Extendable and invertible manifold learning with geometry regularized autoencoders

Authors: Andres F. Duque, Sacha Morin, Guy Wolf, Kevin Moon.
Affiliation: Utah State University, Univ. de Montreal.

Student Name: Zhiyuan Ren
Student e-mail: zhiyuanren@ucsb.edu

Out[1]: Click here to toggle on/off the raw code.

1. Introduction and motivation

The high dimensionality of modern data introduces significant challenges in descriptive and exploratory data analysis.These challenges gave rise to extensive work on dimensionality reduction aiming to provide low dimensional representations that preserve or uncover intrinsic patterns and structures in processed data. A common assumption in such work is that high dimensional measurements are a result of (often nonlinear) functions applied to a small set of latent variables that control the observed phenomena of interest, and thus one can expect an appropriate embedding in low dimensions to recover a faithful latent data representation. While classic approaches, such as principal component analysis (PCA) [1] and classical multidimensional scaling (MDS) [2], construct linear embeddings, more recent attempts mostly focus on nonlinear dimensionality reduction. These approaches include manifold learning kernel methods and deep learning autoencoder methods, each with their own benefits and deficiencies.

Motivated by the complementary advantages provided by AE and kernel methods, we introduce Geometry regularized autoencoders, a general framework which splices the well-established machinery from kernel methods to recover a sensible geometry with the parametric structure of AEs. Thus we gain the benefits of both methods, furnishing kernel methods with efficient OOSE and inverse mapping, and providing the autoencoder with a geometrically driven representation.

2. Background and related work

- **Manifold Learning:** Manifold learning methods for dimensionality reduction typically assume data lie on a low dimensional manifold $\mathcal{M}$ immersed in the high dimensional ambient space. Therefore they aim to map points from $\mathcal{M}$ to a low dimensional Euclidean space that encodes or reveals its intrinsic geometry. However, in practice, such methods only consider a finite set of data points $x_1, \dots, x_n \in \mathbb{R}^{\mathcal{D}}$ (for $\mathcal{D}$ dimensional ambient space), assumed to be sampled from $\mathcal{M}$, and optimize a fixed set of low dimensional points $y_1, \dots, y_n \in \mathbb{R}^d$ (for $d \ll \mathcal{D}$) such that the Euclidean relations between pairs $\left(y_i, y_j\right)$ will reflect intrinsic nonlinear relations between the corresponding $\left(x_i, x_j\right)$. Recent manifold learning kernel methods typically follow the framework introduced in [3] and further extended by t-SNE [4], which are themselves generalization of the metric MDS algorithm, whereby the coordinates in the latent space are optimized by gradient descent to recreate the pairwise similarities (as defined by a kernel) in the input space. Intuitively, the use of a kernel which outputs high similarities for close neighbors enables the capture of the curvature of the underlying manifold in the ambient space. t-SNE, for instance, uses normalized Gaussian similarities in the input space and t-distributed similarities in the latent space. The embedding is optimized so as to minimize the Kullback-Leibler divergence between both distributions.

- **Autoencoders:** PCA naturally provides an extendable and (approximately) invertible embedding function, but falls short in capturing non-linear mappings. To see this, recall that the embedding function is constructed using a matrix $\boldsymbol{M}$ with orthogonal columns consisting of principal components (PCs) such that $\boldsymbol{M}\boldsymbol{M}^T$ is a projection operator, thus acting as an identity on a hyperplane spanned by the PCs. Then, the embedding is given by the linear function $\hat{x} = \boldsymbol{M}^T x$ and its inverse (on the embedded space) is given by $\boldsymbol{M}\hat{x} = \boldsymbol{M}\boldsymbol{M}^T x \approx x$, where the approximation quality (i.e. reconstruction error) serves as the optimization target for computing the PCs in $\boldsymbol{M}$. To generalize this construction to nonlinear embedding functions over a data manifold $\mathcal{M}$, autoencoders (AEs) replace $\boldsymbol{M}$ by an encoder function $f : \mathcal{M} \to \mathbb{R}^d$ and $\boldsymbol{M}^T$ by a decoder function $f^\dagger : \mathbb{R}^d \to \mathcal{M}$, which is an approximate inverse of f. Both functions are parametrized by a neural network and trained via a reconstruction loss to ensure the composite function $f^\dagger \circ f$ acts as an identity on data sampled from $\mathcal{M}$. By considering datasets in matrix notation (i.e., with rows as datapoints), the AE optimization is generally formulated as

