University of Warsaw
Faculty of Mathematics, Informatics, and Mechanics

Piotr Tempczyk

Estimating local intrinsic dimension via density estimation

PhD thesis
in Computer Science

Supervisor:
Marek Cygan
Institute of Informatics
University of Warsaw

Warsaw, September 2025

Supervisor's Statement

I confirm that the presented thesis was prepared under my supervision and that it fulfills the requirements for the doctoral degree in the field of Natural Sciences, in the discipline of Computer Science.

Date Supervisor's Signature

Author's Statement

I declare that the presented thesis was prepared by me and that none of its content was obtained through unlawful means.

The thesis has never before been subject to any procedure for obtaining an academic degree. Furthermore, I declare that the presented version of the thesis is identical to the attached electronic version.

Date Author's Signature

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie stopnia doktora w dziedzinie nauk ścisłych i przyrodniczych w dyscyplinie informatyka.

Data Podpis kierującego pracą

Oświadczenie autora pracy

Oświadczam, że niniejsza rozprawa doktorska została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem stopnia doktora w innej jednostce. Niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Abstract

This dissertation investigates *local intrinsic dimension* (LID) as a principled, pointwise, and scale-aware measure of the effective degrees of freedom in high-dimensional data. Motivated by the Manifold Hypothesis—that observations concentrate near lower-dimensional geometric structures—we argue that "dimension" is best treated as a property of neighborhoods rather than entire datasets. The central problem addressed here is how to estimate this local dimensional structure robustly and efficiently from finite, noisy samples.

The first contribution is *LIDL*, a neural estimator of LID. LIDL perturbs data with isotropic Gaussian noise of variance δ^2, fits a global density model (a normalizing flow) to the perturbed distribution, and reads off LID from how the log-density at a query point changes with $\log \delta$. In an intermediate noise regime this dependence is approximately linear with slope $d - D$, enabling a simple regression-based readout of the local dimension d. The method exposes an explicit *scale dial*, trades brittle kNN statistics for modern density estimation, and scales to high-dimensional settings. We document accuracy on controlled synthetic manifolds and plausible behavior on image data, together with known sensitivities (choice of noise window, density calibration, boundary effects).

Second, we unify recent neural LID estimators under a *Wiener-process* (diffusion) perspective. Adding controlled Gaussian noise corresponds to short-time diffusion of the data distribution; diverse methods then read the same signal in different coordinates: likelihood slopes (LIDL/FLIPD), Jacobian spectra of learned flows (ID-NF/ID-DM), or the geometry of score fields (NB). Within this lens we (i) introduce a taxonomy of *isolated* versus *holistic* estimators depending on whether they rely only on local behavior of the diffused density or on a global transport to a reference; (ii) analyze canonical diffusion cases for lower-dimensional manifolds with non-uniform densities; and (iii) derive closed-form corrections that explain empirical biases away from flat, uniform ideals and clarify when methods agree or diverge.

Third, we propose a *benchmarking framework* that stress-tests LID estimators on traits that routinely cause failure: non-uniform sampling, curvature, boundaries, thin structures, component proximity, and sample-size effects. To bridge analytic toys and real domains, we introduce domain-preserving transformations (Inverse Domain Representation, Monotonic Embedding, Ambient Space Extension, Auxiliary Dimension Injection, and Manifold Synthesis) that carry known-LID manifolds into complex ambient spaces. A side-by-side evaluation of representative classical and neural methods (e.g., ESS, LIDL, NB, FLIPD) reveals complementary strengths, exposes architecture and scale sensitivities, and yields practical guidance for method selection.

Keywords

Local Intrinsic Dimension (LID); Manifold Hypothesis; Neural LID estimation; Normalizing flows; Diffusion/score-based models; Noise perturbation (Wiener process);LID benchmarking framework

Tytuł pracy w języku polskim

Estymacja lokalnej wymiarowości rozmaitości danych za pomocą modeli estymacji gęstości

Streszczenie w języku polskim

Niniejsza rozprawa bada *lokalny wymiar wewnętrzny* (LID) jako zasadniczą, punktową i wrażliwą na skalę miarę efektywnych stopni swobody w danych wysokowymiarowych. Wychodząc od Hipotezy rozmaitości — że obserwacje koncentrują się w pobliżu niżejwymiarowych struktur geometrycznych — argumentujemy, iż „wymiar" najlepiej traktować jako własność *sąsiedztw*, a nie całych zbiorów danych. Głównym problemem podejmowanym w pracy jest to, jak niezawodnie i wydajnie estymować tę lokalną strukturę wymiarową ze skończonych, zaszumionych próbek.

Pierwszym wkładem jest *LIDL*, estymator LID uzywający estymatorów gęstości w postaci sieci neuronowych. LIDL zaburza dane izotropowym szumem Gaussa o wariancji δ^2, dopasowuje globalny model gęstości (normalizing flow) do zaburzonego rozkładu i odczytuje LID z tego, jak logarytm gęstości w punkcie zapytania zmienia się wraz z $\log \delta$. W niskim reżimie poziomu szumu zależność ta jest w przybliżeniu liniowa o nachyleniu $d - D$, co umożliwia prosty, regresyjny odczyt lokalnego wymiaru d. Metoda zapewnia jawną *regulację skali*, zastępuje statystyki oparte na najbliższych sąsiadach nowoczesnym modelowaniem gęstości i skaluje się do danych wysokowymiarowych. Pokazujemy jej dokładność na syntetycznych rozmaitościach oraz wiarygodne zachowanie na zbiorach obrazów, wraz ze znanymi słabościami (dobór okna szumu, kalibracja gęstości).

Po drugie, znajdujemy punkt wspólny dla współczesnych estymatorów LID opartych o modele gęstości uzywającye sieci neuronowych w postaci *procesu Wienera* (dyfuzji). Dodanie kontrolowanego szumu Gaussa odpowiada krótkoczasowej dyfuzji rozkładu danych; różne metody odczytują następnie ten sam sygnał w odmienny sposób: nachylenia logarytmu wiarygodności (LIDL/FLIPD), widma Jacobianu transformacji z modeli normalizing flow (ID-NF/ID-DM) lub geometrię score function (NB). W tej perspektywie (i) wprowadzamy taksonomię estymatorów *izolowanych* kontra *holistycznych*, zależnie od tego, czy opierają się wyłącznie na lokalnym zachowaniu dyfundowanej gęstości, czy na globalnym transporcie do rozkładu odniesienia; (ii) analizujemy kanoniczne przypadki dyfuzji dla niżejwymiarowych rozmaitości o niejednorodnych gęstościach; oraz (iii) wyprowadzamy poprawki w postaci zamkniętej, które wyjaśniają empiryczne odchylenia od płaskich rozmaitości i jednorodnych gęstości prawdopodobieństwa i pokazują analityczną formułę na błąd estymatora LID dla róznych gęstości prawdopodobieństwa.

Po trzecie, proponujemy *porównawcze zbiory danych*, które intensywnie testują estymatory LID pod kątem cech prowadzących do błędów: niejednorodnego próbkowania, krzywizny rozmaitości, brzegów rozkładów, cienkich struktur, bliskości komponentów oraz efektów liczebności próby. Aby zbudować pomost między „analitycznymi syntetycznymi zbiorami danych" a realnymi domenami, wprowadzamy przekształcenia zachowujące strukturę domeny (*Inverse Domain Representation*, *Monotonic Embedding*, *Ambient Space Extension*, *Auxiliary Dimension Injection* oraz *Manifold Synthesis*), które przenoszą rozmaitości o znanym LID do złożonych przestrzeni domenowych. Równoległa ocena reprezentatywnych metod klasycznych i używających estymatorów gęstości (ESS, LIDL, NB, FLIPD) ujawnia mocne i słabe strony każdej z nich.

Słowa kluczowe

Lokalny wymiar wewnętrzny (LID); Hipoteza rozmaitości; Estymacja LID; normalizing flow; Modele dyfuzyjne; Zaburzanie szumem (proces Wienera);

Contents

Chapter 1

Introduction

The purpose of this chapter is to provide the context and motivation for the research presented in this dissertation. It introduces the main problem under investigation, outlines the historical background of the field, highlights the original contributions of this work, and discusses related studies. Together, these elements set the stage for the more detailed chapters that follow.

This chapter is organized as follows. Sec. 1.1 defines the problem and explains why it is of significance. Sec. 1.2 provides a brief overview of the history of the field, tracing the key developments that have shaped current research. Sec. 1.3 presents the specific contributions of this dissertation and situates them within the broader scientific context. Sec. 1.4 sheds more light on the related work, and finally, Sec. 1.5 is the roadmap for this dissertation.

1.1 The Problem

> "The hypothesis that high dimensional data tend to lie in the vicinity of a low dimensional manifold is the basis of manifold learning." [1]

Under this *Manifold Hypothesis*, high-dimensional observations are assumed to concentrate near lower-dimensional geometric structures embedded in an ambient space. In practice, the effective dimensionality is not a single global constant: it varies with location and observational scale, and is influenced by curvature, boundaries, noise, and heterogeneous sampling. This motivates a *local*, scale-aware view of data geometry in which dimension is treated as a property of neighborhoods rather than entire datasets.

In this thesis we adopt precisely that perspective. We study *local intrinsic dimension* (LID) as a pointwise, scale-dependent measure of the effective degrees of freedom exhibited by data in the vicinity of a query point. The central problem is to formulate principled, robust, and computationally feasible ways to assess this local dimensional structure where the Manifold Hypothesis provides an adequate description of real, noisy, finite-sample data, and furnishing a diagnostic for model design and representation choices.

What is a data manifold? Intuitively, a *data manifold* is a subset $S \subset \mathbb{R}^D$ that, when viewed sufficiently close to any point $x \in S$, behaves like a flat d-dimensional plane: locally, the ambient space splits into directions that *move along* the set (tangent) and directions that *move away* from it (normal). Formally, for each $x \in S$ there

[1] This sentence opens the abstract of Fefferman et al. [2016].

is a decomposition $T_x\mathbb{R}^D = T_xS \oplus N_xS$ into tangent and normal spaces; in a small neighborhood, S can be represented as the graph of a smooth map over its tangent plane, so up to second order it "looks like" T_xS.

In this view, the observed data are draws from a smooth probability distribution p_S supported on S, i.e., the *manifold view of data*: high-dimensional observations concentrate near sets that are locally d-dimensional, with d depending on position and scale.

It is often convenient to consider the flat-manifold model where the ambient space factors as $\mathbb{R}^D = \mathbb{R}^d \times \mathbb{R}^{D-d}$ with coordinates (x, y), and S aligns with the first factor.

Below are simple examples of manifolds together with their dimensions:

- **A line in $\mathbb{R}^2$ (ambient $D = 2$, intrinsic $d = 1$).** An affine line can be written parametrically as
$$S = \{(t,\, mt + b) :\ t \in \mathbb{R}\} \subset \mathbb{R}^2,$$
so locally it is exactly flat (a 1D manifold sitting in the plane).

- **A spiral in $\mathbb{R}^2$ (ambient $D = 2$, intrinsic $d = 1$).** A smooth, non-self-intersecting example is the Archimedean spiral with increasing radius:
$$S = \big\{((a + bt)\cos t,\ (a + bt)\sin t) :\ t > -a/b\big\}, \quad a, b > 0.$$
It is a 1D curve in the plane; restricting to the open interval $t > -a/b$ avoids a boundary/cusp at the origin.

- **A paraboloid in $\mathbb{R}^3$ (ambient $D = 3$, intrinsic $d = 2$).** As the graph of a smooth function,
$$S = \{(x, y,\ x^2 + y^2) :\ (x, y) \in \mathbb{R}^2\},$$
this is a 2D surface embedded in 3D space; every small patch is well-approximated by a plane.

- **A d-sphere in $\mathbb{R}^{d+1}$ (ambient $D = d + 1$, intrinsic d).** The unit sphere
$$S_R^d = \{\, x \in \mathbb{R}^{d+1} :\ \|x\| = 1 \,\}$$
is a compact, boundaryless d-dimensional manifold. For $d = 1$ this is a circle; for $d = 2$ a usual sphere in $\mathbb{R}^3$.

Under the manifold view, "dimension" is local and geometric: d is the number of directions that stay on S when we move an infinitesimal step, while movements in the remaining $D - d$ directions leave the set. This picture—tangent vs. normal directions, smooth local charts, unions of pieces, boundaries, and variable thickness—will guide the notions and experiments used throughout this dissertation.

Applications of Local Intrinsic Dimension

Understanding whether data vary along a handful of meaningful directions—and where this number changes—turns an abstract notion of "complexity" into an observable. Local intrinsic dimension (LID) provides precisely such a dial: it summarizes how many degrees of freedom are active in a tiny neighborhood of the point. This perspective connects geometric structure to practical choices in modeling: how big a latent space to use, when a representation has become too entangled, or whether a classifier is operating in an easy or hard region. It also anchors intuitions about learning dynamics: as models train, the

effective dimension of the internal representations of neural networks often contracts or reorganizes, and those shifts may correlate with generalization. Below we outline key areas where LID is used or recommended in the literature cited in our introductions, expanding each with brief context and examples.

Representation learning and network training dynamics. LID has been used as a probe of how deep networks organize information during training. Tracking LID layer–by–layer reveals that internal representations often collapse onto lower–dimensional structures as training proceeds, and that this contraction is not uniform across layers or classes [Ansuini et al., 2019, Li et al., 2018]. Such trends help diagnose under/over–parameterization and can guide architectural or regularization choices (e.g., deciding where to bottleneck or where features remain unnecessarily high–dimensional) [Ansuini et al., 2019, Li et al., 2018, Pope et al., 2020].

Dimensionality reduction. When reducing dimension, a good target dimensionality is crucial. LID estimates inform how aggressively one can compress without destroying neighborhood geometry, and where heterogeneity (varying LID across regions/classes) argues against a single global target. Prior work highlights using ID/LID to set dimensionality reduction targets [Vapnik, 2013, Kleindessner and Luxburg, 2015, Camastra and Staiano, 2016, Loaiza-Ganem et al., 2024].

Manifold learning and density estimation. Classical and neural manifold/density estimators implicitly assume a dimension; LID offers a data–driven way to set or validate that choice. In invertible or score–based models, changes in local density/score rank across noise scales relate to the underlying manifold dimension, and LID helps indicate regimes where density models are faithful versus where curvature/thickness induce bias [Brehmer and Cranmer, 2020, Caterini et al., 2021, Ross and Cresswell, 2021].

Latent space sizing in generative autoencoders (e.g., VAE). Autoencoders require selecting a latent dimensionality; LID provides an interpretable prior for that hyperparameter. Empirically, mis–specifying the latent size (too small or too large relative to data dimension) degrades reconstruction and sample quality [Rubenstein et al., 2018]. In practice, aligning latent size with LID distributions measured on the dataset can reduce trial–and–error and improve training stability [Kingma and Welling, 2014, Tolstikhin et al., 2018].

Generalization and sample efficiency. The intrinsic dimension of a dataset (and of learned features) correlates with how efficiently models learn and how well they generalize. Lower–dimensional effective structure tends to require fewer samples and yields smoother optimization, while high local dimension flags hard regions where models may overfit or need additional inductive bias [Pope et al., 2020]. Monitoring LID thus complements standard learning–curve diagnostics when deciding on data augmentation or capacity.

Memorization analysis in generative models. LID has been used to study when generative models memorize training examples. Regions/classes with elevated LID can coincide with higher memorization propensity, offering a geometry–based lens on privacy and overfitting risks and suggesting targeted regularization or data curation [Ross et al., 2025].

Testing the union–of–manifolds view for images. Beyond single–manifold settings, real data are better modeled as unions of several manifolds of potentially different dimensions. LID helps test this assumption in image domains by revealing dimension shifts between components (e.g., object categories, textures) and by validating whether a union–of–manifolds hypothesis is plausible for a given dataset [Brown et al., 2022].

Out–of–distribution (OOD) detection. Recent methods leverage LID–related cues to improve OOD detection. Intuitively, OOD samples often sit in regions where the local geometry expands or warps relative to in–distribution data; exploiting this signal improves separability and complements likelihood–based scoring in flows and diffusion models [Kamkari et al., 2024]. LID–aware OOD detectors can thus be more robust under covariate shift.

Autoencoder quality diagnostics (reconstruction vs. LID). On different datasets (e.g., MNIST/FMNIST), per–example reconstruction error in VAEs correlates strongly with local dimension: points lying in higher–LID regions are harder to reconstruct, as we show in this work. This makes LID a simple, model–agnostic diagnostic for where reconstructions (or downstream predictors) are likely to struggle.

Prediction uncertainty estimation. We show in this work that we may potentially use LID as a proxy to estimate uncertainty of the machine learning classifier and regression model.

In sum, LID links data geometry to concrete modeling choices. It helps set hyperparameters (e.g., latent size), interpret training dynamics, anticipate generalization, and design evaluation/monitoring tools (e.g., for OOD and memorization). The unifying theme is locality: by treating dimension as a neighborhood property, one can tailor methods to the actual structure of the data.

1.2 History of the Field

The study of *intrinsic dimension* (ID) grew out of two intertwined threads: practical tools for compressing high–dimensional data, and mathematical notions of dimensionality that remain meaningful away from strict linearity. Principal Component Analysis (PCA), introduced in the early 20^{th} century, provided the canonical linear yardstick: the effective number of directions needed to explain variance [Pearson, 1901, Hotelling, 1933]. By the 1970s and 1980s, as pattern recognition matured, researchers began asking for direct estimators of dimensionality from data. A classic early attempt is the near-neighbor approach of Pettis, Bailey, Jain and Dubes, which ties local neighbor geometry to an estimate of the number of degrees of freedom [Pettis et al., 1979]. In parallel, chaos theory brought fractal viewpoints into data analysis: correlation dimension quantified how pairwise counts scale with radius, offering a practical proxy for fractal and information dimensions from finite samples [Gassberger and Procaccia, 1983].

The late 1990s and early 2000s saw two developments that decisively shaped the field. First, manifold learning methods such as Isomap, Locally Linear Embedding, and Laplacian Eigenmaps made the "data lie on a low-dimensional manifold" hypothesis operational for embeddings, sharpening the question of how to estimate the manifold's dimension itself [Tenenbaum et al., 2000, Roweis and Saul, 2000, Belkin and Niyogi, 2003].

Second, a wave of ID estimators built on scale laws of neighborhoods. Kégl's packing-number estimator used capacity (box-counting) ideas but with computationally amenable surrogates [Kégl, 2002]. Levina and Bickel then introduced a landmark maximum-likelihood estimator (MLE) based on k-nearest-neighbor (kNN) distances under locally homogeneous sampling, putting the scaling of neighbor radii into a statistically principled framework [Levina and Bickel, 2004]. Around the same time, Hein and Audibert analyzed ID for submanifolds in $\mathbb{R}^D$, giving finite-sample procedures and clarifying the choice of scale [Hein and Audibert, 2005].

A second wave, roughly 2010–2018, refined both robustness and locality. Johnsson, Soneson, and Fontes proposed ESS (Expected Simplex Skewness), which reads the shape of small simplices to obtain low-bias *local* dimension estimates even at modest sample sizes [Johnsson et al., 2014]. Ceruti and collaborators introduced DANCo, combining the distributions of (normalized) neighbor distances with neighbor angles to stabilize estimation in regimes where concentration phenomena dominate [Ceruti et al., 2014]. A complementary minimalist line culminated in TWO-NN: using only the ratio of the first two neighbor distances, it trades bias control for extreme simplicity and computational speed, while remaining consistent in homogeneous regions [Facco et al., 2017]. In parallel, geometry-aware variants improved the classic MLE by incorporating curvature or metric non-uniformities [Gomtsyan et al., 2019]. These methods collectively established a practical toolbox: distance-based MLEs for generality, angle-aware estimators for concentration regimes, and two-neighbor statistics for fast screening.

Locality became more than an implementation detail with the formalization of *Local Intrinsic Dimensionality* (LID). Amsaleg et al. [2015] showed how tails of the distance distribution around a query encode "how fast the space expands" there, yielding estimators and confidence tools grounded in extreme-value theory. Houle's theory cast LID as a smooth local generalization of expansion/doubling notions, tying it explicitly to the hazard rate of the distance distribution and providing a unifying lens for similarity search and anomaly detection [Houle, 2017a,b]. LID quickly found applications, for instance characterizing adversarial regions in deep networks where effective dimensionality spikes relative to normal data [Ma et al., 2018].

Comprehensive surveys from Camastra and Staiano [2016], Campadelli et al. [2015] consolidate this trajectory—from PCA-inspired heuristics and fractal scalings, through kNN-based likelihood and angle/shape statistics, to modern local models—highlighting open problems such as scale selection, heterogeneity, and finite-sample bias.

1.3 My Contribution

By the early 2020s, despite substantial progress in intrinsic dimension estimation, neural approaches to local intrinsic dimension remained completely unexplored. Nonparametric methods no longer matched the new reality, in which dataset dimensionalities numbered in the hundreds, thousands, or even millions, so a new approach was needed to carry LID estimation into a new era.

We proposed the first LID estimator based on neural density estimators, enabling the method to scale to datasets with thousands of dimensions and beyond. We then presented the first comprehensive theoretical treatment of this algorithm from a new perspective, the one that equips researchers with tools to analyze algorithms grounded in the Wiener process, which underpins most methods currently being developed. Finally, we introduced the first modern test suite for these algorithms to evaluate their behavior

on real, domain-specific data, and we showed that this problem is far more challenging than estimating LID on simple benchmarks such as Gaussian clouds, spheres, or the Swiss roll.

Our results advanced the state of the art in LID estimation and set higher standards for the theoretical analysis of LID algorithms, as well as for how such algorithms should be tested and compared.

A neural method for LID estimation (LIDL)

This thesis introduces *LIDL*, a pointwise estimator that provides explicit control over locality and scales to modern high-dimensional data by leveraging neural density models. Concretely, LIDL perturbs the dataset with isotropic Gaussian noise of variance δ^2, trains a global density model (in practice, a normalizing flow), and infers LID from how the perturbed log-density $\log \rho_\delta(x)$ varies with δ. Under standard regularity conditions and within an intermediate range of δ, this dependence is approximately linear in $\log \delta$ with slope approaching $d - D$; a simple regression across scales then recovers the local dimension d. This design brings two practical advantages: (i) it shifts the dominant difficulty from nonparametric k-NN statistics to modern density estimation; (ii) it introduces a tunable *scale dial* δ to suppress measurement granularity or, conversely, probe fine-scale structure.

Empirically, on controlled synthetic tests (including high-dimensional manifolds, curved manifolds, and multi-component systems), LIDL yields accurate estimates and on image datasets with unknown ground truth its estimate correlates with perceived and experimentally measured complexity of an image. Its limitations are explicit: sensitivity to the chosen δ-range (too much inflation can merge nearby components and introduce bias), dependence on the quality and calibration of the density estimator, and boundary effects where the on-manifold density vanishes or the training data are sparse.

ID-NF [Horvat and Pfister, 2022] appeared few months later as an independent line of attack on the same problem. It also starts from adding noise to the data, but instead of inspecting changes in log-likelihood, it looks directly at the *Jacobian* of the learned flow mapping. When training a normalizing flow on data inflated by Gaussians of different variances, the singular values of the Jacobian along normal and tangent directions evolve in a distinct, predictable way; by fitting a simple dependence to these curves for several noise levels, one can estimate d by counting "large-variance" directions. Conceptually, this follows the same "noise as probe" principle as LIDL but with a different indicator: instead of the decay rate of $\log \rho_\delta(x)$, the decay of singular values in normal directions. In practice, ID-NF trains several flows for increasing noise levels, measures Jacobian spectra, and segments them into "manifold" and "off-manifold" parts.

The advantage is that one measures a quantity close to the geometric decomposition (normal vs. tangent), which has been demonstrated on 64×64 RGB images and on generated distributions (e.g. from StyleGAN), and it naturally extends to OOD detection (where estimated ID tends to increase). Its limitations are: multiple models must be trained and there is a need to calculate SVD for each image, which has high computational complexity (cost); the noise range must be chosen so as not to "merge" manifold structure similar to LIDL; theoretical assumptions simplify topology and may not fully cover strong boundary cases; results can be sensitive to architecture and numerical conditioning of the spectra. Nevertheless, ID-NF provides a coherent picture and—crucially for the narrative of this dissertation—supports the view that "adding controlled noise" is not merely a computational trick but a robust lens for reading local dimension.

A subsequent shift moved from normalizing flows to diffusion. The "normal-bundle" (NB) estimator of Stanczuk et al. [2024] shows that a score-based model trained with variance-exploding noise already contains information useful to estimate LID: near the data manifold, the score field tends to align with the normal directions. By diffusing a point slightly off the manifold, stacking the resulting score vectors, and inspecting the singular-value drop, one can read off the local dimension $d = D - \text{rank}$ without nearest neighbors. On controlled tests (embedded spheres, a nonlinear spaghetti curve in $\mathbb{R}^{100}$, unions of manifolds) and on synthetic image manifolds (squares, Gaussian blobs) the NB method has been shown to outperform classical baselines; on MNIST it produces a plausible ordering of per-digit complexity validated against autoencoder reconstruction curves. The price is computational (many score evaluations per query, sensitivity to diffusion time) and practical (thresholding the spectrum, curvature and boundary effects, and reliance on a well-trained diffusion model).

Almost in parallel, Horvat and Pfister revisited diffusion from a theoretical angle, decomposing the time-dependent vector field into a conservative component and a gauge-free remainder Horvat and Pfister [2024] in the algorithm called *ID-DM*. They show that conservativity is neither necessary nor sufficient for exact sampling or likelihood, yet it can be desirable when inferring *local* manifold properties such as LID. Their estimator analyzes how a small Gaussian perturbation is transported by the learned flow—again embracing the "noise as probe" lens, now inside diffusion—and on illustrative toy Gaussians the Jacobian spectrum recovers d. The trade-off is inverse to NB's: elegant guarantees and diagnostics, but experiments confined to simple settings.

Finally, Kamkari et al. close the loop with *FLIPD* Kamkari et al. [2024], deriving an estimator closely aligned with LIDL in spirit—track how log-density changes with injected noise to read off $D - d$—but evaluating the required rate directly via the Fokker–Planck equation of a *single* pre-trained diffusion model, thereby eliminating multi-model fits. On synthetic LID benchmarks it matches or exceeds prior methods, and on natural images it correlates with non-LID proxies of complexity (e.g., PNG compression) and scales to high-resolution regimes. Familiar caveats remain: sensitivity to architecture and trace-estimation accuracy, the need to choose a noise/scale window that neither merges neighboring components nor overfits microscopic artifacts, and reliance on the diffusion model's capacity to yield stable density evaluations.

A Wiener–process perspective on modern LID estimation

After these explorations, we framed the new wave of LID estimators through the *Wiener–process* lens: adding controlled Gaussian noise can be viewed as a Brownian evolution of the data distribution, and each method in this family—likelihood-based, Jacobian-based, or score-based—reads the same short-time behavior of the diffused density ρ_t near a point x. This perspective describes the shift from classical nonparametric tools—limited by curvature, non-uniform densities, and the curse of dimensionality—to parametric, neural approaches that scale to modern data. In practice, the field has converged on a two-step pattern: first perturb the dataset with noise; then estimate local dimension from how ρ_t changes, whether by tracking likelihood (LIDL/FLIPD), Jacobian spectra (ID-NF/ID-DM), or score geometry (NB). These designs are competitive on large, real datasets when the noise acts "injectively" enough to preserve the dataset's fingerprint, subject to practical constraints that prevent taking t arbitrarily small (e.g., image quantization and instability of training at very low noise levels).

Our contribution is to make this story precise and unified within stated assumptions.

We show that adding Gaussian noise of varying magnitude corresponds to the evolution of a Wiener process in the ambient space, which allows us to invoke Fick's Second Law and replace time derivatives in existing LID objectives with spatial ones. We also show, that LIDL and FLIPD can be viewed as two ways of reading the same short-time signal from ρ_t: one via a log-slope across scales, the other via spatial operators derived from the diffusion equation—turning a shared intuition into a common analytic platform for the second step of these algorithms.

Within this framework, we (i) introduce a simple taxonomy—*isolated* methods (LIDL, FLIPD, NB) that depend only on the local shape of ρ_t up to normalization, and *holistic* methods (ID-NF, ID-DM) whose Jacobian-based readouts depend on the global transformation to a reference Gaussian; (ii) analyze the first step (the perturbation itself) directly in Wiener language, working out canonical diffusion cases for lower-dimensional manifolds with non-uniform densities; and (iii) derive closed-form expressions for key parameters used by LIDL/FLIPD as functions of on-manifold density and dimension—yielding principled corrections away from the flat, uniform ideal and explaining experimental phenomena reported earlier.

Benchmarking LID estimators: motivation and our framework

Stepping back once more, we observed a gap that theory alone could not close: while neural methods for LID estimation advanced quickly, *how* they were being evaluated had not. Most tests either used simple, analytically tractable toy distributions (clean ground truth, but far from real data) or domain datasets like images and audio (realistic complexity, but unknown ground truth). Combined with domain-specific inductive biases of neural architectures, it becomes hard to tell whether an algorithm truly measures LID or exploits quirks of the domain. In short, the field lacked a reliable way to probe *where* methods fail.

Our answer, developed in this work, is a benchmarking toolbox to stress-test LID estimators in a controlled yet domain-faithful way. We first identify manifold traits that routinely trip up algorithms—non-uniform densities, curvature, boundaries, thin structures, nearby components, sample-size effects—and then design datasets that isolate each trait. To bridge the gap between analytic and real-world settings, we introduce transformations that carry manifolds into arbitrary domains while preserving their structure: *Inverse Domain Representation* (IDR) for geometry-preserving embeddings into a target domain; *Monotonic Embedding* (ME) for controlled, smooth distortions; *Ambient Space Extension* (ASE) and *Auxiliary Dimension Injection* (ADI) for ambient-dimension manipulations with known effects; and *Manifold Synthesis* (MS) to create domain-like datasets with known LID. These names and constructions are introduced in this work; precise definitions and implementations are included in Chapter 6.

We conduct a comprehensive, side-by-side evaluation of representative methods—ESS (as a strong classical baseline), LIDL, NB, and FLIPD—on the proposed benchmarks. Each dataset reveals a weakness in at least one method; the comparisons expose where performance shifts are due to density variation, curvature, boundaries, proximity of components, or architectural bias. Taken together, the results argue for more demanding, domain-aware benchmarks and provide a practical path to build them.

1.4 Related Work

We group the literature thematically and cite representative sources. Surveys and benchmarking papers offer broad entry points; methodological references are organized roughly chronologically within each theme.

Surveys and benchmarks. For comprehensive overviews of intrinsic-dimension estimation up to the mid-2010s, see the survey by Camastra and Staiano, which also discusses open problems, and the benchmarking study by Campadelli and collaborators, which compares several families of estimators on synthetic and real data [Camastra and Staiano, 2016, Campadelli et al., 2015]. A recent survey updates the landscape and organizes methods by the geometric information they exploit (tangential, parametric/probabilistic, and topological/metric), but they do not analyze neural-based estimators in their work [Binnie et al., 2025].

Foundations and early estimators. Seminal contributions include the near-neighbor estimator of Pettis, Bailey, Jain, and Dubes, and early PCA–based local approaches; historically earlier work by Bennett and contemporaries formalized the objective of "data dimensionality estimation" in signal collections [Pettis et al., 1979, Bennett, 1969]. The fractal and dynamical-systems line introduced correlation and related dimensions as practically estimable notions of scaling [Gassberger and Procaccia, 1983, Takens, 2006]. Packing/capacity–based approaches, e.g., Kégl's estimator, provided geometric, distribution-agnostic alternatives [Kégl, 2002]. A thorough empirical assessment of several early families can be found in Verveer and Duin [1995].

Likelihood and nearest–neighbor methods (global and local). The maximum-likelihood (MLE) estimator of Levina and Bickel remains a reference point for distance–based approaches [Levina and Bickel, 2004]. Hein and Audibert provided a theoretically grounded estimator for submanifolds, highlighting the role of scale and sample size [Hein and Audibert, 2005]. Work by Sricharan and Hero optimized k–NN–based estimators and analyzed their bias/variance behavior [Sricharan et al., 2010]. More recently, GeoMLE introduced geometry-aware corrections to MLE to mitigate curvature and sampling biases [Gomtsyan et al., 2019].

Angle- and spectrum-aware estimators. The DANCo family combines information from nearest–neighbor distances and interpoint angles to improve robustness at moderate sample sizes [Ceruti et al., 2014]. Expected Simplex Skewness (ESS) uses the skewness of random simplices in local neighborhoods to infer the tangent-space dimension [Johnsson et al., 2014].

Minimal–information and multiscale estimators. The TWO–NN estimator shows that the ratio of the first two neighbor distances suffices, under mild assumptions, to recover dimension, and it pairs naturally with block/multiscale analysis [Facco et al., 2017]. Other multiscale variants generalize classical estimators to operate across neighborhood sizes; see also the comparative discussions in the surveys cited above.

Local Intrinsic Dimensionality (LID) as a formal local statistic. Amsaleg et al. introduced practical LID estimators motivated by similarity search and outlier

detection, while Houle's papers provided a formal, extreme–value–theoretic foundation and multivariate generalization of LID [Amsaleg et al., 2015, Houle, 2017a,b]. EVT-based estimators and analyses followed, consolidating LID as a standard tool for local analysis [Amsaleg et al., 2018]. Applications in robust learning and security further popularized LID (e.g., detection of adversarial examples) [Ma et al., 2018].

Other classical estimators. MADA (Manifold-Adaptive Dimension Estimation) by Farahmand et al. Farahmand et al. [2007] adapts the scale of neighborhood selection to local manifold curvature, improving alignment with the intrinsic geometry of the data. The lPCA method Cangelosi and Goriely [2007] applies localized Principal Component Analysis to estimate intrinsic dimension based on the number of components explaining local variance. Carter et al. Carter et al. [2009] proposed a KNN-based estimator that relies on local neighborhood statistics and density estimation to infer dimensionality. MiND-ML, introduced by Rozza et al. [2012], uses the distribution of minimum neighbor distances under a maximum likelihood formulation. Finally, Fisher Separability (FisherS) by Albergante et al. [2019] exploits class separability in local linear projections via Fisher's discriminant ratio to derive dimension estimates in high-dimensional data.

Additional neural estimators for LID. Beyond the approaches discussed above, several recent works also use neural networks to read off *local* manifold dimension. Yeats et al. propose an *adversarial* readout from a trained score model: by regularizing the score toward harmonicity and then measuring how carefully crafted perturbations move a point in-and-out of the data manifold, they estimate the (topological) local dimension from the number of adversarial directions needed to leave the learned manifold [Yeats et al., 2023]. A complementary autoencoder line makes the Jacobian rank itself a learnable signal: Takhanov et al. add a Ky–Fan antinorm penalty to train autoencoders whose *decoder* has (approximately) rank k, yielding local tangent bases and a practical route to pointwise rank (and thus LID) readout from the learned Jacobian [Takhanov et al., 2023]. Pushing this idea further, Causin and Marta estimate LID by computing the *pullback metric* of a VAE decoder and taking its numerical rank per sample—effectively a neural, pointwise metric–rank estimator of local dimension [Causin and Marta, 2025]. Finally, theoretical analyses of score–Jacobian spectra in diffusion models show that *spectral gaps* in the score's Jacobian across time scales can reveal tangent versus normal directions, providing another neural, model-based route to LID via eigenvalue "drops" [Ventura et al., 2024].

Applications in modern representation learning. Recent work uses ID/LID to probe learned representations and optimization landscapes in deep learning. Representative examples include measuring the intrinsic dimension of objective landscapes and tracking the evolution of representation dimension across layers [Li et al., 2018, Ansuini et al., 2019]. These studies underscore that "dimension" is not merely a preprocessing hyperparameter; it is an observable property of data *and* models that can guide architecture and training choices.

1.5 Roadmap

Most of the content in this work is based on three manuscripts. Two of them were published on ICML and AAAI, and the last one is currently under review. Chap-

ters 2, 4, 5 are based on *LIDL: Local intrinsic dimension using approximate likelihood* [Tempczyk et al., 2022]. Chapter 3 is based on *A Wiener Process Perspective on Local Intrinsic Dimension Estimation Methods* [Tempczyk et al., 2025] and Chapter 6 is based on manuscript titled: *Why do we need new benchmarks for local intrinsid dimension estimation* by Piotr Tempczyk, Dominik Filipiak, Łukasz Garncarek, and Adam Kurpisz.

In **Chapter 2: *LIDL***, we introduce the method formally. We specify the setting, state and prove the core estimate underpinning LIDL, and fix the notation used throughout. We derive closed-form behavior in instructive examples (flats in $\mathbb{R}^D$, anisotropic Gaussians, discrete lines, an "ideal" line), and examine controlled scenarios where ρ_δ is known or numerically integrated to expose boundary, curvature, and interaction effects.

In **Chapter 3: *Wiener Process perspective***, we recast Gaussian perturbations as trajectories of a Wiener process and use Fick's Second Law (the heat equation) to replace time derivatives appearing in modern LID estimators (including LIDL and FLIPD) with spatial derivatives. We derive explicit expressions for the Laplacian of the diffused density on- and off-manifold, introduce the reparameterized slope $\beta_t(x) = t\,\Delta\rho_t(x)/\rho_t(x)$ and its limit $\beta(x)$, and prove its equivalence to asymptotic-slope formulations. Canonical case studies—uniform and Gaussian densities, uniform laws on intervals and hypercubes, parallel and intersecting manifolds—clarify bias sources, operating regimes, and cross-component influence.

