\mathbf{x}}\ln p_t(\mathbf{x})}=\frac{\nu^T\nabla_{\mathbf{x}}p_t(\mathbf{x})}{\mathbf{n}^T\nabla_{\mathbf{x}}p_t(\mathbf{x})}=\frac{\int_{\mathscr{M}}g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_{\mathscr{M}}h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}. \tag{A.12}$$

Split $\mathscr{M}$ into two parts: E and $\mathscr{M}\setminus E$, where $E=B(\mathbf{x},r)\cap\mathscr{M}$ is an open neighbourhood of $\pi(\mathbf{x})$ in $\mathscr{M}$ satisfying the conditions of Lemma A.4.9 and Lemma A.4.5 for a chosen $\varepsilon>0$. Then, we have that (A.12) is equal to

$$\frac{\int_E g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_{\mathscr{M}} h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}} + \frac{\int_{\mathscr{M}\setminus E} g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_{\mathscr{M}} h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}. \tag{A.13}$$

We begin by bounding the first term:

$$\begin{aligned}
\frac{\int_E g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_{\mathscr{M}} h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}} &= \frac{\int_E g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y} + \int_{\mathscr{M}\setminus E} h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}} \\
&= \frac{\dfrac{\int_E g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}}{1 + \dfrac{\int_{\mathscr{M}\setminus E} h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}} \\
&=: \frac{A_t}{1+B_t}.
\end{aligned}$$

For A_t, we have

$$|A_t| = \left|\frac{\int_E g(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}}\right|.$$

Using the fact that $\mathbf{n}^T(\mathbf{y}-\mathbf{x})$ is positive and bounded away from zero and applying the triangle inequality, we obtain

$$\begin{aligned}
|A_t| &\leq \frac{\int_E |v^T(\mathbf{y}-\mathbf{x})|\mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})p_0(\mathbf{y})\,d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\,d\mathbf{y}} \\
&\leq \frac{\varepsilon\, p_{\max}\int_E \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\,d\mathbf{y}}{(\|\pi(\mathbf{x})-\mathbf{x}\|-\varepsilon)\,p_{\min}\int_E \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\,d\mathbf{y}} \\
&= \frac{p_{\max}}{p_{\min}}\frac{\varepsilon}{\|\pi(\mathbf{x})-\mathbf{x}\|-\varepsilon},
\end{aligned}$$

where in the second inequality we used that $0 < p_{\min} < p_0(\mathbf{y}) < p_{\max}$, $|v^T(\mathbf{y}-\mathbf{x})| < \varepsilon$, and $\mathbf{n}^T(\mathbf{y}-\mathbf{x}) > \|\pi(\mathbf{x})-\mathbf{x}\| - \varepsilon$ on E.

Since ε was arbitrary, this term can be made arbitrarily small.

Now we move to B_t. Let $D = \max_{\mathbf{y}\in\mathscr{M}} \|\mathbf{x}-\mathbf{y}\|$. By the triangle and Cauchy-Schwarz inequalities, we have

$$\begin{aligned}|B_t| &\leq \frac{\int_{\mathscr{M}\setminus E} |\mathbf{n}^T(\mathbf{y}-\mathbf{x})| \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I}) p_0(\mathbf{y})\, d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}} \\ &\leq \frac{p_{\max} D}{p_{\min}(\|\boldsymbol{\pi}(\mathbf{x})-\mathbf{x}\|-\varepsilon)} \frac{\int_{\mathscr{M}\setminus E} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\, d\mathbf{y}}{\int_E \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\, d\mathbf{y}},\end{aligned}$$

which goes to zero as t goes to zero.

Finally, we move to the second term in (A.13). We start with the same steps as with the first term:

$$\frac{\int_{\mathscr{M}\setminus E} g(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}}{\int_{\mathscr{M}} h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}} = \frac{\frac{\int_{\mathscr{M}\setminus E} g(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}}}{1+\frac{\int_{\mathscr{M}\setminus E} h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}}} =: \frac{C_t}{1+B_t},$$

so we only need to examine C_t. By the triangle and Cauchy-Schwarz inequalities again, we have

$$\begin{aligned}|C_t| &\leq \frac{\int_{\mathscr{M}\setminus E} |\boldsymbol{\nu}^T(\mathbf{y}-\mathbf{x})| \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I}) p_0(\mathbf{y})\, d\mathbf{y}}{\int_E h(\mathbf{y},\mathbf{x},t)\, d\mathbf{y}} \\ &\leq \frac{p_{\max} D}{p_{\min}(\|\boldsymbol{\pi}(\mathbf{x})-\mathbf{x}\|-\varepsilon)} \frac{\int_{\mathscr{M}\setminus E} \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\, d\mathbf{y}}{\int_E \mathscr{N}(\mathbf{y}|\mathbf{x},\sigma_t^2\mathbf{I})\, d\mathbf{y}},\end{aligned}$$

which goes to zero as t goes to zero.