$$\underset{f, f^\dagger}{\arg\min} \mathcal{L}\left(f, f^\dagger\right) = \mathcal{L}_r\left(X, f^\dagger(f(X))\right) \dots\dots (1)$$

where $f, f^\dagger$ are understood to be applied separately to each row in their input matrix (yielding corresponding output data points organized in matrix form), and $\mathcal{L}_r$ denotes a loss function that measures the discrepancy between the original and reconstructed data points (commonly MSE). It is common to select $d < \mathcal{D}$, forcing the autoencoder to find a representation in latent codes of dimension d while retaining as much information for reconstruction as possible. In this case the autoencoder is undercomplete. Under this formulation, instead of learning new coordinates for the input data, we learn an embedding function f and an inverse function $f^\dagger$. If f is a linear function, the network will project onto the same subspace spanned by the principal components in PCA [5].

Out[2]:

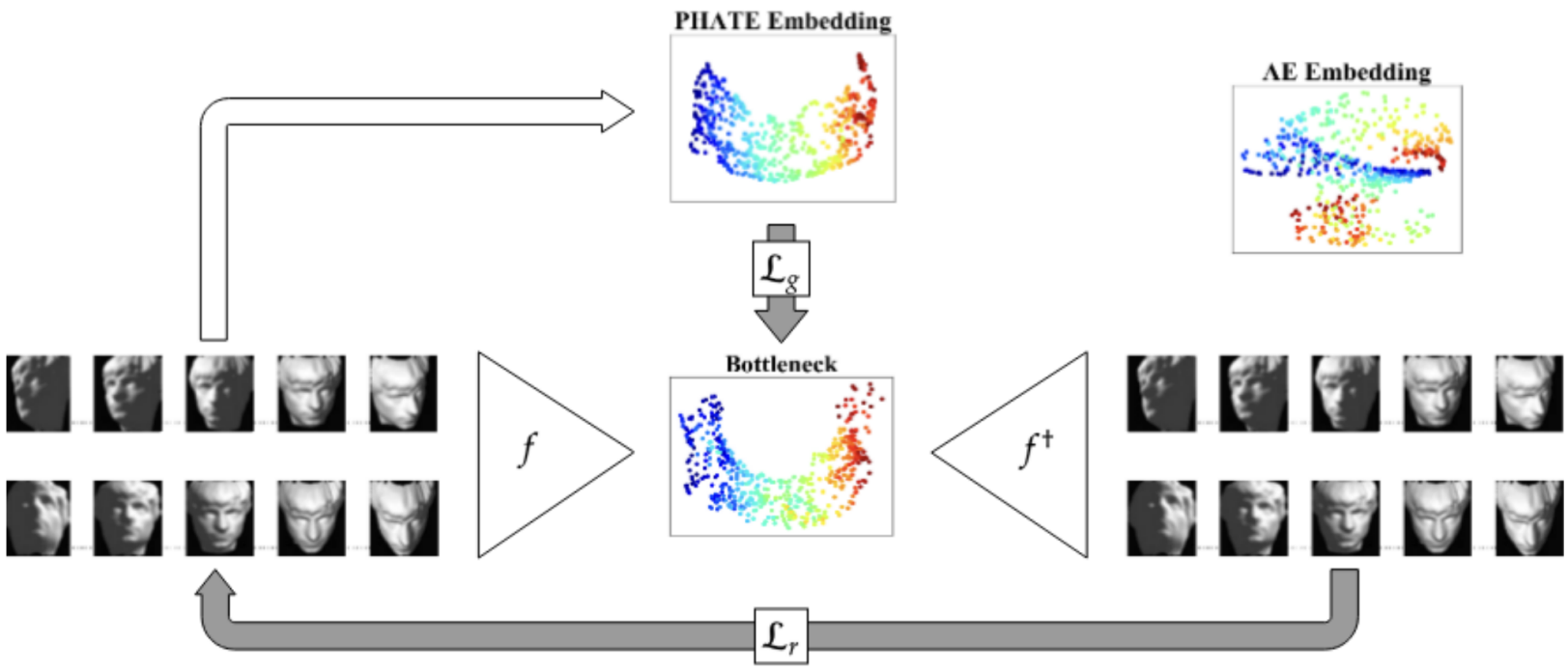


Fig. 1. **Overview of GRAE on the Faces dataset [3]**. Geometric regularization is applied to enforce similarity between GRAE and PHATE embeddings. The vanilla AE embedding (top right) is included for reference.

