Intrinsic dimensionality estimation using Normalizing Flows

Anonymous Author(s)
Affiliation
Address
email

Abstract

How many main invariances are there in a dataset consisting of M samples embedded in $\mathbb{R}^D$? This number, formally known as *intrinsic dimensionality*, can be estimated using nearest neighbor statistics. However, nearest neighbor statistics do not scale to large datasets as their complexity scales quadratically in M, $\mathcal{O}(M^2)$. Additionally, methods based on nearest neighbor statistics perform poorly on datasets embedded in high dimensions where $D \gg 1$. In this paper, we propose a novel method to estimate the intrinsic dimensionality using Normalizing Flows that scale to large datasets and high dimensions. Based on some back-of-the-envelope calculations, we predict how the eigenvalues of the flow's Jacobian evolve when inflating the dataset with different noise magnitudes. We test our method on various datasets, including 64x64 RGB images, where we achieve state-of-the-art results.

1 Introduction

Learning low-dimensional representations of high-dimensional data is becoming increasingly important in the era of big data. Therefore, representation learning is a very active area of research [33] with a wide range of applications ranging from neuroscience [28], molecular biology [29], bioinformatics [13] or image analysis [23]. But how many low-dimensional variables are exactly needed to accurately describe the original data? Unfortunetly, this number is typically not known and needs to be estimated - a task formally known as *intrinsic dimensionality* (ID) estimation.

Moreover, estimating the ID turned out to be an usefull tool for understanding the functioning of Neural Networks and generative models [25, 12, 1, 31]. In the context of generative models, the ID corresponds to the number of latent variables generating the data. Some of these generative models do even rely on knowing the exact number of latent variables generating the data [3, 17].

ID estimation is mathematically challenging. Most approaches rely on the assumption that, locally, the samples are uniformly distributed, and therefore the ID can be estimated using nearest neighbor statistics [4]. However, such nearest neighbor methods do not scale to large datasets consisting of $M \gg 1$ samples as the complexity scales with $\mathcal{O}(DM^2)$ where D is the dimensionality of the embedding space and accounts for calculating the (Euclidean) distance [1] . Thus, this scaling issue is amplified whenever the embedding space D is of the order of the sample size M, $D = \mathcal{O}(M)$. Additionally, nearest neighbor methods suffer from the curse of dimensionality in the sense that they perform badly in high dimensions $D \gg 1$ [5, 4, 32]. In this paper, we propose to overcome these scalability issues by estimating the ID using a standard NF which does scale to large datasets and dimensions.

Our high-level idea is very simple and based on some back-of-the-envelope calculations which predict how the flow's eigenvalues change depending on the amount of noise injected into the data

[1] Therefore, some line of research tries to reduce the complexity of finding the nearest neighbor, see e.g. [27].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

prior to training. For directions of small variability (i.e. off-manifold directions), the corresponding eigenvalues should decrease at a rate predicted by our theory. However, for directions of great variability (i.e. manifold directions), the corresponding eigenvalues will behave differently which allows us to estimate d. We show the soundness of our approach on various datasets, including RGB images of resolution 64×64.

2 Problem statement, background and notations

Here we briefly discuss the problem of estimating the ID, introduce the concept of an NF and additional notations which will be used throughout the paper.

Estimating ID: An $d-$dimensional manifold $\mathcal{M}$ is a set of points that are locally diffeomorphic to subsets of $\mathbb{R}^d$, see e.g. [24] for a formal definition of manifolds. Given M samples $x_1, \ldots, x_M$ from such a $d-$dimensional manifold embedded in $\mathbb{R}^D$, $d < D$, the task is to estimate d. More generally, if the dataset is living on unions of manifolds with different IDs, the task is to estimate d locally for a given point x [2] . For real-world datasets, however, samples are corrupted by noise making the dataset full-dimensional, i.e. $d = D$. In this case, estimating the ID corresponds to estimating how many main invariances there are in the observed dataset. We refer to those main invariances to directions of large variability in the following.

Standard Normalizing Flows: In the simplest case, an NF f_θ is a change of variable transforming a sample from the unknown distribution $p(x)$ to a sample from a standard Normal $\mathcal{N}(0, I_D)$. If $\mathcal{M}$ is full-dimensional, i.e. $d = D$, then the resulting density is given by $p_\theta(x) = |\det J_{f_\theta^{-1}}(u)|^{-1}\mathcal{N}(u; 0, I_D)$ where $u = f_\theta^{-1}(x)$, and the parameters θ are updated to minimize

$$D_{KL}(p(x)||p_\theta(x)) = -\mathbb{E}_{x\sim p(x)}[\log p_\theta(x)] + const. \tag{1}$$

For the general case where $d < D$, the general change of variable formula needs to be considered, see e.g. [19, 7, 11, 3]. From now on we omit the NF's dependence on the parameter θ and write f instead of f_θ.