In **Chapter 4: *Comparison with classical algorithms***, we benchmark LIDL against established classical LID estimators under imperfect density estimation and at scale. We describe our evaluation protocol and baselines, detail when some methods are excluded for practical constraints, and compare scalability (up to thousands of ambient dimensions), robustness on multiscale manifolds (where δ stabilizes estimates), and performance on curved and union manifolds (Swiss roll, helix, sphere, and mixed-dimensional "lollipop" data). We report relative bias/MAE and variance over repeated runs, highlighting when and why LIDL yields unbiased estimates in high dimensions.

In **Chapter 5: *Experiments on image datasets***, we move to real data. Sorting MNIST, FMNIST, and CelebA by per-sample LID, we visualize how estimated dimensionality aligns with perceptual complexity and class structure. We study the operating range of δ: on toy multiscale data it recovers the expected step-like behavior, while on images choosing δ too large can collapse thin manifolds. We analyze the effects of dequantization and δ-sweeps on absolute estimates and rankings, show that increasing the number of models n monotonically reduces error, and link LID to downstream performance via correlations with VAE reconstruction error and with classification accuracy.

In **Chapter 6: *Broad comparison of neural algorithms***, we step back to methodology and propose a principled, domain-aware benchmarking framework that bridges simple analytic toys and messy real data. We formalize a toolkit of transformations and apply it to construct diagnostic datasets probing non-uniformity, curvature, boundaries, thin and nearby manifolds, and sample-size sensitivity. We then summarize quantitative results and distill per-algorithm takeaways, with implementation details and extended analyses deferred to the appendices.

Finally, in **Chapter 7: *Conclusions and future work***, we synthesize our theoretical and empirical findings, reflect on limitations, and chart directions for future work.

Chapter 2

LIDL

In this chapter we describe our method: *Local Intrinsic Dimension estimation using approximate Likelihood* (LIDL) with required proofs, then we show some derivations for a few special cases and conduct numerical experiments showing its behavior in idealized setting, when density estimation is accurate. Specifically, we organize the chapter as follows. In Sec. 2.1 we describe the core insight behind our method. In Sec. 2.2 we formalize the setting, state and prove the core estimate (Thm. 2.2.7), and fix notation used throughout; this section also records simple predictions that later guide our tests (Alg. 1, Fig. 2.1). Sec. 2.3 views δ as an explicit *scale parameter* and illustrates how choosing δ suppresses structures thinner than the operating scale (Fig. 2.2). Sec. 2.4 relaxes connectivity and embedding assumptions by moving to *good immersions* and reducing intersections to the embedded case (Prop. 2.4.3).

We then turn to examples with closed-form calculations. Sec. 2.5 works through instructive cases: flat subsets of $\mathbb{R}^D$; anisotropic Gaussians where the estimate tracks gap structure (Prop. 2.5.1); discrete points along a line with explicit bounds; and an "ideal" line case that makes error terms transparent (Sec. 2.5).

Finally, Sec. 2.6 examines behavior in controlled scenarios where ρ_δ is available explicitly or via numerical integration: boundary effects for a uniform interval (Fig. 2.3); variation across a line with $\mathcal{N}(0,1)$ and distance-dependent errors (Fig. 2.4); curvature on a circle (Fig. 2.5); interactions between nearby components (Fig. 2.6); and the impact of linear regression. We conclude with additional synthetic datasets where the ground truth is known and the estimates match to high precision (Sec. 2.6).

2.1 Intuitive explanation of LIDL

At the core of our method lies the observation that when we add Gaussian noise $\mathcal{N}(0, \delta^2 I)$ to the dataset X embedded in $\mathbb{R}^D$, the rate of change of the log-likelihood at $x \in X$ (at which LID equals d) is approximately linear in the logarithm of δ. Moreover, the proportionality constant is $\beta \approx d - D$ and we can estimate it using linear regression, thus estimating d. We may view δ as a scale parameter of our method, which may be of practical benefit beneficial when dealing with noisy datasets. Intuitive visualization of this concept can be found in Fig. 2.1, and the formal derivation can be found in Sec. 2.2.

To the best of our knowledge, LIDL is the first theoretically grounded method of LID estimation that uses global density estimation methods. In our method we relax the assumptions many algorithms make about uniformity of the density and the manifold flatness in the neighborhood of x. We show theoretically and experimentally that we can

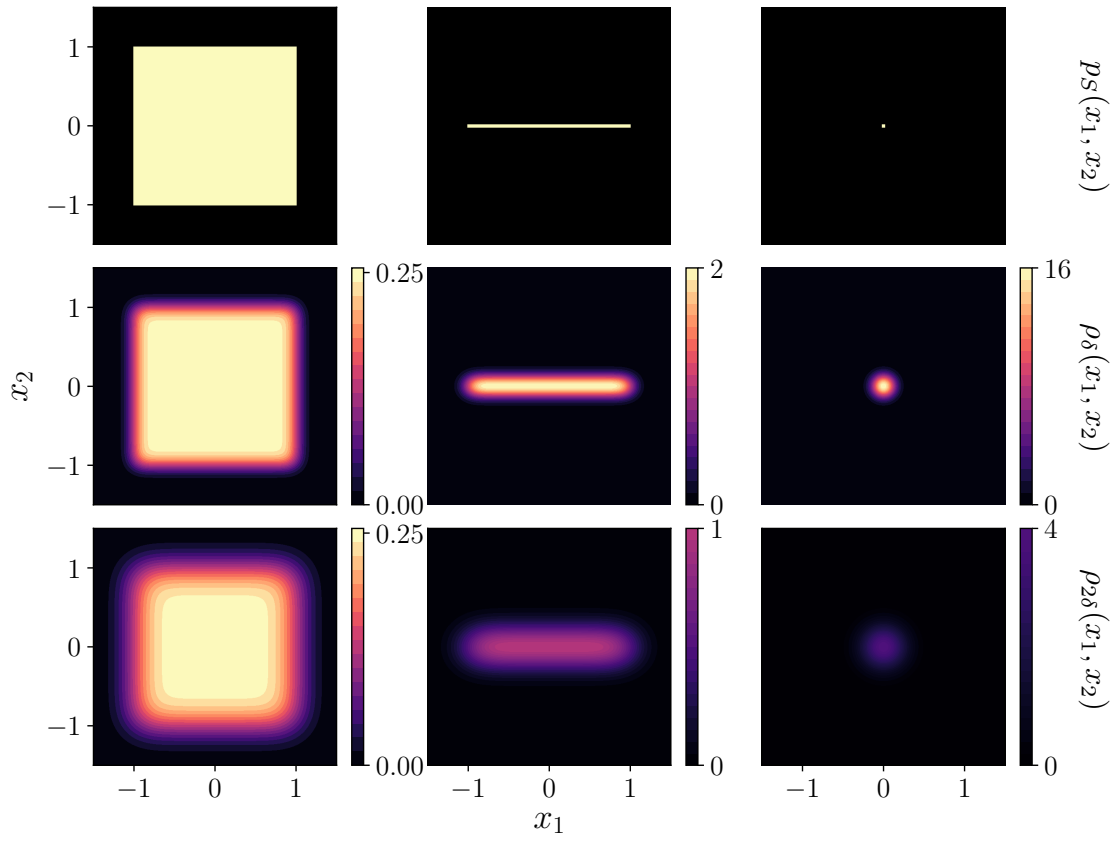


Figure 2.1: Illustration of LIDL's core insight. [Top] Three uniform distributions p_S supported respectively on a square, interval, and a point, with intrinsic dimensions 2, 1, 0. [Middle/bottom] Perturbed densities ρ_δ and $\rho_{2\delta}$ resulting from addition of Gaussian noise with different noise magnitudes (standard deviations): δ and 2δ. Our core insight is that the difference between the densities $\rho_\delta(x)$ and $\rho_{2\delta}(x)$ at any point x depends on the local intrinsic dimension (LID) at that point. Consider point $x = (0, 0)$. For the left column, that difference is zero; for the middle one, the density is halved; for the right one, it is quartered. We leverage this mechanism to estimate LID.

deal with manifolds consisting of multiple multiscale connected components of different dimensions. We also compare our algorithm with a wide range of other LID estimation algorithms, and verify that only our algorithm can give unbiased estimates for high-dimensional datasets. The code to reproduce our results is available at github.com/opium-sh/lidl.

The empirical success of our method was made possible by using neural density estimators called *normalizing flows* (NF) [Rezende and Mohamed, 2015] – described briefly in Sec. 4.1 – which can estimate densities even in high-dimensional spaces as images. Although in this work we use NF as LIDL's density estimator, our method can be used with any density estimation method. Thus, we anticipate that its capabilities to grow further with continuing progress in the area of density estimation.

2.2 Method

In this section, we introduce the problem setting formally and we lay out the LIDL method theoretically, including its derivation and pointing out certain predictions about its behavior that we later verify empirically in Sec. 2.6.

It is often known that a particular dataset X is a subset of some data manifold M equipped with probability measure μ. However, the manifold M and the dataset X need not be directly observable. Instead, there may exist an embedding (or, more generally, an immersion satisfying some regularity conditions, see Sec. 2.4) $j\colon M \to \mathbb{R}^D$ into a Euclidean space of larger dimension, through which we can view X.

The method we propose is based upon the observation, expressed in Theorem 2.2.7, that for a probability measure supported on an embedded submanifold of an Euclidean space, the dimension of its support can be recovered from its asymptotic behavior under small normally distributed perturbations (see Fig. 2.1 for an intuitive illustration).

The formal setting

We first define a class of measures we will restrict our considerations to, which we will call *smooth* measures, including all measures with continuous positive densities.

Definition 2.2.1. A positive measure ν on a manifold N will be called *smooth* if for any chart $\psi\colon U \to V \subset \mathbb{R}^n$ of N, the pushforward $\psi_*\nu$ is absolutely continuous with respect to the Lebesgue measure λ on V, and moreover, its density is locally bounded away from 0, i.e. any $x \in V$ admits a neighborhood on which $dj_*\nu/d\lambda > c$ for some $c > 0$.

Let $S \subset \mathbb{R}^D$ be a smooth connected d-dimensional embedded submanifold of a high-dimensional Euclidean space $\mathbb{R}^D$ (the more general case of a non-connected immersed manifold is dealt with in Sec. 2.4). This is our observable data manifold, embedded in Euclidean space, i.e. $S = j(M)$. Furthermore, suppose we are given a smooth (according to Definition 2.2.1) probability measure p_S on S, representing the data probability distribution. In our notation, this is the pushforward of the probability μ on M, i.e. $p_S = j_*\mu$. We will implicitly treat p_S as a probability distribution on the whole ambient space $\mathbb{R}^D$.

The Gaussian function (i.e. the density of the standard normal distribution) on a Euclidean space V will be denoted by ϕ^V, or ϕ^n in the case where V is the standard $\mathbb{R}^n$ space. Also, for $\delta > 0$, let

$$\phi^V_\delta(x) = \delta^{-\dim V}\phi^V(x/\delta) \tag{2.1}$$

be the density of the normal distribution $\mathcal{N}(0, \delta^2 I)$ with covariance matrix $\delta^2 I$, where I is the identity matrix on V.

Under the above notation, if $X \sim p_S$ is a random vector representing the data, and $N_\delta \sim \mathcal{N}(0, \delta^2 I)$ a normally distributed random noise vector, the distribution of the perturbed random vector $X + N_\delta$ in $\mathbb{R}^D$ is given by the convolution $p_S * \mathcal{N}(0, \delta^2 I)$, and has density

$$\rho_\delta(x) = \int_S \phi_\delta^D(x - y)\, dp_S(y). \tag{2.2}$$

Finally, let us introduce a notation for uniform multiplicative estimates. We will write that $f(x, y) \asymp g(x, y)$ uniformly in x if for every y there exists $C > 0$ such that for all x

$$C^{-1} g(x, y) \le f(x, y) \le C g(x, y). \tag{2.3}$$

This notation extends to any number of variables. We will use it to declutter the proofs from irrelevant constants.

The core estimate

At any $x \in S$ the tangent space of $\mathbb{R}^D$ admits a decomposition $T_x\mathbb{R}^D = T_x S \oplus N_x S$ into a direct sum of the tangent and normal spaces of S. Under the natural identification of $T_x\mathbb{R}^D$ with the underlying $\mathbb{R}^D$ (mapping the origin of $T_x S$ to x), the tangent and normal spaces of S at x become two affine subspaces of $\mathbb{R}^D$ intersecting at x. Denote by $\pi_x \colon \mathbb{R}^D \to T_x S$ and $\pi_x^\perp \colon \mathbb{R}^D \to N_x S$ be the corresponding orthogonal projections. With this notation, the following decomposition of the Gaussian density holds

$$\phi_\delta^D(x - y) = \phi_\delta^{T_x S}(\pi_x(y)) \phi_\delta^{N_x S}(\pi_x^\perp(y)). \tag{2.4}$$

By the Inverse Function Theorem applied to the restriction of π_x to S, in a small neighborhood of any $x \in S$, the manifold S can be represented as the graph of a smooth map $F_x \colon T_x S \to N_x S$. In particular, it follows that in this neighborhood one has $\pi_x^\perp = F_x \circ \pi_x$. Moreover, $F_x(0) = 0$, and since the graph of F_x is tangent to $T_x S$ at the origin, the derivative of F_x at x vanishes. Hence, the Taylor expansion of F_x at 0 starts with the second-order term, and consequently, there exists $C > 0$ such that for small v

$$\|F_x(v)\|_{N_x S} \le C \|v\|_{T_x S}^2. \tag{2.5}$$

We will precede the statement of the core estimate (Theorem 2.2.7) with five lemmas required for its proof. Denote by $B(x, r)$ the ball of radius r in $\mathbb{R}^D$, centered at x.

Lemma 2.2.2. *Let $x \in S$. For sufficiently small δ the projection $\pi_x(S \cap B(x, \delta^{1/2}))$ contains the ball $B(x, \delta) \cap T_x S$.*

Proof. Assume that δ is sufficiently small for F_x to be defined on $B(x, \delta) \cap T_x S$. Let $v \in B(x, \delta) \cap T_x S$. Under our identifications, $y = (v, F_x(v)) \in T_x S \oplus N_x S$ is a point of S such that $v = \pi_x(y)$. Moreover, by eq. (2.5), for sufficiently small δ

$$\|v\|_{T_x S}^2 + \|F_x(v)\|_{N_x S}^2 \le \delta^2(1 + C\delta^2) < \delta, \tag{2.6}$$

so $y \in B(x, \delta^{1/2})$, and $v \in \pi_x(S \cap B(x, \delta^{1/2}))$. □

Lemma 2.2.3. *For $x \in S$ and sufficiently small δ, the estimate*

$$\int_{S \cap B(x, \delta^{1/2})} \phi_\delta^{T_x S}(\pi_x(y))\, dp_S(y) \asymp 1 \tag{2.7}$$

holds uniformly in δ.

Proof. Denote $B = S \cap B(x, \delta^{1/2})$. Integrating by substitution, we obtain

$$\int_B \phi_\delta^{T_x S}(\pi_x(y))\, dp_S = \int_{\pi_x(B)} \phi_\delta^{T_x S}(v)\, d(\pi_x)_* p_S, \tag{2.8}$$

where the pushforward $(\pi_x)_* p_S$ is a smooth measure on $T_x S$. Hence, for sufficiently small δ

$$\int_{\pi_x(B)} \phi_\delta^{T_x S}(v)\, d(\pi_x)_* p_S(v) \asymp \int_{\pi_x(B)} \phi_\delta^{T_x S}(v)\, dv \tag{2.9}$$

uniformly in δ. The integral on the right is at most 1, and simultaneously, by Lemma 2.2.2 we have

$$\int_{\pi_x(B)} \phi_\delta^{T_x S}(v)\, dv \geq \int_{B(x,\delta)\cap T_x S} \phi_\delta^{T_x S}(v)\, dv. \tag{2.10}$$

The last integral is the probability that a normal random variable falls within one standard deviation from the mean, which is a constant independent of δ. □

Lemma 2.2.4. *For sufficiently small δ and $y \in S \cap B(x, \delta^{1/2})$, where $x \in S$, the estimate $\phi_\delta^{N_x S}(\pi_x^\perp(y)) \asymp \delta^{d-D}$ holds uniformly in δ and y.*

Proof. By eq. (2.1),

$$\phi_\delta^{N_x S}(\pi_x^\perp(y)) = \delta^{d-D} \phi^{N_x S}(\delta^{-1}\pi_x^\perp(y)). \tag{2.11}$$

Since π_x is a contraction, we have $\|\pi_x(y)\| \leq \delta^{1/2}$. It follows from eq. (2.5), that for sufficiently small δ

$$\|\pi_x^\perp(y)\| = \|F_x(\pi_x(y))\| \leq C\delta \tag{2.12}$$

for some $C > 0$. Therefore $\delta^{-1}\pi_x^\perp(y)$ lies inside a fixed ball independent of δ, and in consequence

$$\phi^{N_x S}(\delta^{-1}\pi_x^\perp(y)) \asymp 1 \tag{2.13}$$

uniformly in δ, concluding the proof. □

Lemma 2.2.5. *For $x \in S$ and sufficiently small δ,*

$$\int_{S\cap B(x,\delta^{1/2})} \phi_\delta^D(x - y)\, dp_S(y) \asymp \delta^{d-D} \tag{2.14}$$

uniformly in δ.

Proof. Denote $B = S \cap B(x, \delta^{1/2})$. By eq. (2.4) and Lemma 2.2.4, for sufficiently small δ and $y \in B$, we have

$$\phi_\delta^D(x - y) \asymp \delta^{d-D} \phi_\delta^{T_x S}(\pi_x(y)) \tag{2.15}$$

uniformly in δ. It follows that the original integral can be estimated as

$$\int_B \phi_\delta^D(x - y)\, dp_S \asymp \delta^{d-D} \int_B \phi_\delta^{T_x S}(\pi_x(y))\, dp_S. \tag{2.16}$$

The proof concludes by applying Lemma 2.2.3 to the last integral. □

Lemma 2.2.6. *For every $x \in S$*

$$\lim_{\delta\to 0^+} \int_{S\setminus B(x,\delta^{1/2})} \phi_\delta^D(x - y)\, dp_S(y) = 0. \tag{2.17}$$

Proof. Observe that if $\|v\| \geq \delta^{1/2}$, we have

$$\phi_\delta^D(v) \asymp \delta^{-D} \exp\left(-\frac{\|v\|^2}{2\delta^2}\right) \leq \delta^{-D} \exp\left(-\frac{1}{2\delta}\right) \tag{2.18}$$

uniformly in v. This bound on the integrand converges to 0 as $\delta \to 0$, and the measure p_S is finite, proving the convergence of the considered integral. □

Theorem 2.2.7 (The core estimate)**.** *Assume that $S \subset \mathbb{R}^D$ is a connected d-dimensional submanifold endowed with a smooth probability measure p_S. Let ρ_δ be the density of $p_S * \mathcal{N}(0, \delta^2 I)$ on $\mathbb{R}^D$. Then for $x \in S$ and sufficiently small δ, we have*

$$\log \rho_\delta(x) = (d - D) \log \delta + O(1). \tag{2.19}$$

Proof. Since $\delta^{d-D} \geq 1$ for $\delta \leq 1$, given sufficiently small δ, from Lemma 2.2.6 we get

$$\int_{S \setminus B(x, \delta^{1/2})} \phi_\delta^D(x - y)\, dp_S(y) < \delta^{d-D}. \tag{2.20}$$

By combining this with eq. (2.2) and Lemma 2.2.5, we obtain $\rho_\delta(x) \asymp \delta^{d-D}$, which yields the desired estimate after taking log. □

Now, let us consider how to use the core estimate derived above in practice. The core requirement of LIDL is access to the approximate densities $\rho_\delta(x)$, which we have to obtain by fitting a density estimator on the data points from the dataset perturbed with a normally distributed noise of an appropriate magnitude δ. Luckily, these days there exist density estimators which scale to data even as high-dimensional as images. For the purpose of empirical evaluation of our method, in this work we use three models from the family of NF, however we emphasize that our method could use absolutely any density estimation method. A viable alternative could be, for example, using diffusion models [Song et al., 2021], which are likely to lead to further improved accuracy of LIDL estimates.

Given a dataset $X \subset \mathbb{R}^D$, and a point $x \in \mathbb{R}^D$, at which we want to estimate LID (usually $x \in D$, as we want to take a point from the image of the data manifold in $\mathbb{R}^D$), we proceed as follows. First, we choose $n > 1$ values $\delta_1, \ldots, \delta_n$ of perturbation magnitude. We discuss how to choose δ in the following section. Then, we fit n probability densities $\hat{\rho}_i$, which will be our approximations of ρ_{δ_i}. Having estimated the densities $\hat{\rho}_i$, we consider the sequence of points of the form $(\log \delta_i, \log \hat{\rho}_i(x))$. Using linear regression to fit eq. (2.19), we get an estimate β for $d - D$, from which we obtain $\hat{d} = D + \beta$, an estimate for d. To estimate LID for multiple points, we can fit the densities once, and then loop over the points. Full algorithm is presented in Algorithm 1.

It is worth noting that our method fits nicely into the LID estimation framework presented in Amsaleg et al. [2019]. Roughly speaking, it is based on two observations. Firstly, the dimension of an Euclidean space can be recovered from the degree of the polynomial growth rate of its ball volume as a function of its radius. Secondly, this idea can be applied to discrete datasets by replacing the notion of ball volume with the likelihood function of finding a point of the dataset within a given distance from a fixed base point.

In our notation, this likelihood function is $r \mapsto p_S(B(x, r))$, where x is the base point. With reasonable assumptions on the measure p_S, it can be shown that for small r this function behaves like a polynomial of degree d, so the LID value we are estimating is the same as what is defined in Amsaleg et al. [2019], which can be consulted for more details.

Algorithm 1 LIDL algorithm

Require: $X \subset \mathbb{R}^D$; $x_1, \ldots, x_m \in \mathbb{R}^D$; $\delta_1, \ldots, \delta_n \in \mathbb{R}^+$;
  for $j = 1$ to n **do**
    $X_j \leftarrow X$ perturbed with $\mathcal{N}(0, \delta_j^2 I_D)$
    Fit the density model $\hat{\rho}_j$ to X_j
  end for
  for $i = 1$ **to** m **do**
    for $j = 1$ **to** n **do**
      $\xi_j \leftarrow \log \delta_j$
      $\eta_j \leftarrow \log \hat{\rho}_j(x_i)$
    end for
    $\beta \leftarrow$ regression coefficient for a set of n points (ξ_j, η_j)
    $\hat{d}_i \leftarrow D + \beta$
  end for
  return $(\hat{d}_1, \ldots, \hat{d}_m)$

2.3 Viewing δ as a scale parameter

In the previous section, we glossed over the fact that we have to choose the values of δ. From Sec. 2.2 we know that the core estimate is exact for infinitesimally small δ, so the first pressure is for the δ to be small, possibly as small as the numerical precision allows.

However, δ can also be viewed as a length-scale parameter that allows users to choose a certain minimum 'thickness' to be considered, such that dimensions 'thinner' than the threshold will be ignored. Consider an illustrative example: suppose that the probability distribution p_S is concentrated in a tubular neighborhood of another submanifold S' of dimension $d' < d$. In this case, it can be approximated by a probability distribution $p_{S'}$ supported on S'.

Now, if this approximation is 'good', in the sense that the thickness of the considered neighborhood is much smaller than the values of δ used, then intuitively the LIDL estimate should actually reflect the dimension d' of the submanifold S' instead of the true dimension d.

Continuing this example, we present the described behavior empirically. Let $S = \mathbb{R}^D$, and $p_S = \mathcal{N}(0, \Sigma)$, where Σ is a diagonal matrix with entries $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_D^2$. In Fig. 2.2, we plot the LIDL estimates for case $D = 10$ and for σ equally distributed on the

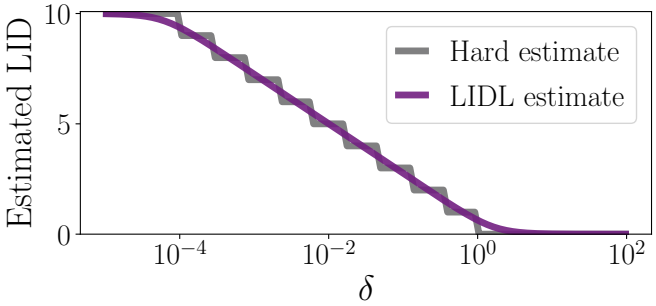


Figure 2.2: LIDL and hard estimates for different values of δ for 10D non-isotripic Gaussian. Notice how LIDL ignores dimensions smaller than δ, as predicted theoretically.

logarithmic scale. We also plot the *hard estimate* which is simply counting the number of entries in σ that are larger than δ. The LIDL estimate follows the hard estimate, and this behavior is predicted theoretically in Sec. 2.5. We further investigate the role of the scale parameter in the case of imperfect density estimates in Sec. 5.2.

As mentioned earlier, having an explicit length-scale parameter can be considered LIDL's feature as compared to other methods. Is allows the user to easily set an operating scale such that to ignore certain amplitude of noise in the original data, e.g. the observation noise if we are able to estimate its magnitude apriori. The empirically observed rule of thumb is to take at least $\delta \gtrsim 10\sigma$, where σ is standard deviation of the noise to be ignored.

Setting such operating scale characteristic can be difficult in many other non-parametric algorithms that calculate statistics based on nearest neighbors. In those approach, there is either of the two natural scale parameters: number of nearest neighbours k or radius r around the point where we search for neighbours.

When using k, our effective operating range depends on a combination of local density and the total number of samples used to run the algorithm. Using r allows to set an operating range. In this case, however, we expose ourselves to the risk of having not enough samples to estimate the local density. Most implementations of those methods set a default k.

2.4 Non-connected data manifolds and intersections

Earlier we assumed that the data comes from a connected manifold M, whose local dimension is constant. Moreover, it was embedded in $\mathbb{R}^D$, precluding self-intersections. These restrictions can be relaxed as follows. Firstly, we may allow M to contain multiple connected components. Secondly, instead of an embedding, we may consider a *good* immersion $j\colon M \to \mathbb{R}^D$, satisfying the following finiteness condition.

Definition 2.4.1. We will call an immersion $j\colon M \to N$ *good*, if M admits an open cover $\mathcal{C}$ such that for every $U \in \mathcal{C}$ the restriction of j to U is an embedding, and moreover, every $x \in N$ has an open neighborhood whose preimage intersects only finitely many sets in $\mathcal{C}$.

In the non-connected case, the dimension is no longer constant on the manifold, but can differ between its components. We will denote by $\dim_x M$ the dimension of M at a point $x \in M$.

Before we proceed, we will prove a simple technical lemma.

Lemma 2.4.2. *Let $j\colon M \to N$ be a good immersion. Then every $x \in N$ has a neighborhood whose preimage intersects only finitely many connected components of M.*

Proof. Let $\mathcal{C}$ be an open cover of M satisfying conditions of Definition 2.4.1. Take $x \in N$, and let $V \subset N$ be a neighborhood of x such that $j^{-1}(V)$ intersects only finitely many sets $U_1, \ldots, U_n \in \mathcal{C}$. On each U_i the restriction of j is an embedding, so there exists a neighborhood $V_i \subset V$ of x whose preimage is contained in a single connected component of U_i, and hence in a single connected component M_i of M.

The intersection $\bigcap_i V_i$ is the required neighborhood of x, as its preimage is contained in the finite union of connected components $\bigcup_i M_i$. □

This more general case reduces to the one studied in Sec. 2.2, as the following reasoning shows.

Proposition 2.4.3. *Suppose $j\colon M \to N$ is an immersion of manifolds. Moreover, let μ be a smooth measure on M. Then there exists a manifold $\tilde{M}$ endowed with a measure $\tilde{\mu}$ and a local diffeomorphism $f\colon \tilde{M} \to M$, such that*

1. *the measure $\tilde{\mu}$ is smooth*
2. *the pushforward $f_*\tilde{\mu}$ equals μ;*
3. *$\tilde{\jmath} = j \circ f\colon \tilde{M} \to N$ restricted to every connected component of $\tilde{M}$ is an embedding.*
4. *if j is good, then so is $\tilde{\jmath}$;*

Proof. Since j is an immersion, there exists an open cover $\mathcal{C}$ of M such that on every $U \in \mathcal{C}$ the restriction of j is an embedding. Let $\{\psi_U : U \in \mathcal{C}\}$ be a partition of unity subordinate to $\mathcal{C}$. Denote $M_U = \{x \in M : \phi_U(x) > 0\}$, and let $f_U\colon M_U \to M$ be the corresponding inclusion map. Finally, let $\tilde{M}$ be the disjoint union of $\{M_U : U \in \mathcal{C}\}$, and define $f\colon \tilde{M} \to M$ by gluing together the inclusions f_U.

The measure $\tilde{\mu}$ can be defined as

$$\tilde{\mu} = \sum_{U \in \mathcal{C}} (f_U^{-1})_*(\psi_U \mu), \tag{2.21}$$

i.e. for every $U \in \mathcal{C}$ we multiply μ by density function ψ_U, restrict it to M_U and pull it to $\tilde{M}$ through f_U. Since by definition ψ_U is continuous and positive on M_U, the measure $\tilde{\mu}$ is smooth. Moreover, by construction we have $f_*\tilde{\mu} = \mu$, and the restriction of $\tilde{\jmath}$ to every M_U (and therefore every to every connected component) is an embedding.

To show the last assertion, assume that j is good. In this case, the cover $\mathcal{C}$ defined above can be chosen in such a way that for every $x \in N$ there exists a neighborhood $V \subset N$ whose preimage $j^{-1}(V)$ intersects only finitely many sets in $\mathcal{C}$. It is then easy to see that the cover $\{M_U : U \in \mathcal{C}\}$ of $\tilde{M}$ satisfies the conditions of Definition 2.4.1, so $\tilde{\jmath}$ is good. □

Now, suppose that in Theorem 2.2.7, instead of an embedded submanifold S, we are dealing with the image of a proper immersion $j\colon M \to \mathbb{R}^D$, and that p_S is the pushforward of a probability measure μ on M. Thanks to Proposition 2.4.3, this reduces to the situation where j restricted to every connected component of M is an embedding.

Proposition 2.4.4. *Suppose $j\colon M \to \mathbb{R}^D$ is a good immersion, and its restriction to every connected component of M is an embedding. Let μ be a smooth probability measure on M, and $p_S = j_*\mu$. For $x \in S = j(M)$ and sufficiently small δ we have*

$$\log \rho_\delta(x) = (d - D) \log \delta + O(1), \tag{2.22}$$

where

$$d = \min_{j(y)=x} \dim_y M. \tag{2.23}$$

Proof. By Lemma 2.4.2, for sufficiently small r the preimage $j^{-1}(B)$ of the ball $B = B(x, r)$ centered at x intersects only finitely many connected components of M. Denote them by $M_1, \ldots, M_k$, and let M_0 be the union of the remaining components. The measure μ can be decomposed as

$$\mu = \sum_{i=0}^{k} \mu(M_i)\mu_i, \tag{2.24}$$

where μ_i is the restriction of μ to M_i, normalized to a probability measure. If we put $p_i = j_*\mu_i$, a similar decomposition holds for p_S.

If we apply Theorem 2.2.7 to $j(M_i)$ endowed with the measure p_i, for $i > 0$, the corresponding perturbed density ρ^i_δ satisfies

$$\rho^i_\delta(x) \asymp \delta^{\dim M_i - D} \tag{2.25}$$

for sufficiently small δ. Moreover, for $\delta < r^2$, we have $j(M_0) = j(M_0) \setminus B(x, \delta^{1/2})$, so by Lemma 2.2.6

$$\lim_{\delta \to 0^+} \rho^0_\delta(x) = 0. \tag{2.26}$$

Consequently, for small $\delta < 1$

$$\rho_\delta(x) = \sum_{i=0}^{k} \mu(M_i)\rho^i_\delta(x) \asymp \sum_{i=1}^{k} \delta^{\dim M_i - D}, \tag{2.27}$$

and the term with the lowest exponent dominates. □

2.5 Examples with explicit derivations

Consider the standard embedding $\mathbb{R}^d \subset \mathbb{R}^D$. Take for S a bounded open subset of $\mathbb{R}^d$, endowed with the uniform probability measure p_S with constant density $\rho \equiv \mathrm{vol}(S)^{-1}$ on S. If we denote by x_1 and x_2 the components of a vector $x \in \mathbb{R}^D$ corresponding to the standard decomposition $\mathbb{R}^D = \mathbb{R}^d \times \mathbb{R}^{D-d}$, it follows from (2.2) and properties of the Gaussian function, that

$$\rho_\delta(x) = \frac{\phi^{D-d}_\delta(x_2)}{\mathrm{vol}(S)} \int_S \phi^d_\delta(x_1 - y_1)\, dy_1. \tag{2.28}$$

Now, if x is an interior point of S, then $x_2 = 0$. Moreover, for sufficiently small δ, the integral above is arbitrarily close to 1, as most of the mass of the integrand falls into a small neighborhood of x_1, which is contained in S. Therefore, for sufficiently small δ

$$\rho_\delta(x) \asymp \phi^{D-d}_\delta(0) = \delta^{d-D}\phi^{D-d}(0) \asymp \delta^{d-D} \tag{2.29}$$

uniformly in δ.

It follows that

$$\log \rho_\delta(x) = (d - D)\log \delta + O(1), \tag{2.30}$$

and hence

$$d - D = \lim_{\delta \to 0} \frac{\log \rho_\delta(x)}{\log \delta}. \tag{2.31}$$

In practice, $d - D$, and in consequence d, can be estimated by considering $\rho_\delta(x)$ for multiple small values of δ, and using linear regression.

Normal distribution in $\mathbb{R}^D$

Suppose that $S = \mathbb{R}^D$, and $p_S = \mathcal{N}(0, \Sigma)$, where Σ is a diagonal matrix with entries $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_D^2$. In this case, the perturbation with $\mathcal{N}(0, \delta^2 I)$ yields another normal distribution $\mathcal{N}(0, \Sigma + \delta^2 I)$, whose density at 0 is

$$\rho_\delta(0) = (2\pi)^{-D/2} \prod_{k=1}^{D} (\sigma_k^2 + \delta^2)^{-1/2}. \tag{2.32}$$

Proposition 2.5.1. *Let $1 \leq d < D$, and denote $\tau = (\sigma_d \sigma_{d+1})^{1/2}$. For $\lambda \geq 1$ and $\delta \in [\lambda^{-1}\tau, \lambda\tau]$ we have*

$$\log \rho_\delta(0) = (d - D) \log \delta + M - C_\lambda, \tag{2.33}$$

where M is independent of δ, and $0 \leq C_\lambda \leq \frac{D\sigma_{d+1}}{2\sigma_d}\lambda^2$.

In other words, the above proposition states that for δ between two consecutive deviations σ_d and σ_{d+1}, our LID estimate is approximately the number d of dimensions in which the Gaussian distribution is 'thicker' than δ, and the approximation error decreases with the growth of the ratio σ_d/σ_{d+1} and the distance of δ from σ_d and σ_{d+1}.

Proof. Let us denote $\eta = \lambda(\sigma_{d+1}/\sigma_d)^{1/2}$. The, for $k \leq d$ we may compute $\lambda\tau = \eta\sigma_d \leq \eta\sigma_k$, which leads to

$$\sigma_k^2 + \delta^2 \leq (1 + \eta^2)\sigma_k^2. \tag{2.34}$$

On the other hand, for $k \geq d + 1$, we have $\lambda^{-1}\tau = \eta^{-1}\sigma_{d+1} \geq \eta^{-1}\sigma_k$, and similarly to the previous case, we have

$$\sigma_k^2 + \delta^2 \leq (1 + \eta^2)\delta^2. \tag{2.35}$$

By applying these two estimates to the formula (2.32) for $\rho_\delta(0)$ we are able to obtain a two-sided estimate

$$M(1 + \eta^2)^{-D/2}\delta^{d-D} \leq \rho_\delta(0) \leq M\delta^{d-D}, \tag{2.36}$$

with $M = (2\pi)^{-D/2} \prod_{k=1}^{d} \sigma_k^{-1}$ independent of δ. Finally, after taking log we can see that

$$\log \rho_\delta(0) = (d - D) \log \delta + \log M - \frac{D}{2} \log(1 + \eta^2), \tag{2.37}$$

and the last term is positive and bounded from above by $D\eta^2/2$, yielding the desired estimate by substituting η. □

From the above Proposition we can see that if there is a large gap between σ_d and σ_{d+1}, then for δ in the neighborhood of their geometric mean, the LID estimate obtained through linear regression should be approximately d, with approximation error decreasing, and the range of viable δ increasing with the growth of the gap size, expressed by the ratio σ_{d+1}/σ_d.

Points along a line

Consider a zero-dimensional manifold M, consisting of N points, endowed with uniform probability measure. Suppose M is embedded into $\mathbb{R}^D$ in such a way that its image $\{x_1, \ldots, x_N\}$ is actually contained in $\mathbb{R} \subset \mathbb{R}^D$, and has the form $x_k = (\xi_k, 0, \ldots, 0)$, where $\xi_{k+1} \geq \xi_k + \eta$ for some $\eta > 0$, i.e. the indexing is chosen in such a way that the points x_k are ordered along $\mathbb{R}$, and the distances between them are at least η.