Putting it all together:

$$\begin{aligned}\frac{|\boldsymbol{\nu}^T\nabla_{\mathbf{x}}\ln p_t(\mathbf{x})|}{|\mathbf{n}^T\nabla_{\mathbf{x}}\ln p_t(\mathbf{x})|} &= \frac{|A_t+C_t|}{|1+B_t|} \\ &\leq \frac{1}{|1+B_t|}\left(\frac{p_{\max}}{p_{\min}}\frac{\varepsilon}{\|\boldsymbol{\pi}(\mathbf{x})-\mathbf{x}\|-\varepsilon}+|C_t|\right) \\ &\xrightarrow[t\to 0]{} \frac{p_{\max}}{p_{\min}}\frac{\varepsilon}{\|\boldsymbol{\pi}(\mathbf{x})-\mathbf{x}\|-\varepsilon}.\end{aligned}$$

Since ε can be chosen arbitrarily small, this finishes the proof. □

Proof of Theorem 1.4.1

Beginning with $\mathbf{n}$ and extending to an orthonormal basis $(\mathbf{n}, v_1, \ldots, v_{d-1})$ of $\mathbb{R}^d$, we have

$$\begin{aligned} \mathrm{S}_{\cos}(\mathbf{n}, \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})) &= \frac{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\|\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})\|} = \frac{\mathbf{n}^T (\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))}{\sqrt{(\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2 + \sum_{i=1}^{d-1} (v_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2}} \\ &= \frac{1}{\sqrt{1 + \sum_{i=1}^{d-1} \left(\frac{v_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}\right)^2}} \xrightarrow[t \to 0]{} 1, \end{aligned}$$

where in taking the limit we applied the Theorem A.4.1. □

Proof of Corollary 1.4.2

Let $(\tau_1, \ldots, \tau_k)$ be an orthonormal basis of the tangent space $T_{\pi(\mathbf{x})}\mathscr{M}$ and extend $\mathbf{n}$ to an orthonormal basis $(\mathbf{n}, \eta_1, \ldots, \eta_{d-k-1})$ of the normal space $\mathscr{N}_{\pi(\mathbf{x})}\mathscr{M}$. Then the projection of the score on the tangent space is given by $\mathbf{T}\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}) = \sum_{i=1}^{k} (\tau_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))\tau_i$, while the projection on the normal space is $\mathbf{N}\nabla_{\mathbf{x}} \ln p_t(\mathbf{x}) = (\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))\mathbf{n} + \sum_{i=1}^{d-k-1} (\eta_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))\eta_i$. Therefore

$$\begin{aligned} \frac{\|\mathbf{T}\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})\|}{\|\mathbf{N}\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})\|} &= \sqrt{\frac{\sum_{i=1}^{k} (\tau_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2}{(\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2 + \sum_{i=1}^{d-k-1} (\eta_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x}))^2}} \\ &= \sqrt{\frac{\sum_{i=1}^{k} \left(\frac{\tau_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}\right)^2}{1 + \sum_{i=1}^{d-k-1} \left(\frac{\eta_i^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}{\mathbf{n}^T \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})}\right)^2}} \xrightarrow[t \to 0]{} 0 \end{aligned}$$

where in taking the limit we applied the Theorem A.4.1. □

A.5 Details on the design of synthetic image manifolds

Squares image manifold: We generated the *k-squares dataset* whose intrinsic dimension is controllable and is set to k. This dataset comprises 32×32 pixel images of squares, generated by first establishing fixed square center locations and side lengths (either 3 or 5 units) across all datapoints. For each square in a given image, we uniformly sampled a brightness value from the unit interval for all pixels within the square's boundary, summing values at points of intersection. The dimension of this manifold is equal to the number of squares k and the ambient space dimension is $32 \times 32 = 1024$. We provide samples in Figure A.2.

Gaussian blobs image manifold: The Squares image manifold is contained in a low dimensional linear subspace which allowed PPCA to estimate the dimension very well. For this reason, we constructed a synthetic image manifold of known dimension, which cannot be contained in any low dimensional linear subspace. We replace the squares by Gaussian blobs (i.e. brightness of pixels withing each blob is proportional to a Gaussian pdf). We randomly pick the centers of the Gaussian blobs which remain fixed for all datapoints. For each datapoint, we sample the standard deviation of each Gaussian blob uniformly from the interval $[1,5]$. The dimension of this manifold is equal to the number of Gaussian blobs. We provide samples in Figure A.3.