Notations: We denote the data density with support on $\mathcal{M}$ as $p(x)$. Given a sample $x \sim p(x)$, if we add some Gaussian noise ε to it, the resulting random variable $\tilde{x} = x + \varepsilon$ has the following density

$$q_{\sigma^2}(\tilde{x}) = \int_{\mathcal{M}} \mathcal{N}(\tilde{x}; x, \sigma^2 I)p(x)dx \tag{2}$$

where I is the $D \times D$ unit matrix.

The Jacobian of a flow $f : \mathbb{R}^D \to \mathbb{R}^D$ has D eigenvalues, which we denote by $\lambda^{(1)}, \ldots, \lambda^{(D)}$.

3 Related work

Estimating the ID of a dataset is a well-studied problem with a vast literature, see [5, 4] for exhaustive overviews.

Nearest Neighbor methods: Historically, two main branches have developed: global and local methods [4]. Global methods estimate the dimensionality of the dataset globally with methods based on PCA is the most famous representative of this class, while local methods do estimate it locally. Consistent local estimators are based on nearest neighbor statistics. Intuitively, the volume of a $d-$dimensional ball scales as r^d with its radius r, and, therefore, neighbors within the $r-$ball of a point should behave similarly. This intuition was formalized by [14] and generalized in [15]. Recently, [9] derived the distribution of the distance to the second nearest neighbor - a power law with the ID as the exponent. We choose this method, coined twoNN, as a representative for nearest neighbor methods and compare its performance with our method in Section 5 .

Methods based on NN: Neural Networks (NN) scale to large datasets. Surprisingly, to the best of our knowledge, next to our method, there exists only one method which attempts to *directly* estimate the ID using an NN based method [32], coined LIDL (see below for a detailed description). However, there exist different *indirect* methods which use performance metrics to estimate the ID. The general

[2] Note, that a local estimator can be used as a global ID estimator by simply averaging over different samples.

idea is to train different models with different latent dimensionalities and compare their performance on a chosen metric, see e.g. [2, 34, 3]. Arguably, the model yielding the best result on this metric corresponds to the right latent dimensionality. However, the estimate will be relative to the chosen performance metric and therefore might not reflect the true ID. As an example of this relativity, we refer to Figure 14 in [3] which use their NF-based method to estimate the ID of the CelebA dataset [26]. They get different results when using the FID-score [16] or reconstruction error as performance metric.

In [30], it was argued that the Jacobian of the encoder $f : \mathbb{R}^D \to \mathbb{R}^d$ of an Autoencoder can give information of directions of contraction. Intuitively, manifold directions correspond to directions of low contraction while off-manifold directions correspond to directions of high contraction. The gist of this idea is very similar to our intuition which we will present Section 4 as directions of contraction can be directly related to the singular values of the encoder's Jacobian. However, the number of contractive directions will only give a lower bound on the ID.

LIDL: [32] propose to use NFs to estimate the ID. LIDL stands for locally intrinsic dimensionality likelihood and is based on the assumption that the inflated log-likelihood $\log q_{\sigma^2}(x)$ scales linearly in σ^2 with the normal space dimension as slope. More precisely, they argue that the inflated distribution $q_{\sigma^2}(x)$ can be written as the product of $p(x)$ and the noise distribution restricted on the normal space, i.e. $q_{\sigma^2}(x) = p(x)\mathcal{N}(x; x, I_{D-d})$ where $\mathcal{N}(x; x, I_{D-d})$ is a $(D-d)-$dimensional Gaussian in the normal space of x. This intuition is mathematically confirmed in [19] under certain conditions on the manifold and generating latent distribution. Thus, [32] estimate the rate at which the log-likelihoods change depending on the noise magnitude σ^2, and therefore train different models with different noise magnitudes $\bar{\sigma}_1^2 < \cdots < \bar{\sigma}_N^2$ to estimate that change.

LIDL heavily relies on the assumption that $q_{\sigma^2}(x) = p(x)\mathcal{N}(x; x, I_{D-d})$, which is generally only true for sufficiently smooth manifolds and latent distributions, see [19] for details. Even if these conditions are fulfilled, the outcome of LIDL depends on the noise magnitudes $\bar{\sigma}_1^2 < \cdots < \bar{\sigma}_N^2$ used to estimate the log-likelihood rate. However, the optimal range again depends on both the manifold and generating latent distribution which is hard to estimate a priori [19]. We will revisit these issues in the experiments, Section 5.

4 Method

How can we estimate the intrinsic dimensionality of data sampled from a manifold using an NF?