In this setting, we will study the quantity $\rho_\delta(x_n)$ more closely, and attempt to understand its relationship with the perturbation magnitude for any δ, not just sufficiently small ones. We have

$$\rho_\delta(x_0) = \frac{1}{N} \sum_{k=1}^{N} \phi_\delta^D(x_n - x_k) = \frac{\phi_\delta^D(0)}{N} \left(1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} \frac{\phi_\delta^D(x_n - x_k)}{\phi_\delta^D(0)}\right) = M\delta^{-D}\left(1 + \epsilon_\delta\right), \tag{2.38}$$

where $M = (N(2\pi)^{D/2})^{-1}$, and

$$\epsilon_\delta = \sum_{\substack{k=1 \\ k\neq n}}^{N} \frac{\phi_\delta^D(x_n - x_k)}{\phi_\delta^D(0)} = \sum_{\substack{k=1 \\ k\neq n}}^{N} \exp\left[-\frac{1}{2}\left(\frac{\xi_n - \xi_k}{\delta}\right)^2\right]. \tag{2.39}$$

After taking log, we get

$$\log \rho_\delta(x_0) = -D\log\delta + \log M + \log(1+\epsilon_\delta), \tag{2.40}$$

where the term $\log M$ is independent of δ, and $0 \leq \log(1+\epsilon_\delta) \leq \epsilon_\delta$.

Proposition 2.5.2. *Let $\lambda \geq 1$. If $\delta < \eta/(\sqrt{2}\lambda)$ then $\epsilon_\delta \leq 4e^{-\lambda^2}$. In particular, for $\epsilon > 0$, we have $\epsilon_\delta < \epsilon$ provided that*

$$\delta < \frac{\eta}{(-2\log(\epsilon/4))^{1/2}}, \tag{2.41}$$

i.e. the threshold value for δ depends logarithmically on ϵ.

Proof. We have $|\xi_i - \xi_j| \geq \eta\,|i-j|$, and therefore

$$\epsilon_\delta \leq \sum_{\substack{k=1 \\ k\neq n}}^{N} \exp\left[-\frac{1}{2}\left(\frac{\eta(n-k)}{\delta}\right)^2\right] \leq \sum_{\substack{k=1 \\ k\neq n}}^{N} e^{-\lambda^2(n-k)^2}. \tag{2.42}$$

For an upper estimate, we may also extend the summation over all integers except n, obtaining

$$\epsilon_\delta \leq \sum_{k\neq n} e^{-\lambda^2(n-k)^2} = 2\sum_{j=1}^{\infty} e^{-\lambda^2 j^2} \leq 2\sum_{j=1}^{\infty} e^{-\lambda^2 j} = \frac{2}{1-e^{-\lambda^2}} e^{-\lambda^2}. \tag{2.43}$$

For $\lambda \geq 1$ we have $(1-e^{-\lambda^2})^{-1} \leq 2$, so in the end $\epsilon_\delta \leq 4e^{-\lambda^2}$. By solving $\epsilon = 4e^{-\lambda^2}$ for λ we obtain $\lambda = (-\log(\epsilon/4))^{1/2}$, yielding the last assertion. □

Ideal LIDL for normal distribution on a line

Suppose our submanifold S is the image of the standard embedding $\mathbb{R} \subset \mathbb{R}^D$, and let $p_S = \mathcal{N}(0,1)$. In this case, the perturbed distribution is $\mathcal{N}(0,\Sigma)$, where Σ is a diagonal matrix with entries $(1+\delta^2, \delta^2, \ldots, \delta^2)$. The density ρ_δ at a point $x = (t, 0, \ldots, 0) \in S$ is therefore

$$\rho_\delta(x) = \frac{\delta^{1-D}}{(2\pi)^{D/2}(1+\delta^2)^{1/2}} \exp\left(-\frac{t^2}{2(1+\delta^2)}\right), \tag{2.44}$$

and its logarithm can be decomposed into the following sum

$$\log \rho_\delta(x) = (1-D)\log\delta - \frac{D}{2}\log(2\pi) - \frac{1}{2}\log(1+\delta^2) - \frac{t^2}{2(1+\delta^2)}. \tag{2.45}$$

Let us now apply the trivial case of linear regression involving only two points, amounting to computing the slope of the line passing through two points. We have

$$\frac{\log \rho_{\delta_1}(x) - \log \rho_{\delta_2}(x)}{\log \delta_1 - \log \delta_2} = 1 - D - \epsilon(t), \tag{2.46}$$

where the error term expands to

$$\epsilon(t) = \frac{1}{2(\log\delta_1 - \log\delta_2)}\left[\log\frac{1+\delta_1^2}{1+\delta_2^2} - t^2\left(\frac{1}{1+\delta_1^2} - \frac{1}{1+\delta_2^2}\right)\right], \tag{2.47}$$

yielding a LID estimate $\hat{d}_x = 1 - \epsilon(t)$ at x. We can see that the error ϵ decomposes into two terms of opposite signs. The first term depends only on δ, and the second one, grows quadratically with t.

If we put $\delta_1 = \eta\delta$, and $\delta_2 = \delta$, the coefficient of t^2 can be further rewritten as

$$\frac{1}{2(\log\delta_1 - \log\delta_2)}\left(\frac{1}{1+\delta_1^2} - \frac{1}{1+\delta_2^2}\right) = \frac{\delta^2(1-\eta^2)}{2\log\eta(1+\delta^2)(1+(\delta\eta)^2)} \asymp \frac{\delta^2(1-\eta^2)}{2\log\eta}, \tag{2.48}$$

where the estimate holds uniformly in δ if δ is bounded δ from above. Although for fixed δ and η the error is unbounded as a function of t, if we were allowed to adjust δ based on t (with fixed η), for the error $\epsilon(t)$ to be bounded in t it is necessary and sufficient that $\delta \leq C/t$ for some constant C.

Finally, the expected error for the LID estimate (computed in the above manner) at a random x drawn from our distribution can be computed

$$\begin{aligned}\int_{\mathbb{R}} \epsilon(t)\phi^1(t)\,dt &= \frac{1}{2(\log\delta_1 - \log\delta_2)}\left[\log\frac{1+\delta_1^2}{1+\delta_2^2} - \int_{\mathbb{R}} t^2\phi^1(t)\,dt\left(\frac{1}{1+\delta_1^2} - \frac{1}{1+\delta_2^2}\right)\right] = \\ &= \frac{1}{2(\log\delta_1 - \log\delta_2)}\left[\log\frac{1+\delta_1^2}{1+\delta_2^2} - \left(\frac{1}{1+\delta_1^2} - \frac{1}{1+\delta_2^2}\right)\right] = \epsilon(1),\end{aligned} \tag{2.49}$$

where the last integral is just the variance of $\mathcal{N}(0,1)$, i.e. 1.

2.6 Empirical Behavior of the Proposed Method

In this section we examine the behaviour of our method when confronted with certain isolated difficulties. Instead of relying on a computed approximation $\hat{\rho}_\delta$, we assume we are given the actual perturbed density ρ_δ explicitly or we compute it through numerical integration. This ensures that any error observed during this analysis is caused directly by our LIDL method and not the density estimator. However, it comes at a price of restricting us to relatively simple examples where we can efficiently compute ρ_δ.

Uniform density on an interval

We assume, that in the neighborhood of x the density is bounded from below by a positive constant. But for some real-world cases, this assumption is not fulfilled. To investigate how LIDL behaves in this case we ran it on $\mathcal{U}(0,1)$. It can be seen as a distribution on the real line, whose density vanishes outside $[0,1]$ interval, violating this assumption. Alternatively, in the vicinity of the interval endpoints, the size of the neighborhood admitting the parametrization required for the proof of the core estimate decreases to 0.

We analytically calculated the convolution of $\mathcal{U}(0,1)$ with $\mathcal{N}(0,\delta^2)$ and used it to estimate LID at 1000 points between 0 and 1. We used just two points for linear regression, corresponding to $\delta_1 = \delta$ and $\delta_2 = 1.05\delta$. The estimates for different values

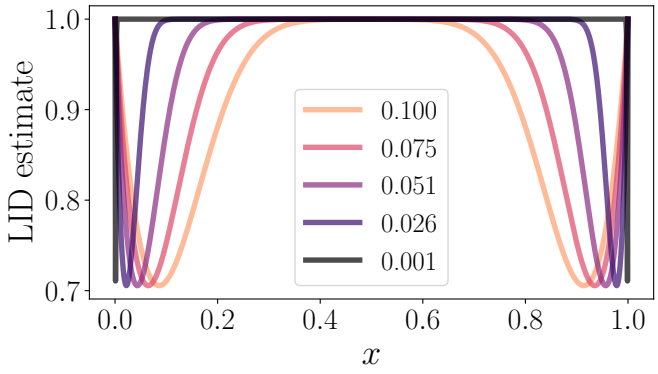


Figure 2.3: LIDL estimates for points from $\mathcal{U}(0,1)$ for different values of δ marked with different colors, as explained in the legend of the plot.

of δ are plotted in Fig. 2.3. We can see that an error is introduced near the boundary as expected. In this case, its maximum value does not depend on the value of δ, and points affected by this problem lie in the part closer than $\sim 4\delta$ to the endpoints of the distribution support.

Normal distribution on a line

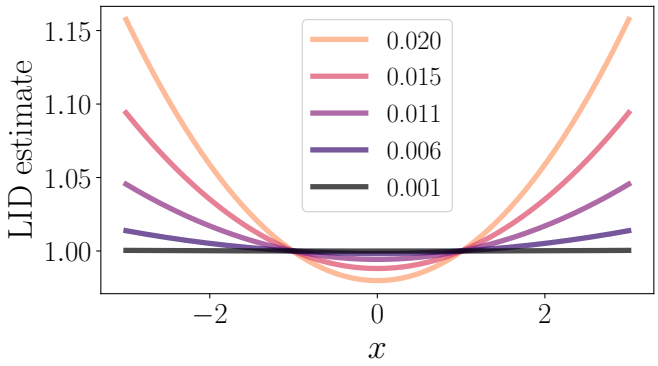


Figure 2.4: LIDL estimates for points from $\mathcal{N}(0,1)$ for different values of δ marked with different colors, as explained in the legend of the plot.

In this example, we study the LID estimates at different points of a line embedded in $\mathbb{R}^D$. In Fig. 2.4 we can see the estimates computed as per the previous example, for a few values of δ. At first glance, it is worrying that the error seems to explode with distance from the mean of the distribution. In Sec. 2.5, we show that the error is quadratic in this distance, and, reassuringly, that its expected value over the whole distribution can be controlled. The reason for this behavior can be traced back to the proof of Lemma 2.2.3 (more specifically eq. (2.9) in the appendix), which depends on the positive constant locally bounding the density from below. In our example, the density decreases as $e^{-t^2/2}$, which produces the quadratic error term (the final error is bounded by $\sum_i |\log C_i|$, where C_i are the multiplicative estimate constants appearing in all the steps of the proof).

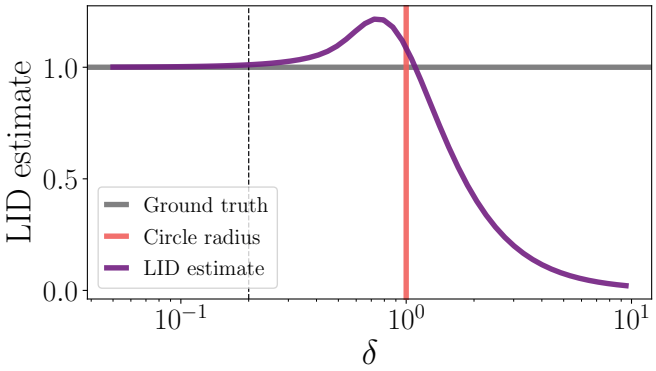


Figure 2.5: LIDL estimate as a function of δ for a uniform density on a unit circle. Vertical line at $\delta = 0.2$.

Uniform density on a curved manifold

The LIDL estimate is affected by the curvature of the manifold, which manifests in the constant C appearing in eq. (2.5), subsequently used in the proofs of Lemmas 2.2.2 and 2.2.4. To see empirically how the curvature influences the LIDL estimate, we numerically computed the convolution of the uniform density on the unit circle embedded in $\mathbb{R}^2$ with the noise distribution $\mathcal{N}(0, \delta^2 I)$ for 2 values of δ similarly as in the previous examples. We calculated LIDL for the range of $\delta \in (0.05, 10)$. We plot the estimate dependence on δ in Fig. 2.5.

We can see that for $\delta \lesssim 0.2$ the estimate error is relatively small. After the positive bias for $\delta < 1$ we can observe a monotonic drop in the estimate until it reaches nearly 0. This is by the effect described in Sec. 2.3 where LIDL was observed to ignore the directions in which the standard deviations were lower than δ.

Manifolds with neighboring components

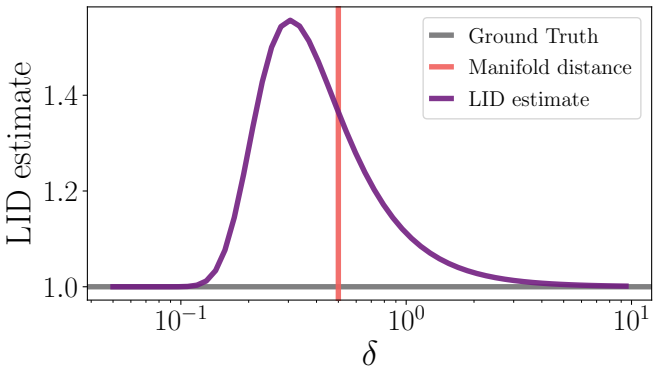


Figure 2.6: LIDL estimate as a function of δ for 2 long 1-dimensional manifolds parallel to each other.

In a real-world setting, it is possible for some connected components of the data manifold S to be close to each other in the observable data space $\mathbb{R}^D$, especially when some features in the dataset have discrete distribution (e.g. height and sex in a medical dataset). In those settings, for values of δ comparable to the distance between the

components, additional bias may be introduced to the estimate. To investigate this we ran an experiment similar to the previous example, but with a uniform distribution supported on the union of two long parallel segments. We then calculated LIDL estimates for the midpoints of those segments, to minimize the error caused by proximity to the boundary. We present the results in Fig. 2.6. We can see positive bias in LIDL estimate appearing as δ is close to the distance between the segments, while for δ much larger than this distance, LIDL seems to view those two segments as a single line.

Synthetic datasets

We ran evaluations of LIDL with density estimates computed using numerical integration on Swiss roll, uniform distribution on a helix, and Gaussians from 10 up to 4000 dimensions. We got almost exact estimates with mean absolute error (MAE) lower than 10^{-4} for every dataset.

2.7 Conclusion

In thic chapter we presented *LIDL*, a simple, theoretically grounded method for estimating local intrinsic dimension from noisy data. The core result (Thm. 2.2.7) shows that

$$\log \rho_\delta(x) \approx (d-D)\, \log \delta + \text{const},$$

so the LID at x is obtained as the slope of a line fitted to $(\log \delta, \log \rho_\delta(x))$ across a few noise scales (Alg. 1; Fig. 2.1). The method is density model-agnostic: we used normalizing flows for density estimation, but any suitable estimator can be plugged in.

A practical advantage is that δ acts as a *scale knob* (Sec. 2.3): increasing δ suppresses structures thinner than the operating scale, which helps ignore small-amplitude noise (Fig. 2.2).

Controlled examples and numerical studies confirmed the theory and highlighted predictable biases near boundaries, in very low-density regions, under curvature, and when nearby components interact (Figs. 2.3–2.6). In practice, using several logarithmically spaced δ's and checking linear fit quality suffices for robust estimates. Overall, LIDL offers a concise, scalable route to LID with clear diagnostics and a tunable operating scale.

Chapter 3

Wiener Process perspective

In this chapter, we point out and exploit the fact that adding Gaussian noise of varying magnitudes can be seen as studying the evolution of the Wiener process describing the diffusion of particles (points of the dataset) in the ambient space. This point of view enables us to employ Fick's Second Law of Diffusion to eliminate time derivatives from mathematical descriptions of state-of-the-art LID algorithms like LIDL and FLIPD Kamkari et al. [2024], and replace them with spatial derivatives.

We begin by recasting dataset perturbations as trajectories of a Wiener process, which unifies the first stage of LIDL, NB, ID–NF, ID–DM, and FLIPD and lets us replace time derivatives with spatial ones via Fick's Second Law (heat equation). In Sec. 3.2 we derive an explicit formula for the Laplacian of the diffused density both off- and on-manifold (Lemma 3.2.1, Cor. 3.2.2). Next, in Sec. 3.3 we connect this PDE view to practical estimators by introducing the reparameterized slope $\beta_t(x) = t\,\Delta\rho_t(x)/\rho_t(x)$ and its limit $\beta(x)$, showing the equivalence with asymptotic slope formulations (Prop. 3.3.1) and giving a closed-form expression in terms of the data density on $\mathbb{R}^d$ (Prop. 3.3.3). Sec. 3.4 then works through canonical cases to expose bias and operating regimes: the constant ("uniform") case on $\mathbb{R}^d$; Gaussian densities with location- and anisotropy-dependent effects (recovering the parabola and stair-step phenomena, Fig. 3.1); uniform laws on an interval and on a hypercube, highlighting boundary effects (Fig. 3.2a); parallel hyperplanes and general convex combinations, where exponential coefficients control cross-component influence (Fig. 3.2b); and intersecting manifolds. We close with brief conclusions outlining how this perspective suggests extensions to curved manifolds and to settings with nearby components.

Wiener process is a stochastic process modeling particle diffusion. Its increments over disjoint time intervals are independent and normally distributed, with variance proportional to time increments. Since in the machine learning community the term *diffusion* is already overloaded, we will stick to Wiener process when speaking of particle diffusion process.

3.1 The new perspective on existing algorithms

In this section we present a new perspective on perturbing datasets, unifying the approaches seen in the algorithms presented by LIDL, NB, ID-NF, ID-DM, FLIPD [Stanczuk et al., 2024, Horvat and Pfister, 2022, 2024, Kamkari et al., 2024]. All these algorithms consist of two stages, the first of which amounts to perturbing the dataset with normally distributed random noise of fixed variance t. In the second stage, each of the algorithms

utilizes the behavior of the perturbed density in the neighborhood of a fixed point under changes in the noise variance.

The first phase of each algorithm can be interpreted as applying the Wiener process to the points in the dataset. Afterward, the resulting set of points is used to train some type of generative model (or models) to estimate the distribution of the dataset undergoing the Wiener process at time t. From the point of view of differential equations, the distribution density function of the diffused dataset is described by Fick's Second Law of Diffusion.

Fick's Second Law of Diffusion. *Let $\rho_t : \mathbb{R}^D \mapsto \mathbb{R}$ denote the probability density function modeling particles undergoing diffusion at time t. Then ρ_t satisfies the differential equation*

$$\frac{d}{dt}\rho_t = C\Delta\rho_t, \tag{3.1}$$

where $C \in \mathbb{R}$, and Δ stands for the standard Laplacian in $\mathbb{R}^D$.

Now, given a dataset embedded in $\mathbb{R}^D$, we assume that it has been drawn from some latent union of submanifolds S endowed with a probability measure p_S (which can be naturally treated as a probability measure on $\mathbb{R}^D$). The goal of Local Intrinsic Dimension estimation is to find out the dimension of S at any point of the dataset.

To model the Wiener process with initial distribution p_S (which is not a function on $\mathbb{R}^D$), let us first define

$$\phi_t^D(x) = (2\pi t)^{-D/2} e^{-\|x\|^2/2t}. \tag{3.2}$$

This is the density of normal distribution on $\mathbb{R}^D$ with covariance matrix tI. It is the fundamental solution of the differential equation given by Fick's Second Law of Diffusion (3.1) with $C = 1/2$. Here, this means that the convolution

$$\rho_t = p_S * \phi_t^D \tag{3.3}$$

is the solution of (3.1) for $t > 0$ and hence it describes the Wiener process starting from the initial probability distribution p_S.

To limit the complexity introduced by curvature, from now on we will consider only flat manifolds. This means that, without loss of generality, we may assume that S is the first factor in product decomposition $\mathbb{R}^D = \mathbb{R}^d \times \mathbb{R}^{D-d}$. We will denote the coordinates of $\mathbb{R}^d$ and $\mathbb{R}^{D-d}$ by x and y, respectively. We will moreover assume that p_S, now a probability distribution on $\mathbb{R}^d$, has a density $\psi\colon \mathbb{R}^d \to \mathbb{R}$.

3.2 Laplacian of the diffused density

Assuming that p_S, considered as a probability distribution on $\mathbb{R}^d$, has a density $\psi\colon \mathbb{R}^d \to \mathbb{R}$, we can separate variables in (3.3), obtaining

$$\rho_t(x, y) = \psi * \phi_t^d(x)\phi_t^{D-d}(y) \tag{3.4}$$

for $(x, y) \in \mathbb{R}^d \times \mathbb{R}^{D-d}$. Moreover, the decomposition $\mathbb{R}^D = \mathbb{R}^d \times \mathbb{R}^{D-d}$ gives rise to the decomposition of the Laplacian on $\mathbb{R}^D$ into Laplacians on the factors, namely $\Delta = \Delta_x + \Delta_y$, by simply grouping the derivatives with respect to x_i and y_i coordinates.

Lemma 3.2.1. *For $t > 0$ and $(x, y) \in \mathbb{R}^D$ we have*

$$\begin{aligned}\Delta\rho_t(x,y) = &\left(\frac{\|y\|^2}{t^2} + \frac{d-D}{t}\right)\rho_t(x,y) \\ &+ \phi_t^{D-d}(y)\Delta_x(\psi * \phi_t^d)(x).\end{aligned} \tag{3.5}$$

Proof. By using the product decomposition (3.4), we get

$$\begin{aligned}&\Delta\rho_t(x,y) = \\ &\psi * \phi_t^d(x)\Delta_y\phi_t^{D-d}(y) + \Delta_x(\psi * \phi_t^d)(x)\phi_t^{D-d}(y)\end{aligned} \tag{3.6}$$

To derive the first term, we note that a direct computation yields

$$\Delta_y\phi_t^{D-d}(y) = \phi_t^{D-d}(y)\left(\frac{\|y\|^2}{t^2} + \frac{d-D}{t}\right). \tag{3.7}$$

□

As a consequence, by putting $y = 0$ and using $\phi_t^{D-d}(0) = (2\pi t)^{(d-D)/2}$ we obtain the following.

Corollary 3.2.2. *For $t > 0$ and $x \in \mathbb{R}^d$ we have*

$$\Delta\rho_t(x,0) = \frac{d-D}{t}\rho_t(x,0) + (2\pi t)^{(d-D)/2}\Delta_x(\psi * \phi_t^d)(x). \tag{3.8}$$

Let us stop here for a moment to discuss the expression $\Delta_x(\psi * \phi_t^d)$. When applying a differential operator to a convolution, under certain regularity conditions we can move it to either of the factors, which can turn out to be quite helpful. As it will turn out in the examples, $\Delta_x\psi * \phi_t^d$ will be the most desired form of $\Delta_x(\psi * \phi_t^d)$.

More precisely, given a convolution $f * g$, and a k-th order differential operator L, the following conditions guarantee that $L(f * g) = Lf * g$ (conditions for the other equality follow from commutativity of convolution):

1. All partial derivatives of f up to order k (including 0-th order, i.e. f itself) exist and are bounded.
2. The function g is integrable, i.e. $\int|g| < \infty$.

The second condition is clearly satisfied by densities of probability distributions, and ϕ_t^d satisfies the first condition. Hence, for any ψ we can write

$$\Delta_x(\psi * \phi_t^d) = \psi * \Delta_x\phi_t^d. \tag{3.9}$$

The second equality however requires additional assumptions;

$$\Delta_x(\psi * \phi_t^d) = \Delta_x\psi * \phi_t^d, \tag{3.10}$$

holds if ψ is bounded and has bounded partial derivatives up to order 2.

3.3 From Wiener process to LID estimation

The findings from the last section can be used to analyze how the LIDL and FLIPD [Kamkari et al., 2024] behaves for some particular cases. The main contribution of Kamkari et al. [2024] is a substantial improvement on the side of density estimation. Therefore, when dealing with perfect density estimators and very small noise differences, both algorithms estimate the same quantity and give the same results from the theoretical perspective. Due to this fact from now on we will be analyzing LIDL, as we want to analyze the aspects of those implementations that do not depend on the problems with density estimation itself.

Reformulating LIDL

Given a point $x \in S$ and a set of times $t_1, \ldots, t_n$, LIDL estimates the linear regression coefficient α of the set of points $(\log \delta_i, \log \rho_{t_i}(x))$, where $\delta_i = \sqrt{t_i}$. We proved that

$$\log \rho_t(x) = (d - D) \log \sqrt{t} + O(1), \tag{3.11}$$

and therefore $\alpha \approx d - D$. We show that if t is small enough, this estimate is accurate.

This procedure can be seen as approximating the asymptotic slope of the parametric curve $(\log \sqrt{t}, \log \rho_t(x))$. In other words, the graph of $s \mapsto \log \rho_{e^{2s}}(x)$ for $s \to -\infty$. Another approach would consider the its derivative. Let us define its reparameterized derivative (with $t = e^{2s}$)

$$\beta_t(x) = \frac{2t}{\rho_t(x)} \frac{d}{dt} \rho_t(x) = \frac{t \Delta \rho_t(x)}{\rho_t(x)}, \tag{3.12}$$

where the last equality comes from the diffusion equation (3.1) with $C = 1/2$. Moreover, denote the asymptotic slope of the aforementioned curve by

$$\beta(x) = \lim_{s \to -\infty} \frac{d}{ds} \log \rho_{e^{2s}}(x) = \lim_{t \to 0^+} \beta_t(x). \tag{3.13}$$

The results presented below are proved in Appendix A.1. The next Proposition shows that the two approaches discussed above are equivalent.

Proposition 3.3.1. *Given a strictly positive differentiable function $f\colon (0, a) \to (0, \infty)$ and a positive real number $\alpha > 0$, the following conditions are equivalent.*

1. *The function f explodes at 0 like $t^{-\alpha}$, i.e. for some positive constants $c, C > 0$ one has $c < t^\alpha f(t) < C$ for some $\epsilon > 0$ and $t \in (0, \epsilon)$.*
2. $\log f(t) = -\alpha \log t + O(1)$.
3. $\lim_{t \to 0^+} \log f(t) / \log t = -\alpha$.
4. $\lim_{t \to 0^+} t f'(t) / f(t) = -\alpha$.

As a consequence, the estimation of Local Intrinsic Dimension using LIDL can be achieved by computing $\beta(x)$, yielding $d = D + \beta(x)$.

Proposition 3.3.2. *For t near 0 the following estimate holds*

$$\log \rho_t(x) = \beta(x) \log \sqrt{t} + O(1). \tag{3.14}$$

The next proposition provides an elegant expression for $\beta_t(x)$, and consequently for $\beta(x)$, expressed in terms of the density ψ on $\mathbb{R}^d$.

Proposition 3.3.3. *For $t > 0$ and $x \in S = \mathbb{R}^d \subseteq \mathbb{R}^D$ we have*

$$\beta_t(x) = d - D + \frac{\Delta_x(\psi * \phi_t^d)(x)}{\psi * \phi_t^d(x)} \cdot t. \tag{3.15}$$

3.4 Examples

From the theoretical considerations of LIDL it follows that $\beta(x) = d - D$ if ψ is sufficiently regular and positive near x. In other words,

$$\lim_{t \to 0^+} \frac{\Delta_x(\psi * \phi_t^d)(x)}{\psi * \phi_t^d(x)} \cdot t = 0. \tag{3.16}$$

Now, we will try to obtain this conclusion directly and calculate bias of LIDL for $t > 0$ in a few special cases by analyzing the behavior of $\beta_t(x)$.

The "uniform distribution" on Euclidean space.

There is no such thing as the uniform distribution on $\mathbb{R}^d$. However, from a purely theoretical viewpoint, in our differential equation approach we don't need the assumption of ϕ being a probability density; it could be any function. And since constant functions are usually the simplest examples, we will now investigate what happens if we put $\psi(x) \equiv 1$ on the whole $\mathbb{R}^d$ space.

Using Proposition 3.3.3 and the fact that ψ has bounded derivatives, this case leaves us with

$$\beta_t(x) = d - D + \frac{\Delta_x \psi * \phi_t^d(x)}{\psi * \phi_t^d(x)} \cdot t = d - D, \tag{3.17}$$

since $\Delta_x \psi \equiv 0$. This expression is constant in t, and in particular its limit at 0 is $\beta(x) = d - D$. In this case, LIDL estimator is not biased for all $t > 0$.

Normal distribution.

Now consider the normal distribution on $\mathbb{R}^d$ with covariance matrix $\Sigma = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, and denote its density function by ψ. The convolution $\psi * \phi_t^d$ is the density of the normal distribution with covariance matrix $\Sigma + tI$. If we simplify notation be putting $\phi_i = \phi^1_{\sigma_i^2 + t}$, we get

$$\psi * \phi_t^d(x) = \prod_{i=1}^{d} \phi_i(x_i). \tag{3.18}$$

To compute the Laplacian of this convolution, note that

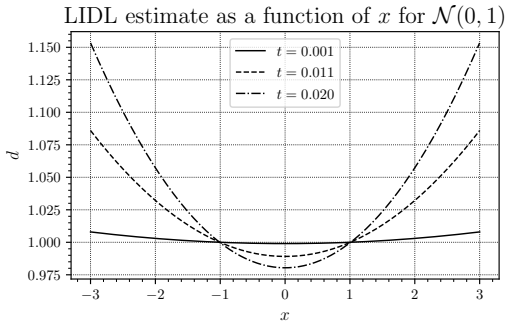
$$\psi * \phi_t^d(x) = \frac{\psi * \phi_t^d(x)}{\phi_i(x_i)} \cdot \phi_i(x_i), \tag{3.19}$$

where the first factor does not depend of x_i, and therefore

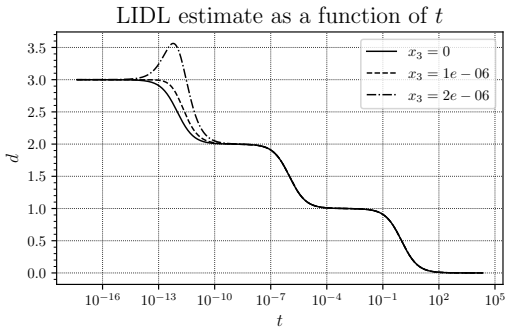
$$\frac{\partial^2(\psi * \phi_t^d)}{\partial x_i^2}(x) = \psi(x) * \phi_t^d(x) \cdot \frac{1}{\phi_i(x_i)} \frac{\partial^2 \phi_i}{\partial x_i^2}(x_i), \tag{3.20}$$

leading to

$$\begin{aligned} \beta_t(x) &= d - D + t \frac{\Delta(\psi * \phi_t^d)(x)}{\psi * \phi_t^d(x)} = \\ &= d - D + t \sum_{i=1}^{d} \frac{1}{\phi_i(x_i)} \frac{\partial^2 \phi_i}{\partial x_i^2}(x_i) \\ &= d - D + t \sum_{i=1}^{d} \frac{x_i^2 - (\sigma_i^2 + t)}{(\sigma_i^2 + t)^2}. \end{aligned} \tag{3.21}$$



(a) Example of the bias of a LIDL estimate for different points from $\mathcal{N}(0,1)$ and for different values of t. This plot recreates a numerical calculations presented in Fig. 2.4.



(b) Plot of a LIDL estimate as a function of t for the distribution $\mathcal{N}(\mathbf{0}, \mathrm{diag}(1, 10^{-6}, 10^{-12}))$ and for three different points $\mathbf{x} = (0, 0, x_3)$, which represents a distance of 0, 1 and 2 σ_3 from 0 on 3rd dimension. Should be compared with Fig. 2.2.

Figure 3.1: LIDL estimates for Gaussian distributions.

It is easy to see that the second derivatives of ϕ_i are continuous in $t > -\sigma_i^2$, so the sum in the above expression has finite limit for $t \to 0$, and therefore $\beta(x) = d - D$.

In the special case where $\Sigma = \sigma^2 I$, these calculations simplify further, as $\psi * \phi_t^d = \phi_{\sigma^2+t}^d$, and since

$$\Delta_x \phi_{\sigma^2+t}^d(x) = \left(\frac{\|x\|^2}{(\sigma^2+t)^2} - \frac{d}{\sigma^2+t} \right) \phi_{\sigma^2+t}^d(x), \tag{3.22}$$

we have

$$\beta_t(x) = d - D + \left(\frac{\|x\|^2}{(\sigma^2+t)^2} - \frac{d}{\sigma^2+t} \right) t. \tag{3.23}$$

These results express analytically the experimental observations from Sec. 2.6 and FLIPD paper, as can be verified by looking at Fig. 3.1a. We can observe, that if we move to the regions of very low probability for a Gaussian, it generates very high positive bias, which may highly overestimate the true LID (also observed as a *bump* at $t = 10^{-12}$ in Fig. 3.1b). Luckily, most of the points in our dataset come from the region of high probability, but we should be less certain of the estimates for points from low probability regions.

Additionally, one can observe that curves obtained numerically are somewhat flatter than one in this study. The fact that the derivative was approximated by linear regression on numerically calculated densities – which may lead to slightly different results – might be a possible reason.

Arbitrary distribution with sufficiently *nice* density.

By this point, the notion of *nice* density is shall be more clear. We want to be able to use the equality

$$\Delta_x(\psi * \phi_t^d) = \Delta_x \psi * \phi_t^d. \tag{3.24}$$

To do so, we need ψ to be bounded, twice differentiable, and have bounded first and second-order partial derivatives. We will also require ψ to have *continuous* second-order partial derivatives. This is not a severe restriction, as numerous distributions satisfy these properties – including the normal distribution or more generally, mixtures of Gaussians.

In this case, we have

$$\beta_t(x) = d - D + \frac{\Delta_x \psi * \phi_t^d(x)}{\psi * \phi_t^d(x)} \cdot t, \tag{3.25}$$

however this time $\Delta_x \psi$ is some arbitrary continuous function. Being differentiable, ψ is also continuous, and we can use the general fact that for a bounded continuous function, f on $\mathbb{R}^d$ one has

$$\lim_{t \to 0^+} f * \phi_t^d(x) = f(x). \tag{3.26}$$

This gives us, for x such that $\psi(x) > 0$,

$$\lim_{t \to 0^+} \frac{\Delta_x \psi * \phi_t^d(x)}{\psi * \phi_t^d(x)} \cdot t = \frac{\Delta_x \psi(x)}{\psi(x)} \lim_{t \to 0^+} t = 0, \tag{3.27}$$

and again $\beta(x) = d - D$. It has been already proven that in this case β yields a correct estimate of dimension, circumventing complexities of LIDL proofs.

It is worth noting, that when $\Delta_x \psi = 0$, the estimate is accurate. It is the case for the aforementioned "uniform distribution" on $\mathbb{R}^d$, but it is also true if locally the density is a linear function of x. In Fig. 3.1a we can observe that for $x \approx \pm 1$ (Laplacian of a Gaussian density equals 0 at these points) and small values of t, the estimate is accurate.

Uniform distribution supported on an interval.

Now consider an example where the density is not differentiable – the uniform distribution on an interval $[a, b] \subset \mathbb{R}$, i.e.

$$\psi(x) = \frac{1}{b-a} \chi_{[a,b]}(x), \tag{3.28}$$

where $\chi_A(s)$ is the indicator function of the set A, equal to 1 on A and 0 outside A. In the next example, we will generalize this to a hypercube, but the core observations can be made in this simpler 1-dimensional case.

The difficulty introduced by the non-differentiability of ψ is we are no longer allowed to move the Laplacian inside the convolution to get

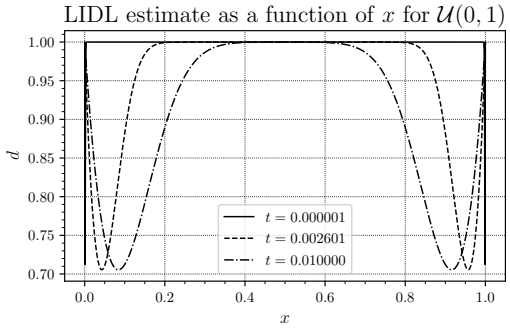
$$\Delta_x(\psi * \phi_t) = \Delta_x \psi * \phi_t \tag{3.29}$$

(we omit the superscript $d = 1$ from ϕ_t) – as tempting as it might be. Therefore, a different manner of proceeding is needed. We may still move the Laplacian to ϕ_t. In the 1-dimensional case, Δ_x is simply the second derivative, and since

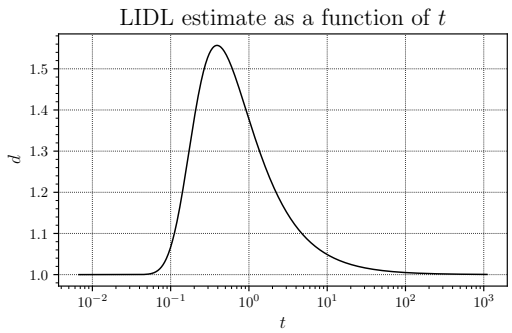
$$\phi_t'(u) = -u\phi_t(u)/t \tag{3.30}$$

we have

$$\begin{aligned} \Delta_x(\psi * \phi_t)(x) &= \frac{1}{b-a} \int_{x-b}^{x-a} \phi_t''(u)\, du \\ &= \frac{\phi_t'(x-a) - \phi_t'(x-b)}{b-a} = \\ &= \frac{(x-b)\phi_t(x-b) - (x-a)\phi_t(x-a)}{t(b-a)}, \end{aligned} \tag{3.31}$$



(a) Example of the bias of a LIDL estimate for different points from $\mathcal{U}(0,1)$ and values of t. This plot recreates a numerical calculations presented in Fig. 2.3



(b) LIDL estimate as a function of t for a point from parallel 1D manifolds separated by a distance of 1 with uniform distribution on them. Similar to result from Fig. 2.6

Figure 3.2: LIDL estimates.

