Revealing Intrinsic Dimensionality: A Survey on Estimators and Searching for Potential Utilities

Yerim Kim Eunyi Lyou Hee-Hwan Wang Dongyeon Woo
Seoul National University

Abstract

There are ongoing works trying to understand the representation space of deep neural network. In this work, we are investigating the representation space using a geometric property, Intrinsic Dimension (ID), the minimal number of coordinates required for describing the distribution of representations. We first compare existing ID estimators and find the accurate and robust estimator. Then, we perform overall experiments to derive useful correlation between ID and other general metrics on various deep learning scenarios, especially data augmentation and transfer learning.

1. Introduction

It is well known that high-dimensional data (e.g. natural languages and images) consist of low-dimensional structure [28, 31]. Deep Neural Networks (DNN) is a powerful tool for extracting low-dimensional representations from high-dimensional data, such as Convolutional Neural Networks (CNN) in image classification [6].

Though DNN could precisely predict output from unseen data (i.e. generalizability), it still remain unclear why they perform well and what does the representation imply. A large body of prior works try to answer the unfolded question [29, 35]. Intrinsic Dimensionality (ID) is one of them which starts to give us quantitative explanations of representation manifold, derived when high-dimensional data passes forward neural networks [31].

Intrinsic Dimension (ID), a geometric property of data representations, is the minimal number of coordinates required for describing the distribution of data representations with minimal information loss. By measuring ID, we could measure the dimensionality of the representation manifold, and it leads to the estimation of representation complexity. Since ID is one of the major physical characteristic of the representation space, it could be used as a key for unveiling DNN's hidden part.

It is shown that the dimensionality plays an important role in deciding the number of samples to train machine learning models (i.e. sample complexity) [25], which is related to the generalization capacity of models [12, 28]. It is also known that measured IDs of data representations are negatively proportional to the model performance [2] with regards to overparameterization. Considering the redundant number of parameters in models reduces generalization capacity [8], ID of data representations may be a useful tool to compare redundancy and generalization capacity between models.

In this work, we look into Intrinsic Dimension from investigating the existing ID estimators to various applications using ID estimation. First, we compare various ID estimators to give understanding of the existing ID measurement methods. Considering the accuracy and robustness of the estimators on real and synthetics images, we choose TwoNN [9] as our base method on following experiments.

Then, we perform extensive experiments to find correlation between ID and other metrics. We focus on the most elementary scenarios on deep learning tasks, data augmentation and transfer learning. We find patterns of ID at the intermediate layers during training with data augmentation or finetuning, and show the relationship between ID estimation and the validation accuracy for each scenario.

Here is a summary of our contributions:

- We introduce various ID estimators and show their accuracy and robustness on the real and the synthetic dataset. Considering the robustness and the computational efficiency, we use TwoNN [9] for the remaining parts.
- We show how ID estimation could be utilized for analyzing data augmentation in respect of data model accuracy and the role of each CNN layer.
- We show the pattern between the ID estimation values and the validation accuracy, which is also related with transferred model's performance and overfitting.

2. Related works

Intrinsic Dimensionality and generalization capacity. ID of data representation could help us to reveal requirements

(a) Yorkshire Terrier, dim=8 (b) Banana, dim=8 (c) Yorkshire Terrier, dim=32 (d) Banana, dim=32

Figure 1. Synthesized images with varying latent dimensionality by BigGAN [5]

for generalizable neural networks. First, ID gives explanations linking sample complexity to the model performance of CNN. The larger the number of ID (i.e. manifolds are more complex), the larger the number of samples are required (i.e. sample complexity increases) for the learning function to approximate underlying manifolds of data [25, 26]. Building on these theoretical literatures, recent study with synthetic and natural image datasets show that the number of training samples, required for CNN to achieve the highest accuracy in test samples, increases as ID increases [28]. Second, ID reveals the difference between generalization and memorization in training CNN. [36] shows that CNN has a risk of memorizing all the training samples and poorly predicting the test samples by comparing the accuracy of models trained with original image data to randomly re-labeled image data. A recent study adopts this framework and compares ID of each layer in two different models [2]. As a result of this study, a model trained with original data shows an increasing tendency of ID in earlier layers and vice versa in later layers. In contrast, a model trained with randomly labels data shows a peak of the number of ID in later layers, which reflects poor generalization capacity of the model [2, 23].These experimental results imply the utility of ID as a quantitative metric for evaluating the generalization capacity of CNN in classification tasks.