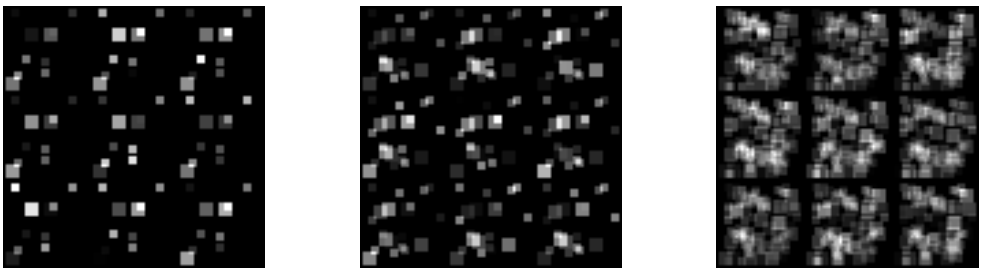


Fig. A.2 Nine samples from Squares image manifolds of dimensions 10, 20 and 100 (from left to right).

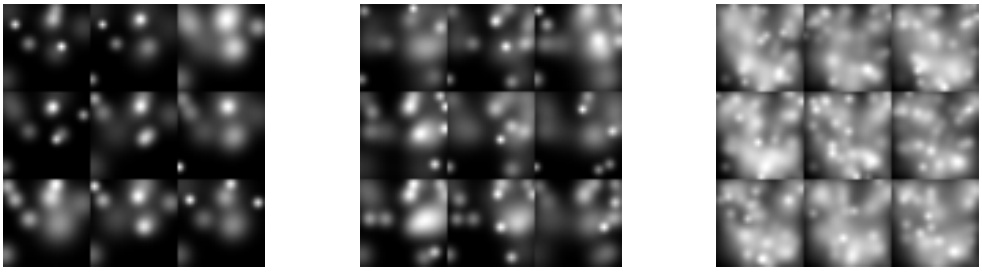


Fig. A.3 Nine samples from Gaussian blob image manifolds of dimensions 10, 20 and 100 (from left to right).

A.6 Additional Experimental Results

A.6.1 Euclidean Data

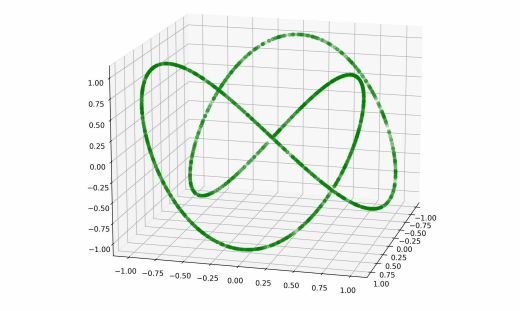


Fig. A.4 Projection of the spaghetti line on the first three dimensions.

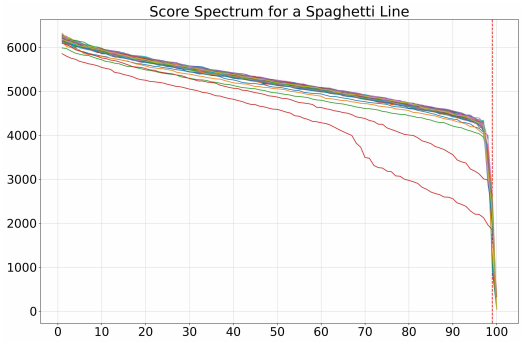


Fig. A.5 Score spectrum of the spaghetti line. The last singular value clearly vanishes indicating that the intrinsic dimensionality of the manifold is equal to one.

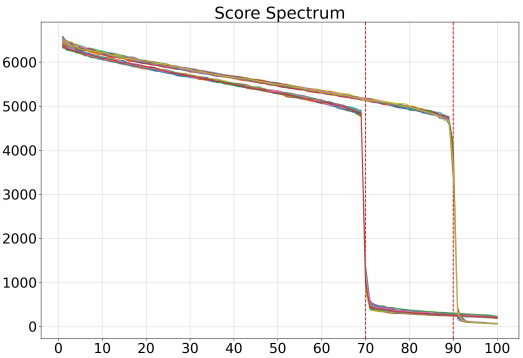


Fig. A.6 Score spectrum for the union of k-spheres ($k_1 = 10, k_2 = 30$). The separated drops in the spectra clearly show that the data comes form the union of two manifolds of different dimensions.

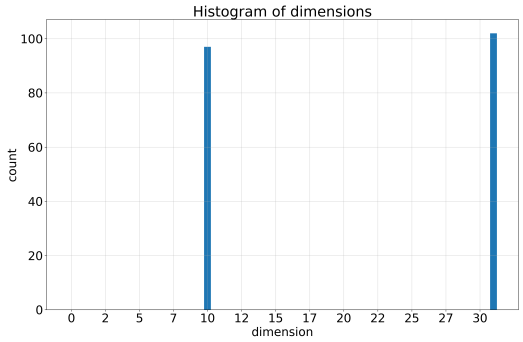


Fig. A.7 The histogram of estimated dimensions for the union of k-spheres ($k_1 = 10, k_2 = 30$). The counts are taken over estimates $\hat{k}(\mathbf{x}_0^{(i)})$ at different points $\mathbf{x}_0^{(i)}$.