Expanding the denominator in a similar fashion yields

$$\beta_t(x) = d - D + \frac{(x-b)\phi_t(x-b) - (x-a)\phi_t(x-a)}{\Phi_t(x-a) - \Phi_t(x-b)}, \tag{3.32}$$

where Φ_t is the cumulative distribution function corresponding to the density ϕ_t. In particular for $x \in (a,b)$ we see that since $x-b < 0 < x-a$, when $t \to 0^+$, the denominator tends to 1, while both terms of the numerator tend to 0, leaving us with $d - D$. LIDL estimate curves for this case for different values of t are plotted in Fig. 3.2a.

Uniform distribution supported on a hypercube.

Let us now consider a more general case – the uniform distribution on a hypercube $[a_1, b_1] \times \cdots \times [a_d, b_d] \subset \mathbb{R}^d$. We have

$$\psi(x) = \prod_{i=1}^{d} \frac{1}{b_i - a_i} \chi_{[a_i,b_i]}(x_i), \tag{3.33}$$

Denote

$$\psi_i(s) = \frac{1}{b_i - a_i} \chi_{[a_i,b_i]}(s), \tag{3.34}$$

and observe that since $\phi_t^d(x)$ is the product of $\phi_t(x_i)$, we have

$$\psi * \phi_t^d(x) = \prod_{i=1}^{d} \psi_i * \phi_t(x_i). \tag{3.35}$$

By directly computing the derivatives, we obtain

$$\frac{\Delta_x \psi * \phi_t^d(x)}{\psi * \phi_t^d(x)} = \sum_{i=1}^{d} \frac{(\psi_i * \phi_t)''(x_i)}{\psi_i * \phi_t(x_i)} = \sum_{i=1}^{d} \frac{\psi_i * \phi_t''(x_i)}{\psi_i * \phi_t(x_i)}, \tag{3.36}$$

reducing our problem to the 1-dimensional variant we have dealt with in the preceding example. Summing up, we have

$$\begin{aligned} \beta_t(x) &= d - D \\ &+ \sum_{i=1}^{d} \frac{(x_i - b_i)\phi_t(x_i - b_i) - (x_i - a_i)\phi_t(x_i - a_i)}{\Phi_t(x_i - a_i) - \Phi_t(x_i - b_i)}, \end{aligned} \tag{3.37}$$

and the asymptotic behavior with $t \to 0^+$ follows the 1-dimensional case.

Union of two parallel hyperplanes.

Suppose that S is a union of two parallel hyperplanes, $S_1 = \mathbb{R}^d$ and $S_2 = v + \mathbb{R}^d$, where $v \perp \mathbb{R}^d$. Moreover, assume that $p_S = (1-\lambda)p_1 + \lambda p_2$ is a convex combination of probability measures p_i supported on S_i, with densities $\psi_i \colon \mathbb{R}^d \to \mathbb{R}$ (we identify S_2 with $\mathbb{R}^d$ through the map $x \mapsto x + v$). In this case, for $x \in S_1$ we have

$$\beta_t^1(x) = d - D + \frac{\Delta_x(\psi_1 * \phi_t^d)(x)}{\psi_1 * \phi_t^d(x)} \cdot t, \tag{3.38}$$

and by Lemma 3.2.1, and the observation that

$$\rho_t^2(x) = \psi_2 * \phi_t^d(x)\phi_t^{D-d}(v), \tag{3.39}$$

we get

$$\beta_t^2(x) = d - D + \frac{\|v\|^2}{t} + \frac{\Delta_x(\psi_2 * \phi_t^d)(x)}{\psi_2 * \phi_t^d(x)} \cdot t. \tag{3.40}$$

Here, we see that for an off-manifold point $x \notin S_2$, the expression for $\beta_t^2(x)$ contains a summand $\|v\|^2/t$ that explodes at 0, and if the last term is under control, $\beta^2(x)$ is infinite. However, by Lemma 3.4.2, the coefficient of $\beta_t^2(x)$ in the expansion of $\beta_t(x)$ from Lemma 3.4.1 decreases exponentially in $1/t$, neutralizing this divergence.

For the remainder of this example, let us assume $\psi_1 = \psi_2 = \psi$. In this case, we may apply multiple simplifications, in particular

$$\beta_t^2(x) = \beta_t^1(x) + \frac{\|v\|^2}{t}, \tag{3.41}$$

and moreover

$$\begin{aligned}
\frac{\rho_t^2(x)}{\rho_t(x)} = & \\
&= \frac{\psi * \phi_t^d(x)\phi_t^{D-d}(v)}{(1-\lambda)\psi * \phi_t^d(x)\phi_t^{D-d}(0) + \lambda\psi * \phi_t^d(x)\phi_t^{D-d}(v)} \\
&= \frac{\phi_t^{D-d}(v)}{(1-\lambda)\phi_t^{D-d}(0) + \lambda\phi_t^{D-d}(v)} \\
&= \frac{1}{(1-\lambda)e^{\|v\|^2/2t} + \lambda}.
\end{aligned} \tag{3.42}$$

Since $\frac{\lambda\rho_t^1(x)}{\rho_t(x)} = 1 - \frac{\lambda\rho_t^2(x)}{\rho_t(x)}$ we can simplify the expression for $\beta_t(x)$ from Lemma 3.4.1, yielding

$$\begin{aligned}
\beta_t(x) &= \beta_t^1(x) + \frac{\|v\|^2}{t} \cdot \frac{\lambda\rho_t^2(x)}{\rho_t(x)} \\
&= \beta_t^1(x) + \frac{\lambda\|v\|^2}{t\left((1-\lambda)e^{\|v\|^2/2t} + \lambda\right)}.
\end{aligned} \tag{3.43}$$

To give a concrete example, if $\psi = \phi_{\sigma^2}^d$, then

$$\begin{aligned}
\beta_t(x) = d - D + &\left(\frac{\|x\|^2}{(\sigma^2+t)^2} - \frac{d}{\sigma^2+t}\right)t \\
&+ \frac{\lambda\|v\|^2}{t\left((1-\lambda)e^{\|v\|^2/2t} + \lambda\right)}.
\end{aligned} \tag{3.44}$$

Another interesting example occurs when the data on both parallel manifolds follow uniform density functions. Although the derivation in Eq. (3.56) requires the density functions to be probability distributions, this scenario can be simulated by considering two Gaussian distributions with relatively large standard deviations. This yields the following formula for $\beta_t(x)$, presented in Fig. 3.2b for $v = 1$, $\lambda = \frac{1}{2}$:

$$\beta_t(x) = d - D + \frac{\lambda \|v\|^2}{t\left((1-\lambda)e^{\|v\|^2/2t} + \lambda\right)}. \tag{3.45}$$

Union of two intersecting manifolds.

In this example we will consider a manifold S decomposing into a union of two components S_1 and S_2, intersecting at a point $x \in S_1 \cap S_2$. Denote by d_i the dimension of S_i. As before, let $p_S = \lambda p_1 + (1-\lambda)p_2$. Moreover, suppose that

$$\beta_t^i(x) = d_i - D + E_i(t), \tag{3.46}$$

where $E_i(t)$ expresses the error of β_t^i in estimating the dimension of S_i, and $\lim_{t\to 0^+} E_i(t) = 0$. By Lemma 3.4.1 we have

$$\beta_t(x) = \left(\frac{\lambda\rho_t^1(x)}{\rho_t(x)}d_1 + \frac{(1-\lambda)\rho_t^2(x)}{\rho_t(x)}d_2\right) - D + E(t), \tag{3.47}$$

where the error term is a convex combination of E_1, and E_2, and thus is bounded by their maximum,

$$\begin{aligned} E(t) &= \frac{\lambda\rho_t^1(x)}{\rho_t(x)}E_1(t) + \frac{(1-\lambda)\rho_t^2(x)}{\rho_t(x)}E_2(t) \\ &\leq \max\{E_1(t), E_2(t)\}. \end{aligned} \tag{3.48}$$

In particular, it also vanishes as $t \to 0^+$.

The value of LID at x estimated by $\beta_t(x)$ lies between d_1 and d_2, and is controlled by the asymptotic of $\lambda\rho_t^1(x)/\rho_t(x)$. If $d_1 = d_2 = d$, then it is also equal to d.

Convex combinations of distributions

Our new approach allows to easily analyze the behavior of LIDL on unions of manifolds. Suppose that $S = \bigcup_i S_i$ is a finite union of manifolds. The measure p_S can be then decomposed into a convex combination of probability measures p_i supported on S_i, namely

$$p_S = \sum_i \lambda_i p_i, \tag{3.49}$$

where $\lambda_i > 0$ and $\sum_i \lambda_i = 1$. With each (S_i, p_i) we may associate their corresponding diffused density ρ_t^i, reparameterized slope β_t^i, and its limit β^i through equations (3.3), (3.12), and (3.13). Due to bilinearity of convolution, analogous decomposition holds for ρ_t, namely

$$\rho_t(x) = \sum_i \lambda_i \rho_t^i(x). \tag{3.50}$$

Using these, we may now attempt to reduce the problem to the study of individual components.

In the following discussion it turns out that the decomposition of S into a union of manifolds, and the fact that p_i are supported on the components of this decomposition, are of no importance. The only essential assumption is the convex combination decomposition (3.49).

Lemma 3.4.1. *A convex combination decomposition $p_S = \sum_i \lambda_i p_i$, gives rise to a decomposition of $\beta_t(x)$ into a convex combination of $\beta_t^i(x)$,*

$$\beta_t(x) = \sum_i \frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} \beta_t^i(x). \tag{3.51}$$

In particular, if all the limits below exist, one has

$$\beta(x) = \sum_i \left(\lim_{t \to 0^+} \frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} \right) \beta^i(x). \tag{3.52}$$

Proof. The desired decomposition can be obtained through direct expansion of the definition of β_t. We have

$$\begin{aligned} \beta_t(x) &= \sum_i \frac{t\lambda_i \Delta \rho_t^i(x)}{\rho_t(x)} \\ &= \sum_i \frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} \cdot \frac{t \Delta \rho_t^i(x)}{\rho_t^i(x)} \\ &= \sum_i \frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} \beta_t^i(x). \end{aligned} \tag{3.53}$$

The second assertion follows by taking the limit. □

Now take a closer look at the coefficients $\lambda_i \rho_t^i(x)/\rho_t(x)$.

Lemma 3.4.2. *Suppose that $p_S = \sum_i \lambda_i p_i$ is a convex combination decomposition, and for some $R > r > 0$ there exist $i \neq j$ such that $p_i(B(x,R)) = 0$ and $p_j(B(x,r)) = C > 0$. Then the following estimate holds.*

$$\frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} \leq \frac{\lambda_i}{\lambda_i + C\lambda_j e^{(R^2 - r^2)/2t}}. \tag{3.54}$$

Moreover,

$$\lim_{t \to 0^+} \frac{\lambda_i \rho_t^i(x)}{\rho_t(x)} = 0, \tag{3.55}$$

and p_i does not contribute to $\beta(x)$.

Proof. First, observe that the integral defining $\rho_t^i(x)$ can be estimated by the supremum of the integrand $\phi_t^D(x-y)$ over the support of p_i, contained in $B(x,R)^c$, namely

$$\begin{aligned} \rho_t^i(x) &= \int \phi_t^D(x-y)\, dp_i(y) \\ &\leq \sup_{y \in B(x,R)^c} \phi_t^D(x-y) = (2\pi t)^{-D/2} e^{-R^2/2t}. \end{aligned} \tag{3.56}$$

Therefore, the ratio $\rho_t^j(x)/\rho_t^i(x)$ can be bounded from below as

$$\begin{aligned}\frac{\rho_t^j(x)}{\rho_t^i(x)} &\geq \int e^{(R^2-\|x-y\|^2)/2t}\, dp_j(y) \\ &\geq \int_{B(x,r)} e^{(R^2-\|x-y\|^2)/2t}\, dp_j(y) \\ &\geq \int_{B(x,r)} e^{(R^2-r^2)/2t}\, dp_j(y) = Ce^{(R^2-r^2)/2t}.\end{aligned} \tag{3.57}$$

Combining both these estimates, we end up with

$$\begin{aligned}\frac{\rho_t(x)}{\lambda_i\rho_t^i(x)} &= 1 + \sum_{k\neq i} \frac{\lambda_k\rho_t^k(x)}{\lambda_i\rho_t^i(x)} \\ &\geq 1 + \frac{\lambda_j\rho_t^j(x)}{\lambda_i\rho_t^i(x)} \geq 1 + \frac{\lambda_j}{\lambda_i} Ce^{(R^2-r^2)/2t}.\end{aligned} \tag{3.58}$$

Inverting and taking the limit yields the desired assertions. □

3.5 Conclusion

This chapter recasts Gaussian perturbations as trajectories of a *Wiener process* and uses Fick's Second Law (heat equation) to replace time derivatives with spatial ones. This unifies the first stage of LIDL, NB, ID–NF, ID–DM, and FLIPD, and leads to closed-form expressions for the Laplacian of the diffused density on- and off-manifold (Sec. 3.2). We introduce the reparameterized slope

$$\beta_t(x) = t\,\Delta\rho_t(x)/\rho_t(x), \qquad \beta(x) = \lim_{t\to 0^+} \beta_t(x),$$

and show its equivalence to asymptotic slope formulations used in LID estimators, with a simple expression on $\mathbb{R}^d$ (Sec. 3.3).

Canonical examples (Sec. 3.4) recover and explain observed behaviors: the Gaussian "parabola" and anisotropy "stair-steps," boundary effects for uniform laws on intervals/hypercubes, cross-component influence that decays exponentially with separation for parallel manifolds, and weighted effects at intersections. Practically, this PDE view (i) turns LID estimation into ratios of spatial derivatives that any density model can supply, and (ii) clarifies operating regimes and biases at finite t. Extensions to curved manifolds and more complex multi-component settings follow naturally from the same framework.

Chapter 4

Comparison with classical algorithms

In this chapter, we compare LIDL with other classical algorithms on classical benchmarks, investigate its behavior with imperfect density estimators, and run it on synthetic datasets. Details of the training procedure can be found in Sec. A.2. For clarity, we benchmark on datasets with known ground truth, use normalizing flows (unless stated otherwise), and report relative bias/MAE and variance over multiple runs; methods that do not scale in practice are noted where excluded.

We proceed as follows: first, Sec. 4.1 briefly recalls the normalizing-flow models we use as density estimators (MAF, RQ-NSF, Glow) and our training setup. Next, we lay out the evaluation protocol and baselines drawn from `scikit-dimension`, noting that FisherS and DANCo were excluded due to memory/runtime limits at scale. We then present three focused comparisons: (i) *Scalability* on high-dimensional uniform/Gaussian data up to 4K dimensions (Fig. 4.1; Tables 4.1, 4.2, A.2); (ii) *Multiscale manifolds*, showing how anisotropy and sample size affect estimates and how LIDL's explicit scale stabilizes results (Fig. 4.2); and (iii) *Curved and union manifolds*—Swiss roll, helix, sphere, and the lollipop dataset mixing 0-, 1-, and 2-D components—highlighting per-component performance (Fig. 4.3; Tabless. 4.1, 4.2).

4.1 Normalizing Flows

In our work we use Normalizing Flows (NF) as a density estimators for most of the time. NF are very flexible tools for approximating probability distributions. They use parametrized nonlinear invertible transformation f_θ and change of variable formula to transform a simple density $\pi(z)$ into a more complicated one. NF are trained using gradient-based methods (e.g. SGD) to maximize log-likelihood of the data

$$\max_\theta \sum_i \log q(x_i)$$

where

$$q(x) = \pi(f_\theta(x)) \left| \det \frac{\partial f_\theta(x)}{\partial x} \right|.$$

We used MAF [Papamakarios et al., 2017], RQ-NSF [Durkan et al., 2019] and Glow [Kingma and Dhariwal, 2018] models in our experiments. More detailed introduction to normalizing flows can be found in Dinh et al. [2014].

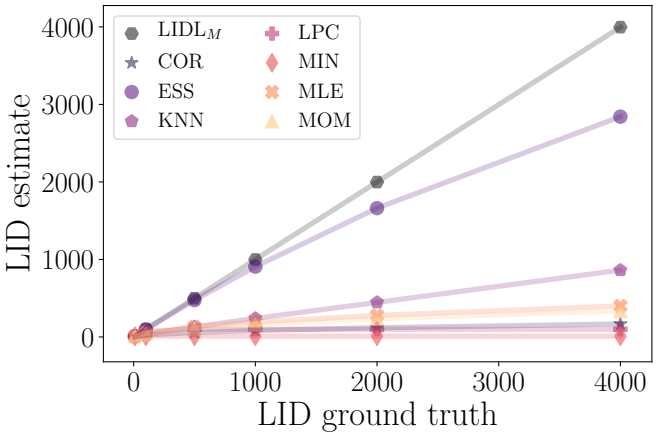


Figure 4.1: LID estimates for d dimensional uniform distribution on a hypercube. More results and abbreviation explanations can be found in Tables 4.1, 4.2 and A.2. The dimensionality d of the distribution is plotted on the horizontal axis and the estimates for different algorithms on the vertical axis.

4.2 Comparison on synthetic datasets

We collated LIDL with other LID estimation algorithms from `scikit-dimension` Python library [Bac et al., 2021b], which covers all of the important algorithms for LID estimation, and compared them in three different aspects: *1.* Scalability, *2.* Multidimensional and curved manifolds, *3.* Multiscale manifolds.

We excluded FisherS and DANCo algorithms because they do not scale well to higher-dimensional settings. FisherS suffered from memory problems on medium datasets, and DANCo had unfeasibly long runtimes (multiple weeks) on the thousand-dimensional datasets. According to the convention in the field, we choose to make comparison only on synthetic datasets, because we have ground truth for them.

Impact of linear regression on LIDL estimate

Because our estimate depends on linear regression algorithm in order to estimate β, it may suffer from the same issues as any regression coefficient estimation algorithm Li [1985], so in the future, more robust algorithm for linear regression estimation may be considered. Because we estimate only the rate of change, and not the constant from linear regression equation, LIDL is prone to biased log-likelihood estimates, and noise added to log-likelihood estimate only affects the variance of the estimate.

Scalability

To test scalability we ran all algorithms on standard multidimensional uniform and normal distributions up to 4K dimensions. Detailed results of the comparison are gathered in Tables 4.1 and 4.2 (starting from the 7-th row). Each dataset consisted of 10K data points and each algorithm was run 5 times on different samples from the distribution. For each run, we calculated differences between true LID and estimate and averaged it over 5 runs. Then we divided the result by the average manifold dimensionality for each dataset, getting a relative bias of each algorithm. In subsequent tables, we report relative MAE and estimate standard deviation for the same procedure. From those tables, we can

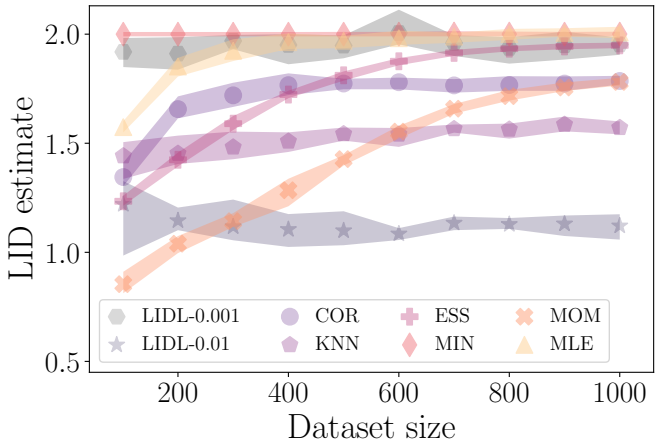


Figure 4.2: LID estimates for uniform distribution on a rectangle with edge lengths equal to 0.1 and 0.01. The size of the dataset is plotted on the horizontal axis and the estimate (with respective 95% confidence intervals) on the vertical axis. For most algorithms (except LIDL, KNN and MIN), we can see a disturbing phenomenon: the estimate depends on the sample size. LIDL-δ stands for LIDL with MAF density estimator and scale parameter δ. Other abbreviations are explained in Table A.2.

clearly see, that although in many cases LIDL does not have the lowest error and bias, for almost all datasets the results are in the $\pm 5\%$ range. Other algorithms fail to accurately estimate dimensions exceeding 100. One exception is ESS, which stands out from the rest but remains inferior to LIDL. We plot LID estimates for some of the algorithms (we omitted few for the sake of clarity) for multidimensional uniform distributions in Fig. 4.1. All the abbreviations used in the plot are explained in Table A.2.

Multiscale manifolds

A useful LID estimation algorithm should operate properly on multiscale manifolds. In this section, we compare existing LID methods and LIDL on highly non-isotropic datasets. We observed that most of the algorithms with the same scale parameters (or those without such parameters, like ESS) give different results for different sizes of the dataset. We hypothesize that this may be caused by violating assumptions about the local uniformity of the distribution, but we did not investigate it further. Only LIDL, MiNDML, DANCo, and KNN give stable estimates for different dataset sizes. We plot those results for selected algorithms in Fig. 4.2. For both scale parameter values, LIDL gives stable estimates for different dataset sizes. The rest of the unplotted algorithms also give unstable estimates, and we omitted them only to make the plot more readable.

Curved manifolds and unions of manifolds

We tested LIDL and other algorithms on some smaller but more complicated manifolds. We used three classical benchmarks from Kleindessner and Luxburg [2015]: the Swiss roll dataset, uniform density on a helix, and uniform density on a sphere. These datasets lie on a curved manifolds (2-, 1- and 7-dimensional respectively) which may cause difficulties with fitting density estimators. Results of those experiments can be found in rows 4-7 of Tables 4.1 and 4.2. The results for LIDL are decent (Relative bias less than 0.05 and

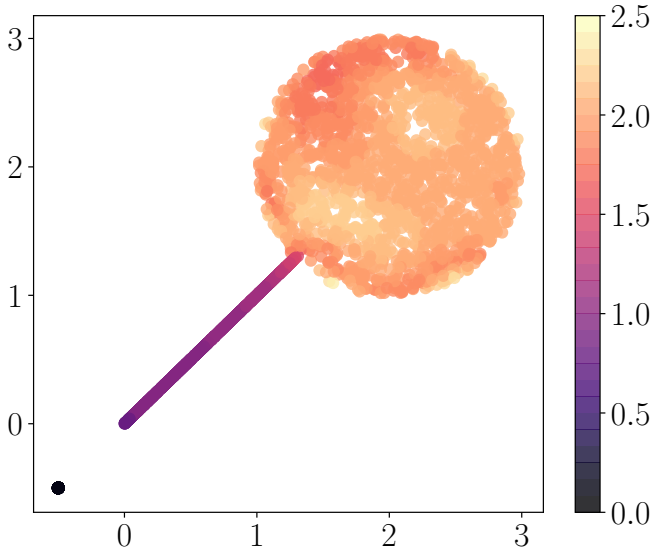


Figure 4.3: Points from the lollipop benchmark dataset and LIDL (with MAF) estimates for those points.

relative MAE less than 0.06), but lPCA and ESS gave estimates with relative bias and MAE less than 0.01. For perfect density estimates LIDL gives almost perfect estimates on those datasets, as presented in Sec. 2.6.

None of the above datasets however consisted of components of different dimensions, which may be the case for many real-world datasets. We used a *lollipop dataset*, which is composed of 0, 1, and 2-dimensional components. The dataset and its corresponding LIDL estimates are plotted in Fig. 4.3. On the 2- and 1-dimensional parts, many algorithms achieved good results, some even better than LIDL, but the 0-dimensional component, which consisted of replicas of the same point, caused most problems for other algorithms.

When algorithms tried to estimate LID for this 0-dimensional part, only lPCA and LIDL were able to estimate its dimensionality properly, and almost all other algorithms failed to converge. When we jittered those points a little with $\mathcal{N}(0, 10^{-6})$, almost all of the algorithms converged but all of them yielded estimates close to 2. Thanks to the possibility of setting operating scale in LIDL, we could estimate the latter dimension correctly, regardless of noise in the data. Results for each component of the manifold treated separately can be found in the first 3 rows of Tables 4.1 and 4.2.

Table 4.1: Relative bias of LID estimates. All algorithm names explained in Table A.2

Distribution	LID	LIDL_M	LIDL_R	COR	ESS	KNN	LPC	MAD	MIN	MLE	MOM	TLE	TWO
Lollipop in $\mathbb{R}^2$	0	**0.00**	**0.00**	1.67	1.67	1.60	1.67	1.82	1.67	1.80	1.74	-	1.65
Lollipop in $\mathbb{R}^2$	1	**0.00**	**0.01**	**0.00**	**0.00**	0.58	**0.01**	0.10	**0.00**	**0.05**	**0.00**	-	**0.00**
Lollipop in $\mathbb{R}^2$	2	**-0.00**	**-0.00**	**-0.01**	**-0.00**	-0.07	**0.00**	0.10	**-0.00**	0.08	**0.01**	-	**-0.03**
$\mathcal{U}$ on helix in $\mathbb{R}^3$	1	**0.01**	**0.00**	**0.00**	**0.00**	0.68	**0.00**	0.12	**0.00**	0.06	**0.00**	**0.00**	**0.00**
$\mathcal{U}$ on $S^7 \subseteq \mathbb{R}^8$	7	**-0.00**	**0.00**	-0.28	**0.00**	-0.37	**0.00**	**0.03**	-0.18	**0.02**	**-0.04**	0.08	-0.13
Swiss roll in $\mathbb{R}^3$	2	**0.04**	**0.01**	**-0.00**	**0.00**	**-0.05**	**0.00**	0.06	**0.00**	0.05	**0.00**	0.05	**-0.01**
$\mathcal{N}_{10} \subseteq \mathbb{R}^{10}$	10	**0.00**	**0.00**	-0.40	**-0.00**	-0.47	**0.00**	**0.02**	-0.25	**0.01**	-0.07	**0.01**	-0.16
$\mathcal{N}_{100} \subseteq \mathbb{R}^{100}$	100	**-0.00**	**0.00**	-0.78	**-0.01**	-0.66	-0.28	-0.51	-0.90	-0.50	-0.57	-0.56	-0.60
$\mathcal{N}_{1000} \subseteq \mathbb{R}^{1000}$	1000	**0.00**	**0.00**	-0.93	-0.09	-0.74	-0.90	-0.83	-0.99	-0.82	-0.85	-0.85	-0.86
$\mathcal{N}_{4000} \subseteq \mathbb{R}^{4000}$	4000	**-0.00**	-	-0.96	-0.29	-0.77	-0.98	-0.91	-1.00	-0.91	-0.92	-0.93	-0.93
$\mathcal{N}_{10} \subseteq \mathbb{R}^{20}$	10	**0.00**	**0.01**	-0.40	**-0.00**	-0.25	**0.00**	**0.02**	-0.25	**0.01**	-0.07	**0.01**	-0.16
$\mathcal{N}_{100} \subseteq \mathbb{R}^{200}$	100	**0.04**	**0.03**	-0.78	**-0.01**	-0.46	-0.28	-0.51	-0.90	-0.50	-0.57	-0.56	-0.60
$\mathcal{N}_{1000} \subseteq \mathbb{R}^{2000}$	1000	0.11	0.30	-0.93	-0.09	-0.52	-0.90	-0.83	-0.99	-0.82	-0.85	-0.85	-0.86
$\mathcal{N}_{2000} \subseteq \mathbb{R}^{4000}$	2000	0.11	-	-0.95	-0.17	-0.55	-0.95	-0.88	-0.99	-0.87	-0.89	-0.90	-0.90
$\mathcal{U}_{10} \subseteq \mathbb{R}^{10}$	10	**-0.04**	**-0.04**	-0.39	-0.07	-0.47	**0.00**	-0.10	-0.29	-0.11	-0.17	-0.07	-0.22
$\mathcal{U}_{100} \subseteq \mathbb{R}^{100}$	100	**0.00**	**0.01**	-0.75	**-0.02**	-0.67	-0.28	-0.50	-0.90	-0.49	-0.56	-0.53	-0.59
$\mathcal{U}_{1000} \subseteq \mathbb{R}^{1000}$	1000	**-0.00**	**0.00**	-0.92	-0.09	-0.76	-0.90	-0.81	-0.99	-0.81	-0.84	-0.84	-0.85
$\mathcal{U}_{4000} \subseteq \mathbb{R}^{4000}$	4000	**-0.00**	-	-0.96	-0.29	-0.78	-0.98	-0.90	-1.00	-0.90	-0.92	-0.92	-0.92

Table 4.2: Relative MAE of LID estimates. All algorithm names explained in Table A.2

Distribution	LID	LIDL_M	LIDL_R	COR	ESS	KNN	LPC	MAD	MIN	MLE	MOM	TLE	TWO
Lollipop in $\mathbb{R}^2$	0	**0.00**	**0.00**	1.67	1.67	1.60	1.67	1.82	1.67	1.80	1.74	-	1.65
Lollipop in $\mathbb{R}^2$	1	**0.00**	**0.01**	**0.00**	**0.00**	0.58	**0.01**	0.12	**0.00**	**0.05**	**0.00**	-	**0.00**
Lollipop in $\mathbb{R}^2$	2	**0.01**	**0.01**	**0.01**	**0.00**	0.62	**0.00**	0.43	**0.00**	0.25	**0.03**	-	**0.05**
$\mathcal{U}$ on helix in $\mathbb{R}^3$	1	**0.01**	**0.00**	**0.00**	**0.00**	0.68	**0.00**	0.14	**0.00**	0.06	**0.00**	**0.00**	**0.00**
$\mathcal{U}$ on $S^7 \subseteq \mathbb{R}^8$	7	**0.00**	**0.00**	0.28	**0.00**	0.44	**0.00**	0.27	0.18	0.18	0.08	0.17	0.15
Swiss roll in $\mathbb{R}^3$	2	0.06	**0.01**	**0.00**	**0.00**	0.37	**0.00**	0.25	**0.00**	0.14	**0.02**	0.06	**0.03**
$\mathcal{N}_{10} \subseteq \mathbb{R}^{10}$	10	**0.00**	**0.01**	0.40	**0.00**	0.47	**0.00**	0.27	0.25	0.19	0.11	0.16	0.17
$\mathcal{N}_{100} \subseteq \mathbb{R}^{100}$	100	**0.00**	**0.02**	0.78	**0.02**	0.66	0.28	0.52	0.90	0.50	0.57	0.56	0.60
$\mathcal{N}_{1000} \subseteq \mathbb{R}^{1000}$	1000	**0.00**	**0.01**	0.93	0.09	0.74	0.90	0.83	0.99	0.82	0.85	0.85	0.86
$\mathcal{N}_{4000} \subseteq \mathbb{R}^{4000}$	4000	**0.01**	-	0.96	0.29	0.77	0.98	0.91	1.00	0.91	0.92	0.93	0.93
$\mathcal{N}_{10} \subseteq \mathbb{R}^{20}$	10	**0.00**	**0.01**	0.40	**0.00**	0.68	**0.00**	0.27	0.25	0.19	0.11	0.16	0.17
$\mathcal{N}_{100} \subseteq \mathbb{R}^{200}$	100	**0.04**	**0.03**	0.78	**0.02**	0.87	0.28	0.52	0.90	0.50	0.57	0.56	0.60
$\mathcal{N}_{1000} \subseteq \mathbb{R}^{2000}$	1000	0.12	0.30	0.93	0.09	0.95	0.90	0.83	0.99	0.82	0.85	0.85	0.86
$\mathcal{N}_{2000} \subseteq \mathbb{R}^{4000}$	2000	0.12	-	0.95	0.17	0.96	0.95	0.88	0.99	0.87	0.89	0.90	0.90
$\mathcal{U}_{10} \subseteq \mathbb{R}^{10}$	10	**0.04**	**0.04**	0.39	0.07	0.47	**0.00**	0.27	0.29	0.20	0.18	0.17	0.22
$\mathcal{U}_{100} \subseteq \mathbb{R}^{100}$	100	**0.00**	**0.02**	0.75	**0.02**	0.67	0.28	0.50	0.90	0.49	0.56	0.53	0.59
$\mathcal{U}_{1000} \subseteq \mathbb{R}^{1000}$	1000	**0.00**	**0.02**	0.92	0.09	0.76	0.90	0.81	0.99	0.81	0.84	0.84	0.85
$\mathcal{U}_{4000} \subseteq \mathbb{R}^{4000}$	4000	**0.01**	-	0.96	0.29	0.78	0.98	0.90	1.00	0.90	0.92	0.92	0.92

4.3 Conclusions

In thic chapter we benchmarked LIDL against classical LID estimators on synthetic datasets with ground truth, using normalizing flows (MAF, RQ–NSF, Glow) for density estimation, and we reported relative bias/MAE and variance over multiple runs.

On high–dimensional uniform/Gaussian data up to 4K dimensions, LIDL stays within $\pm 5\%$ while most baselines degrade beyond $\sim$100 dimensions; ESS is the strongest classical baseline yet remains inferior (Fig. 4.1; Tables. 4.1, 4.2, A.2). On anisotropic multiscale data, LIDL's explicit scale δ yields stable estimates across sample sizes, whereas many methods exhibit sample-size dependence; KNN and MiND–ML are comparatively stable (Fig. 4.2). On curved and union manifolds (Swiss roll, helix, sphere, lollipop), LIDL is competitive and near-perfect with exact densities; with imperfect densities lPCA/ESS can outperform on some cases, but on the lollipop's 0D component only lPCA and LIDL recover the correct dimension, and LIDL remains correct under small jitter via its operating scale (Fig. 4.3; Tables 4.1, 4.2).

Because LIDL estimates a slope, bias in log-likelihood primarily shifts the intercept while noise raises variance, suggesting further gains from robust regression and stronger density models.

Chapter 5

Experiments on image datasets

In this chapter we evaluate LIDL on real images. At a high level, we (i) visualize how per-sample LID aligns with perceptual complexity and class structure, (ii) examine the role of the operating scale δ and the effect of dequantization, (iii) reduce estimation error via ensembling over multiple models n, and (iv) relate LID to downstream performance in generative reconstruction and classification.

We organize the section as follows. We begin with qualitative image results by sorting MNIST, FMNIST, and CelebA by their LIDL estimates (Fig. 5.1) and plotting per-class empirical CDFs for MNIST and FMNIST (Figs. 5.2, 5.3). Next, Sec. 5.2 characterizes the operating range of δ: on controlled multiscale data LIDL reproduces the expected step-like behavior (Fig. 5.8), whereas on images an overly large δ collapses thin manifolds (e.g., dark garments in FMNIST; Fig. 5.7). We then study how dequantization and δ-sweeps affect estimates—both absolute values and sample rankings—using class-wise curves (Figs. 5.9, 5.10). Sec. 5.3 quantifies error reduction from increasing the number of models n (ensembling along the same δ range), showing monotonic MSE improvements (Fig. 5.11). Finally, Sec. 5.4 links LID to downstream performance: higher LID correlates positively with VAE reconstruction error (Fig. 5.12) and negatively with classification accuracy (Fig. 5.13).

5.1 Experiments on MNIST, FMNIST and Celeb-A

We ran LIDL on MNIST, FMNIST, and Celeb-A ($D = 1\text{K}, 1\text{K}, 12\text{K}$ respectively) datasets using Glow as a density estimator. We sorted those datasets according to LIDL estimates and observed that visually more complex examples have higher LID. Some small, medium, and high dimensional images from those datasets are shown in Fig. 5.1. We present cumulative distribution function (CDF) for MNIST and FMNIST in Fig. 5.2 and 5.3. More samples from MNIST, FMNIST, Celeb-A sorted by their LID can be found in Fig. 5.4, 5.5, and 5.6.

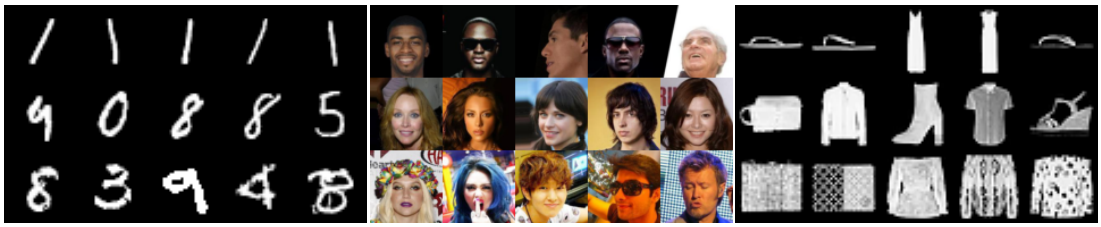


Figure 5.1: Samples from different image datasets (MNIST, Celab-A, FMNIST from left to right) presented according to their LIDL estimates (top to bottom). Those results are highly correlated with the complexity of an image.