Simplest scenario: Assume the simple case where the target distribution $p(x)$ is a normal distribution, i.e. $p(x) = \mathcal{N}(\mu, \Sigma)$ with $\mu \in \mathbb{R}^D$ and

$$\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_D^2), \quad \sigma_i^2 > 0. \tag{3}$$

Thus, we assume that the data manifold is given by the entire embedding space, $\mathcal{M} = \mathbb{R}^D$. Note, however, that if $\sigma_i^2 \to 0$, $i = 1, \ldots, D-d$, this corresponds to samples from a $d-$dimensional manifold embedded in $\mathbb{R}^D$. Then, the true NF f transforming $x \sim \mathcal{N}(\mu, \Sigma)$ into $u = f^{-1}(x)$ distributed according to $\mathcal{N}(0, I_D)$ is given by

$$f_i^{-1}(x) = \frac{x_i - \mu_i}{\sigma_i}, \quad i = 1, \ldots, D, \tag{4}$$

with Jacobian

$$J_{f^{-1}}(x) = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_D}\right) \tag{5}$$

Therefore, an eigenvector of $J_{f^{-1}}(x)$ corresponding to a large eigenvalue $\lambda^{(i)}$ (i.e. small σ_i) is in direction of small variability in the data.

It is easy to see that the same holds for more general covariance matrices Σ. The true NF f transforming $x \sim \mathcal{N}(\mu, \Sigma)$ into $u = f^{-1}(x)$ distributed according to $\mathcal{N}(0, I_D)$ is given by

$$f^{-1}(x) = \Sigma^{-1}(x - \mu) \tag{6}$$

with Jacobian

$$J_{f^{-1}}(x) = \Sigma^{-1} = S^T D^{-1} S. \tag{7}$$

where $\Sigma = S^T D S$, $D = \text{diag}(\sigma_1^2, \dots, \sigma_D^2)$ and S is orthonormal and consists of the eigenvectors of Σ. Thus, the eigenvalues of $J_{f^{-1}}(x)$ are given by $D^{-1} = \text{diag}\left(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_D}\right)$ as in Equation (5).

From simple to the generic scenario: In general, the NF f^{-1} non-linearly transforms a sample $x \sim p(x)$ such that $u = f^{-1}(x)$ is distributed according to $\mathcal{N}(0, I_D)$. Hence, Equation (6) does not hold anymore. However, Equation (6) is a good first-order approximation as locally the flow acts linearly,

$$f^{-1}(x) \approx f^{-1}(x^*) + J_{f^{-1}}(x^*)(x - x^*). \tag{8}$$

Thus, the flow of arguments from the simplest scenario takes on and we note the following observation:

Observation 1: Large eigenvalues of the NF's Jacobian evaluated at x correspond to directions of small variability in the data. Directions of large variability, however, correspond to small eigenvalues.

How can we make use of this observation to estimate the ID of a data manifold? When training an NF on clean manifold data (i.e. samples from a manifold without intrinsic noise), Observation 1 predicts a clear change in magnitude between eigenvalues corresponding to off-manifold and on-manifold directions [3] . Indeed, we can observe such a change in magnitude in various toy examples, see Section 4.1. However, to estimate d based on this observation only would require setting an arbitrary threshold on the magnitude of those eigenvalues. On the other hand, in a more realistic scenario, real world data have some intrinsic noise.

Data with intrinsic noise: With the same setting as above, we now additionally assume that the intrinsic noise is coming from a standard Gaussian with magnitude σ_0, i.e. $p(x) \sim \mathcal{N}(\mu, \Sigma + \sigma_0^2 I)$. Then, Equation (5) turns into

$$J_{f^{-1}}(x) = \text{diag}\left(\frac{1}{\sqrt{\sigma_0^2 + \sigma_1^2}}, \dots, \frac{1}{\sqrt{\sigma_0^2 + \sigma_D^2}}\right). \tag{9}$$

Now, what happens if we inflate the data-manifold with Gaussian noise of variance σ^2? For directions of small variability in the data, σ_i^2 is small. Thus, only if σ^2 exceeds the intrinsic noise magnitude σ_0^2, we expect to see a change in $\lambda^{(i)}$. For directions of large variability, however, we expect to see a change in $\lambda^{(j)}$ only if σ^2 is greater than $\sigma_0^2 + \sigma_j^2$ as in this case σ_j^2 is not small. To account for the more general case where the intrinsic noise is not isotropic, we denote the intrinsic noise in the direction of the eigenvector corresponding to eigenvalue $\lambda^{(i)}$ as $\sigma_0^2(i)$ in the following.

Observation 2: If we were to inflate the noisy data-manifold with a Gaussian of variance σ^2 in direction of

(i) *small variability*, we expect the corresponding eigenvalue $\lambda^{(i)}$ to change significantly only if $\sigma^2 \gg \sigma_0^2(i)$.

(ii) *large variability*, we expect the corresponding eigenvalue $\lambda^{(i)}$ to change significantly only if $\sigma^2 \gg \sigma_0^2(i) + \sigma_i^2$.

Given this observation, we propose to train a separate NF for different noise magnitudes $\bar{\sigma}_1^2 < \dots < \bar{\sigma}_N^2$, calculate for all of those models the flow's eigenvalues on a sample x from the manifold, and, according to Equation (9), find parameters α_i and β_i such that $\hat{\lambda}(\sigma^2) = \frac{\beta_i}{\sqrt{{\alpha_i}^2 + \sigma^2}}$ fits the set of pairs $\{(\bar{\sigma}_n^2, \lambda_n^{(i)}(x))\}_{n=1,\dots,N}$. Note that β_i/α_i corresponds to $\lambda^{(i)}$ if no noise was used to inflate the manifold and that α_i^2 corresponds to the onset of strong decay.