3. Method

The AE formulation presented (1) departs from manifold learning approaches as it lacks an explicit condition to recover geometrical interactions between observations. To fill that gap, we propose a general framework called Geometry regularized autoencoders which explicitly penalizes misguided representations in the latent space from a geometric perspective. Thus, we add a soft constraint in the bottleneck of the autoencoder as follows:

$$\arg\min_{f,f^\dagger} \mathcal{L}\left(f, f^\dagger\right) = \mathcal{L}_r\left(X, f^\dagger(f(X))\right) + \lambda \mathcal{L}_g(f(X), \mathcal{E}) \ldots\ldots (2)$$

The $\mathcal{L}_g$ term in (2) is the geometric loss, penalizing the discrepancy between the latent representation and the embedding $\mathcal{E}$ previously learned by a manifold learning algorithm. Specifically, given an embedding of training points $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$, we define the geometric loss as $\mathcal{L}_g(f(X), \mathcal{E}) = \sum_{i=1}^{n} \|e_i - f(x_i)\|^2$.

The parameter $\lambda \geq 0$ determines how strongly the latent space of the AE should match the embedding $\mathcal{E}$. Thus the network will implicitly force the latent space of the autoencoder to preserve the relationships learned by the manifold learning technique, resulting in a non-linear embedding function f and its inverse $f^\dagger$ that are consistent with sensible geometric properties. This regularization can be applied with any manifold learning approach, whether it be Isomap, t-SNE, etc. The resulting latent space will then inherit the corresponding strengths and weaknesses of the selected approach.

To generate $\mathcal{E}$ in this work, we use PHATE [6] as it has proven to preserve long-range relationships (global structure) in a low-dimensional representation beyond the capabilities of spectral methods such as Laplacian eigenmaps, Diffusion Maps, LLE, and Isomap, especially when the dimension d is required to be 2 or 3 for visualization. PHATE is built upon diffusion geometry [47]. PHATE first computes an α-decay kernel with an adaptive bandwidth, which captures local geometry while remaining robust to density changes. The kernel matrix is normalized to obtain a probability transition matrix P (diffusion operator) between every pair of points. Various scales of the geometry can then be uncovered by computing a t-step random walk over P, with a higher t implying more diffusion, pushing transition probabilities to a more global scale.

The parameter t is automatically chosen by studying the entropy of a distribution over the eigenvalues of P (known as the von Neumann Entropy) as t increases. Typically, the first few t-steps lead to a sharp drop in the entropy, which is thought in [4] to be the process of denoising the transition probabilities, whereas later steps will reduce entropy at a lower rate, thus, slowly losing meaningful information in the structure. Subsequently PHATE computes the so-called potential distances D'_t, which have proven to be adequate distances between the transition probabilities encoded in P^t. Finally, metric MDS is applied to D'_t to optimally preserve the potential distances in a low-dimensional representation. Fig. 1 shows an overview of GRAE using the PHATE embedding.