Utility of ID. Recent studies have explored ID, and revealed its utility for DNN. With data composed of heterogeneous manifolds, ID has been used to assess the complexity of a search query [16, 19], to control the early termination of a search [14, 15], and to detect outliers [32]. Also, it has been extensively validated that the performance of neural networks depends on the ID, not the high extrinsic dimensionality of data [24]. Therefore, ID is regarded as an important component to overcome the curse of dimensionality [4, 21, 22, 34]. [24] shows that the rates of approximation and generation errors only depend on the ID. [12] shows the usefulness of estimating ID at a fundamental level and a practical level. At a fundamental level, ID determines the capacity and complexity of the representation data variation and analyzes generalization capability by estimating the redundancy of the representation. At a practical level, ID may help us to increase the generalization performance of neural networks with reduced memory and time since ID contains low dimensional and essential representations of images.

ID estimation methods. The efficiency and effectiveness of ID in describing much higher dimensions of real-world data distributions highly depend on the choice of estimators [1]. There are two primary approaches to estimating the ID, the global approach and the local approach. Global ID defined by [3] is the minimum number of parameters to describe the data while minimizing the information loss, and is estimated globally with a whole data set. These global estimation methods try to linearly estimate the ID of manifolds underlying the whole data distribution. However, global estimation methods have not worked well because it has limitations in preserving the local structures of manifolds [27]. Local estimation methods simply try to estimate the topological dimension of the data manifolds using only the information contained in the sample neighborhood [17]. [10] proposed the fundamental algorithm, and some variants of Fukunaga-Olsen's algorithm have been proposed to locally estimate ID. Among many local methods, Nearest Neighbor Algorithm [27] and its variants are widely adopted in various works [7].

Nearest neighbor(NN)-based estimation methods describe the distribution of neighboring data points as functions of the ID, d with an assumption that neighborhood points are uniformly drawn from the small dimensional manifold. Among NN-based estimation methods, kNN [13] uses k-nearest data points, and twoNN [9] uses the distance of the first two nearest data points. Estimation with smaller neighborhood size allows to lower the influence of dataset inhomogeneities [9]. Maximum Likelihood Estimator(MLE) [20] calculates the NN estimators with a Poisson process approximation, maximizing the log-likelihood

	dim=4	dim=8	dim=16	dim=32
kNN	3.00	4.00	5.00	5.33
GeoMLE	0.30	14.52	52.08	104.26
TwoNN	**4.66**	10.68	18.30	**24.71**
MLE(k=14)	4.91	**9.41**	**14.44**	17.87

Table 1. ID estimation with various latent dimension variability

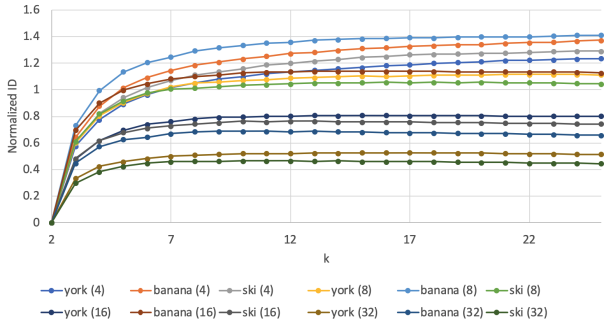


Figure 2. Normalized ID of MLE estimator with varying k

	TwoNN			MLE		
Img Resize	ID (cifar10)	ID (svhn)	ID (cub200)	ID (cifar10)	ID (svhn)	ID (cub200)
32	17.62	18.27	9.18	25.92	18.31	28.55
64	17.05	18.17	9.26	25.92	18.31	29.50
128	17.06	18.17	9.38	25.92	18.31	29.69
256	17.07	18.17	9.38	25.92	18.31	29.67
std	0.28	0.05	**0.10**	**0.00**	**0.00**	0.54

Table 2. ID estimation with image resizing

	TwoNN		MLE	
Sampling Rate	ID (cifar10)	ID (svhn)	ID (cifar10)	ID (svhn)
0.05	22.92	16.89	26.28	18.04
0.1	20.78	17.03	25.99	18.23
0.2	18.33	18.44	26.24	18.32
0.5	17.58	18.38	26.06	18.29
1.0	17.62	18.27	25.92	18.31
std	2.34	0.77	**0.16**	**0.11**

Table 3. ID estimation with various sampling rate

of the observed process. GeoMLE [11] is a kind of MLE method with geodesic distance [18], calculating the distance along the surface of a manifold.