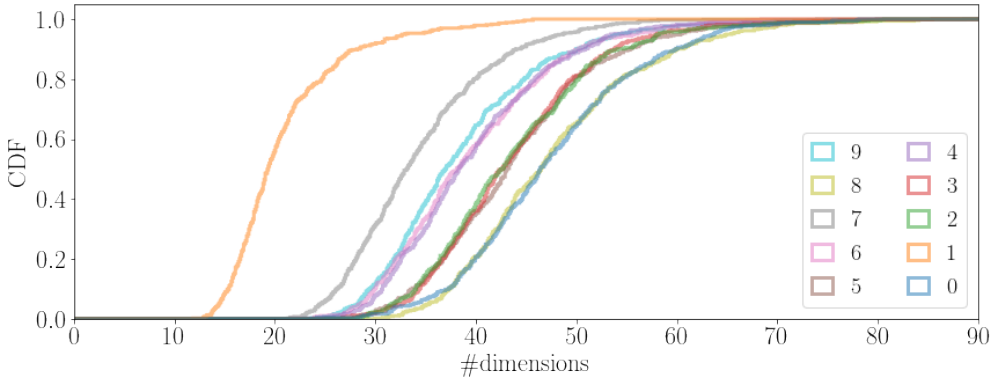


Figure 5.2: Empirical CDF of 5000 examples from MNIST dataset. Each line represents CDF for separate class in the dataset. Class number (which also is a represented digit in this case) can be found in the legend.

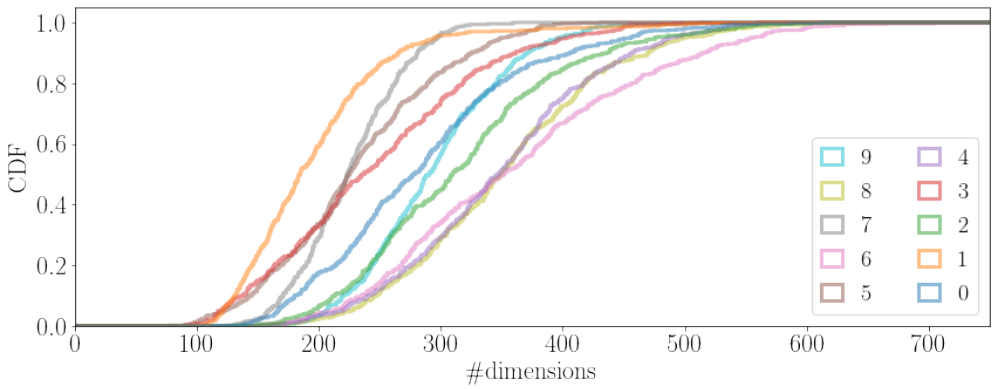


Figure 5.3: Empirical CDF of 5000 examples from FMNIST dataset. Each line represents CDF for separate class in the dataset. Class number can be found in the legend.

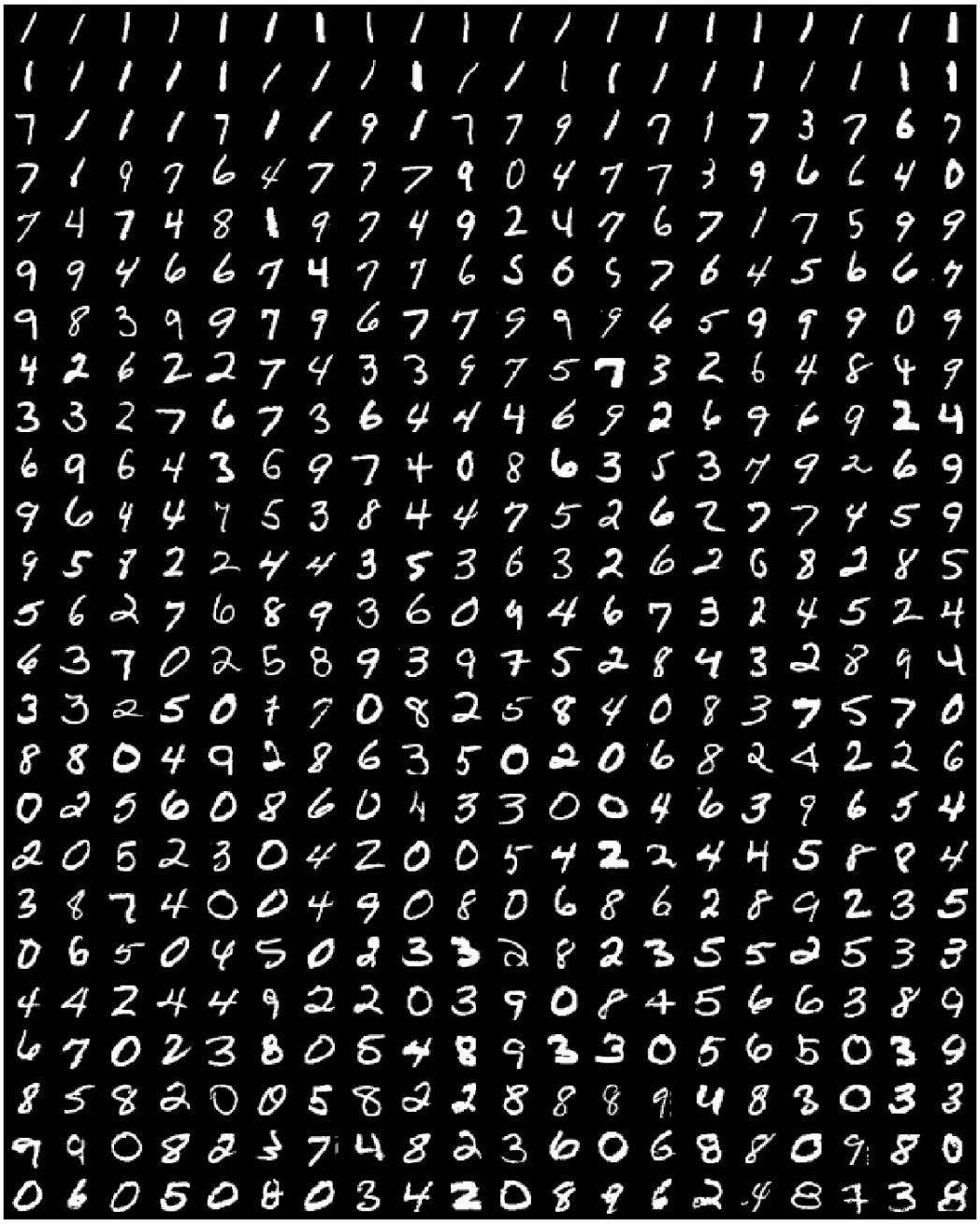


Figure 5.4: Samples from MNIST sorted by their LID estimates.

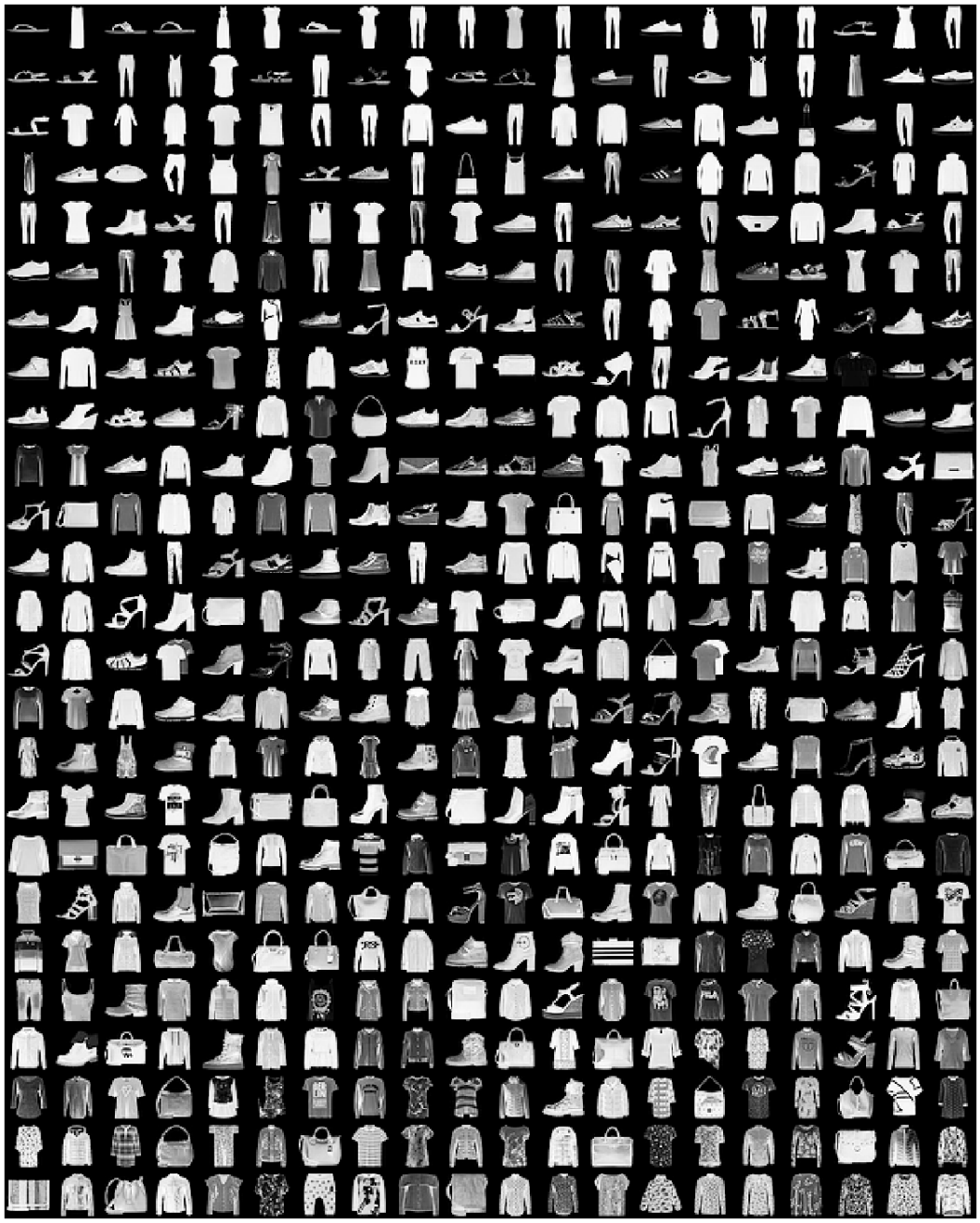


Figure 5.5: Samples from FMNIST sorted by their LID estimates.

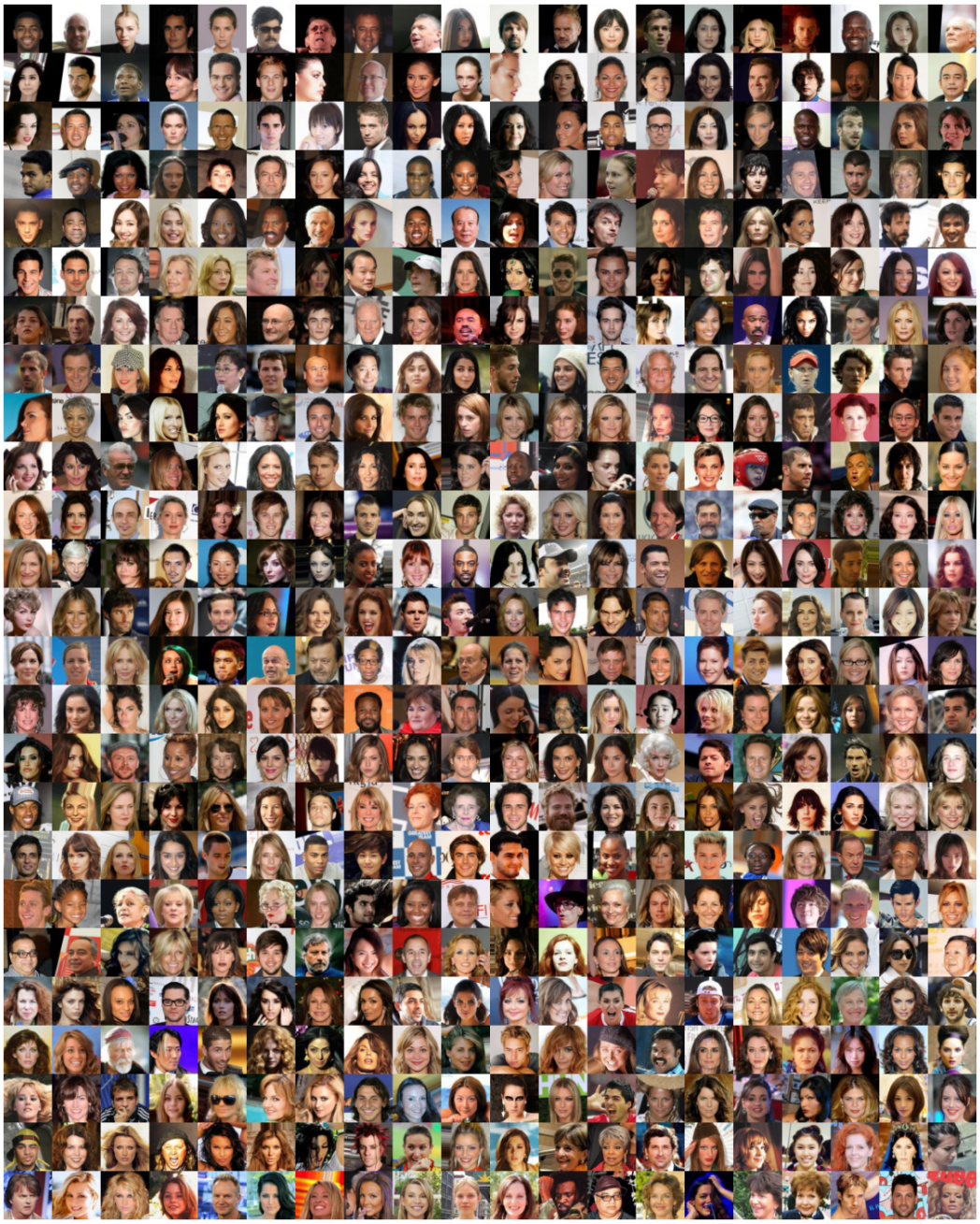


Figure 5.6: Samples from Celeb-A sorted by their LID estimates.

5.2 Operating range

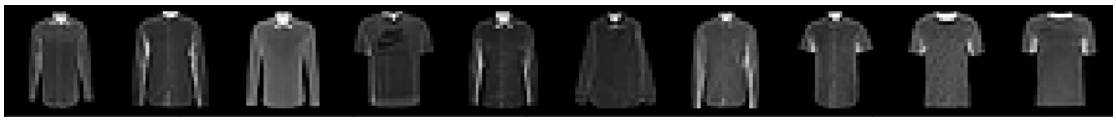


Figure 5.7: Images from the FMNIST dataset, for which the LID estimate is close to 0. This effect occurred when we used to high δs for this thin data manifold.

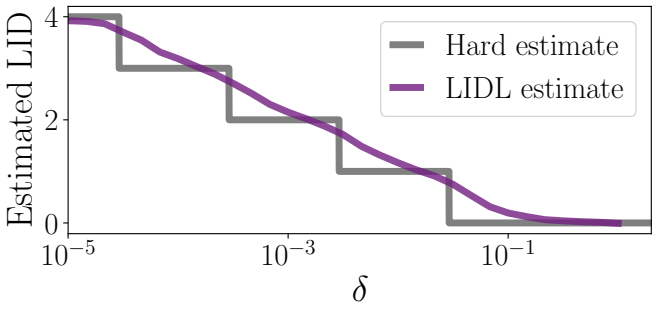


Figure 5.8: LIDL and hard estimate for different values of δ for 4-dimensional multiscale uniform distribution. We can see that LIDL ignores dimensions that are much smaller than δ even with imperfect density estimators.

As stated in Sec. 2.3, δ can be seen as a scale parameter. We introduced some numerical and theoretical results to support this hypothesis, and in this section, we are going to present some experiments investigating this topic. In Fig. 5.8 we present a similar experiment to that from Fig. 2.2, but this time with 4-dimensional uniform density. Results seem quite similar to previous theoretical results. For similar Gaussian distribution, we get an almost identical relation between dimension variance, LIDL estimate and δ.

We also tuned a δ range on image MNIST and FMNIST to reduce dequantization noise (added by us to the dataset before training, described in subsequent paragraphs) influence on the LIDL estimate. In this experiment on FMNIST (normalized to values between -0.5 and 0.5) for values of $\delta > 0.1$ we observed that the whole cluster of darker clothes had been estimated as being 0-dimensional. We present some samples from this cluster in Fig. 5.7. This experiment shows us that we have to be careful with using bigger values of δ.

Relation between examples for different range of δs We observed that for image dataset LID estimates for two disjoint sets of 4 δs have similar ranks (they on average differ between 10%-15%), and relations between points in each set (i.e. if LID estimate for x_j is lower than LID estimate for x_i) are preserved in 80-90% of cases.

LID estimate dependence on δ and effect of dequantization We present MNIST and FMNIST LID estimates (averaged per class) dependence on δ in Fig. 5.9 and 5.10. Images present wide range of δs (from 10^{-4} to 10^{1}) for original datasets and datasets with dequantization used during and after training. Black dashed line indicates a theoretical δ, above which LIDL should not calculate dequantization dimensions into LIDL estimate. This is 10 times standard deviation of dequantization noise $\mathcal{U}(0, 1/255)$. We can see that slightly above this threshold estimates for quantized and dequantized datasets aligh with each other. We can also observe, that for dequantized datasets and very small δ LID estimate is close to the dimensionality of the space, and for very big δs, LID estimates are close to 0 as expected.

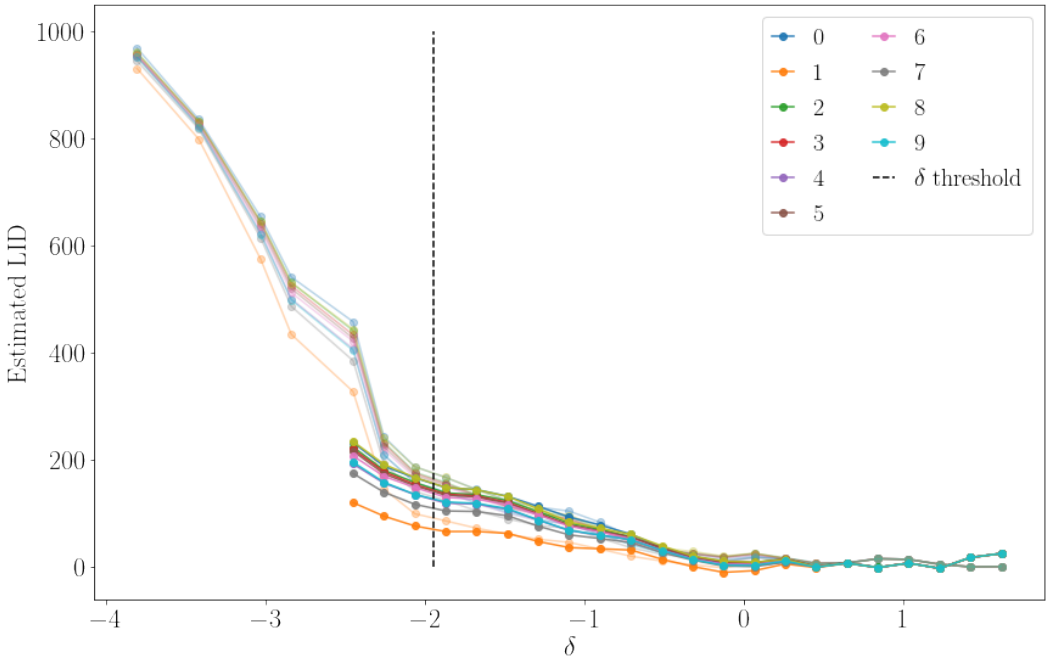


Figure 5.9: MNIST average LID estimates for each class for quantized (strong color) and dequantized (faded colors) as a function of δ.

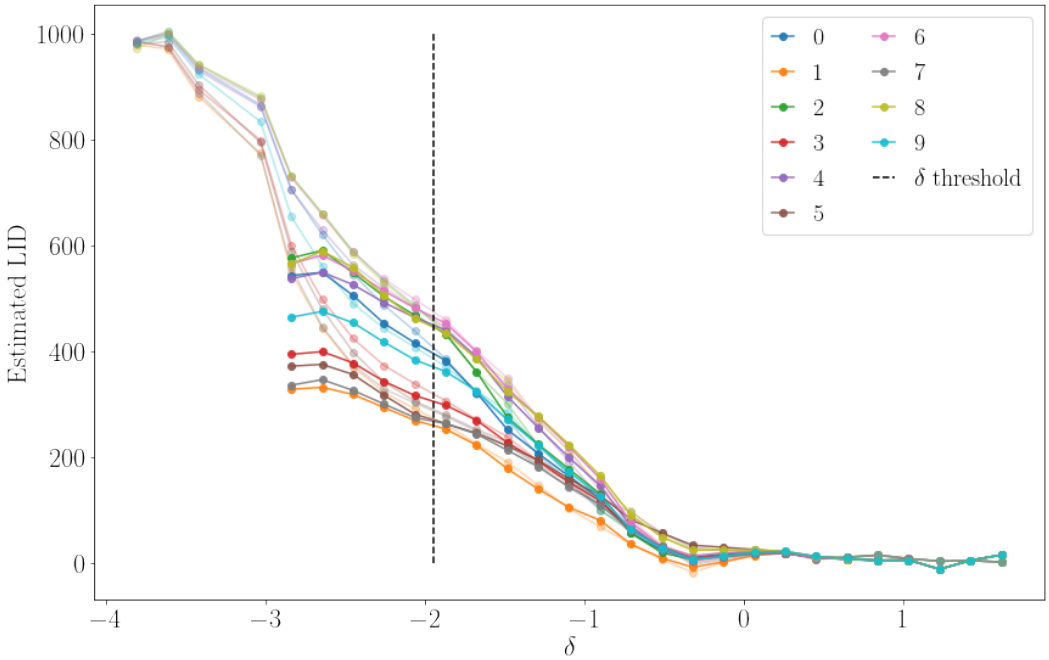


Figure 5.10: FMNIST average LID estimates for each class for quantized (strong color) and dequantized (faded colors) as a function of δ.

5.3 Reducing the error of the density estimate

Because model ensemble methods [Opitz and Maclin, 1999] often reduces prediction error in many machine learning models, and most of LIDL error comes from the imperfect density estimators, we applied it to our problem by increasing the number of models n used in LIDL. We were able to reduce an error of each estimate by simply adding more models between the same range of δs. An example of this behavior for 10-dimensional Gaussian embedded in 20-dimensional space is plotted in Fig.5.11.

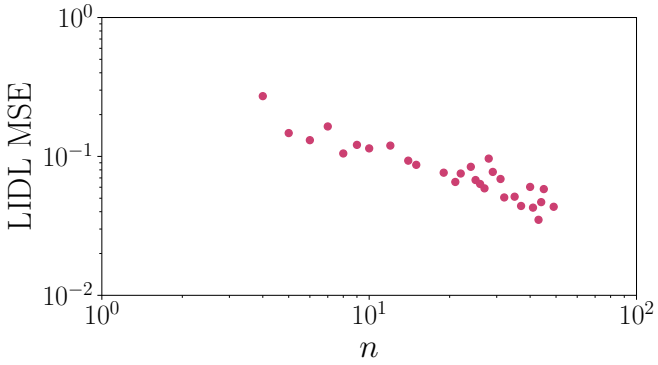


Figure 5.11: The dependence of mean-square error (MSE) on the number of models used in LIDL (n from Algorithm 1). We can observe monotonic decrease of the estimate error with the increase of n.

5.4 LID connection with ML model performance

In this section, we show, that LID estimates are connected with model behavior for autoencoder and classification deep neural networks. Our result suggests, that the connection between LID and model performance is significant, so LIDL estimates can potentially be used in problems like semi-supervised learning, active learning, uncertainty estimation, and curriculum learning.

Reconstruction error vs LID for autoencoders In this experiment we wanted to investigate if the estimate from LIDL is correlated with reconstruction error for the image in VAE [Kingma and Welling, 2014]. We trained VAE on MNIST with latent space sizes 50 and 150, and observed that there is a high correlation (Pearsons $R > 0.7$ in both cases) between MSE and LID. We plotted LIDL estimates against MSE for 5K images in Fig. 5.12. We can see an almost linear relationship between those quantities.

LID and classification accuracy We observed that classifiers can achieve better accuracy on data points with lower LID estimates. We trained neural networks on a subset of MNIST (300 images) and FMNIST (50K images) datasets and noticed a negative correlation of LID and an accuracy on a test set. Results are presented in Fig 5.13. What is more, we observed similar behavior inside a majority of classes (9 classes for MNIST, and 8 for FMNIST).

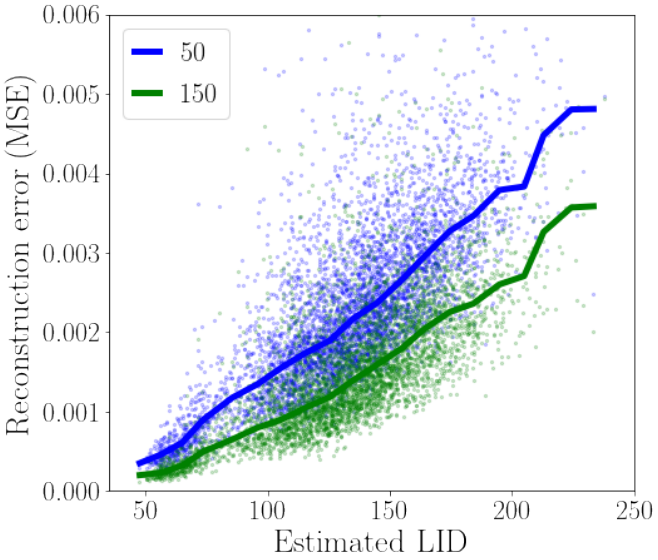


Figure 5.12: LIDL estimates and VAE MSE scatterplot for a sample from MNIST dataset. Lines are running medians for those point clouds. Values in legend are VAE latent space size.

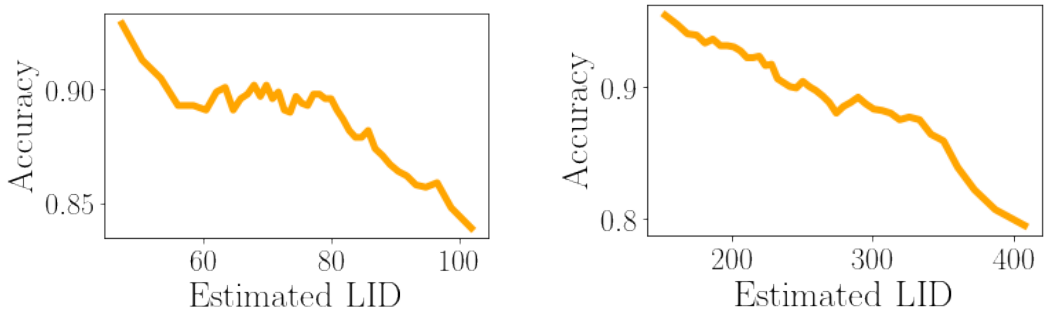


Figure 5.13: Average classifier accuracy per LID value on MNIST(left) and FMNIST(right) datasets. We can see a very strong negative correlation between those values.

5.5 Conclusions

Across MNIST, FMNIST, and CelebA, our empirical study shows that LIDL produces per-point LID estimates that align with perceptual complexity and class structure (Fig. 5.1; CDFs in Figs. 5.2, 5.3). The operating scale δ behaves as a true scale parameter: on controlled multiscale data LIDL recovers step-like dimensionality as δ crosses intrinsic scales (Fig. 5.8), whereas on images an overly large δ can collapse thin manifolds—e.g., dark garments in FMNIST appear nearly 0-dimensional (Fig. 5.7). Dequantization effects follow theory: slightly above the threshold set by $\approx 10\times$ the noise standard deviation, estimates for quantized and dequantized data align; very small δ recovers the ambient dimension, while very large δ drives estimates toward zero (Figs. 5.9, 5.10). Rankings are reasonably stable across disjoint δ-ranges (pairwise order preserved in 80–90% of cases; average rank differences of 10–15%).

We showed, that ensembling over more models n within a fixed δ-range monotonically reduces estimation MSE (Fig. 5.11), offering a simple route to improved accuracy.

Finally, we showed that per-point LID from LIDL closely tracks other machine learning models behavior: on MNIST, VAE reconstruction error is strongly and nearly linearly correlated with LID for latent sizes 50 and 150 (Pearson $R > 0.7$), indicating that higher local complexity predicts harder reconstructions (Fig. 5.12); for classification on MNIST and FMNIST, average test accuracy decreases sharply with LID and the trend persists within most classes, linking lower LID to easier, more reliable predictions (Fig. 5.13). These findings point to LID as a practical signal for semi-/active learning, curriculum design, and uncertainty estimation, with the caveat that correlations are not causation and that estimator quality and the chosen operating scale δ can modulate the effect.

Chapter 6

Broad comparison of neural algorithms

In this chapter we take a closer look at the procedure of testing new LID estimation algorithms, we identify the problems with current approach and propose a solution. We proceed as follows: Sec. 6.1 motivates the need for rigorous, domain-aware benchmarks, identifies gaps in current practice (simple analytic toy data vs. real datasets with unknown LID), and frames our transformation-based evaluation strategy that bridges these regimes. Sec. 6.2 formalizes the transformation toolkit (IDR, ME, ASE, ADI, MS). Sec. 6.3 applies it to construct diagnostic datasets and reports per-method behavior across: (i) non-uniform densities, (ii) manifold curvature, (iii) boundary effects, (iv) thin manifolds, (v) nearby manifolds, and (vi) estimate sensitivity to sample size, followed by controlled real-world transformations (ADI/ASE/ME) and a real-like dataset with known LID (MS). We then summarize quantitative results in Sec. 6.4 and distill per-algorithm takeaways in Sec. 6.5. Implementation details and additional analyses are deferred to the appendices (Appendix A.3, A.4).

6.1 Motivation

Advancements in neural methods enabled the rapid development of neural algorithms for LID estimation. These parametric approaches leverage the power of deep learning, including generative models that analyze density changes, variations in the singular values of the Jacobian under different noise magnitudes, the rank of scores from diffusion models and using adversarial attacks on generative models..

Despite the rapid development of LID estimation algorithms, scant attention has been paid to evaluating their performance. Most existing benchmarks for assessing LID estimation methods follow two common strategies. The first involves datasets sampled from well-defined and simplistic distributions where the LID is known and well-understood. The second strategy evaluates LID estimation methods on domain datasets.

The main advantage of the first method lies in the well-understood ground truth of LID, as the distributions have analytical form, making the evaluation of LID estimation methods reliable. This makes it possible to conduct an extended mathematical analysis as presented in Chapter 3. Unfortunately, those datasets fail to capture the complexity of the real-world manifolds. Moreover, most existing works do not consider edge cases such as varying manifold thickness or neighboring manifolds, nor do they analyze results in a

way that reveals inefficiencies of their algorithms even on simple datasets like Gaussian distributions.

On the other hand, the second approach for testing LID estimation methods, which involves using real-world domain datasets, matches the desired level of complexity but suffers from a significant drawback: in most cases, the ground truth of LID is unknown. This makes it impossible to assess a method's performance reliably, leading to possibly erroneous conclusions. Furthermore, the underlying neural architectures are often tailored to specific domains, incorporating inductive biases which limit their transferability across domains. This makes it challenging to determine whether the final method can handle various datasets effectively.

In this part we identify key manifold characteristics that pose significant challenges for LID estimation methods and propose datasets specifically designed to test these characteristics in isolation. Subsequently, we evaluate these datasets using a broad range of state-of-the-art LID estimation algorithms, demonstrating that each dataset exposes weaknesses in at least one method.

In this chapter we bridge the gap between benchmarks based on well-understood analytical distributions and real-world datasets by employing several data transformations. As a result, we are able to stress-test algorithms on datasets with unknown LID by evaluating their performance before and after transformation and comparing it to the ground truth LID difference imposed by the transformations. This, combined with the introduction of a technique that maps the dataset into a different domain representation without altering the underlying manifold, equips us with a comprehensive toolbox for testing algorithms across various datasets and domains.

Finally, we address the issue of the need for new benchmarks for LID estimation. Our results from this chapter underscore the importance of more challenging benchmarks, including domain-specific evaluation that addresses potential biases in neural network-based methods. This work also highlights the need for further development of benchmarks to keep pace with the rapid and diverse advancements in LID estimation techniques.

6.2 Methods for creating domain datasets

We use a set of transformations and methods that can be used to create challenging benchmarks for any continuous domain like images, audio, video, EEG, etc.

Inverse Domain Representation (IDR) The goal of this method is to bridge the gap between datasets sampled from analytical distributions with known ground truth of LID and real-world datasets on arbitrary domains. While producing an artificial dataset sampled from an arbitrary manifold is not challenging (one can embed the manifold in $\mathbb{R}^D$ and sample from the embedding's range), this becomes more complex when one wants the dataset to resemble some real-world data. We need a method to embed an arbitrary manifold into the ambient space of a given dataset $X \subset \mathbb{R}^D$ in a manner such that the image is "close to" X.

Take, for example, a dataset $X \subset \mathbb{R}^D$ of face images, and a manifold $M \subseteq \mathbb{R}^d$, already embedded in $\mathbb{R}^d$. If we take d eigenvectors of the covariance matrix of the distribution of face images computed from X, their linear combinations will yield a d-dimensional space of face-like images in which we can embed a copy of M. Sampling from this copy gives us a dataset of face-like images arising from a prescribed manifold. For the purpose of experiments class 7 images from FashionMNIST dataset were used as a basis for IDR transformation, samples for Gaussians dataset are presented in the Fig. 6.1.

More formally, suppose we are given a manifold M smoothly embeddable in $\mathbb{R}^d$ through $\phi\colon M \to \mathbb{R}^d$. Given any dataset X embedded in $\mathbb{R}^D$ with $D \geq d$, we may artificially create a dataset diffeomorphic to M, interpolating the points of X. The procedure starts with computing the mean $\mu_X \in \mathbb{R}^D$ of X and applying PCA to the centered dataset $X - \mu_X$. Then, we take the principal component vectors $u_1, \ldots, u_d \in \mathbb{R}^D$ corresponding to d largest eigenvalues. After that, we embed M in $\mathbb{R}^D$ through $\hat{\phi}(p)\colon M \to \mathbb{R}^D$ given by the formula $\hat{\phi}(p) = \mu_X + \sum_{i=1}^{d} \phi_i(p) u_i$. Finally, we take $\hat{\phi}(M)$ as the new (continuous) dataset, from which we may now sample points.

The embedding $\hat{\phi}$ is just the composition of ϕ, followed by the embedding of $\mathbb{R}^d$ into $\mathbb{R}^D$, taking the standard basis to the vectors u_i. Note that the vectors $u_1, \ldots, u_d$ are unit vectors; therefore, the geometry of the manifold remains unchanged. Nevertheless, the quality of the outcome, measured by visual similarity to the target image domain, is highest when the first d eigenvalues are large, i.e., the original dataset X is not "squeezed" in the corresponding directions.

While the technique is applicable to any target domain, in this paper, we present it for image representation only, as we find it the most insightful and the closest to real datasets on which LID estimation methods have been tested in the past.

In this work we use images from class 7 of FMNIST dataset to fit PCA with $D = 784$. A sample from such dataset is depicted in Fig. 6.1.

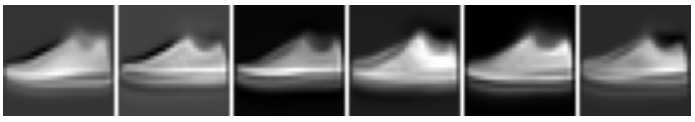


Figure 6.1: Few samples from Gaussian (IDR) dataset.

A possible alternative for the Inverse Domain Representation could utilize the vectors $u_1, \ldots, u_d$ scaled with the corresponding eigenvalues. This approach would ensure that the resulting dataset visually better matches the target image domain; however, it could alter the geometry of the dataset, especially when the first d eigenvalues are significantly different in magnitude. In extreme cases, this could lead to the deformation of the dataset and affect the LID of the transformed data.

For this reason, since preserving the LID of the transformed dataset is crucial for our study, we do not follow this approach. Moreover, the results presented in this paper demonstrate that a transformation that preserves the geometry of the dataset while mapping it to another domain still leads to different LID estimations by neural network-based algorithms, posing a significant challenge for the evaluation of modern methods.

Monotonic Embedding (ME) The goal of this method is to assess the robustness of LID estimation algorithms to smooth geometric deformations of the data manifold. It is particularly useful for datasets with unknown intrinsic dimensionality. The procedure involves applying continuous, monotonic transformations to the coordinates of the data in the ambient space, effectively stretching or compressing the geometry in a controlled manner. As long as the derivative of the applied function remains within a reasonable range, we expect the LID estimates to remain stable before and after the transformation. Notably, different functions may be applied independently to each coordinate, allowing for highly flexible distortions.

Ambient Space Extension (ASE) This method alters the ambient dimensionality of a dataset without modifying the LID. Similar to the previous approach, it is suitable for

datasets with unknown LID and can be used to confirm algorithm stability on a dataset. The extension is performed by introducing new dimensions as deterministic, continuous, and monotonic functions of the original coordinates. In the case of image data, this could be achieved through deterministic upscaling techniques. For audio signals, analogous transformations include increasing the sampling rate. At first glance, such modifications may appear trivial. However, when viewed through the lens of deep learning models, especially convolutional neural networks, they can introduce significant complexity. The hierarchical structure of learned convolutional features may differ considerably between models trained on original versus extended datasets.