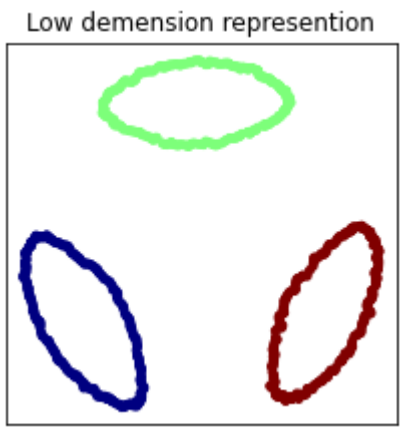
4. Implement

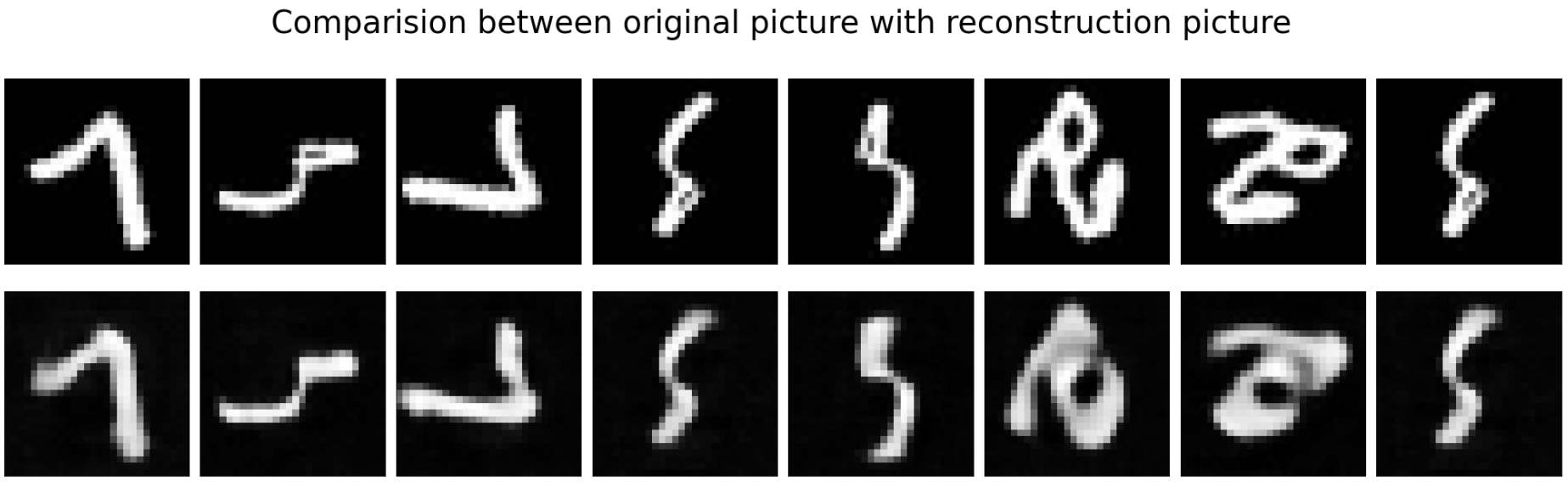
```
packages (from matplotlib>=3.3.4->geomstats) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site
-packages (from matplotlib>=3.3.4->geomstats) (4.33.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site
-packages (from matplotlib>=3.3.4->geomstats) (1.4.2)
Requirement already satisfied: packaging>=20.0 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-p
ackages (from matplotlib>=3.3.4->geomstats) (21.3)
Requirement already satisfied: cycler>=0.10 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-pack
ages (from matplotlib>=3.3.4->geomstats) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-pac
kages (from matplotlib>=3.3.4->geomstats) (9.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/s
ite-packages (from matplotlib>=3.3.4->geomstats) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-pack
ages (from pandas>=1.1.5->geomstats) (2022.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/s
ite-packages (from scikit-learn>=0.22.1->geomstats) (3.1.0)

Requirement already satisfied: six>=1.5 in /Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-packages
(from python-dateutil>=2.7->matplotlib>=3.3.4->geomstats) (1.16.0)


/Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-packages/tqdm/auto.py:22: TqdmWarning: IProgress no
t found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.h
tml (https://ipywidgets.readthedocs.io/en/stable/user_install.html)
  from .autonotebook import tqdm as notebook_tqdm
/Users/zhiyuanren/miniforge3/envs/grae/lib/python3.9/site-packages/numba/cpython/hashing.py:525: UserWarning:
FNV hashing is not implemented in Numba. See PEP 456 https://www.python.org/dev/peps/pep-0456/ (https://www.p
ython.org/dev/peps/pep-0456/) for rationale over not using FNV. Numba will continue to work, but hashes for b
uilt in types will be computed using siphash24. This will permit e.g. dictionaries to continue to behave as e
xpected, however anything relying on the value of the hash opposed to hash as a derived property is likely to
not work as expected.
  warnings.warn(msg)
INFO: Using numpy backend
```