3. Comparison of ID Estimators

3.1. Estimation Accuracy

We first compare the accuracy of baseline ID estimators, kNN [13], GeoMLE [11], TwoNN [9], and MLE [20]. We generate synthetic images using pretrained BigGAN [5]. Each image has 128×128 RGB data, and 1,500 samples from 3 categories (Yorkshire Terrier, Banana, Ski) are sampled with limiting the number of varying latent variables to 4, 8, 16, and 32. Figure 1 shows several samples from the generated images.

Table 1 shows the estimated ID value for various estimators with varying latent dimension. It is shown that TwoNN and MLE show much accurate results compared with other estimators, specifically MLE performs better at the intermediate latent dimension regime while TwoNN is more accurate for large latent dimensional images.

Note that MLE estimator needs additional hyperparameter k to determine the number of neighborhood samples used for measurement. We plot Figure 2 to find the correlation between k value and the ID estimation accuracy, and empirically find that MLE estimator shows stable ID value at $k > 8$.

3.2. Estimation Robustness

Robustness to the variable factors is another element for estimator. We measure the standard deviation of estimated value for TwoNN and MLE. Note that other methods are emitted due to the low accuracy discovered at Section 3.1.

We compare the variation of estimation with resized input dataset. *cifar-10*, *svhn*, and *cub200* [33] are chosen as the base dataset, and their default image size is (32×32), (32×32), and (150×150), respectively. We resize the input image with bicubic interpolation into (32×32), (64×64), (128×128), and (256×256).

Followed by the comparison with various resizing option, we also compare the ID esimation with different sampling rate to find the impact with the dataset size. Varying the sampling rate as $[0.05, 0.1, 0.2, 0.5, 1.0]$, standard deviation is calculated at *cifar-10* and *svhn*.

Table 2 and Table 3 illustrate the robustness of ID estimation for TwoNN and MLE. It is found that both estimators show reasonable estimation robustness while MLE is slightly better at small and scarse dataset.

3.3. TwoNN

For the following sections, we use TwoNN [9] to estimate the Intrinsic Dimension of data and representation manifold, since TwoNN method shows both accurate and robust measurements throughout Section 3.1 and Section 3.2. There is also an engineering benefit that it uses only the distances of the first and the second nearest neighbor for each sample, so it can efficiently estimate the ID of high-dimensional data. Note that considering the high computational complexity, MLE is not used for the rest of our experiments despite of better robustness. We will briefly introduce the basic idea of TwoNN and illustrate the implementation of the methods in this chapter.

Let $r_l^{(i)}$ be the l-th nearest neighbor of the sample i and

d be the intrinsic dimensionality. Then the the volume of space enclosed by two successive shell with radius $r_l^{(i)}$ and $r_{l-1}^{(i)}$ is given by

$$\Delta v_l = w_d \left({r_l^{(i)}}^d - {r_{l-1}^{(i)}}^d \right)$$

where w_d is the volume of the unit d-sphere. Assuming the the constant density ρ around the sample i,

$$P(\Delta v_l^{(i)} \in [v, v+dv]) = \rho e^{-\rho v} dv$$

Let R be $\frac{\Delta v_i}{\Delta v_j}$, then pdf for R is derived as:

$$g(R) = \frac{1}{(1+R)^2}$$

and by defining a quantity $\mu = \frac{r_2}{r_1}$ and letting $R = \frac{\Delta v_2}{\Delta v_1}$

$$\begin{aligned} R &= \mu^d - 1 \\ f(\mu) &= d\mu^{-d-1} \\ F(\mu) &= (1 - \mu^{-d}) \end{aligned}$$

where $f(\mu)$ and $F(\mu)$ are cdf, pdf of μ, respectively. So, by estimating the cumulative density function of the value μ, we can derive the intrinsic dimension d through the following equation:

$$\frac{\log(1 - F(\mu))}{\log(\mu)} = d \tag{1}$$

Note that equation 1 can be calculated from the estimation of μ and $F(\mu)$.