Auxiliary Dimension Injection (ADI) This method increases the ambient dimensionality of datasets with unknown LID by adding informative features derived from parametric transformations of the original data. Parameters are sampled from known distributions, ensuring the result remains structurally related to the source dataset. The approach is highly flexible. For example, an audio signal may be filtered with a random low-pass cutoff and blended with the original at a random ratio, yielding two additional, non-trivial dimensions. In the image domain, concatenating random pairs of MNIST digits produces samples whose dimensionality equals the sum of the individual image dimensions, as the new dataset effectively forms a Cartesian product of the original space with itself.

Manifold Synthesis (MS) This generates datasets with known intrinsic dimensionality but complex, non-trivial geometric structure by applying a deterministic, continuous, and bijective transformation to a parametrized manifold. The requirement of bijectivity ensures the transformation is a diffeomorphism, preserving topological and differential properties of the original space. For images, one could define a manifold by parametrizing object attributes such as position, orientation, and other features, then rendering corresponding images based on these parameters. For audio data, one could generate samples by combining audio fragments according to controlled parameters such as start time, filter cutoff, duration, and volume. This results in datasets with a well-defined intrinsic dimensionality but with appearance and structure that are significantly more complex. Examples of such dataset are shown in Fig. 6.2.

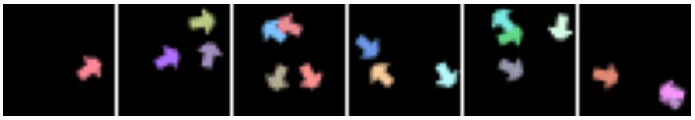


Figure 6.2: Sample from Arrows (MS) dataset.

6.3 Algorithm analysis demonstrated using image datasets

In this section, we use methods presented before to create datasets designed to test various interesting aspects of LID estimation algorithms, along with a discussion of their construction and characteristics. The details of experimental setting can be found in Appendix A.3. In captions of the figures in this section we include the reasoning behind scoring for algorithms summarized in Table 6.4.

Moreover, for the introduced datasets, we present the estimated LID for the tested algorithms. We focus on presenting one plot of interesting results for selected most interesting case, while the remaining ones are shown in the end of each section. All the results are summarized in tables in Sec. 6.4. We present results for LIDL in two different

scenarios: before the IDR transformation marked "LIDL (org. manifold)" and after IDR marked "LIDL". We evaluated it this way to show how much different the results may be in those two settings.

Non-uniform densities

As experiments in this work and in Stanczuk et al. [2024], Kamkari et al. [2024] and our theoretical considerations show, that LID estimation for non-uniform densities may be close to correct value when averaged on the whole dataset but biased in certain areas, e.g., in LIDL this bias is a function of the laplacian of the density. The main reason is that existing algorithms make specific assumptions during derivation — such as density smoothness, manifold flatness, or even local density constancy. Therefore, it is necessary to more thoroughly examine their behavior under non-uniform densities.

Gaussians (IDR) The dataset is a mixture of four 5-dimensional Gaussian distributions with means located at $(-3, 3, 0, 0, 0)$, $(3, -3, 0, 0, 0)$, $(-3, -3, 0, 0, 0)$, $(3, 3, 0, 0, 0)$ with standard deviations $1/27$, $1/9$, $1/3$, and 1 respectively, transformed with the IDR method. For each point, we calculated the distance from the closest mean and divided this distance by the appropriate standard deviation. Results for each algorithm are presented in Fig. 6.3 and 6.4, ans samples are presented in Fig. 6.1.

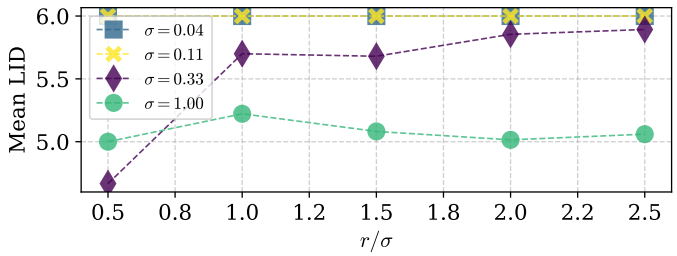
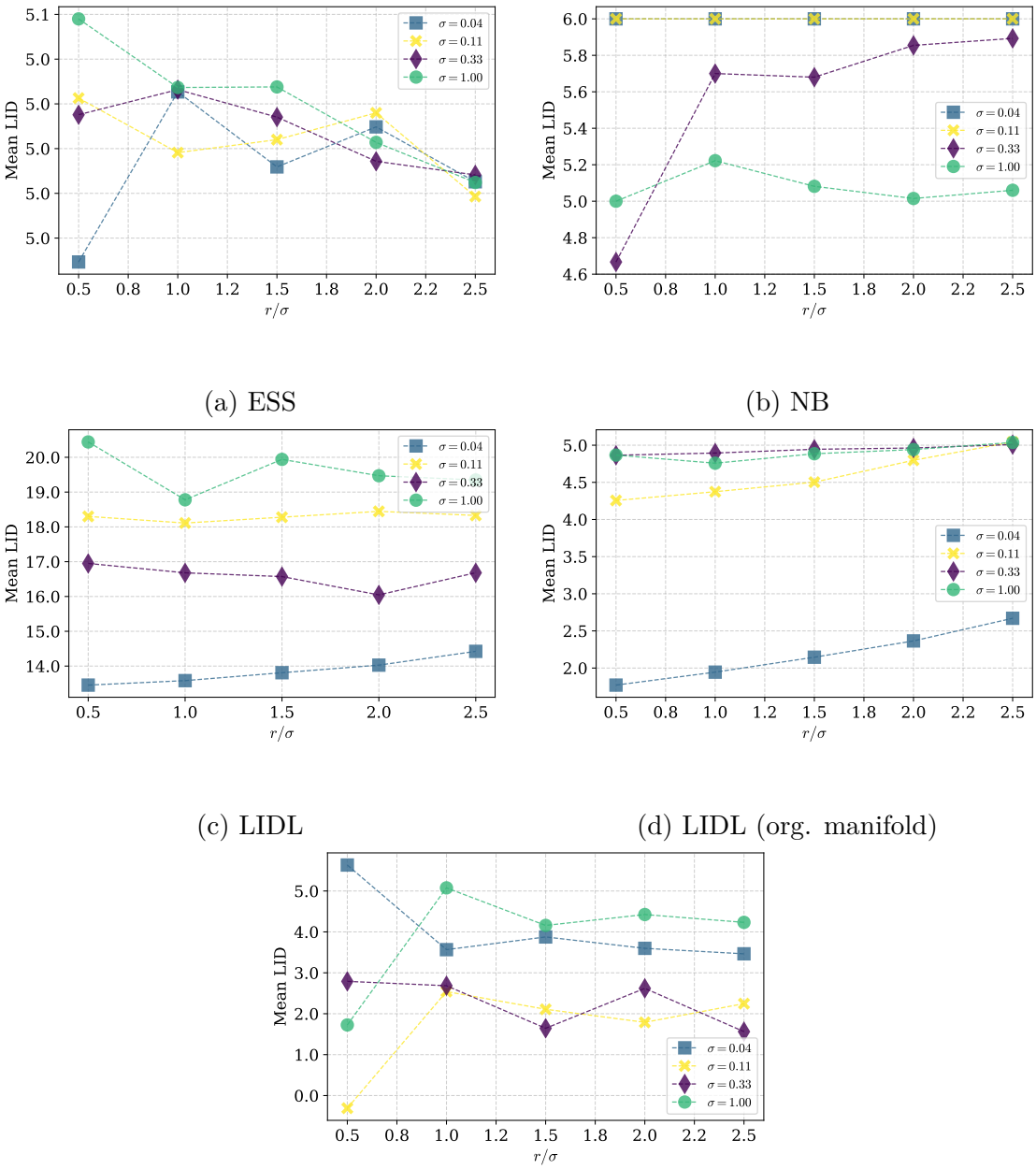


Figure 6.3: LID estimates calculated using NB. This dataset – Gaussians (IDR) – is composed of a mixture of four 5-dimensional Gaussians. Each line represents the average LID estimate as a function of a standardized distance of a point from its corresponding component mean.

We would expect that no matter from which component of the mixture we sample and no matter how far we are from the mixture component mode, the estimate should be equal 5. We can observe that only ESS was able to solve this task.



(a) ESS

(b) NB

(c) LIDL

(d) LIDL (org. manifold)

(e) FLIPD

Figure 6.4: Results for gaussians dataset for different algorithms. ESS ranked H because estimate is almost perfect; NB ranked L because it overestimates for most of the cases; LIDL ran on original manifold yields accurate estimates for more than half of the cases, so we ranked it M.

Manifold curvature

Building on the motivation from the previous section and the fact that the majority of algorithms are tailored for flat, uniform manifolds, we aim to test the behavior of LID estimation algorithms on curved manifolds. Curved manifolds are commonly present in real-world scenarios.

Spheres (IDR) We used 4 disjoint spheres S^5 with origins located in $(-3, 3, 0, 0, 0)$, $(3, -3, 0, 0, 0)$, $(-3, -3, 0, 0, 0)$, $(3, 3, 0, 0, 0)$ and with radia of $1/27$, $1/9$, $1/3$, 1 respectively, transformed with the IDR method. For each point in the dataset, we calculated the distance from the four sphere origins and assigned those points to the closest sphere. Estimate distributions for each algorithm and each sphere separately are presented in Fig. 6.5 and 6.6. The only algorithm unaffected by the curvature of this dataset was ESS.

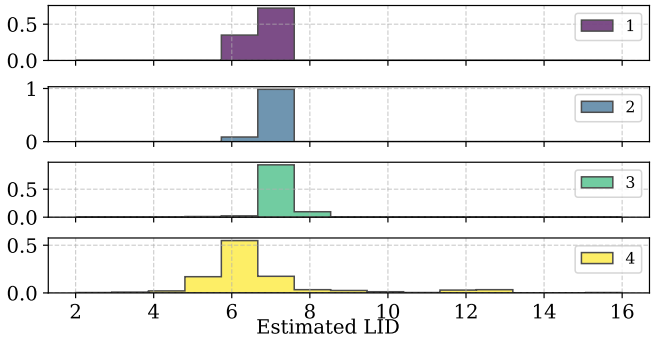
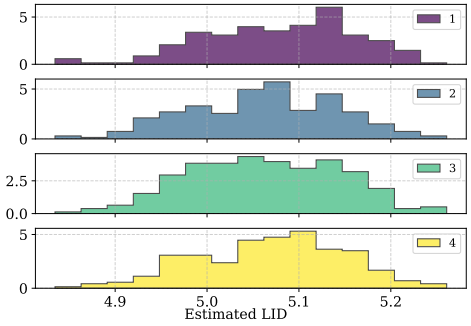


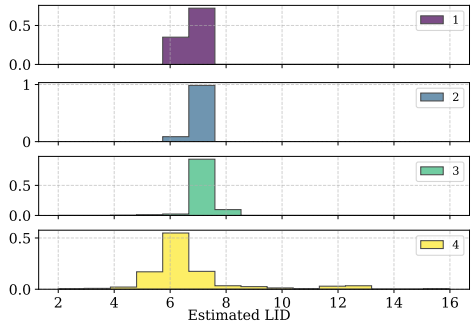
Figure 6.5: LID estimate distribution computed using NB for the Sphere (IDR) dataset composed of four disjoint spheres of different radii. Numbers from 1 to 4 indicate the sphere number order from smallest to largest sphere.

Spaghetti (IDR) The dataset used in this analysis is the spaghetti line dataset introduced by Stanczuk et al. [2024] but transformed into an image domain using IDR. It is a 1-dimensional manifold homeomorphic with the circle twisted and folded that it occupies $k = 20$ dimensions. Points from this manifold are sampled as follows: $\theta \sim \mathcal{U}(0, 2\pi)$; $x_i = sin((i + 1)\theta)$, for $i = 1, \ldots, k$.

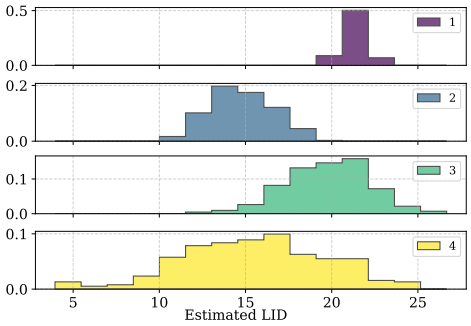
Results are presented in Table 6.1 and reveal that this dataset can pose a serious challenge for some algorithms. Notably, as shown by Stanczuk et al. [2024], the NB algorithm could solve the dataset without IDR domain transformation up to an embedding into $k = 100$ dimensions with high accuracy. However, the IDR transformation introduces significant challenges, leading to higher (but still reasonable) errors even for $k = 20$.



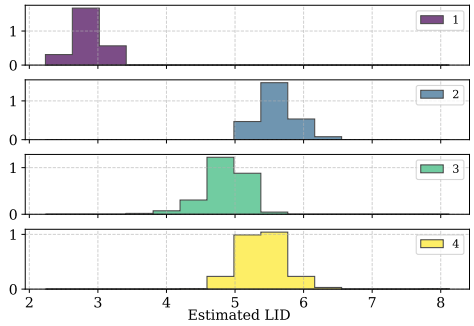
(a) ESS



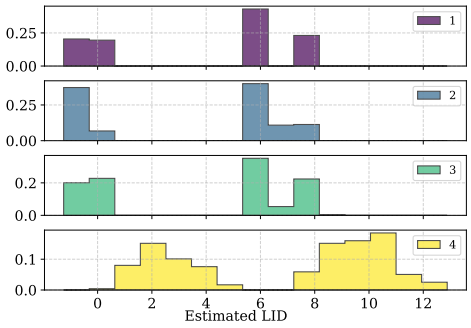
(b) NB



(c) LIDL



(d) LIDL (org. manifold)



(e) FLIPD

Figure 6.6: Results for spheres dataset for different algorithms. ESS ranked H because estimate is almost perfect; NB ranked L because it overestimates for all of the cases; LIDL ran on original manifold yields accurate estimates for around 30% of the cases, so we ranked it M.

Boundaries of manifolds

An extreme case of non-uniform density is a distribution with sharp edges, like a uniform distribution on a hypercube. Such distributions are common in real-world datasets due to measurement limitations. For example, cameras can't record light intensity beyond a threshold, so image datasets often lie within a hypercube, with many points on its boundary — as in images with white or black pixels.

Many algorithms work under the assumption that when measuring LID at a point x, we are considering a sufficiently small neighborhood of x where density has some *nice* properties, e.g., being sufficiently smooth. However, in practice, the neighborhood under consideration is contained in a ball of some radius r, whose center can lie in the proximity of a boundary or precisely on it.

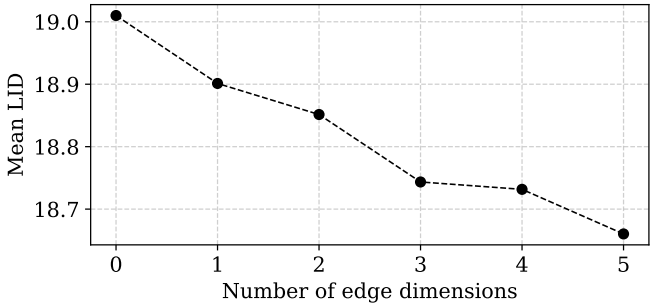
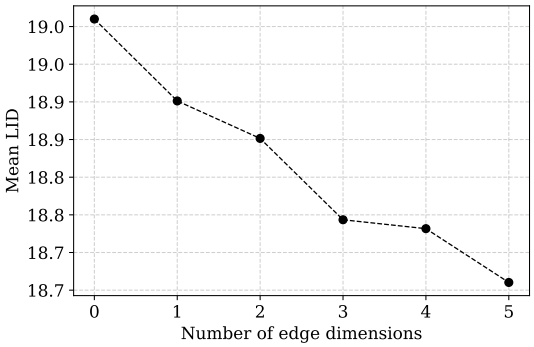
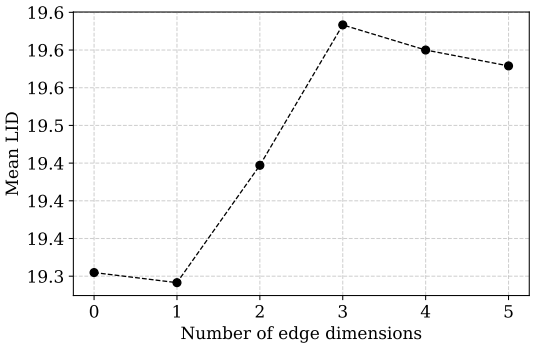


Figure 6.7: LID as a function of edge dimensions for ESS on 20-dimensional Uniform (IDR) distribution between -3 and 3.

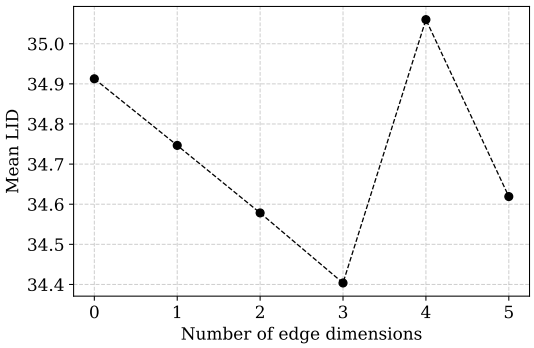
Uniform (IDR) The dataset is a 20-dimensional uniform distribution between -3 and 3 on each dimension, transformed with the IDR method. In this test, we are grouping points that are in the proximity of m edges. A point is said to be close to an edge if it lies on the original manifold closer than 0.25 from the edge located at 3 or -3. We test LID estimation for various values of parameter m. Results are presented in Fig. 6.7 and 6.8. In the former, a monotonic relationship between the average LID and a number of edge dimensions for the ESS algorithm can be observed. The pattern is less visible for other algorithms though.



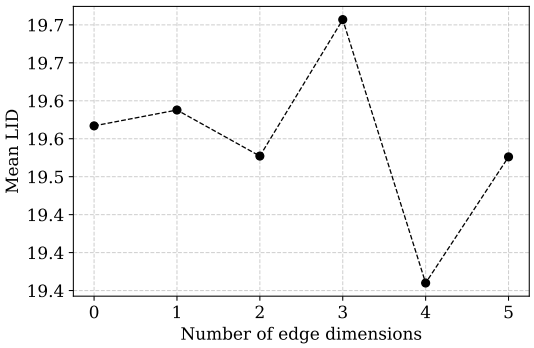
(a) ESS



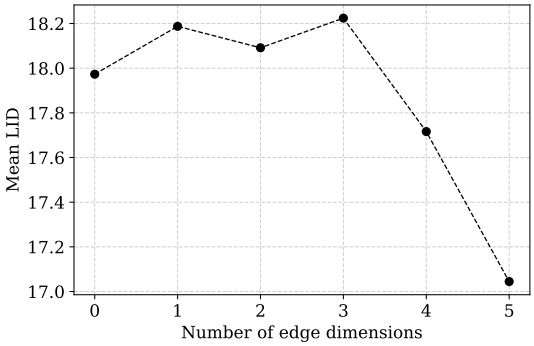
(b) NB



(c) LIDL



(d) LIDL (org. manifold)



(e) FLIPD

Figure 6.8: Results for edge points from uniform dataset for different algorithms. ESS underestimates the true value in all cases, so we ranked it L; NB has better estimate when closer to the edge, but for most of the points further from edges it underestimates so we ranked it M; LIDL on original manifold underperformed only in cases, where points are close to 4 or more edges, so we ranked it H.

Thin manifolds

An interesting scenario arises when moving along the manifold on a certain path, where the local intrinsic dimension remains unchanged. However, in an orthogonal direction to this movement, the manifold becomes thinner. In such cases, we would expect to observe consistent LID estimations across all observations. Nevertheless, some algorithms might mistakenly identify the manifold as having a lower local dimensionality.

Moon (IDR) The manifold we study is 3 dimensional. It is moon-shaped in the first two dimensions and a uniform interval in the third one. More formally, it is sampled uniformly at random from the points $(x_1, x_2, x_3) \in \mathbb{R}^3$ intersected with the set $\mathcal{M}$ defined as

$$\mathcal{M} := \{\|(x_1, x_2)\| \leq r,\ \|(x_1, x_2 + 0.1)\| \geq 0.899r,\ x_3 z| \leq r\}$$

where r is a radius hyperparameter that can be chosen arbitrarily. For our experiments, we used the value $r = 3$. The resulting manifold has a thickness of $0.201r$ on the bottom part and $0.001r$ on the upper part. Finally, the dataset is transformed using the IDR method.

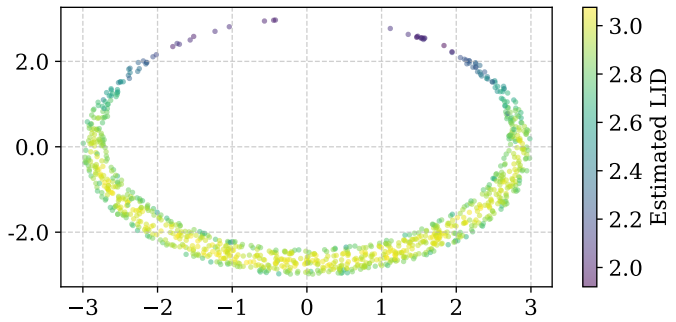
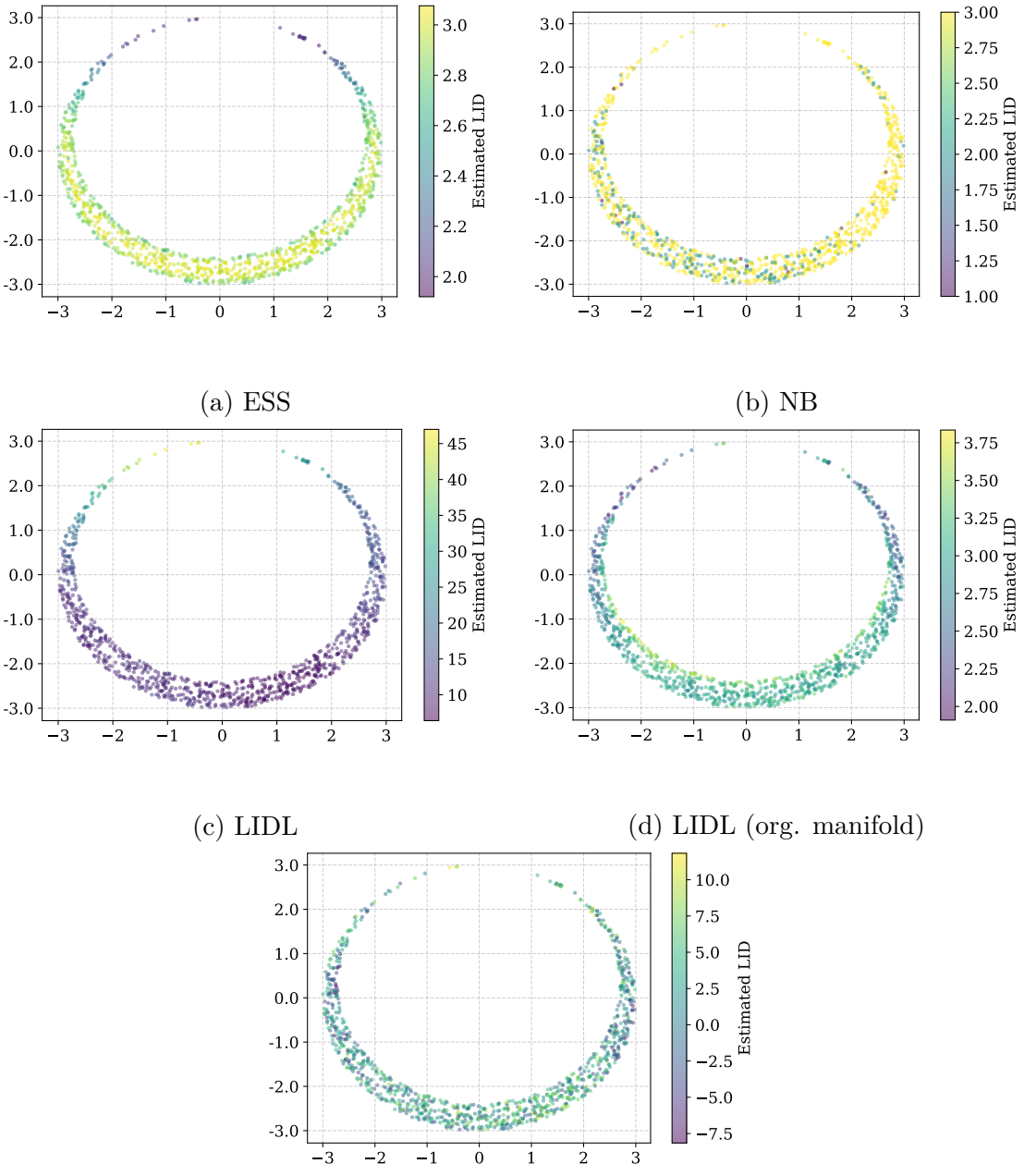


Figure 6.9: LID estimates for the Moon (IDR) dataset using ESS.

Fig. 6.9 and 6.10 show results for different algorithms. ESS has the best performance among them. It was able to maintain the correct estimate until the manifold was very thin; on the other hand, the NB estimates were close to ground truth but with errors distributed quite independently of manifold thickness. What is interesting is that we can observe a slight drop in the estimate for ESS close to the border of the moon, which is the same effect that was observed on the edge of uniform distribution.



(a) ESS

(b) NB

(c) LIDL

(d) LIDL (org. manifold)

(e) FLIPD

Figure 6.10: Results for moon dataset for different algorithms. We ranked ESS H due to the accurate estimates for most of the time; We ranked NB and LIDL on original manifold L due to the high variance in the estimate.

Nearby manifolds

We showed that when the expected distance to the nearest neighbor in the ambient space is smaller than the expected distance to the neighbor on the same manifold, it may lead to a bias in the estimate. The ability to recognize separate manifolds close to each other for finite sample size is a desirable property for LID estimation algorithms. To showcase this phenomenon, we introduce the Funnel and Spiral datasets.

Funnel (IDR) We consider a 2-dimensional funnel embedded in 3 dimensions, visualized in Fig. 6.11. The first two coordinates of the manifold original space: x_1 and x_2, and generated using this set of equations:

$$t = \mathcal{U}(0, 8);\ r = 3\exp(-t);\ \theta = \mathcal{U}(0, 2\pi);$$

$$x_1 = t - 4;\ x_2 = r\sin\theta;\ x_3 = r\cos\theta$$

Such a dataset is then transformed with the IDR method.

Results are presented in Fig. 6.11 and 6.12. We can observe that even when the algorithm gives a proper estimate when the radius of the funnel is high, for a low radius, the estimate is distorted. The ESS algorithm behaves as expected. For wide parts of the funnel, the estimate is exact. For the narrower parts, it goes up because manifolds are close to each other, and the algorithms start to detect points in all 3 dimensions. On the right end, it goes down when it starts to resemble a 1-dimensional line. The rest of the algorithms give more or less biased and noisy estimates compared to ESS.

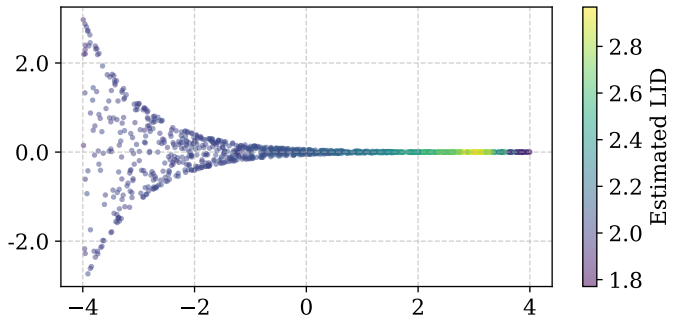
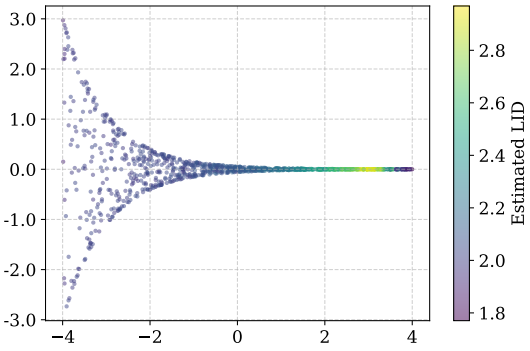
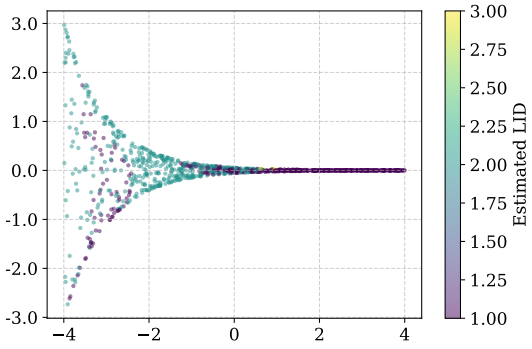


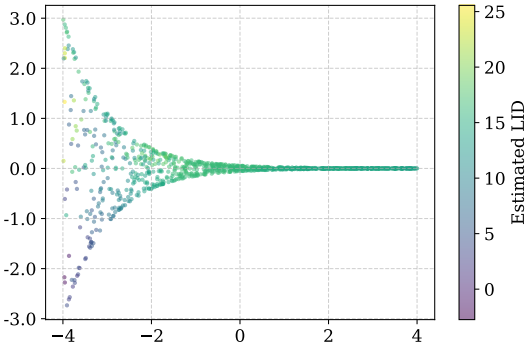
Figure 6.11: LID estimates for the Funnel (IDR) dataset using ESS.



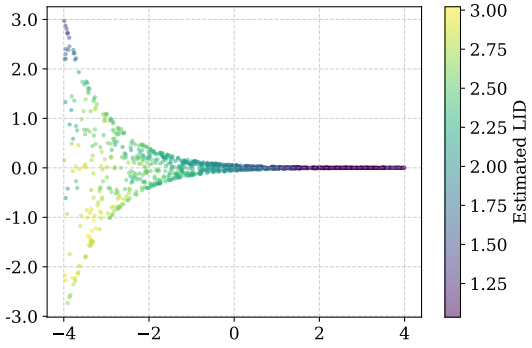
(a) ESS



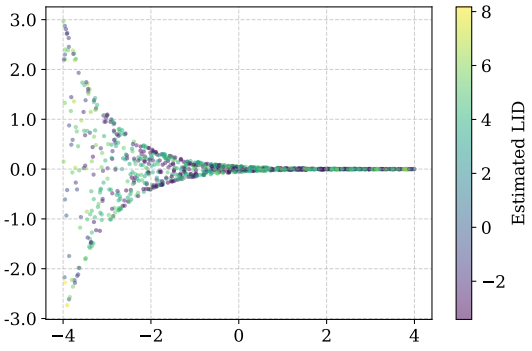
(b) NB



(c) LIDL



(d) LIDL (org. manifold)



(e) FLIPD

Figure 6.12: Results for the funnel dataset for different algorithms. We ranked ESS H due to the accurate estimates for most of the time; We ranked NB and LIDL on original manifold L due to the high variance in the estimate.

Spiral (IDR) A data set we consider is a spiral dataset visualized in Fig. 6.13 for the first two coordinates x_1 and x_2, generated using the set of equations:

$$t = \mathcal{U}(1, 100);\ r = 1/t; x_1 = r\sin(t/r);\ x_2 = r\cos(t/r)$$

where the distance to the closest point from the next revolution of the spiral gets smaller when we go down the spiral. The dataset has a useful property: the expected distance to the nearest neighbor calculated on the manifold is the same at any point on the manifold. Such a constructed dataset is then transformed with the IDR method.

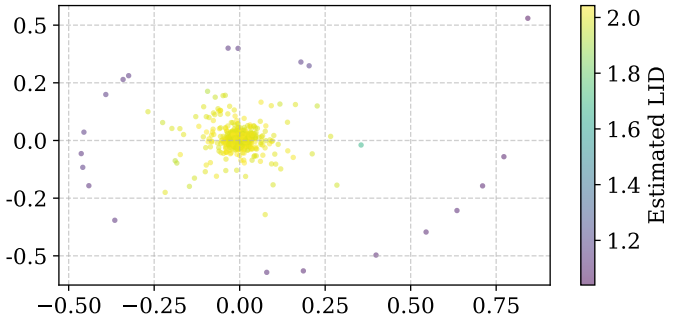
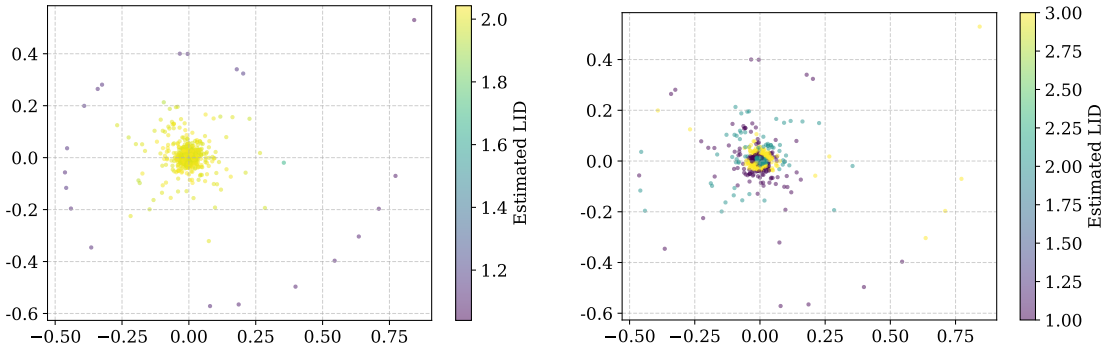


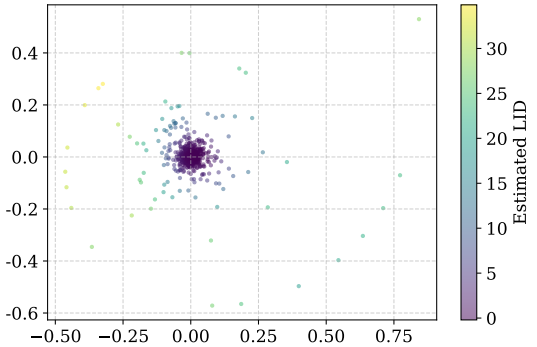
Figure 6.13: LID estimates for the Spiral (IDR) dataset using ESS.

An interesting observation for ESS algorithm, showcased in Fig. 6.13, is that the algorithm shows an LID estimate close to 2 only during the first revolution, where the manifold visually has an LID equal to 1 for a few more revolutions, especially when looking at full dataset which is 100× bigger than the test set. We want to highlight the fact that the algorithm was run with a hyperparameter of 100 neighboring points, which is a standard value for this parameter. We observed in our experiments that reducing the parameter responsible for the number of neighbors in ESS can reduce the estimate error by giving correct estimates further down the spiral. However, this dataset dependent hyperparametrization is a problematic approach for datasets with unknown LID that cannot be inspected visually. We also note that the ESS algorithm performed very well compared to the remaining methods, for which the results are presented in Fig. 6.14.

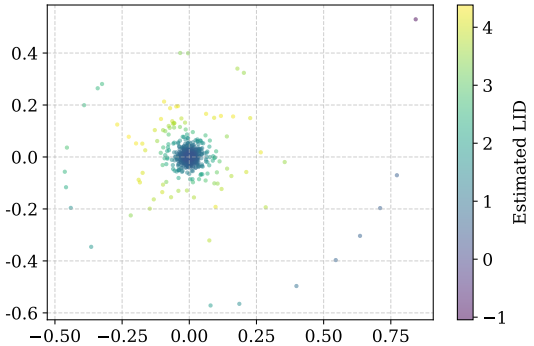


(a) ESS

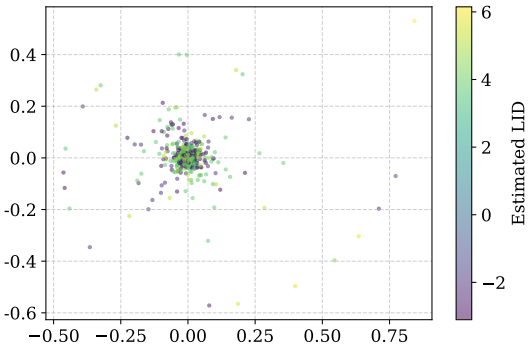
(b) NB



(c) LIDL



(d) LIDL (org. manifold)



(e) FLIPD

Figure 6.14: Results for the spiral dataset for different algorithms. We ranked ESS M due to the accurate estimates at the beginning of the spiral; We ranked NB and LIDL on original manifold L due to the high variance in the estimate.

Lack of network architecture invariance

In our experiments we observed, that for LIDL and FLIPD use of convolutional networks in the algorithm (Glow[Kingma and Dhariwal, 2018] and U-net[Ronneberger et al., 2015] respectively) yields worse results than using feedforward based networks (MAF[Papamakarios et al., 2017] and MLP) on the same manifold but transformed using IDR. We can see in Table 6.3 that some of the effect may be just from widening the ambient space, but similar effect were reported in Kamkari et al. [2024]. Our experiments for NB show that on Gaussian and Sphagetti datasets transformed by IDR we obtain different results than in the original paper without IDR transformation (see Sec.6.3).