A.6.2 Synthetic Image Data

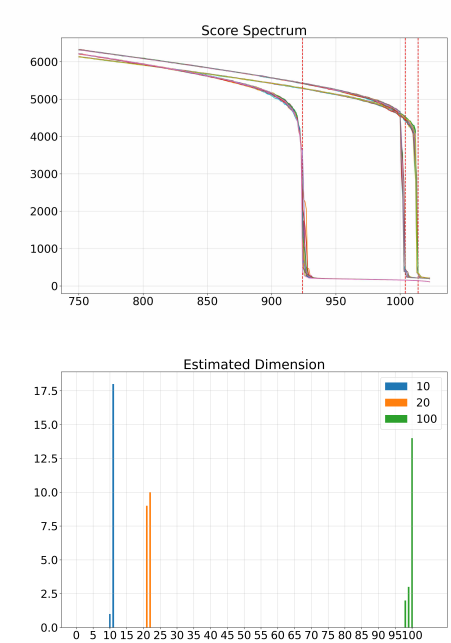


Fig. A.8 Score spectra and histogram of estimated dimension based on the score spectrum of the Squares image manifold of dimensions 10, 20 and 100.

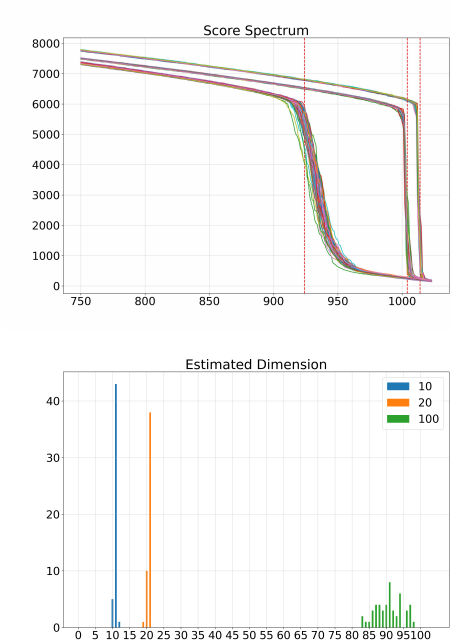


Fig. A.9 Score spectra and histogram of estimated dimension based on the score spectrum of the Gaussian blobs image manifold of dimensions 10, 20 and 100.

A.6.3 MNIST

In Figure A.10 we present 500 score spectra evaluated at 500 different instances for each digit. We observe that a considerable number of spectra indicate lower manifold dimension than the dimension documented in Table 1.1. This deviation can be attributed to amplified geometrical and statistical error at the respective evaluation points.

We choose the maximum estimated dimension as our conjecture of the intrinsic dimension under the premise that a collapse of the spectrum at a smaller dimension than the dimension of the normal space is extremely unlikely from a probabilistic standpoint. However, it is feasible to encounter a spectrum collapse suggesting a higher normal space dimension, hence a lower intrinsic dimension, due to the intensified geometric and approximation error at the point of evaluation. It is noteworthy that our estimation strategy locates the drop at the position of the maximal gradient, a practice that may marginally inflate the intrinsic dimension estimate, as exemplified in the Squares and Gaussian blobs manifolds. Therefore, if our reported