Algorithm 1 ID measurement with mini-batch

Require: $X = (\text{data samples}), s = (\text{mini-batch size})$
  $N \leftarrow (\text{sample number})$
  $m \leftarrow ceil(N/s)$
  $dists = Zeros(N, 3m)$
  for i in range(m) **do**
    for j in range(m) **do**
      $X_i \leftarrow X_{si:\min(s(i+1),N)}$
      $X_j \leftarrow X_{sj:\min(s(j+1),N)}$
      $dist \leftarrow EuclideanDist(X_i, X_j)$
      $dists_{si:\min(s(i+1),N),3j:3(j+1)} \leftarrow Top3s(dist)$
    end for
  end for
  for i in range(N) **do**
    $\mu_i \leftarrow Top3(dists_i)/Top2(dists_i)$
  end for
  $\mu \leftarrow Sort(\mu)$
  $cdf \leftarrow Arange(N)/N$
  $id \leftarrow LinearReg(\log(\mu), -\log(1 - cdf))_{slope}$
  return id

Augmentation Method	Dataset ID
Raw dataset	11.4
RandFlip&Rotate	27.6
RandCrop&Resize	43.1
ColorJittering	39.7
All	56.6

Table 4. Dataset ID with various image augmentation method

ID measurement with mini-batch. Since measuring pairwise distances of samples requires $O((nm)^2)$ memory size with n samples with m representation dimension, it is nearly impossible to measure ID at dataset with large number of samples and the high dimensionality. So, we suggest to split the measuring process into small mini-batches.

We describe the detailed process at Algorithm 1.

4. Application of ID Estimation

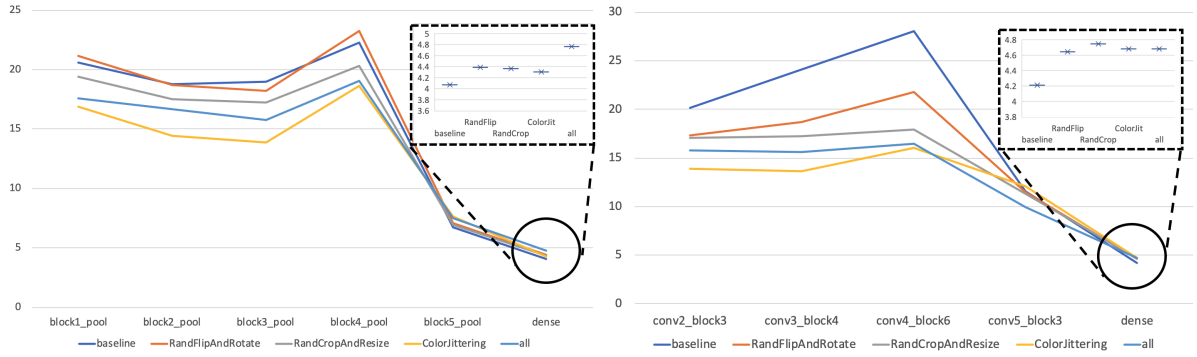
4.1. ID at Data Augmentation

Data augmentation encompasses various techniques that enhance the size of training data without changing their semantics, enabling models to robustly learn features of data [30]. Diverse data augmentation methods, such as flipping, cropping, and changing the intensity of color space, are widely applied to computer vision [30]. In these experiments, we explore the relationship between data augmentation methods and generalizability of models in terms of ID. We apply three types of widely-used data augmentation methods, i.e., random flipping-rotating, random cropping, and random color jittering, to image classification tasks with *cifar-10* dataset. We use VGG-16 and ResNet-50 as our training models. We train models with non-augmented (e.g., baseline) or augmented images (e.g., RandFlipAndRotate, RandCropAndResize, ColorJittering, and all three types of augmentation methods) and estimate ID of features extracted from test datasets to which augmentation methods are not applied.

As seen in Table 4, training dataset without any data augmentation is constituted by the lowest number of ID, whereas the training dataset to which all types of data augmentation methods are applied shows the highest number of ID. It could be said that data augmentation enhanced the complexity of the data space of images.