Our method for estimating the ID: Given (α_i, β_i) for all $i = 1, \dots, D$, we expect to see a clear pattern separating eigenvalues corresponding to directions of small and large variability, respectively. According to Observation 1, β_i/α_i must be much larger for large variability directions and according to Observation 2, the onsets α_i^2 must be smaller compared to small variability directions. More concretely, the latter suggests that counting the number of eigenvalues with onset $\leq \alpha$,

$$F(\alpha) = \#\left\{i \in \{1, \dots, D\} \text{ s.t. } \alpha^{(i)} \leq \alpha\right\}, \tag{10}$$

[3] In fact, theoretically, eigenvalues corresponding to off-manifold directions need to become infinity for $d-$dimensional manifolds. Indeed, training an NF on clean manifold data creates numerical instabilities as observed in [19, 22, 6].

$F(\alpha)$ should have a plateau when varying α from $-\infty$ to ∞ corresponding to the number of eigenvalues in direction of small variability. Ideally, this plateau can be seen clearly when plotting $F(\alpha)$ and thus the number of off-manifold directions can be directly read out. Alternatively, a sum of two sigmoidal functions can be fitted to approximate F,

$$\hat{F}(\alpha) = \frac{a_1}{b_1 + e^{-(\alpha - c_1)}} + \frac{a_2}{b_2 + e^{-(\alpha - c_2)}}, \tag{11}$$

where $c_1 < c_2$ and hence a_1 corresponds to the number of small variability directions and a_2 to the number of large variability directions, see Algorithm 4 for an algorithmic overview of the method.

Algorithm 4: Estimating the intrinsic dimensionality given a set of data points.

Require: Dataset $\mathcal{D} = \{x_i\}$ with $x_i \in \mathbb{R}^D$. Samples $\{x_1^*, ..., x_K^*\} \subset \mathcal{D}$ to use for estimating the ID. Noise magnitudes $\bar{\sigma}_1^2 < ... < \bar{\sigma}_N^2$.

for $n = 1$ **to** N **do**
  $\rightarrow$ add noise to dataset: $\tilde{\mathcal{D}} = \{x_i + \varepsilon_i\}$, where $\varepsilon_i \sim \mathcal{N}(0, \bar{\sigma}_n^2 I_D)$
  $\rightarrow$ learn $q_{\bar{\sigma}_n^2}$ using an NF f_n
  $\rightarrow$ calculate and store eigenvalues $\{\lambda_n^{(i)}(x_k^*)\}_{i=1,\dots,D}$ of $J_{f_n^{-1}}(x_k^*)$, $k = 1, \dots, K$
end for
$\rightarrow$ calculate mean eigenvalues for all i and n, $\hat{\lambda}_n^{(i)} := \frac{1}{K}\sum_{k=1}^{K} \lambda_n^{(i)}(x_k^*)$
for $i = 1$ **to** D **do**
  $\rightarrow$ store (α_i, β_i) such that $\hat{\lambda}(\sigma^2) = \frac{\beta_i}{\sqrt{{\alpha_i}^2 + \sigma^2}}$ fits $\{(\bar{\sigma}_n^2, \hat{\lambda}_n^{(i)})\}_{n=1,\dots,N}$
end for
$\rightarrow$ calculate $F(\alpha) = \#\{i \text{ s.t. } \alpha_i \leq \alpha\}$ for $\alpha \in (-\infty, \infty)$
$\rightarrow$ fit $\hat{F}(\alpha)$ given by Equation (11) and store parameter a_2
$\rightarrow$ estimator is given by $\hat{d} = a_2$

4.1 Pedagogical example

We illustrate our method on a pedagogical example - a uniform distribution on a sphere corrupted by some intrinsic Gaussian noise with magnitude $\sigma_0^2 = 10^{-6}$. In Figure 1 left, we show how the three eigenvalues change in log-log-scale depending on various noise magnitudes. For the error bars, we use 200 clean training examples (i.e. not corrupted by noise). After the intrinsic noise magnitude is surpassed (first dashed line from the left), the large eigenvalue changes with slope ≈ -0.5 whereas the two smallest eigenvalues barely change. After the noise magnitude exceeds the radius of the sphere (second dashed line from the left), the eigenvalues approach each other. In the right figure, we show $F(\alpha)$ and its estimate $\hat{F}(\alpha)$. The title of the right figure shows the output of Algorithm 4, the parameter a_2 of $\hat{F}(\alpha)$.

4.2 Special case: Images

So far, our theory assumes unbounded data such that if the inflation noise σ^2 tends to infinity, $\sigma^2 \rightarrow \infty$, the eigenvalues of the flow's Jacobian tend to 0, $\lambda \rightarrow 0$. According to our theory, this decay happens much later for eigenvalues corresponding to manifold directions than for off-manifold directions manifesting as a plateau for the eigenvalue onsets α. According to Equation (9), those onsets reflect the amount of variability in the manifold directions.