Estimated LID vs sample size

We showed in Chapter 4 that for numerous algorithms the bias of the estimate is dependent on sample size. While it is natural that an algorithm error gets smaller for bigger sample sizes, an introduced bias leads to a situation where we don't know if we have enough data for our estimate to be correct. To test that we created a series of training datasets by drawing samples of different sizes from the FMNIST dataset. The validation set used for early stopping and the test set was held the same. One may wonder why we chose a real-world dataset rather than an artificial one with known LID. Fig. 4.2 shows that such a dependency does not occur for LIDL on artificial data, and this is the reason why we aimed for more challenging dataset. Our experiments demonstrated that LIDL exhibited a noticeable bias for small sample sizes on the FMNIST dataset.

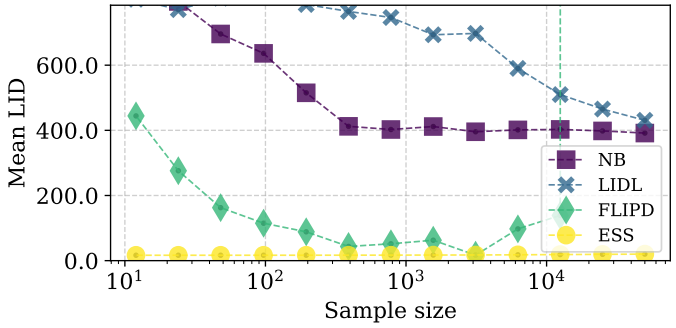
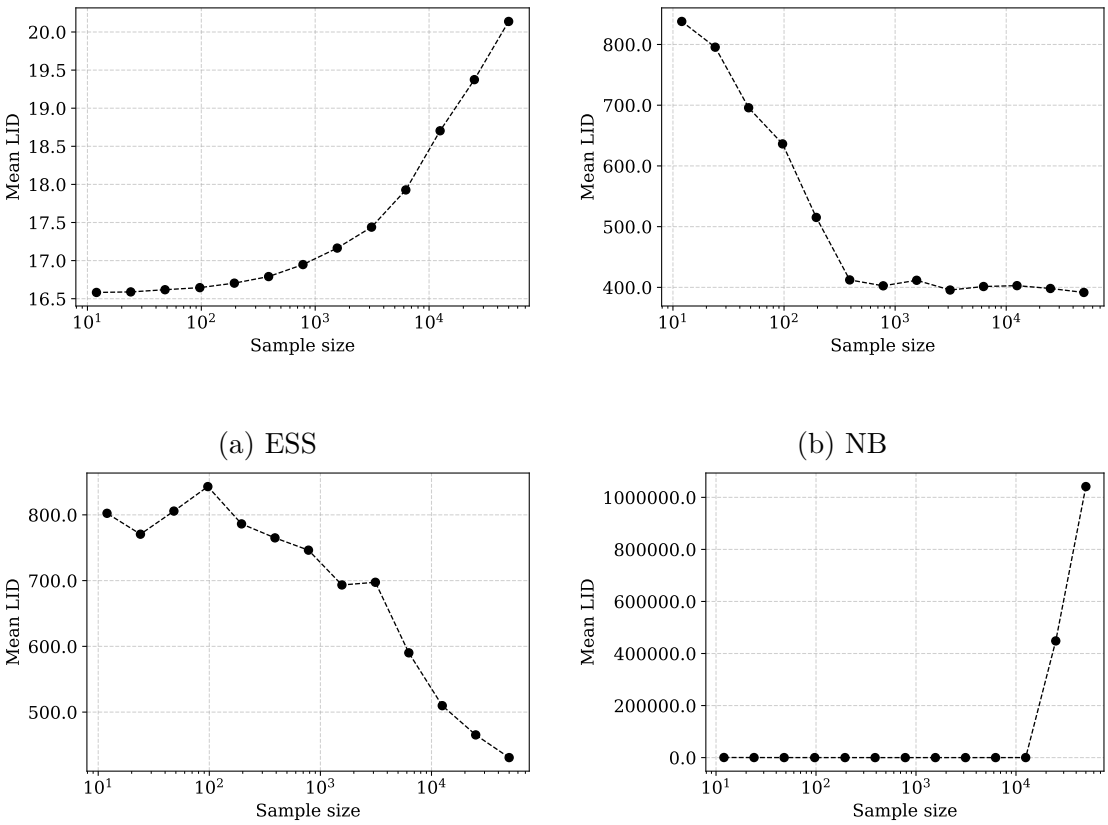


Figure 6.15: LID estimate in sample size on FMNIST.

The results are presented in Fig 6.15 and 6.16. We observe a rather significant dependence of the estimate on the sample size. Among all the algorithms, the NB algorithm achieves interesting characteristics. It stabilizes average estimate values for datasets bigger than 1000 samples, which is a good result compared to other algorithms. One interesting algorithm in this context is Erba et al. [2019], which is designed to deal with undersampled regions, but we did not test it during our experiments.



(a) ESS

(b) NB

(c) LIDL

(e) FLIPD

Figure 6.16: Per-method results for various sample sizes from FMNIST dataset. We ranked ESS L because the estimate dependence on sample size looks worring; We ranked NB H because the estimate stabilizes with sample size; We ranked LIDL L, because the estimate do not stabilize as sample size grows.

Real-world dataset transformations

For most real-world datasets used to evaluate the performance of LID estimation methods, the ground truth of LID is unknown. The absence of ground truth makes it impossible to reliably assess the quality of LID estimates produced by these algorithms in such domains. In what follows, we propose a set of transformations on such datasets that modify dimensionality in a controlled manner, enabling a rigorous evaluation of algorithm performance by measuring the difference in LID before and after applying the transformation. In all experiments in this subsection, we used a downscaled version of FMNIST as our base dataset. The original images were resized to 16×16 pixels. Results from this section are presented in Table 6.2.

Added dimensions (ADI) In this dataset, we added an 8-pixel-wide frame using mirror padding in eight directions, creating images of size 32×32 pixels. We generated three variants of the dataset with 0, 4, or 8 added dimensions by applying random brightness changes to a respective number of reflections. The results of LID estimation for the ESS algorithm are presented in Fig. 6.17. The estimates for datasets with added dimensions were calibrated by subtracting the number of added dimensions from the respective estimates, ensuring that the results should align with the identity line.

For ESS, points where LID estimation indicated a lower dimensionality in the original dataset remain close to the identity line, suggesting a small relative error after image transformation. In contrast, for points where LID estimation in the original dataset was high, the estimates after transformation fall significantly below the identity line, indicating large estimation errors.

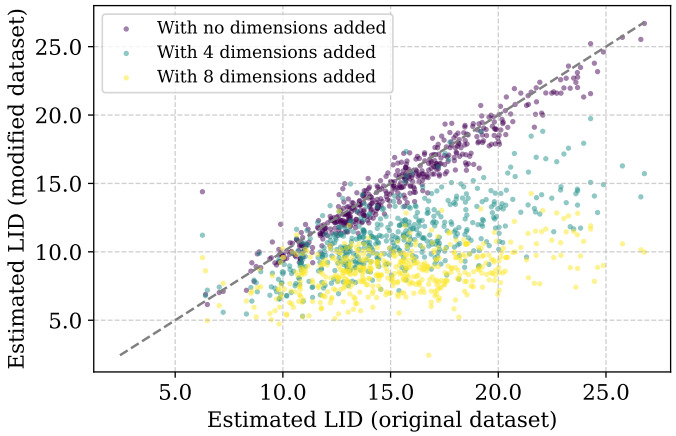
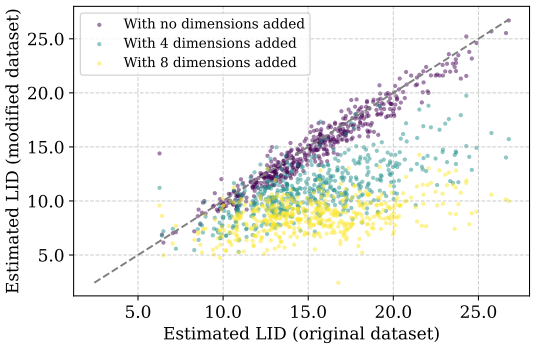
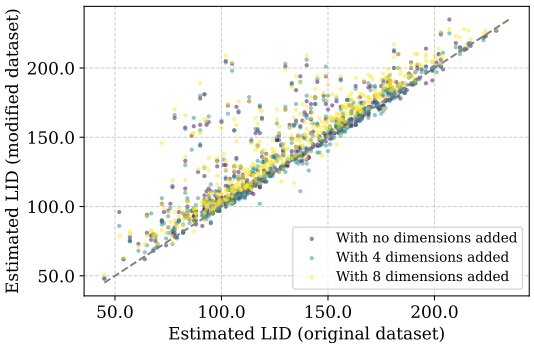


Figure 6.17: LID estimated on modified dataset minus the added dimensions versus the LID estimate on the original data set, the FMNIST 16×16, for the ESS algorithm. The number of added dimensions in the legend.

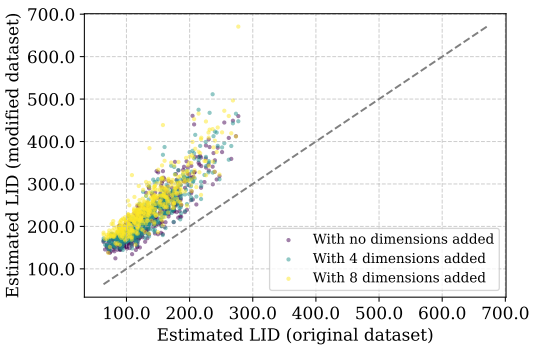
The LIDL algorithm, in turn, adds dimensions even when the dataset has zero additional noisy reflections, meaning no additional dimensions were introduced. NB maintains the estimate on the modified dataset close to the identity line for some points, but for others, it overestimates the LID regardless of the number of added noisy reflections. Another observation is that the estimates vary significantly, sometimes by more than ±50% compared to the same estimate on the base dataset. All results are presented in Fig. 6.18.



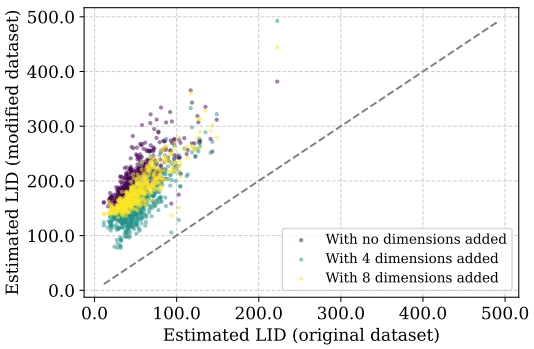
(a) ESS



(b) NB



(a) LIDL



(c) FLIPD

Figure 6.18: Estimated LID for FMNIST datasets with extra artificial dimensions. Estimates for datasets with added dimensions were calibrated in a way, that number of added dimensions were subtracted from respective estimates, so that on average results should lie on identity line ($y = x$). We ranked NB M because it is the only algorithm that have some points at identity line.

Upscaled (ASE) The dataset used was an upscaled 32×32 version of the base dataset. For upscaling, we used a torch function interpolate with bilinear mode.

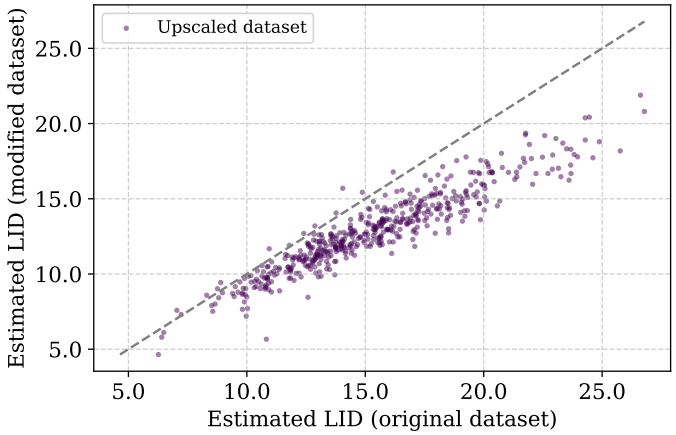
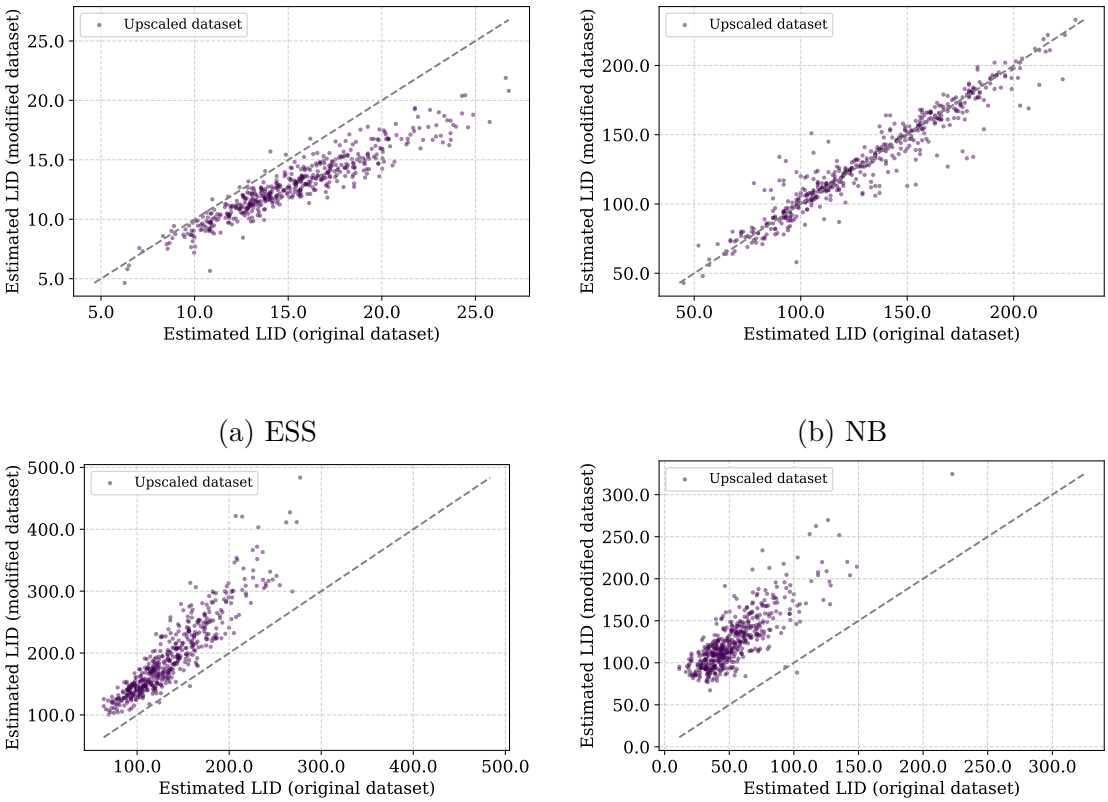


Figure 6.19: LID before upscaling vs LID after upscaling of the FMNIST images for ESS algorithm.

NB's performance in this task is quite impressive, despite the variability of the estimate, which remains of a similar magnitude as in the previous experiment involving added dimensions. ESS returns estimates lower than the original ones, while FLIPD and LIDL exhibit the opposite behavior, outputting higher estimates. All results are presented in Fig 6.19 and 6.20.



(a) ESS

(b) NB

(c) LIDL

(d) FLIPD

Figure 6.20: Effect of upscaling on FMNIST data. We ranked NB M because it is the only algorithm that have some points at identity line.

Stretched (ME) We create a ME by using a polynomial transformation applied to each pixel after normalizing its values to the range $[0, 1]$. We performed two transformations using different exponents: $x \in [0, 1] \mapsto y = x^l$, for $l \in \{0.25, 4\}$. The best-performing algorithm in this case is ESS, which maintains a similar mean LID before and after the transformation, albeit with a high variance. NB and LIDL estimates drop significantly after the spatial transformation, while FLIPD heavily overestimates the LID after transformation. Full results in Fig. 6.21 and 6.22.

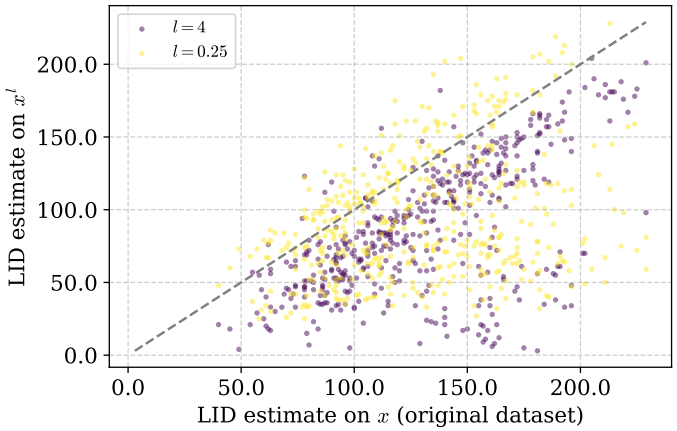
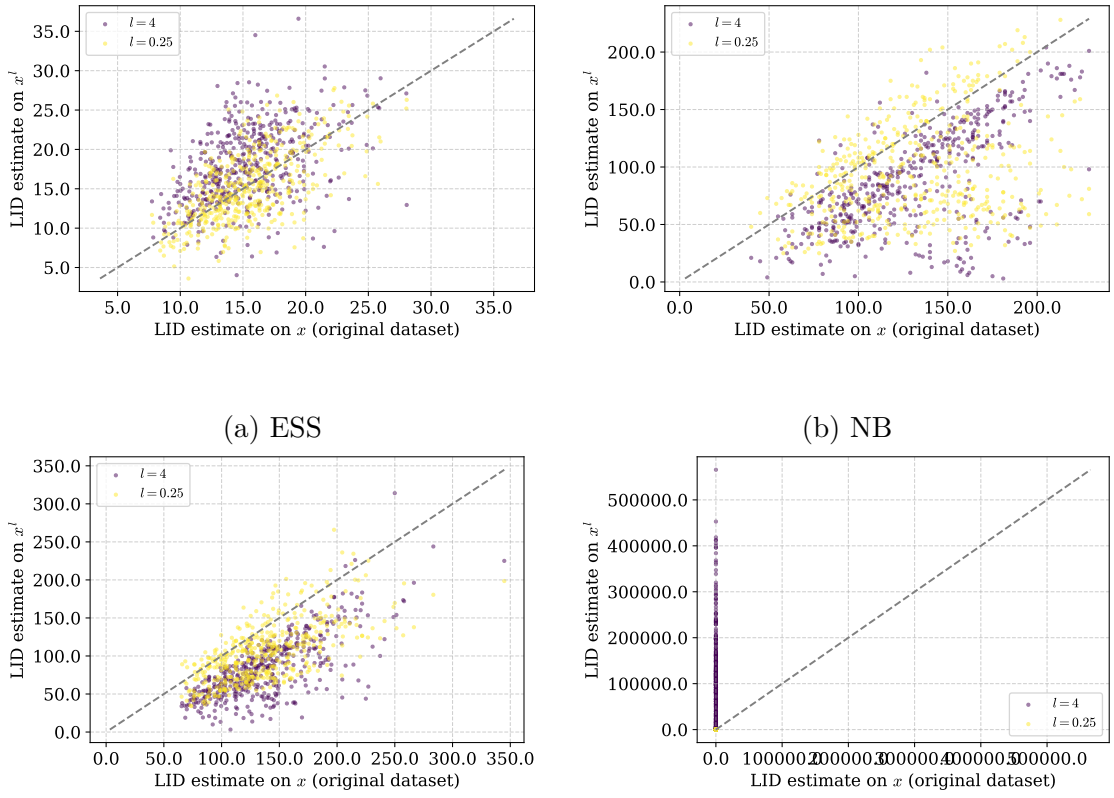


Figure 6.21: LID estimates for NB on FMNIST before and after the transformation x^ℓ, with two values of $l \in \{0.25, 4\}$.



(a) ESS

(b) NB

(c) LIDL

(d) FLIPD

Figure 6.22: Estimated LID under a FMNIST spatial stretching transformation. Exponent of the transformation is in the legend. We ranked ESS M, because despite the high variance, at least the points were distributed approximately in equal numbers on both sides of the identity line.

Real-like dataset with known LID

There are almost no datasets that simultaneously resemble real-world images and have a known underlying LID. Two notable exceptions are the Gaussian blobs from Stanczuk et al. [2024] and 3DIdent dataset introduced by Zimmermann et al. [2021]. The latter one was too big our computational budget and available GPU resources. Therefore, we created similar dataset using MS approach.

Arrows (MS) The dataset consists of 32×32 images of arrows placed on a black background. Each arrow is described by six variables: horizontal and vertical position, rotation, and color (three variables for RGB). The manifold dimensionality is six times the number of arrows in the image. A sample is shown in Fig. 6.2. There is a small possibility of manifold collapse when arrows perfectly overlap, but the probability of this occurring is less than 10^{-3}, making it negligible.

In this experiment, none of the algorithms produced results close to the ground truth. The ESS estimate failed to distinguish between different manifolds, consistently outputting values around 15 for all cases, with variations appearing only in an uncontrolled manner. The NB algorithm significantly overestimated LID values and produced some outlier estimates for LID of a value around 3k. FLIPD produced estimates ranging from -20 to 40. Figures 6.23 and 6.24 present full results.

This dataset is far more challenging for existing algorithms because the manifold may have some nasty properties that others presented manifolds don't. E.g. our theoretical considerations show that the manifold has many V-shape corners as an artifact of translating and rotating the arrow on the image. This may pose a challenge for the existing algorithms designed with the simpler manifold shapes in mind.

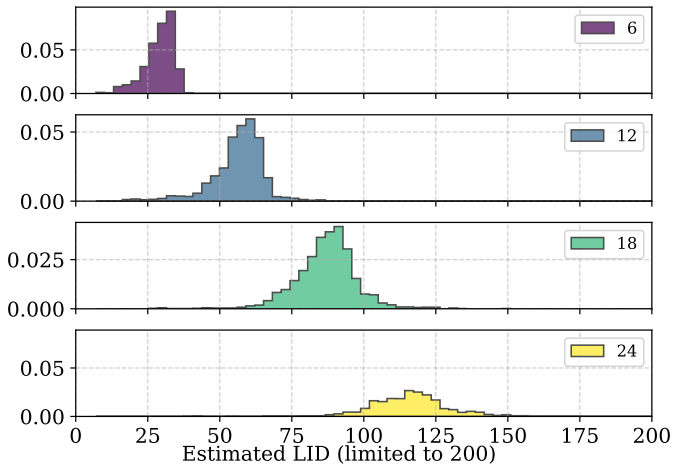
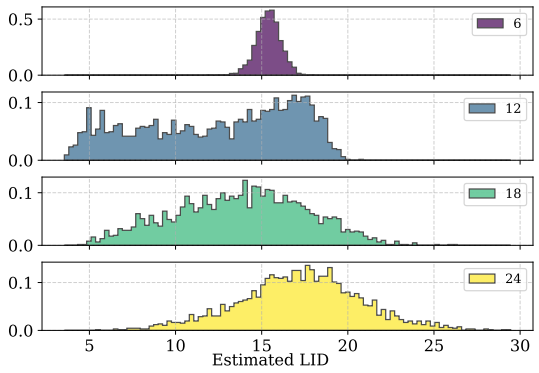
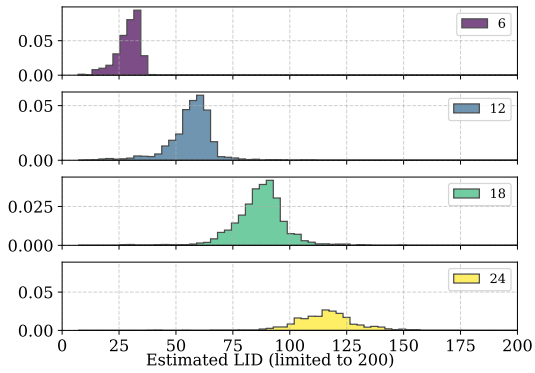


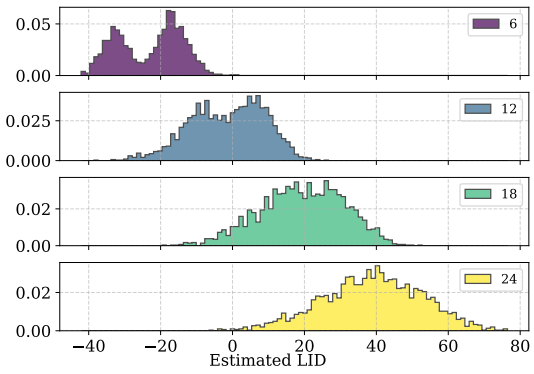
Figure 6.23: LID distribution estimated using NB for different manifolds in arrows dataset. Manifold dimensionality in legend.



(a) ESS



(b) NB



(d) FLIPD

Figure 6.24: Estimated LID for arrows dataset. We ranked ESS and FLIPD as L, only because at least some of the estimates are close, but we can see that with high probability it is purely by accident and not because those algorithms are performing well on this task.

6.4 Tables with the results

In this section we present tables with the results. We present three different metrics. μ is the average estimate for the dataset, σ is the standard deviation of the estimates, and MAE means mean average error of the estimate if the ground truth is known to us.

Table 6.1: LID estimations with MAE for the datasets with known dimensionality.

algorithm	ESS			FLIPD			LIDL			NB		
	μ	σ	MAE	μ	σ	MAE	μ	σ	MAE	μ	σ	MAE
Gaussians	4.99	0.09	0.07	3.25	3.57	3.08	17.24	3.09	12.24	5.72	0.66	0.81
Spheres	5.07	0.08	0.09	4.49	3.77	3.24	17.90	3.77	12.90	6.80	1.06	1.83
Spaghetti	1.03	0.03	0.03	1.12	4.19	3.54	10.25	6.40	9.53	2.12	1.54	1.12
Uniform	18.87	0.43	1.13	18.08	3.30	2.80	34.70	3.61	14.70	19.41	1.78	1.07
Moon	2.86	0.21	0.15	2.14	3.96	3.42	12.31	5.45	9.31	2.75	0.46	0.25
Funnel	2.20	0.26	0.21	1.73	3.46	3.24	14.18	3.09	12.20	1.48	0.51	0.54
Spiral	1.99	0.13	0.99	1.47	3.36	3.34	1.98	5.38	2.26	2.03	0.73	1.03
Arrows	14.73	3.88	6.23	8.57	25.80	17.43	–	–	–	455.85	1,007.53	440.82

Table 6.2: LID estimations for the modified real-world datasets with unknown dimensionality.

algorithm	ESS		FLIPD		LIDL		NB	
	μ	σ	μ	σ	μ	σ	μ	σ
FMNIST (base)	15.32	3.75	55.92	24.62	138.73	44.11	133.45	38.34
FMNIST (add dim +0d)	14.87	3.70	210.03	45.92	227.01	62.37	142.40	37.57
FMNIST (add dim, +4d)	15.21	2.40	166.43	45.91	235.48	67.07	144.89	37.68
FMNIST (add dim, +8d)	16.72	1.65	193.58	38.85	257.39	61.89	154.77	38.91
FMNIST (upscaled)	12.86	2.67	129.57	38.12	197.13	65.71	132.78	39.44
FMNIST (stretched $x^{0.25}$)	14.92	4.44	49.96	23.21	105.17	42.25	98.10	58.18
FMNIST (stretched x^4)	17.98	4.85	86,355.13	94,294.77	81.68	40.36	85.74	43.34

Table 6.3: LID estimations with MAE for the datasets with known dimensionality.

algorithm	LIDL (w/ IDR)			LIDL (org. manifold)			LIDL (org. manifold + padding)		
	μ	σ	MAE	μ	σ	MAE	μ	σ	MAE
Gaussians	17.24	3.09	12.24	4.34	1.17	0.90	18.68	19.38	22.14
Spheres	17.90	3.77	12.90	4.69	1.10	0.83	27.49	12.99	23.28
Spaghetti	10.25	6.40	9.53	5.30	1.94	4.31	36.25	11.41	35.25
Uniform	34.70	3.61	14.70	19.57	1.06	0.88	32.46	10.59	13.58
Moon	12.31	5.45	9.31	2.97	0.31	0.24	-21.31	18.99	25.57
Funnel	14.18	3.09	12.20	1.87	0.61	0.54	11.20	5.98	9.56
Spiral	1.98	5.38	2.26	0.85	0.88	0.66	–	–	–

6.5 Takeaways for each algorithm

Algorithm / LID estimation aspect	Non-uniform density	Manifold curvature	Boundaries of manifolds	Thin Manifolds	Nearby Manifolds	Inductive bias invariance	Real-world dataset size dependence	Artificially added dimensions (RWD)	Upscaled manifold (RWD)	Stretched manifolds (RWD)	Synthesized real-like datasets
ESS [Johnsson et al., 2014]	**H**	**H**	L	**H**	**H**	-	L	L	L	M	L
NB [Stanczuk et al., 2024]	L	L	M	L	L(M)	L	**H**	M	M	L	O
LIDL	O	O	O	O	O	L	L	L	L	L	-
LIDL (org. manifold)	M	M	**H**	L	L	-	-	-	-	-	-
FLIPD [Kamkari et al., 2024]	O	O	O	O	O	L(L)	L	L	L	O	L

Table 6.4: Summary of our experiments. Columns: LID-estimation aspects tested by our benchmarks; rows: neural-based algorithms (ESS as a classical benchmark). Performance of methods was classified following the legend – H: high, M: moderate, L: low, O: out-of-range (unassessable). More information about the reasoning behind each score can be found in the captions of images with per-algorithm results in Sec. 6.3. Gray color marks cells where similar aspects in the method's original paper (or in earlier experiments in case of LIDL) were investigated; parentheses give that paper's reported performance (assumed H if absent). We show that algorithms that passed original simple tests for many of aspects did worse on our benchmarks; many aspects – especially on real-world datasets (RWD) – remain untested and some only partly explored.

In Table 6.4 we shows the summary of the results presented in this chapter, and we present per-algorithm takeaways below. More information about the reasoning behind each score can be found in the captions of images with per-algorithm results in Sec. 6.3

ESS algorithm performed very well for datasets with low-dimensional manifolds. From the experiments in Chapter 4 we know, that its performance deteriorates if manifold dimensionality raises, which can be observed on 20D uniform datasets. We observed in preliminary experiments that its behavior can change when working on smaller samples, and the `n_neighbours` parameter is crucial to the performance, but there is no prior way of setting it right. We could go with the highest value possible due to the computational and memory constraints, but in the case of the Spiral dataset the smaller values works better, especially for smaller sample sizes like $10K$, so there is no right answer to that problem.

NB performed well, especially compared to other algorithms using neural networks. It failed on arrows, stretch, gaussian and spheres and had higher error than ESS on low-dimensional manifolds. It will surely beat ESS on higher-dimensional manifolds due to the ESS underestimation bias for higher-dimensions, but compared to ESS it lacks the precision and robustness on simpler manifolds. This perhaps may be corrected by the person skilled in training diffusion models, but in practice this method will benefit from more stable and less noisy estimates, especially because some tests were failed due to the high variance and not high bias like in the case of other methods like LIDL and FLIPD.

LIDL performed much worse than expected based on its performance on datasets with known LID from Chapter 4. In the first experiments, we used LIDL with Glow Normalizing Flow Kingma and Dhariwal [2018], but it performed worse to MAF Papamakarios et al. [2017], and its training time was an order of magnitude slower than MAF so finally we sticked with MAF. To further investigate this algorithm, we ran LIDL on selected datasets in three different scenarios: before the IDR transformation, before the IDR transformation but padded with zeros to reach 784 ambient space dimensions,

and after the IDR transformation. This analysis provided deeper insight into where the accuracy of LIDL's estimates deteriorates. In the first case, where the original ambient space before IDR was 30-dimensional, LIDL performed well. However, the IDR transformation or expanding the ambient space caused LIDL's performance to deteriorate. The results of these experiments are presented in Table 6.3.

Yet there is another problem with LIDL: although theoretical results in Chapter 3 show that when $\delta \to 0$ we should get an unbiased estimate, while in practice we always get the estimate close to ambient space dimensionality. This is an unsolved problem of how to choose this parameter in real-world scenarios. Those results are presented in Sec. A.4, where we can observe how much the LIDL estimate varies with δ. For presenting the results we have chosen one δ range which had the best performance on IDR datasets, still not being even close to the underlying LID value.

FLIPD This algorithm suffered from the same problems as LIDL, but to a greater extent. Authors of the FLIPD describe a "knee" on the plot where the estimate is the closest to the ground truth, but in practice, those models many times had problems converging and producing unreliable estimates with a hard-to-find "knee" structure. When we knew the underlying LID we were able to present better results when choosing the value of t with smaller MAE, but it is not a practical scenario for real-world datasets. The results for different values of t are presented in Sec. A.4, where we can observe how much FLIPD estimate varies with t.

We show in Chapter 3 that the theoretical foundations behind LIDL and FLIPD are solid and give tools to calculate reasonable ranges of delta for the given problem, so it suggests that the problem is the non-ideal density estimator.

6.6 Conclusions

In this chapter we introduced a transformation-based benchmarking toolkit (IDR, ME, ASE, ADI, MS) that bridges analytic toy data and real domains, and used it to stress-test LID estimators with an emphasis on LIDL's limitations.

Our experiments show that LIDL's accuracy is fragile under domain-changing operations: performance deteriorates after IDR or even zero-padding to higher ambient dimensions, ADI can induce spurious extra dimensions even when none are added, ASE (upscaling) tends to inflate estimates, and monotone pixel-wise stretching (ME) systematically shifts them.

Beyond transformations, LIDL is sensitive to non-uniform densities, curvature, boundaries, thin manifolds, and nearby components, and it exhibits notable sample-size dependence on real data. A central practical weakness is the operating-scale choice: while theory predicts unbiasedness as $\delta \to 0$, in practice small δ drives estimates toward the ambient dimension, yielding overestimation without a principled, data-driven selection rule. Finally, LIDL's dependence on density modeling quality is acute—both the choice of density estimator and the training budget materially affect outcomes—making estimator imperfections the dominant failure mode on challenging, real-like data such as Arrows (MS). In comparison, ESS is strong on low-dimensional, clean geometries but underestimates in higher dimensions and is sensitive to neighborhood hyperparameters; NB is comparatively stable under ASE and moderate sample sizes but drifts under ME/ADI and struggles on Arrows; FLIPD inherits LIDL's small-t sensitivity and requires selecting a "knee" in t, which is unreliable without ground truth. These findings argue for domain-aware, transformation-grounded benchmarks and for future LIDL variants with

robust multi-scale fitting, explicit δ diagnostics, and stronger density estimators before deployment on real datasets.

Chapter 7

Conclusions and Future work

In this thesis we proposed LIDL, a theoretically grounded estimator of local intrinsic dimension based on neural density models; we introduced a Wiener–process (heat–equation) perspective that recovers and explains estimator behavior via spatial derivatives; and we designed a transformation-based benchmarking toolkit (IDR, ME, ASE, ADI, MS) that bridges analytic toy data and real domains. Together, these pieces show that LID can scale to thousands of dimensions and that per-point LID aligns with model behavior in practice, while also revealing important limitations that must be addressed before reliable deployment on complex, real-world datasets.

A clear lesson is that LIDL's accuracy hinges on two coupled choices: the operating scale δ and the quality of the underlying density estimates. Although the theory guarantees unbiasedness as $\delta \to 0$, in practice for real-world dataset very small scales drive estimates toward the ambient dimension, and there is no principled, data-driven rule for picking δ per point. Moreover, LIDL is fragile under domain-changing transformations (e.g., IDR, ASE, ADI, ME) and shows biases with non-uniform densities, curvature, boundaries, thin structures, nearby components, and finite sample sizes. These limitations motivate multi-scale estimation, explicit diagnostics for scale stability, and estimator-agnostic safeguards (e.g. robust slope fitting).

The Wiener–process view suggests concrete remedies at finite t. Because $\beta_t(x) = t\,\Delta\rho_t(x)/\rho_t(x)$ isolates curvature and density-variation effects, one can design debiasing terms, boundary corrections, and even anisotropic or geometry-aware perturbation kernels. Extending the analysis beyond flat embeddings to curved manifolds (explicit dependence on the second fundamental form), mixtures and intersections (mixture-weight asymptotics), and quantized data should translate into practical correction recipes. A complementary avenue is to estimate Laplacians and scores with modern generative models (score-based/diffusion, consistency models) to improve robustness for non-uniform regions and larger t.

Beyond methodology, LID shows promise as a utility signal. The observed links between LID and model performance indicate that LID can guide semi-supervised and active learning (LID-aware sampling), curriculum learning (per-point difficulty schedules), and uncertainty estimation and calibration. Additional opportunities include monitoring domain shift and out-of-distribution behavior via shifts in the LID landscape, as well as using LID maps to steer data augmentation and editing.

Evaluation must keep pace. The transformation toolkit should evolve into domain-aware, principled benchmarks with metrics that quantify transformation consistency (ME/ASE/ADI invariance), scale stability (agreement across δ ranges), curvature and

boundary robustness, mixture separability, non-uniformity sensitivity, and real-to-IDR transfer fidelity. These tests should extend beyond images to audio, video, EEG and to harder, real-like generators (e.g., 3DIdent-style datasets), and to higher-dimensional manifolds. Although this work provides detailed, plot-based assessments, future iterations should attach concise scores so that methods are consistently comparable across aspects; our goal is to raise methods to at least the M level in Table 6.4 across most axes.

Concretely, we see four near-term priorities. First, principled δ selection and diagnostics: per-point multi-scale fitting with linearity checks, bias–variance tracking, and uncertainty quantification (confidence intervals for the slope, finite-sample rates, and sample–complexity bounds). Second, partial differential equation(PDE)-informed corrections: curvature- and boundary-aware debiasing using $\Delta\rho_t/\rho_t$. Third, stronger and more reliable density back-ends: score-based/diffusion models and consistency training to stabilize $\log \rho_\delta(x)$ across scales, plus quantization-aware likelihoods for discrete sensor data. Fourth, standardized, reproducible testbeds and cross-method studies: analyze non-density methods (e.g. ESS) under the same PDE lens, identify complementary strengths, and develop hybrids that inherit robustness across regimes.