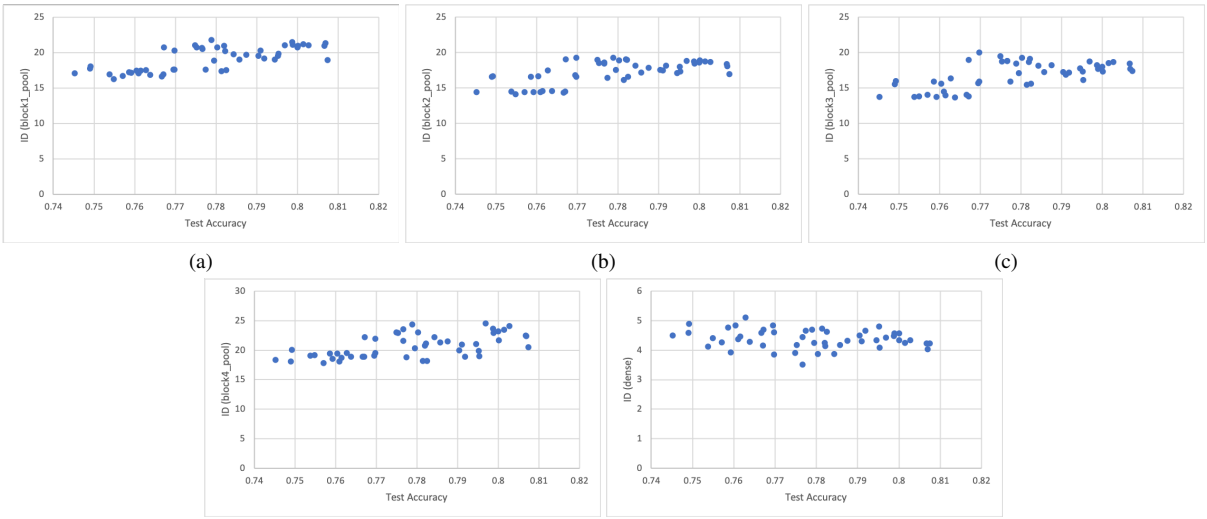
In Figure 3a and Figure 3b, ID of features extracted from intermediate layers reaches the peak point at right before the last pooling layer (i.e., conv4 block6 in ResNet-50 and block4 pool in VGG-16), and they become similar at dense layer. Of note, after the dense layer, ID of features extracted by models trained with augmented images shows higher ID than features extracted by models trained with non-augmented images.

Interestingly, in ResNet-50, IDs of features extracted



(a) VGG-16 (b) ResNet-50

Figure 3. ID estimation at intermediate layers



(a) (b) (c)

(d) (e)

Figure 4. Test accuracy vs. ID measure at intermediate layers of VGG-16

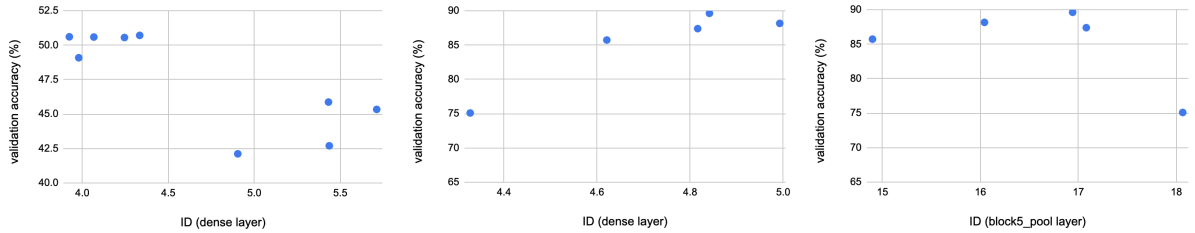
by the baseline model at each pooling layer are higher than IDs of extracted features by models trained with augmented images. Whereas, at the dense layer, ID of the baseline model's features is the lowest compared to augmented images' features. In VGG-16, except for *RandFlipAndRotate* model, IDs of features extracted by models trained with augmented images are lower than the baseline model's ID. Regarding changing patterns of the number of ID across the layers, both model architectures show quite distinct configurations. While ResNet-50 shows an increasing-to-decreasing pattern, VGG-16 shows a decreasing-to-increasing pattern. In short, regardless of whether models were trained with non-augmented images or augmented images, ID changing patterns across pooling layers are quite different according to model architecture.