However, some real world datasets are bounded. For example, RGB images consist of pixels which take values in $[0, 255]$. Thus, for $\sigma^2 \rightarrow \infty$, inflated images will become too noisy and all the information about the manifold is lost. Now, assume that $\sigma_{\max}^2$ is the maximal amount of noise that can be tolerated before all the images become so noisy that they become meaningless. Then, all onsets greaeter or equal to $\sigma_{\max}^2$ will correspond to manifold directions as the amount of variability in manifold directions must be greater than $\sigma_{\max}^2$.

How can we set this $\sigma_{\max}^2$ in the least arbitrary way? In the supplementary, we compute this maximal amount of noise from the following rationale. If σ^2 is too large, then essentially all the pixels will saturate to either the lower or the upper bounds, so a reasonable maximal amount of noise is such that 50% of the pixels are saturated. If the pixel values fall in the interval $[0, x_{\max}]$ (and if we assume that pixel values are uniformly distributed on this interval), then $\sigma_{\max} = 0.68 x_{\max}$ (see supplementary).

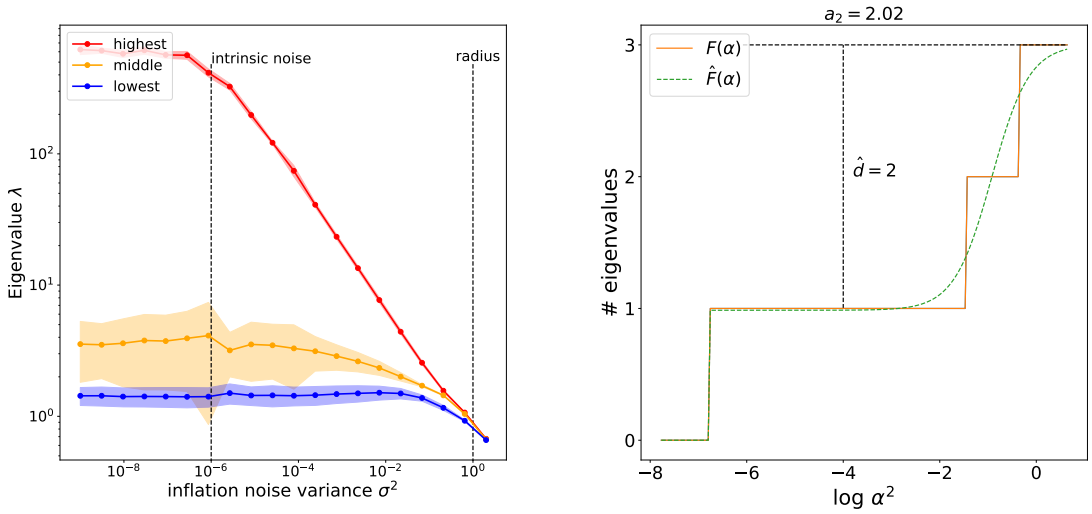

Figure 1: **Left:** λ vs. σ^2 plot as described in the main text. **Right:** $F(\alpha)$ and $\hat{F}(\alpha)$ as defined in Equations (10) and (11).

Remark 1

(i) Note, for $K = 1$, Algorithm 4 estimates the ID locally at a given point x_1^*.

(ii) There is no need to calculate $F(\alpha)$ for all $\alpha \in \mathbb{R}$ since $F(\alpha) \in \{0, 1, \dots, D\}$. Given α_i for all $i = 1, \dots, D$, we set $\alpha \in [\alpha_{min}, \alpha_{max}]$ where $\alpha_{min} = argmin_{i=1,\dots,D}\{\alpha_i\}$ and $\alpha_{max} = argmax_{i=1,\dots,D}\{\alpha_i\}$.

(iii) Our theory was derived for the flow's Jacobian eigenvalues. However, equivalent derivations apply for the flow's singular values as well. Indeed, for the experiments in Section 5.1 and 5.2, we used both the singular- and eigenvalues and did not find any significant difference. However, effectively, the singular values are more convenient to analyze as they are always positive and real. Therefore, we use the singular values of the flow's Jacobian for the experiments presentend in Section 5.

5 Experiments

We benchmark our method with twoNN and LIDL, see Section 3. The latter is very similar to our method as we both inflate the manifold with different values of σ^2 and then use an NF to learn q_{σ^2}. However, [32] estimate the rate at which the log-likelihoods change depending on the noise magnitude σ^2, whereas we study how the flow's Jacobian eigenvalues evolve. Hence, we rely on the NF being able to transform a sample of $p(x)$ to a standard Gaussian - a much easier task than learning the inflated distribution q_{σ^2} exactly. Also, we don't rely on the assumption that $q_{\sigma^2}(x) = p(x)\mathcal{N}(x; x, I_{D-d})$ and thus we don't need to fine-tune the different noise magnitudes $\bar{\sigma}_1^2, \dots, \bar{\sigma}_N^2$, see Section 3 for more details.