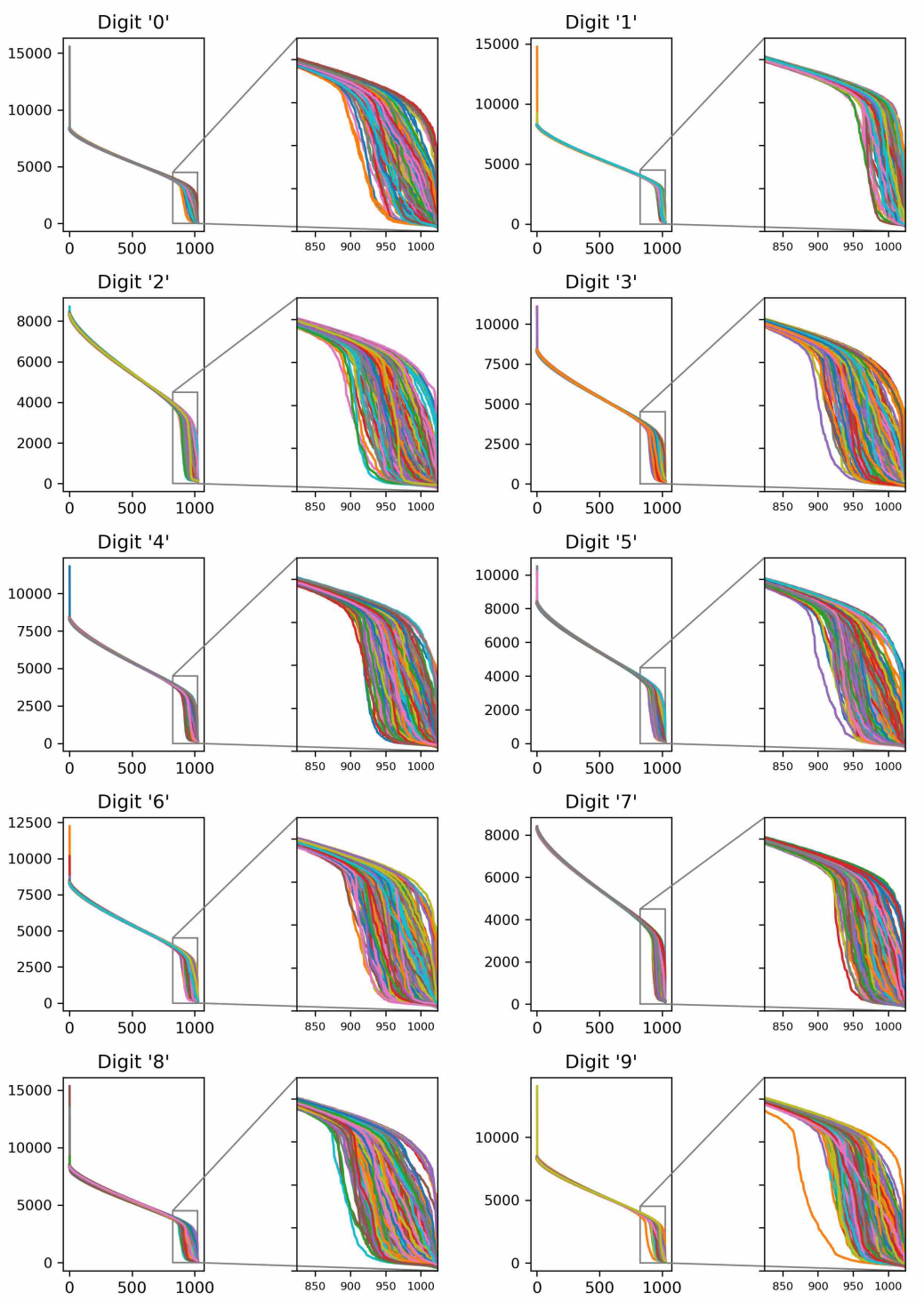


Fig. A.10 MNIST score spectra for all digits

dimension for each MNIST digit is not precise, it is either a minor overestimation or a lower limit of the true dimension.

A.7 Robustness analysis

A.7.1 Robustness to score approximation error

As we discussed in the previous sections, our method is guaranteed to work given a perfect score approximation for sufficiently small t. However, in practice there will be an approximation error resulting from finite training data, limited model capacity and imperfect optimization. Therefore, we conduct an empirical analysis of the influence of the error in score approximation on the produced estimate of the dimension. We train a model s_θ on a uniform distribution on 25-sphere and then we corrupt the output of the model with a Gaussian perturbation $e \sim \mathcal{N}(0, \sigma_e^2 \mathbf{I})$. Then, we produce score spectra by applying our method to the resulting corrupted scores. We repeat this experiment for different intensities of noise σ_e. We pick σ_e so that the noise norm to score norm ratio $r = \mathbb{E}[\|e\|]/\mathbb{E}_{x_{t_0} \sim p_{t_0}(\mathbf{x}_{t_0}|\mathbf{x}_0)}[\|s(\mathbf{x}_{t_0}, t_0)\|]$ has a prescribed value. We find that as we increase the intensity of noise singular values corresponding to the tangential component start to increase causing the gap in the score spectrum to diminish. This is expected since the noise added to the score vectors has a tangential component. However, for values of $r < 0.5$ our method is still producing a visible drop in the spectrum at the correct point. The results are presented in Figure A.11.

A.7.2 Robustness to non-uniform distribution on the manifold

	Uniform	$\alpha = 1$	$\alpha = 0.75$	$\alpha = 0.5$
Ground Truth	10	10	10	10
Ours	11	10	10	7
MLE (m=5)	9.61	5.37	4.83	4.12
MLE (m=20)	9.46	4.99	4.49	3.82
Local PCA	11	5	4	3
PPCA	11	11	11	11

Table A.2 Dimensionality detection for non-uniform distribution. For our method the maximum over pointwise estimates $\hat{k}(\mathbf{x}_0)$ is considered.