We find another interesting result at Figure 4. After aggregating the overall relationship between the test accuracy and the ID estimation at each intermediate layers, we find that there are positive relationship at pooling layers and the negative relationship at dense layer. It might imply that the well-trained model should preserve high ID value for the intermediate dimension and the dimension should be successfully compressed at the last layer.

4.2. ID at Transfer Learning

In this section, we show how ID estimation works during transfer learning, one of the most common scenario in deep learning. We observe the ID measurement with parametrizing datasets, models, finetuning layers, but there are few consistent results over our experiments.

We assume the transfer learning case that the pretrained VGG-16 and Resnet-50 models are being finetuned on *cifar-10* and *cub200* dataset. Note that each dataset represents 'easy' and 'hard' transfer scenario, since finetuning at *cifar-10* easily achieves high validation accuracy while finetuning at *cub200* usually fails due to the categories with



(a) cub200 / ResNet-50 (b) cifar-10 / VGG-16 (c) cifar-10 / VGG-16

Figure 5. ID vs. test accuracy after finetuning

similar features (distinguishing the species of birds) and the small number of samples in the dataset. During finetuning, we tested scenarios with freezing/unfreezing model layers in various way. And the ID estimations are performed at featured pooling layers of each model. Each finetuning task continues until the early stopping occurs with validation accuracy metic, or it reaches the maximum number of epochs, 100.

Interestingly, we observed contradicting results at two distinct experiments, one with cub200, ResNet50 on Figure 5a, and the other with cifar-10, VGG16 on Figure 5b. At Figure 5a, we could observe the negative relationship between ID at dense layer and the validation accuracy, while we find positive relationship at Figure 5b. It seems like the different patterns are occurred from the different validation state, the first one is at converging / non-converging state and the second experiment is at non-overfitting / overfitting regime. That is, it might be possible that ID at the last layer increases while the model is being trained, but starts to be decrease when the model is being overfitted.

Figure 5c shows that the correlation between ID at block5_pool layer is slightly negative which is reversed at the next layer, dense layer.

We also investigate the change of ID estimation during the finetuning VGG-16 model on cifar-10. As it is shown at Figure 6, ID values are dynamically changing during finetuning. During the experiment, the highest validation accuracy is measured at 5 20 epochs. So we can find the pattern that all intermediate layers' ID change during training phase, and only a single block3_pool3 layer's ID is changing when the model performs best. After that, ID values are start to being diverged.

5. Conclusions

In this research, we cover a wide range of tasks regarding ID estimation from comparing estimators to applying into common deep learning scenarios. By comparing previously suggested ID estimators of the estimation accuracy and the robustness, we find both TwoNN and MLE show impressive performance. During the training with data augmentation, we find that models trained with augmented images are capable of extracting features represented by less number of ID than models trained with non-augmented-images at the intermediate layers while being opposite at the output. This may imply that data augmentation helps models to capture simpler manifold structures at the earlier stage, while more complex but sufficient to represent the variation of each category at the last stage. On analyzing ID the transfer learning, we find that there are distinct correlations of ID and the validation accuracy at training and overfitting regime. Also, we find a clue to detect overfitting by utilizing the ID values at intermediate layers of model.

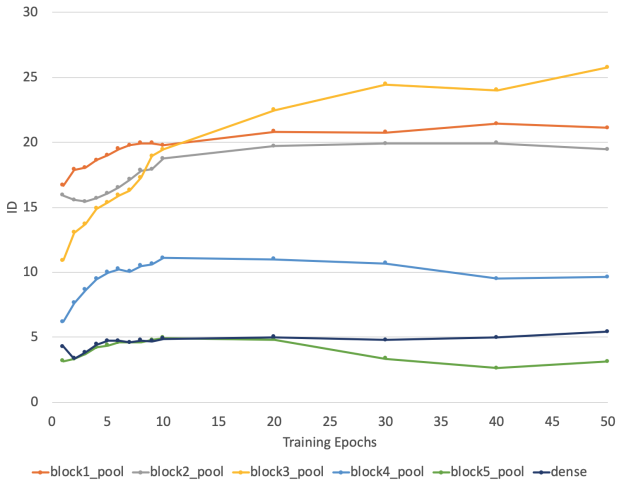


Figure 6. ID at intermediate layers during finetuning

Since we have performed only few experiment for each concept, it needs large-scale experiments to validate the hypothesis concretely. But, we suggest that our research provides possibilities of using ID estimation as a tool of judging the state of training neural network which might lead to unveil the hidden black box.