As an abbreviation for our method, we use ID-NF. We refer to the supplementary for corresponding training details and additional figures.

5.1 Low-dimensional synthetic datasets

We test our method on various synthetic datasets with known ID: a sphere, torus, hyperboloid, thin spiral, swiss roll, and Stiefel manifold, see Table 1. We use different distributions on those manifolds to test the sensitivity on the sampling distribution $p(x)$. We compare our method with the LIDL and

Distribution	D	ID	ID-NF	LIDL	twoNN
mixture on sphere	3	2	2.01	1.91±0.06	1.98
correlated on sphere	3	2	2.04	1.66±0.07	1.99
mixture on torus	3	2	2.02	2.05±0.04	1.97
correlated on torus	3	2	2.02	2.07±0.05	2.02
correlated on hyperboloid	3	2	2.02	2.01±0.07	1.99
unimodal on hyperboloid	3	2	2.01	1.92±0.1	1.96
exponential on thin spiral	2	1	1	1.08±0.06	1
mixture on swiss roll	3	2	2.02	2.26±0.03	1.98
correlated on swiss roll	3	2	2.02	2.48±0.03	1.94
mixture on stiefel	4	1	1.07	1.19±0.01	0.99

Table 1: Performance of different ID estimation methods (ID-NF, LIDL, twoNN), on various datasets.

twoNN estimator introduced in Section 3. For those low-dimensional examples, all methods perform well. The twoNN and our method estimate all ID exactly. The LIDL method, however, shows some sensitivity towards the sampling distribution (see the highlighted estimate for the correlated on sphere distribution), and slightly overestimates the swiss roll dimensionality (see the highlighted estimate for the swiss roll). However, we did not try to find the optimal range for σ^2 and use the same across all distributions. By this, we want to demonstrate the disadvantage of having to estimate the correct range for every distribution separately.

5.2 High-dimensional synthetic datasets

Next, we study how the methods scale to higher embedding dimensions. For that, we sample uniformly from $\mathcal{S}(D/2)$ embedded in $\mathbb{R}^D$ for different even values of D. For a fair a comparison, we only use a training set of size 10^4 for all methods. In Figure 2 left, we display how the different methods estimate the dimensionality. ID-NF is on par with LIDL and outperforms twoNN.

In Figure 2 right, we repeat the experiment using only 10^3 samples. ID-NF still performs very well. LIDL, however, has greater variability and significantly underestimates the ID for $d = 200$ (i.e. $D = 400$). This indicates another benefit of our method: as opposed to LIDL, we do not need to learn the density exactly, a task that requires more samples for higher dimensions.

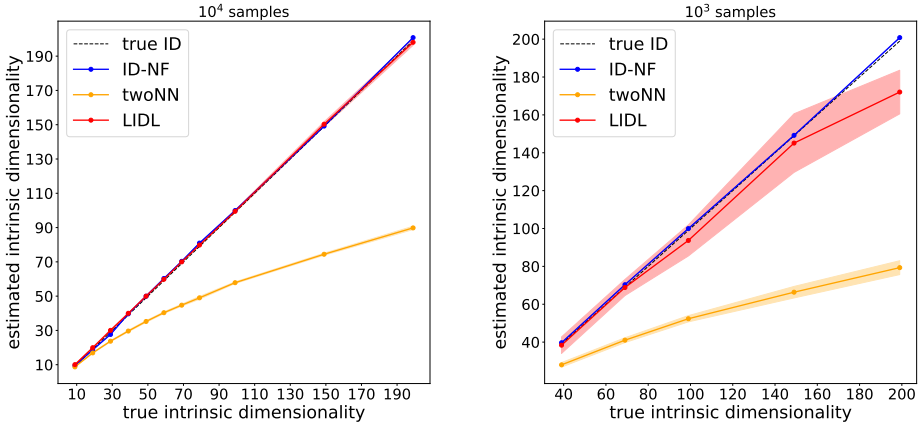

Figure 2: **Left:** Performance of different ID-estimation methods on uniform samples from $\mathcal{S}(D/2)$ embedded in $\mathbb{R}^D$ for D ranging from 20 to 400. **Right:** Corresponding plot using 10^3 instead of 10^4 samples for traning.

5.3 StyleGan image manifolds

In [3], a $d-$dimensional image manifold was created by using a recent variant of a generative adversarial network, StyleGan2 [21], trained on a high-quality image dataset, the FFHQ dataset [20]. When generating new images, only d of total 512 latent variables were varied while keeping the others fixed. This generates a $d-$dimensional image manifold. We downsample those images to a resolution of $64 \times 64 \times 3$, i.e. the embedding space has dimension $D = 12288$.