We examine the robustness to our method to non-unifomity in data distribution on the manifold surface. Under perfect score approximation and sufficiently small t_0 our method is guaranteed to work, but we conduct an empirical study to investigate the behaviour in the presence of score approximation error and $t_0 > 0$ used in practice in diffusion models. We consider a k-sphere and sample the surface of the sphere in a non-uniform fashion. We obtain points on the k-sphere by sampling vectors $\boldsymbol{\theta}$ of $k-1$ angles (in radians) from a

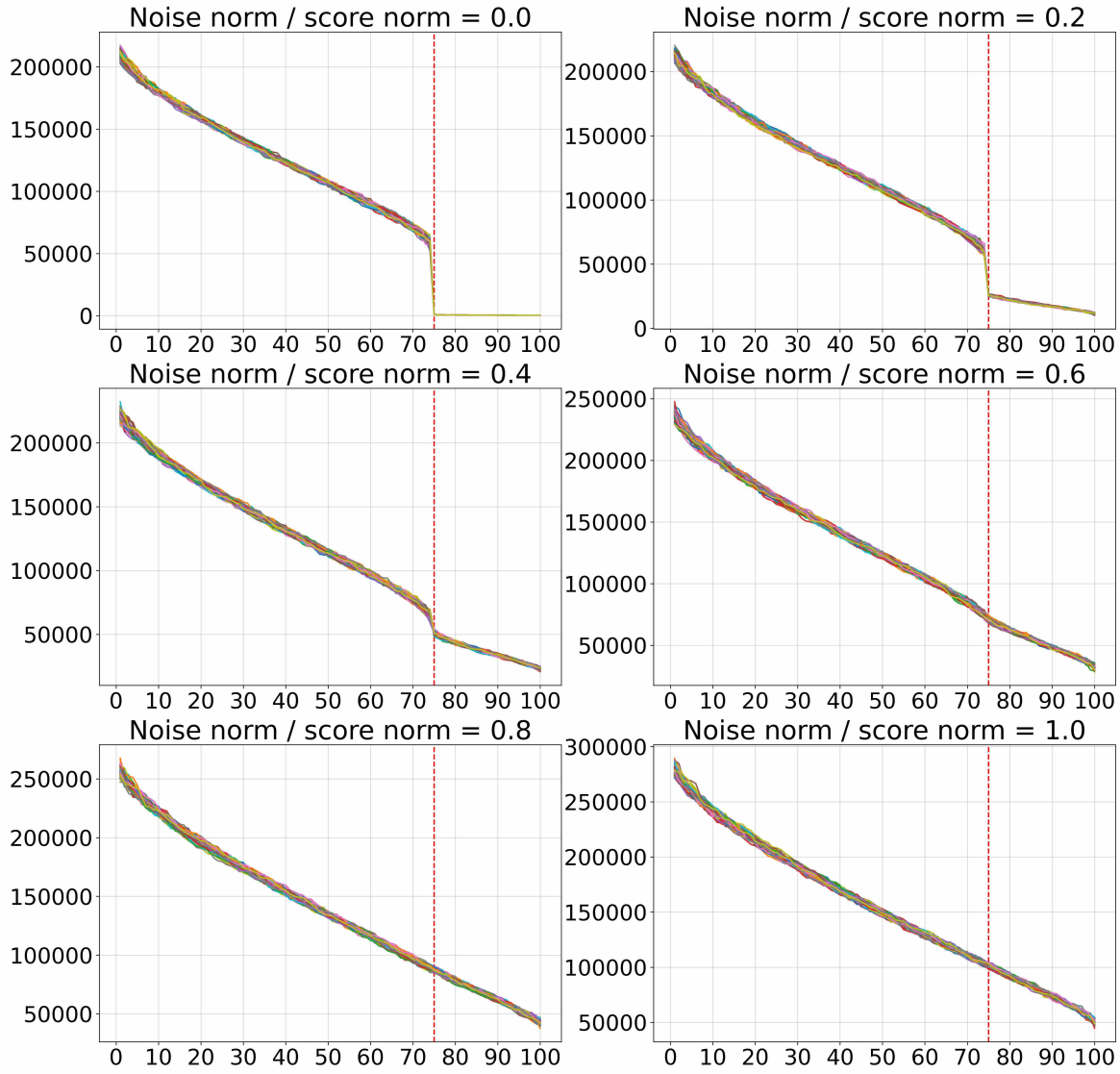


Fig. A.11 Score spectra for noise corrupted score model on 25-sphere.

Gaussian distribution $\mathcal{N}(0, \alpha \mathbf{I})$, where $\alpha \in \mathbb{R}$ is a constant that determines the degree of non-uniformity. Then, we embed the resulting points via a random isometry in a 100 dimensional ambient space. We sample $n = 1000$ points $\mathbf{x}_0^{(j)}$ from the manifold and at each point we estimate the dimensionality $\hat{k}(\mathbf{x}_0^{(j)})$ via the score spectrum. The pointwise estimates are presented in Figure A.12 and final estimates are shown in Table A.2. For values of $\alpha \in \{1, 0.75\}$, we can still obtain an accurate estimate of the dimension if we take $\hat{k} = \max_{j=1,\ldots,1000} \hat{k}(\mathbf{x}_0^{(j)})$ the maximum over point-wise estimates. For an extremely concentrated distribution with $\alpha = 0.5$ the method underestimates the dimension, which indicates that the tangential component of the score was not approximately constant in the neighborhoods used to sample the scores. This problem could be further mitigated by ap-

proximating the score closer to the manifold and using smaller sampling neighborhoods (i.e. for smaller t_0) or sampling more points $\mathbf{x}_0^{(j)}$. Notice that taking the maximum over $\hat{k}(\mathbf{x}_0^{(j)})$ is theoretically justified (as long as we assume we are dealing with a connected manifold) since our method can underestimate due to geometric or approximation error (cf. sections 1.4 and 1.5, but it is unlikely to significantly overestimate.