We expect progress along these lines to make LID estimation substantially more dependable across domains and operating conditions, turning it into a practical tool for modern large-scale datasets.

Editorial Note

This thesis was edited and corrected for grammatical and stylistic errors using LLM-based systems. These systems also helped ensure the consistency and clarity of the text. We verified the suggested changes before incorporating them to ensure the meaning remained unchanged.

Acknowledgments

First and foremost, I would like to thank my supervisor, Marek Cygan, for his constant support throughout my PhD and during the work on this dissertation. His guidance, encouragement, and constructive feedback were invaluable at every stage.

The second person to whom I owe a great debt is Adam Kurpisz. Without his support, this dissertation would certainly not be what it is today. Thank you for our joint research and the countless discussions about the problems addressed here, as well as for the other projects we pursued together during this time.

Third, I wish to thank Adam Goliński, who was also a major source of support during my PhD, especially on the first paper, and who consistently pushed me to aim as high as possible.

I am grateful to all of my co-authors, without whom neither this dissertation nor the included articles would exist. In particular, I wish to acknowledge Jacek Tabor and Przemysław Spurek for steering me toward the research directions in which this work ultimately developed. I also thank Łukasz Garncarek, Rafał Michaluk, Dominik Filipiak, and Ksawery Smoczyński (listed in order of appearance) for their help and collaboration.

I further thank Michał Karpowicz, Tomasz Odrzygóźdź, Antoni Kijowski, Tomasz Odrzygóźdź, Maciej Śliwowski, Maciej Dziubiński, and Piotr Kozakowski for their valuable and constructive comments.

Finally, I am deeply grateful to my family: Kasia, Pola, Kalina, Wacława and Michał for their patience and help while I wrote this dissertation. Without them, balancing these studies with professional work would not have been possible.

Many of the experiments reported in this dissertation were performed on the *Entropy* cluster at the Institute of Informatics, University of Warsaw, funded by NVIDIA, Intel, the Polish National Science Center grant UMO2017/26/E/ST6/00622, and the ERC Starting Grant TOTAL. Additional computations were carried out on the *GUŚLARZ 9000* workstation at the Polish National Institute for Machine Learning.

Bibliography

Luca Albergante, Jonathan Bac, and Andrei Zinovyev. Estimating the effective dimension of large biological datasets using fisher separability analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2015.

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Extreme-value-theoretic estimation of local intrinsic dimensionality. *Data Mining and Knowledge Discovery*, 32(6):1768–1805, 2018.

Laurent Amsaleg, Oussama Chelly, Michael E Houle, Ken-Ichi Kawarabayashi, Miloš Radovanović, and Weeris Treeratanajaru. Intrinsic dimensionality estimation within tight localities. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 181–189. SIAM, 2019.

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*, pages 6111–6122, 2019.

Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: a python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021a.

Jonathan Bac, Evgeny M. Mirkes, Alexander N. Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: A python package for intrinsic dimension estimation. *Entropy*, 23(10), 2021b. ISSN 1099-4300.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Robert Bennett. The intrinsic dimensionality of signal collections. *IEEE Transactions on Information Theory*, 15(5):517–525, 1969.

James A. D. Binnie, Paweł Dłotko, John Harvey, Jakub Malinowski, and Ka Man Yim. A survey of dimension estimation methods, 2025. URL https://arxiv.org/abs/2507.13887.

Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. 2020.

Bradley CA Brown, Anthony L Caterini, Brendan Leigh Ross, Jesse C Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. In *The Eleventh International Conference on Learning Representations*, 2022.

Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Information Sciences*, 328:26–41, 2016.

Paola Campadelli, Elena Casiraghi, Claudio Ceruti, and Alessandro Rozza. Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015:1–21, 2015.

Richard Cangelosi and Alain Goriely. Component retention in principal component analysis with application to cdna microarray data. *Biology direct*, 2(1):1–21, 2007.

Kevin M Carter, Raviv Raich, and Alfred O Hero III. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, 2009.

Anthony L. Caterini, Gabriel Loaiza-Ganem, Geoff Pleiss, and John P. Cunningham. Rectangular Flows for Manifold Learning. *NeurIPS*, 2021.

Paola Causin and Alessio Marta. Estimating dataset dimension via singular metrics under the manifold hypothesis: Application to inverse problems. *arXiv preprint arXiv:2507.07291*, 2025.

Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern Recognition*, 47(8):2569–2581, 2014.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Vittorio Erba, Marco Gherardi, and Pietro Rotondo. Intrinsic dimension estimation for locally undersampled data. *Scientific reports*, 9(1):17133, 2019.

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.

Amir Massoud Farahmand, Csaba Szepesvári, and Jean-Yves Audibert. Manifold-adaptive dimension estimation. In *Proceedings of the 24th international conference on Machine learning*, pages 265–272, 2007.

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

P Gassberger and I Procaccia. Measuring the strangeness of the strange attractor. *Physica D*, 189, 1983.

Marina Gomtsyan, Nikita Mokrov, Maxim Panov, and Yury Yanovich. Geometry-aware maximum likelihood estimation of intrinsic dimension. In Wee Sun Lee and Taiji Suzuki, editors, *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pages 1126–1141. PMLR, 17–19 Nov 2019. URL https://proceedings.mlr.press/v101/gomtsyan19a.html.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Christian Horvat and Jean-Pascal Pfister. Intrinsic dimensionality estimation using normalizing flows. *Advances in Neural Information Processing Systems*, 35:12225–12236, 2022.

Christian Horvat and Jean-Pascal Pfister. On gauge freedom, conservativity and intrinsic dimensionality estimation in diffusion models. *arXiv preprint arXiv:2402.03845*, 2024.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *J. Ed. Psych.*, 24:417–441, 1933.

Michael E Houle. Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications. In *International Conference on Similarity Search and Applications*, pages 64–79. Springer, 2017a.

Michael E Houle. Local intrinsic dimensionality ii: multivariate analysis and distributional support. In *International Conference on Similarity Search and Applications*, pages 80–95. Springer, 2017b.

Kerstin Johnsson, Charlotte Soneson, and Magnus Fontes. Low bias local intrinsic dimension estimation from expected simplex skewness. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):196–202, 2014.

Hamidreza Kamkari, Brendan Leigh Ross, Rasa Hosseinzadeh, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. A geometric view of data complexity: Efficient local intrinsic dimension estimation with diffusion models. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024.

Balázs Kégl. Intrinsic dimension estimation using packing numbers. *Advances in neural information processing systems*, 15, 2002.

D.P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114*, 2014.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Matthäus Kleindessner and Ulrike Luxburg. Dimensionality estimation without distances. In *Artificial Intelligence and Statistics*, pages 471–479, 2015.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17:777–784, 2004.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Guoying Li. Robust regression. *Exploring data tables, trends, and shapes*, 281:U340, 1985.

Gabriel Loaiza-Ganem, Brendan Leigh Ross, Rasa Hosseinzadeh, Anthony L Caterini, and Jesse C Cresswell. Deep generative models through the lens of the manifold hypothesis: A survey and new connections. *arXiv preprint arXiv:2404.02954*, 2024.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

Karl Pettis, Thomas A. Bailey Jr., Anil K. Jain, and Richard C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(1):25–37, 1979. doi: 10.1109/TPAMI.1979.4766873. URL http://doi.ieeecomputersociety.org/10.1109/TPAMI.1979.4766873.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Brendan Leigh Ross and Jesse C. Cresswell. Tractable Density Estimation on Learned Manifolds with Conformal Embedding Flows. *NeurIPS*, 2021.

Brendan Leigh Ross, Hamidreza Kamkari, Tongzi Wu, Rasa Hosseinzadeh, Zhaoyan Liu, George Stein, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. A geometric framework for understanding memorization in generative models, 2025. URL https://arxiv.org/abs/2411.00113.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Alessandro Rozza, Gabriele Lombardi, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. Novel high intrinsic dimensionality estimators. *Machine learning*, 89(1): 37–65, 2012.

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

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Kumar Sricharan, Raviv Raich, and Alfred O Hero. Optimized intrinsic dimension estimator using nearest neighbor graphs. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5418–5421. IEEE, 2010.

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *Forty-first International Conference on Machine Learning*, 2024.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pages 366–381. Springer, 2006.

Rustem Takhanov, Y Sultan Abylkairov, and Maxat Tezekbayev. Autoencoders for a manifold learning problem with a jacobian rank constraint. *Pattern Recognition*, 143: 109777, 2023.

Piotr Tempczyk, Rafał Michaluk, Lukasz Garncarek, Przemysław Spurek, Jacek Tabor, and Adam Golinski. Lidl: Local intrinsic dimension estimation using approximate likelihood. In *International Conference on Machine Learning*, pages 21205–21231. PMLR, 2022.

Piotr Tempczyk, Łukasz Garncarek, Dominik Filipiak, and Adam Kurpisz. A wiener process perspective on local intrinsic dimension estimation methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20859–20866, 2025.

Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Enrico Ventura, Beatrice Achilli, Gianluigi Silvestri, Carlo Lucibello, and Luca Ambrogioni. Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion. *arXiv preprint arXiv:2410.05898*, 2024.

Peter J. Verveer and Robert P. W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on pattern analysis and machine intelligence*, 17(1): 81–86, 1995.

Eric Yeats, Cameron Darwin, Frank Liu, and Hai Li. Adversarial estimation of topological dimension with harmonic score maps. *arXiv preprint arXiv:2312.06869*, 2023.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International conference on machine learning*, pages 12979–12990. PMLR, 2021.

Appendix A

Appendix

A.1 Equivalent formulations of LIDL

The following proposition allows us to conclude that the two approaches to LID estimation we discussed are equivalent.

Proof of Proposition 3.3.1.

Proof. All limits will be taken with $t \to 0$, and we will omit this from notation.

If $t < t^\alpha f(t) < C$, then $\log(t^\alpha f(t)) = O(1)$. Rearranging leads to point 2. Going further, dividing by $\log x$ and observing that $\lim O(1)/\log t = 0$ yields point 3.

Now, assume condition 3. Since $\lim \log t = -\infty$, for the whole expression to tend to $-\alpha < 0$, we need $\lim \log f(t) = \infty$. In this case however we may apply the de l'Hospital rule giving rise to point 4.

It remains to show the final implication. Let $g(t) = t^\alpha f(t) > 0$ be the function we want to bound. Substituting $f(t) = t^{-\alpha} g(t)$ into $\lim_{t\to 0} t f'(t)/f(t) = -\alpha$, we obtain

$$\begin{aligned} \lim \frac{t(-\alpha t^{-\alpha-1} g(t) + t^{-\alpha} g'(t))}{t^{-\alpha} g(t)} &= \\ &= \lim \left(-\alpha + \frac{g'(t)}{g(t)} \right) \\ &= -\alpha, \end{aligned} \tag{A.1}$$

amounting to $\lim(\log g(t))' = 0$. But this means that for some $\epsilon > 0$, the derivative $(\log g(t))'$ is bounded on $(0, \epsilon)$, implying that $\log g(t)$ is Lipschitz and therefore bounded on this interval. This yields the desired estimates. □

Under the notation of Sec. 3.3 we have the following two propositions.

Proposition A.1.1. *For t near 0 the following estimate holds*

$$\log \rho_t(x) = \beta(x) \log \sqrt{t} + O(1). \tag{A.2}$$

Proof. By using Proposition 3.3.1 with $f(t) = \rho_t(x)$, we can see that its last condition holds with $\alpha = -\beta(x)/2$. Thus, the equivalent condition 2. yields the desired equality. □

The next proposition provides an elegant expression for $\beta_t(x)$, and consequently for $\beta(x)$, expressed in terms of the density ψ on $\mathbb{R}^d$.

Proposition A.1.2. *For $t > 0$ and $x \in S = \mathbb{R}^d \subseteq \mathbb{R}^D$ we have*

$$\beta_t(x) = d - D + \frac{\Delta_x(\psi * \phi_t^d)(x)}{\psi * \phi_t^d(x)} \cdot t. \tag{A.3}$$

Proof. By (3.12) and Lemma 3.2.1 we obtain

$$\begin{aligned} \beta_t(x) &= \frac{t}{\rho_t(x)} \Delta \rho_t(x) \\ &= d - D + \frac{t(2\pi t)^{(d-D)/2} \Delta_x(\psi * \phi_t^d)(x)}{\rho_t(x)}, \end{aligned} \tag{A.4}$$

where in the second term $(2\pi t)^{(d-D)/2}$ cancels out after expanding ρ_t in the denominator. □

A.2 Experimental details from LIDL experiments

When using LIDL with parametric density estimators on non-synthetic datasets, choosing hyperparameters is a challenge. We cannot directly estimate the error of the algorithm because we does not have access to ground truth LID. However, we observed in our experiments that choosing the hyperparameters leading to models minimizing negative log-likelihood on the validation set is a good strategy for minimizing the error of the LID estimate. We apply this approach in all our experiments; as density estimators we employ MAF [Papamakarios et al., 2017], RQ-NSF [Durkan et al., 2019] and Glow [Kingma and Dhariwal, 2018].

In scalability experiments we used 3 types of datasets. Uniform distribution on interval $(0, 1)$ on a hypercube (denoted by $\mathcal{U}_N$, where N is dimensionality of a cube), multivariate Gaussian ($\mathcal{N}_N \subseteq \mathbb{R}^N$) where N is dimensionality of a distribution and data space, and ($\mathcal{N}_N \subseteq \mathbb{R}^{2N}$), where we embedded N-dimensional Gaussian in $2N$-dimensional space by duplicating each coordinate. In each experiment we used 11 δs between 0.025 and 0.1.

A.3 Experimental details from algorithm comparison

ESS setup

In our experiments we used ESS implementation from Bac et al. [2021a] which can be found under scikit-dimension.readthedocs.io with default hyperparameters if not stated otherwise in the text. Because this implementation cannot calculate LID on unseen data and we can only get predictions for the training set, we have made a decision to jointly train on training and test datasets, but only present the results for the test part of the data.

NB setup

For NB experiments, we used the PyTorch implementation along with the environment setup provided by Stanczuk et al. [2024], which is publicly available under github.com/GBATZOLIS/ID-diff. The conducted experiments have been carried out using DDPM Ho et al. [2020] with Adam optimizer ($\alpha = 2\text{e}{-4}$, $\beta = 0.9$, $\epsilon = 1\text{e}{-8}$). The convergence has been assessed using the validation holdout dataset. For a more detailed

description of hyperparameters for all conducted experiments (except for Arrows), please refer to MNIST/config.py, which is available in the aforementioned repository. The only hyperparameter adjustments we made were connected to the shape of the input data. For the Arrows experiment, we used based our config on the celebA/ddpm.py file.

LIDL setup

To perform experiments with LIDL, we utilised the official PyTorch implementation (github.com/opium-sh/lidl) released by Tempczyk et al. [2022]. All the experiments (except the ones on the original manifold and padding) has been performed on MAE Papamakarios et al. [2017] with 5 layers and 5 hidden units. For the version with the original manifold and padding, we had to went down with the number of layers to 4 due to unfavourable scalability and performance constraints. Each experiment has been perofrmed with the diffent set of δ triplets, such that $\delta \in \left\{\left(2^{-(n-1)}, 2^{-n}, 2^{-(n+1)}\right) | n \in \{1, \ldots, 7\}\right\}$. The rest of the hyperparameters remained in line with the version used on image datasets.

FLIPD setup

To obtain the results presented in this article we utilized a fork from August 12, 2024, of the repository listed under github.com/layer6ai-labs/flipd by the authors of FLIPD Kamkari et al. [2024]. As for now authors recommend utilizing github.com/layer6ai-labs/dgm_geometry for LID estimation experiments. For all datasets, the architecture of the diffusion model was the same up to input layer dimensionality. We used the same MLP architecture for diffusion models as authors which is described in Section C appendix of FLIPD article Kamkari et al. [2024]. For all datasets we ran 1000 epochs of training for each network, choosing the best model measured by its' validation loss. For the datasets with known LID values we chose t which yielded lowest MAE error given all samples from test set. For datasets with unknown LID values we used heuristic presented with original article with the caveat that we searched for the *knee* for values of t larger than the maximum average estimated LID. We did it as for many of the datasets for certain t values instabilities occurred which resulted in extreme LID estimates. Those values prevented the kneed package from performing correctly as it assumes monotonicity of the function.

Table A.1: Approximate duration for model training on RTX 2080 Ti GPUS.

Dataset	NB	FLIPD	LIDL
FMNIST (upscaled)	1d @ 2GPU	5.5h @ 2 GPU	9h @ 1 GPU
FMNIST (downscaled)	0.5d @ 2GPU	4h @ 2 GPU	6h @ 1 GPU
FMNIST (stretched x^4)	1.5d @ 2GPU	4.5h @ 2 GPU	7.5h @ 1 GPU
FMNIST (stretched $x^{0.25}$)	2d @ 2GPU	4.5h @ 2 GPU	7.5h @ 1 GPU
FMNIST (add dim, +4d)	2d @ 2GPU	5h @ 2 GPU	8h @ 1 GPU
FMNIST (add dim, +8d)	4d @ 2GPU	6h @ 2 GPU	10h @ 1 GPU
Spiral	5.5d @ 2GPU	10.5h @ 2 GPU	22h @ 1 GPU
Uniform	3.5d @ 2GPU	6h @ 2 GPU	9.5h @ 1 GPU
Funnel	4.5d @ 2GPU	6h @ 2 GPU	9.5h @ 1 GPU
Moon	5d @ 2GPU	6.5h @ 2 GPU	10h @ 1 GPU
Gaussians	1d @ 2GPU	7.5h @ 2 GPU	11h @ 1 GPU
Spheres	1d @ 2GPU	7h @ 2 GPU	10h @ 1 GPU
Spaghetti	1.5d @ 2GPU	6h @ 2 GPU	9h @ 1 GPU
Arrows	10d @ 8GPU	10h @ 2 GPU	–

A.4 Some other results for LIDL and FLIPD from the comparison

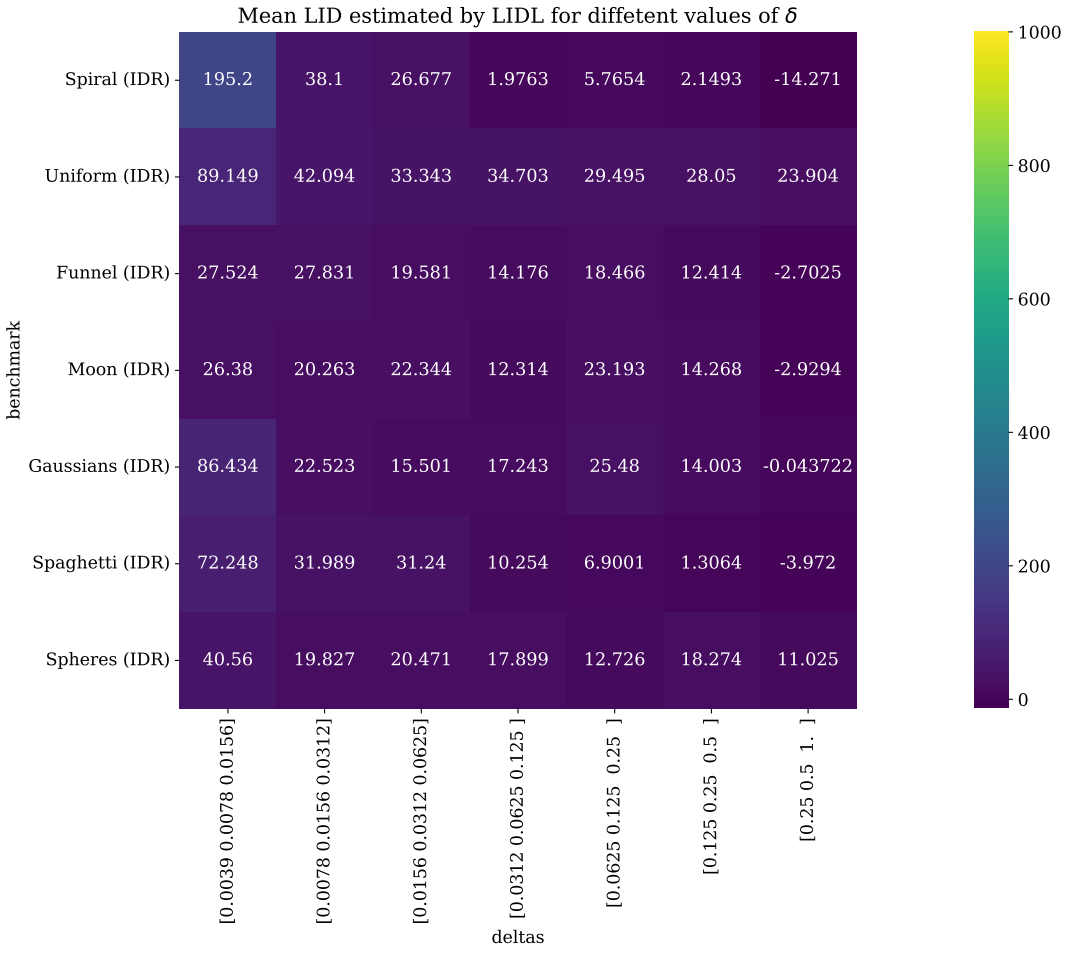


Figure A.1: On this plot we present how values of average LID estimate changes for different values of δ parameter and different IDR datasets for LIDL algorithm.

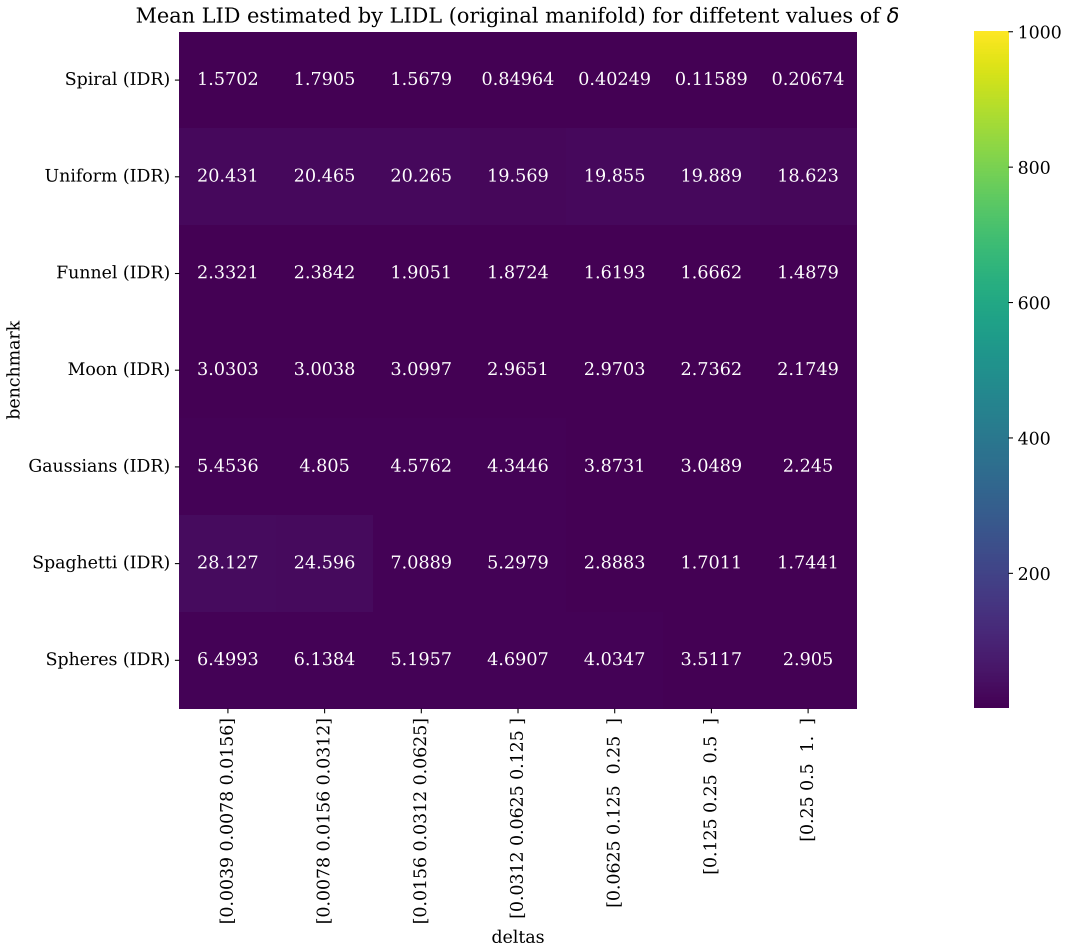


Figure A.2: On this plot we present how values of average LID estimate changes for different values of δ parameter and different datasets for LIDL algorithm. Those datasets are made of original manifold coordinates before IDR transformation. Ambient space of those datasets is 30.

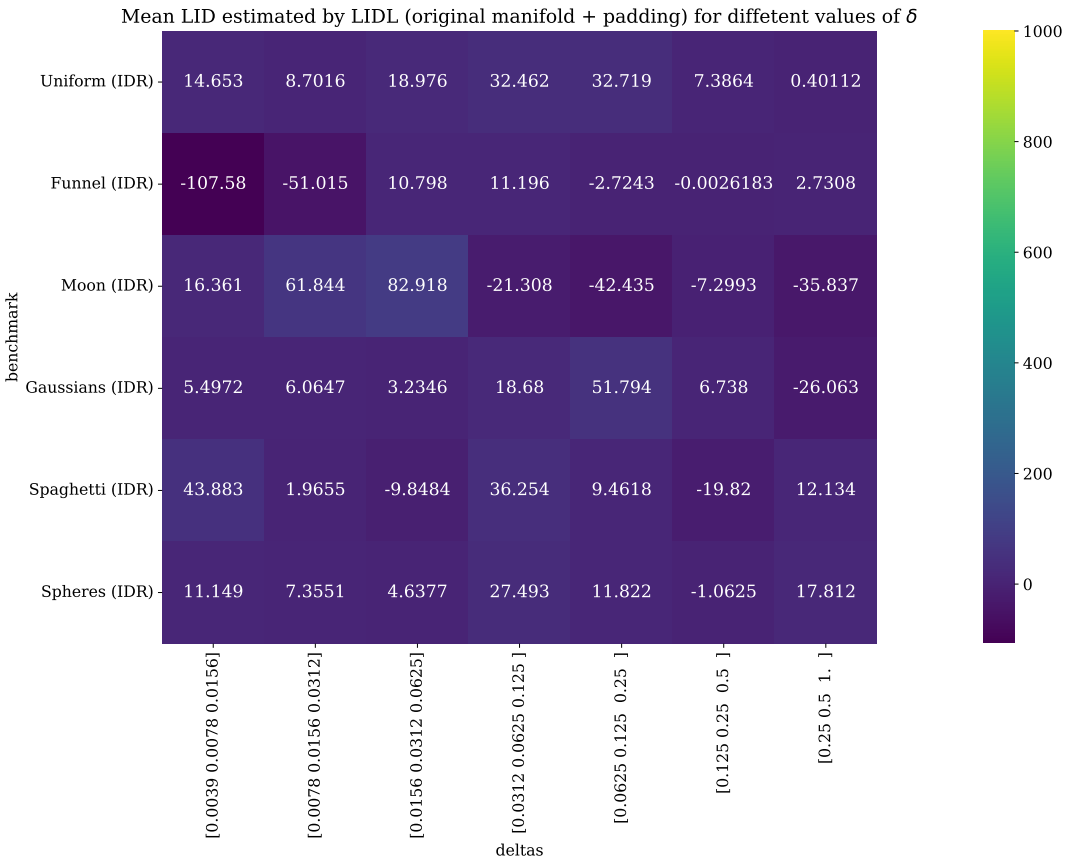


Figure A.3: On this plot we present how values of average LID estimate changes for different values of δ parameter and different datasets for LIDL algorithm. Those datasets are made of original manifold coordinates before IDR transformation padded with 0 to be of higher dimension. Ambient space of those datasets is 784.

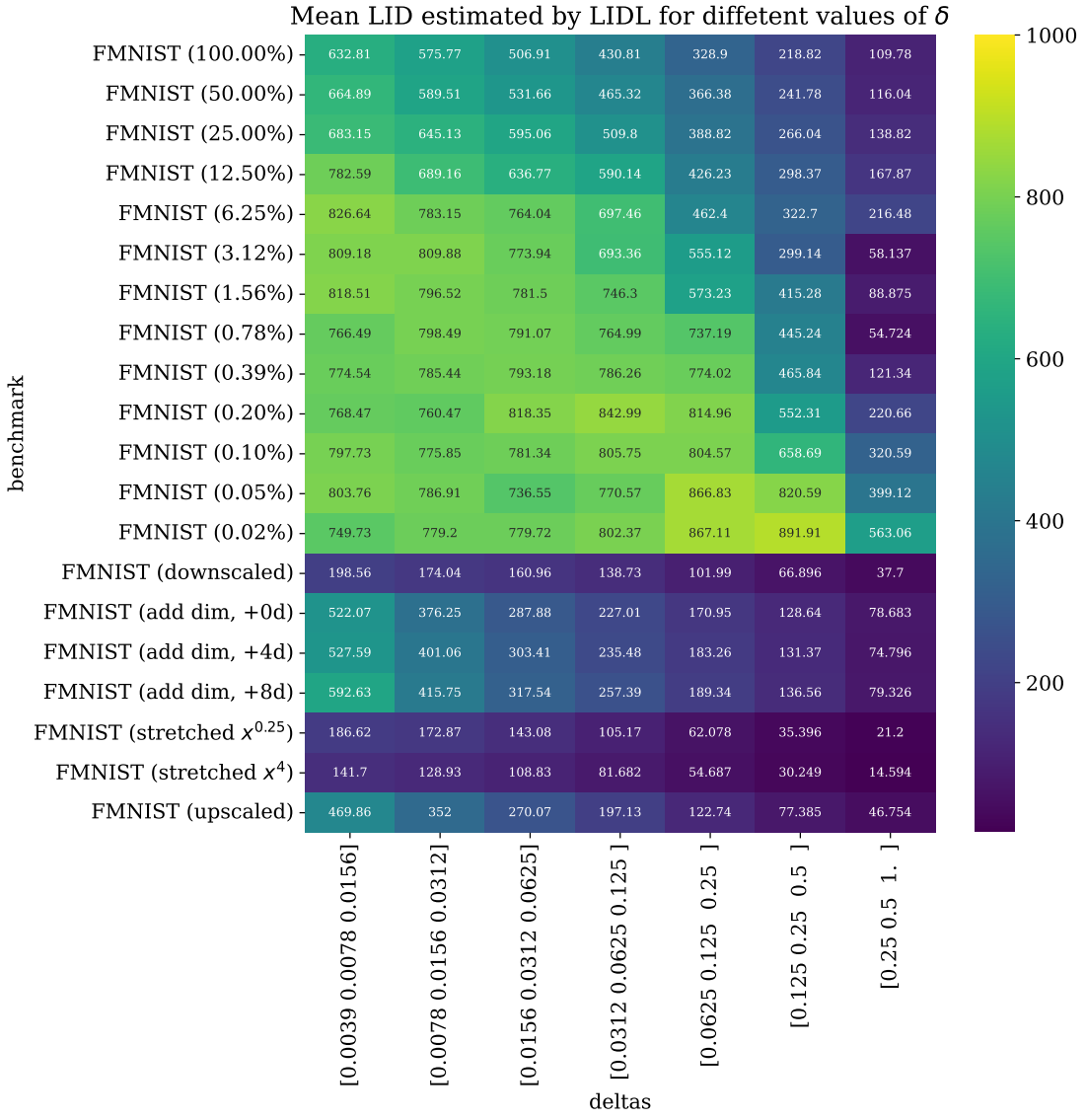


Figure A.4: On this plot we present how values of average LID estimate changes for different values of δ parameter and different modified FMNIST datasets for LIDL algorithm.

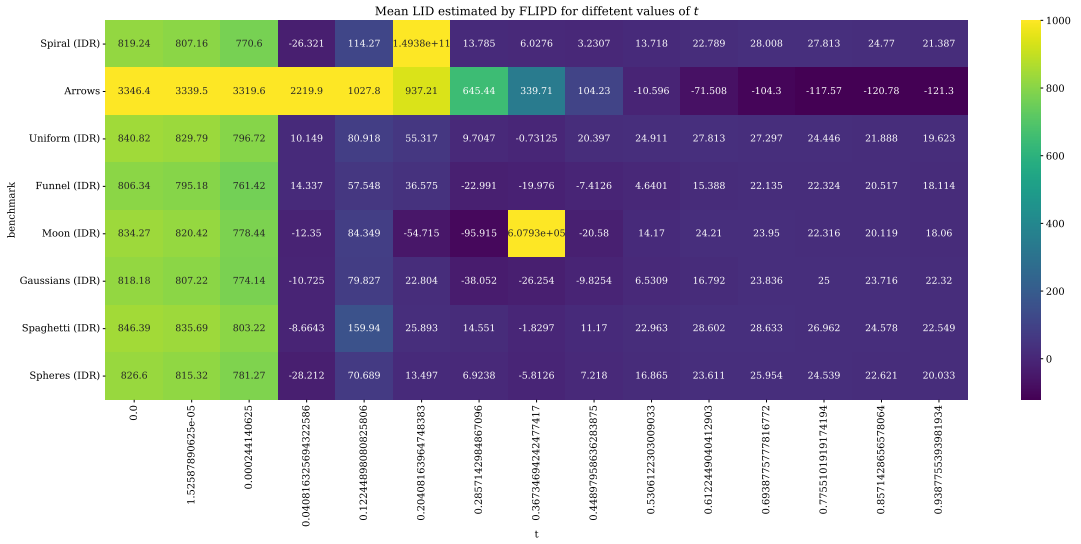


Figure A.5: On this plot we present how values of average LID estimate changes for different values of t parameter and different IDR datasets for FLIPD algorithm. To improve clarity, we present results for every 4th t.

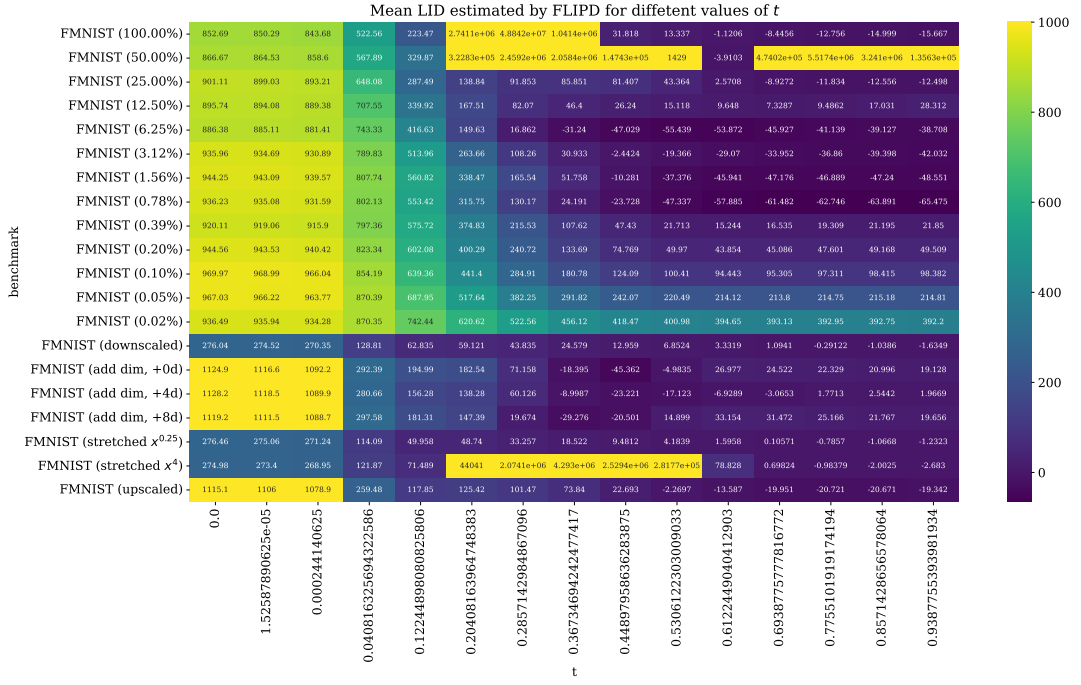


Figure A.6: On this plot we present how values of average LID estimate changes for different values of t parameter and different modified FMNIST datasets for FLIPD algorithm. To improve clarity, we present results for every 4th t.

A.5 Classical algorithms compared with LIDL

Table A.2: Algorithms used for comparison. All implementations from scikit-dimension library [Bac et al., 2021b].

Name	Shortcut	citation
CorrInt	COR	Gassberger and Procaccia [1983]
MADA	MAD	Farahmand et al. [2007]
MLE	MLE	Levina and Bickel [2004]
lPCA	LPC	Cangelosi and Goriely [2007]
KNN	KNN	Carter et al. [2009]
DANCo	DAN	Ceruti et al. [2014]
MiND_ML	MIN	Rozza et al. [2012]
ESS	ESS	Johnsson et al. [2014]
MOM	MOM	Amsaleg et al. [2018]
FisherS	FIS	Albergante et al. [2019]
TwoNN	TWO	Facco et al. [2017]
TLE	TLE	Amsaleg et al. [2019]