We apply our method on the StyleGan $d = 2$ and $d = 64$ image manifold consisting of 10^4 and $2 \cdot 10^4$ images, respectively. In Figure 3, we show the evolution of singular values depending on different noise magnitudes σ^2 in log-log-scale for $d = 2$. The dashed vertical line is located at $\sigma_{\max} = 255 \cdot 0.68$ corresponding to the maximal amount of noise that can be tolerated before all the images become so noisy that they become meaningless as described in Section 4.2 . As we can see, the smallest two eigenvalues, highlighted with red markers, are nicely separated.

In Figure 4, we show the heights and onsets for all singular value curves for $d = 2$ on the left, and $d = 64$ on the right. We highlight the points corresponding to the 2 smallest and 64 smallest singular values, respectively, in red. The dashed vertical line is again located at $\sigma_{\max} = 255 \cdot 0.68$. For $d = 2$, we count $\hat{d} = 2$ singular values which have onsets greater than $\sigma^2_{\max}$. For $d = 64$, we count $\hat{d} = 53$.

We also estimated the ID using LIDL, however, we did not get a consistent estimator. In fact, depending on the range one uses for the inflation noise σ^2, the estimate varies greatly.

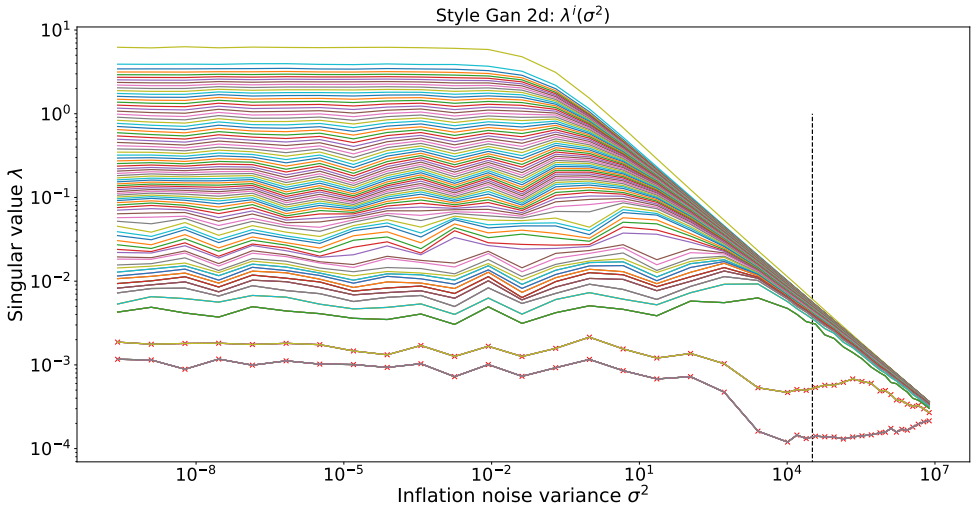

Figure 3: λ vs. σ^2 plot in log-log-scale for StyleGan 2d image manifold. We show only 1% of all possible eigenvalues including the 10 smallest eigenvalues. The two smallest eigenvalues are plotted dashed and highlighted in red.

6 Discussion

We have introduced ID-NF, a new method to estimate the ID exploiting the ability of NFs to transform data into samples from a Gaussian random variable. Based on some simple back-of-the-envelope calculations, we derived how the eigenvalues of the flows Jacobian evolve when inflating the data with Gaussian noise before training. Crucially for our estimator, eigenvalues corresponding to the directions of large variability (i.e. manifold directions) evolve significantly different compared to eigenvalues corresponding to the directions of small variability (i.e. off-manifold directions).

We demonstrated that we can estimate the ID for different manifolds with different sampling distribution. We compared our method to a state-of-the-art ID estimator based on nearest neighbor statistics,

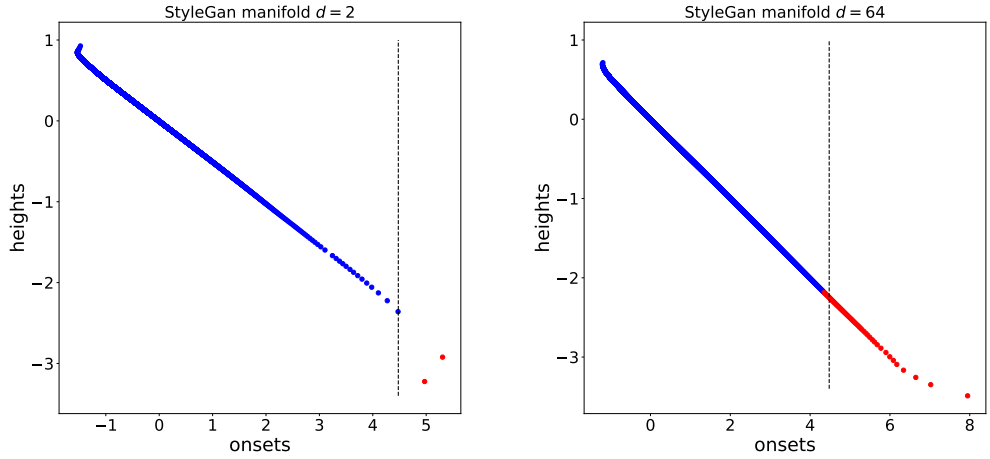

Figure 4: Scatter plots of the onsets and heights, $(\alpha, \beta/\alpha)$, for all singular values in log-log scale. In red, the d smallest singular values are highlighted. The dashed line corresponds to the $\sigma^2_{\max}$ as described in Section 4.2, i.e. $\sigma_{\max} = 0.68255$.