A.7.3 Relaxing the strict manifold assumption

The proof of the Theorem 1.4.1 assumes that p_0 is strictly contained on the data manifold $\mathscr{M}$, however in practice it is possible that the data distribution is concentrated around $\mathscr{M}$ rather then being contained within. Therefore, we conduct an empirical analysis, which examines how our method works in the case of the data contained around the manifold. We start with p_0 a uniform distribution on a unit 25-sphere embedded in a 100 dimensional space and convolve it with a Gaussian distribution $\mathscr{N}(0, \sigma\mathbf{I})$ to obtain a distribution p_0^σ that concentrates around $\mathscr{M}$. As σ increases the distribution is blurred out more and more into the ambient space. We train score model on each of the resulting distributions and use our method to estimate the dimension. We find that our method produces correct estimation for for small values of σ i.e. when p_0^σ is still concentrated tightly around $\mathscr{M}$. This is expected since of high values of σ the distribution isn't really concentrated around any manifold and therefore the notion of intrinsic dimension doesn't make any sense. The results are presented in Figure A.13.

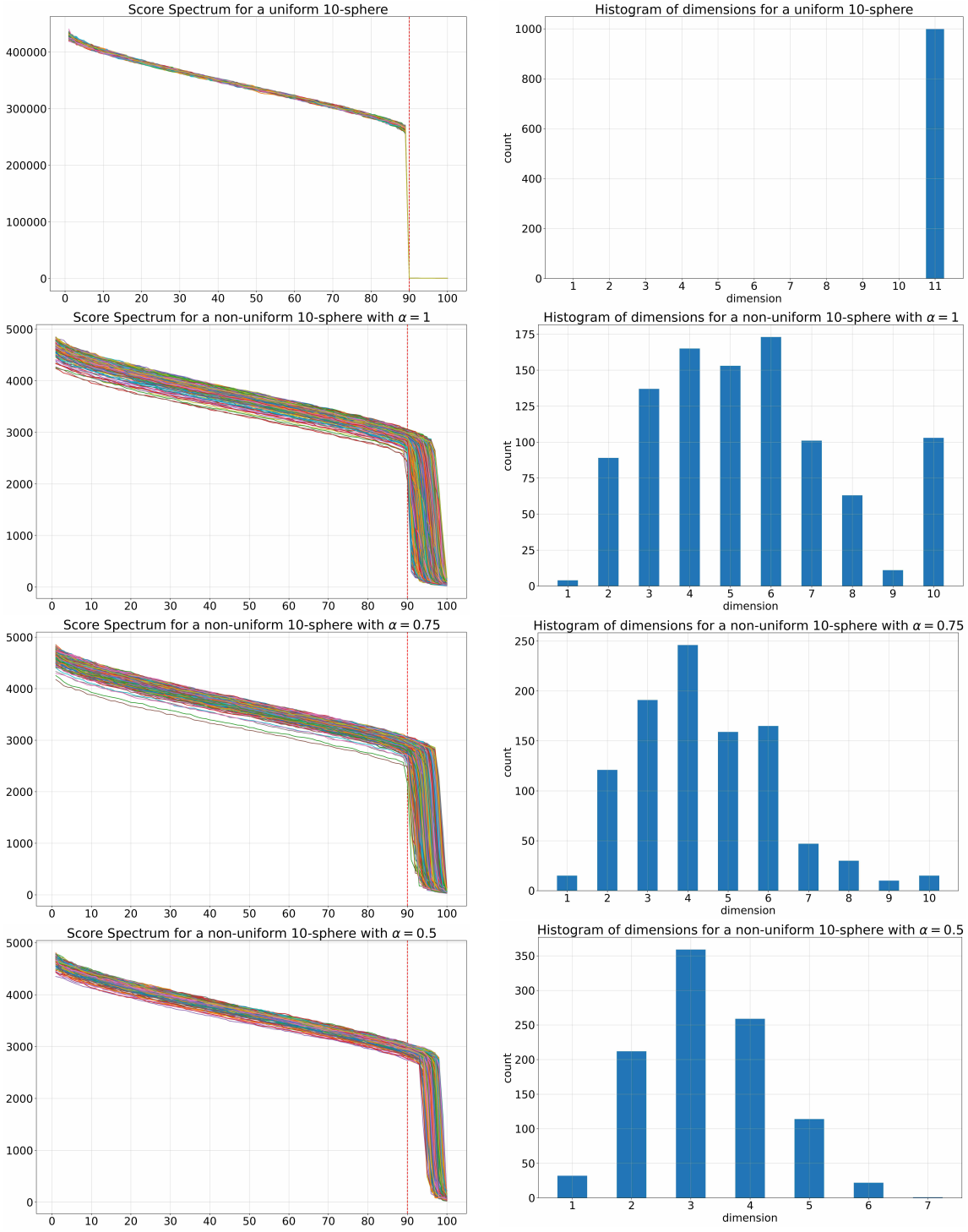


Fig. A.12 Dimensionality estimates for uniform and non-uniform distributions on a 10-sphere. On the right, we present histograms showing how many points $\mathbf{x}_0^{(j)}$ result in a given $\hat{k}(\mathbf{x}_0^{(j)})$. Taking $\hat{k} = \max_j \hat{k}(\mathbf{x}_0^{(j)})$ allows for robust estimation for moderate values of the concentration parameter α.

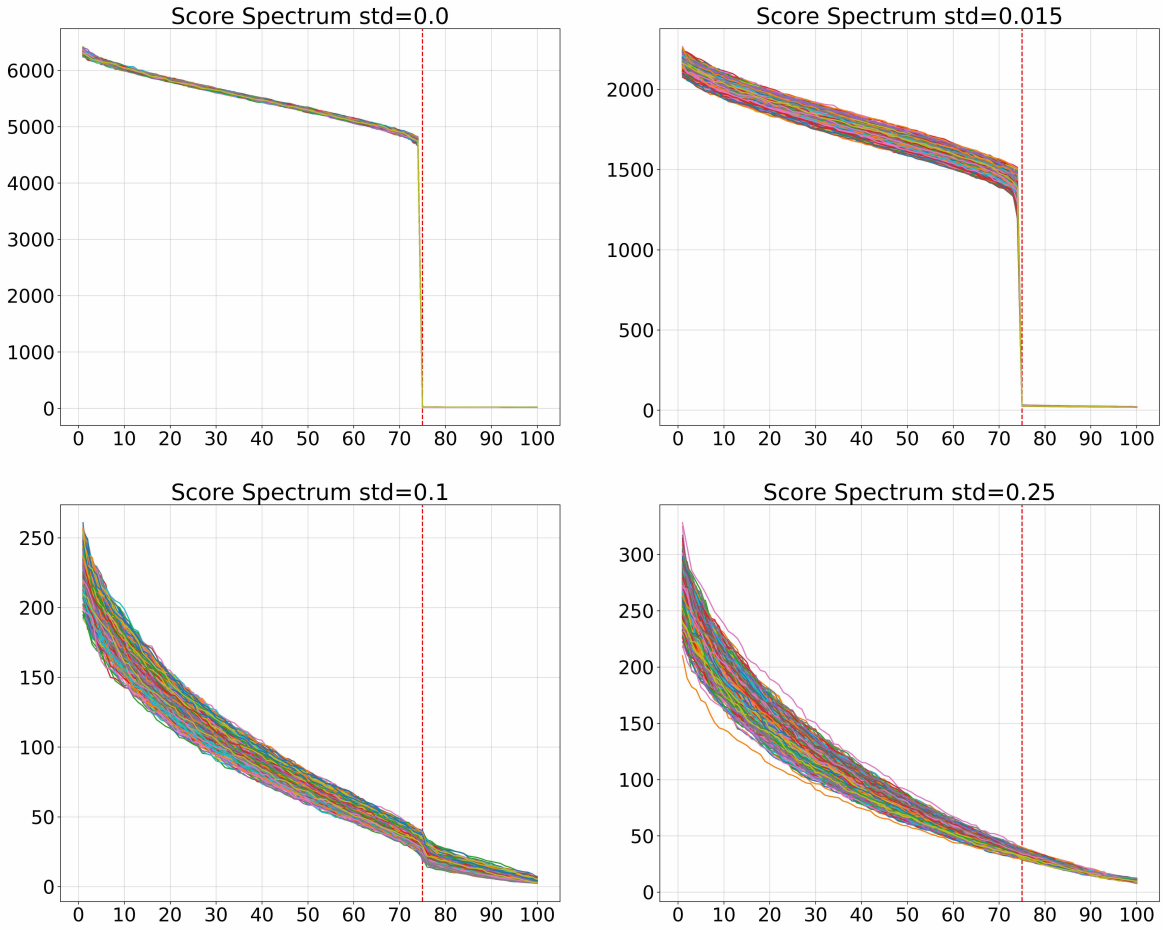


Fig. A.13 Score spectra for score models on 25-sphere trained on noisy manifold data.