References

[1] 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. 2

[2] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *arXiv preprint arXiv:1905.12784*, 2019. 1, 2

[3] Robert Bennett. The intrinsic dimensionality of signal collections. *IEEE Transactions on Information Theory*, 15(5):517–525, 1969. 2

[4] Peter J Bickel and Bo Li. Local polynomial regression on unknown manifolds. In *Complex datasets and inverse problems*, pages 177–186. Institute of Mathematical Statistics, 2007. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. 2, 3

[6] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 1

[7] Francesco Camastra. Data dimensionality estimation methods: a survey. *Pattern recognition*, 36(12):2945–2954, 2003. 2

[8] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *arXiv preprint arXiv:1306.0543*, 2013. 1

[9] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7, 2017. 1, 2, 3

[10] Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 100(2):176–183, 1971. 2

[11] Marina Gomtsyan, Nikita Mokrov, Maxim Panov, and Yury Yanovich. Geometry-aware maximum likelihood estimation of intrinsic dimension. In *ACML*, 2019. 3

[12] Sixue Gong, Vishnu Naresh Boddeti, and Anil K Jain. On the intrinsic dimensionality of image representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3987–3996, 2019. 1, 2

[13] Daniele Granata and Vincenzo Carnevale. Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets. *Scientific Reports*, 6, 2016. 2, 3

[14] Michael E Houle, Xiguo Ma, Michael Nett, and Vincent Oria. Dimensional testing for multi-step similarity search. In *2012 IEEE 12th International Conference on Data Mining*, pages 299–308. IEEE, 2012. 2

[15] Michael E Houle, Xiguo Ma, Vincent Oria, and Jichao Sun. Efficient algorithms for similarity search in axis-aligned subspaces. In *International Conference on Similarity Search and Applications*, pages 1–12. Springer, 2014. 2

[16] Michael E Houle and Michael Nett. Rank-based similarity search: Reducing the dimensional dependence. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):136–150, 2014. 2

[17] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. 2

[18] Rasa Karbauskaitė, Gintautas Dzemyda, and Edmundas Mazėtis. Geodesic distances in the maximum likelihood estimator of intrinsic dimensionality. *Nonlinear analysis: modelling and control*, 16(4):387–402, 2011. 3

[19] David R Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 741–750, 2002. 2

[20] Jisu Kim, Alessandro Rinaldo, and Larry A. Wasserman. Minimax rates for estimating the dimension of a manifold. *ArXiv*, abs/1605.01011, 2019. 2, 3

[21] Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. *arXiv preprint arXiv:1110.4300*, 2011. 2

[22] Samory Kpotufe and Vikas K Garg. Adaptivity to local smoothness and dimension in kernel regression. In *NIPS*, pages 3075–3083, 2013. 2

[23] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018. 2

[24] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *J. Mach. Learn. Res.*, 21:174–1, 2020. 2

[25] Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*, pages 1786–1794, 2010. 1, 2

[26] Hariharan Narayanan and Partha Niyogi. On the sample complexity of learning smooth cuts on a manifold. In *COLT*, 2009. 2

[27] Karl Pettis, Thomas A. Bailey, Anil K. Jain, and Richard C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:25–37, 1979. 2

[28] Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021. 1, 2

[29] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–52, 2021. 1

[30] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. 4

[31] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000. 1

[32] Jonathan Von Brünken, Michael E Houle, and Arthur Zimek. Intrinsic dimensional outlier detection in high-dimensional data. *NII Technical Reports*, 2015(3):1–12, 2015. 2

[33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 3

[34] Yun Yang and David B Dunson. Bayesian manifold regression. *The Annals of Statistics*, 44(2):876–905, 2016. 2

[35] CL Zachary. The mythos of model interpretability. *Communications of the ACM*, pages 1–6, 2016. 1

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 2