twoNN, and to a related method which is also based on NFs, LIDL. We outperform twoNN for high dimensions, Table 1, and LIDL for small data regimes, Figure 2 and images, Section 5.3. As opposed to LIDL, we don't have to fine-tune the noise magnitudes $\bar{\sigma}_1^2, \ldots, \bar{\sigma}_N^2$. However, we need $\bar{\sigma}_1$ to be small and $\bar{\sigma}_N$ to be sufficiently large which may be computationally expensive - especially when applying our method to images.

We showed that our method scales to RGB images of resolution 64×64 populating a $d = 2$ and $d = 64$ dimensional manifold. For $d = 2$ our estimate is exact, and for $d = 64$ we estimate 53. To the best of our knowledge, this is the first method which estimates the ID on such high-resolution image manifolds consistently.

Finally, our theoretical derivations rely on the assumption that data is unbounded, although we demonstrated that our approach can be adapted for bounded data too. However, for this case more research is needed to understand the exact behaviour of the flow's Jacobian eigenvalues/ singular values when approaching the boundary. An interesting direction to pursue is to adapt the target distribution and inflation noise to the data topology. For images, for example, this amounts to changing the NF's target distribution to be uniform instead of Gaussian, and using uniform instead of Gaussian noise.

Broader Impact: As a theoretical paper, we don't foresee any direct negative social impacts of our work.

References

[1] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[2] Nitish Bahadur and Randy Paffenroth. Dimension estimation using autoencoders. *arXiv preprint arXiv:1909.10702*, 2019.

[3] Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. *Advances in Neural Information Processing Systems*, 33, 2020.

[4] Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.*, 328:26–41, 2016.

[5] Paola Campadelli, Elena Casiraghi, Claudio Ceruti, and Alessandro Rozza. Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015, 2015.

[6] Eike Cramer, Felix Rauh, Alexander Mitsos, Raúl Tempone, and Manuel Dahmen. Nonlinear isometric manifold learning for injective normalizing flows. *arXiv preprint arXiv:2203.03934*, 2022.

[7] Edmond Cunningham, Renos Zabounidis, Abhinav Agrawal, Ina Fiterau, and Daniel Sheldon. Normalizing flows across dimensions. *arXiv preprint arXiv:2006.13070*, 2020.

[8] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[9] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):1–8, 2017.

[10] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[11] Mevlana C Gemici, Danilo Rezende, and Shakir Mohamed. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

[12] Sixue Gong, Vishnu Naresh Boddeti, and Anil K. Jain. On the intrinsic dimensionality of image representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] Daniele Granata and Vincenzo Carnevale. Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets. *Scientific reports*, 6(1):1–12, 2016.

[14] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9:189–208, 1983.

[15] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in rd. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296, 2005.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6629–6640, 2017.

[17] Christian Horvat and Jean-Pascal Pfister. Denoising Normalizing Flow. In *Advances in Neural Information Processing Systems*, volume 34, pages 9099–9111. Curran Associates, Inc., 2021.

[18] Christian Horvat and Jean-Pascal Pfister. Density estimation: an inflation-deflation approach, 2021.

[19] Christian Horvat and Jean-Pascal Pfister. Density estimation on low-dimensional manifolds: an inflation-deflation approach. *arXiv:2105.12152 [cs, stat]*, February 2022. arXiv: 2105.12152.

[20] T Karras, S Laine, and T Aila. A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116. IEEE, 2020.

[22] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33, 2020.

[23] N. Krueger and M. Felsberg. A continuous formulation of intrinsic dimension. In *Proc. BMVC*, pages 27.1–27.10, 2003. doi:10.5244/C.17.27.

[24] J.M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019.

[25] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *arXiv preprint arXiv:1809.03113*, 2018.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[27] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

[28] Rich Pang, Benjamin J Lansdell, and Adrienne L Fairhall. Dimensionality reduction in neuroscience. *Current Biology*, 26(14):R656–R660, 2016.

[29] Stefano Piana and Alessandro Laio. Advillin folding takes place on a hypersurface of small dimensionality. *Phys. Rev. Lett.*, 101:208101, Nov 2008.

[30] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. page 8.

[31] Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018.

[32] Piotr Tempczyk, Przemysław Spurek, and Jacek Tabor. LIDL: LOCAL INTRINSIC DIMENSION ESTIMATION USING APPROXIMATE LIKELIHOOD. page 10, 2021.

[33] S Velliangiri, S Alagumuthukrishnan, et al. A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science*, 165:104–111, 2019.

[34] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomput.*, 184(C):232–242, apr 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 5.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] But it is planned.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1.

(d) Did you include the total amount of comute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]