5. Demonstration and Analysis

A - Rotated Digits Testing

```
Fitting GRAE...
    Fitting manifold learning embedding...
    Fitting encoder & decoder...


Fitting GRAE...
    Fitting manifold learning embedding...
    Fitting encoder & decoder...
```



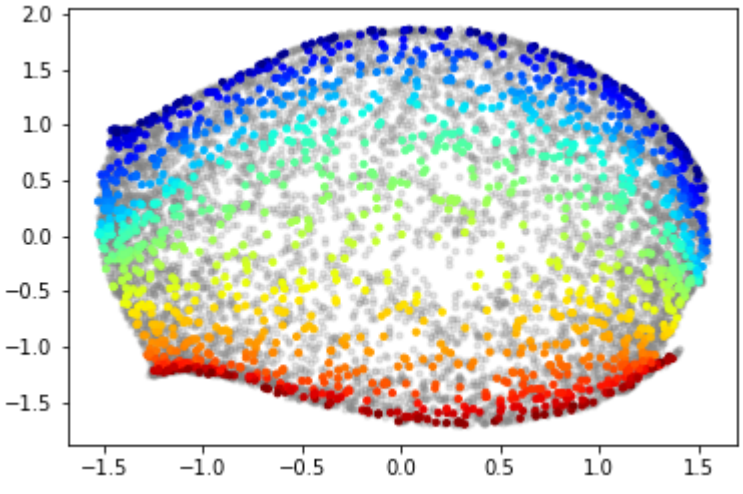


We can see that after geometry regularized autoencoders training, the model's latent represention will be very clearly seperate in 2-d space, which means the model training process works a lot.
And from second figure above, originial pictures in first row are very similar to the pictures generated by decoder.
So in summary, this method present a general approach to leverage traditional dimensionality reduction and manifold learning methods, providing them with a natural OOSE and an invertible mapping.

B - SwissRoll Manifold Testing

```
Fitting GRAE...
    Fitting manifold learning embedding...
    Fitting encoder & decoder...
```

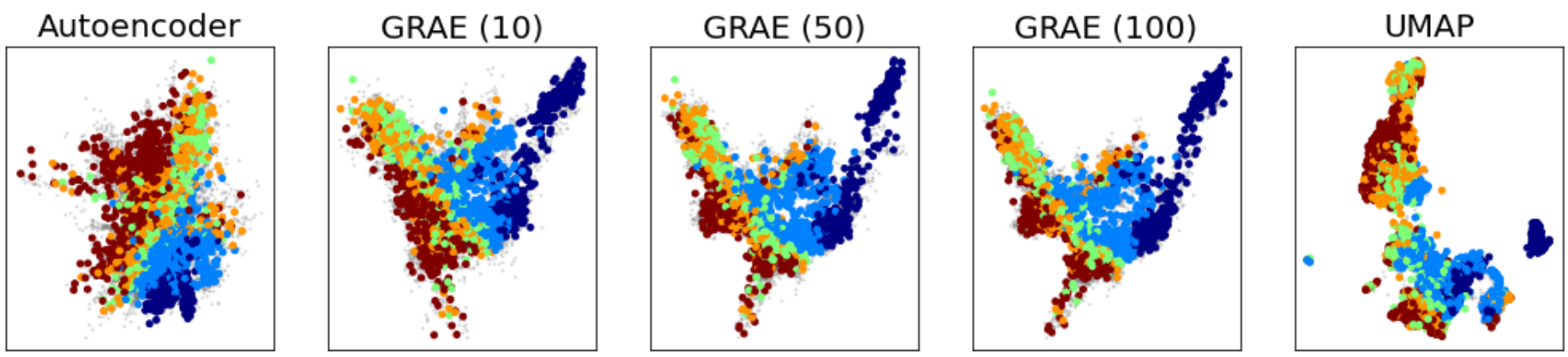
Out[10]: <matplotlib.collections.PathCollection at 0x28fbb8550>



In SwissRoll manifold space, date with different categories are clearly seperated into different represention shape in 2d-space.

C - Comparsion with other methods

Out[11]:



From this fig, we can see that geometry regularized autoencoders have more clear seperation in low dimension space than Autoencoder and UMAP. As training epochs increase, the seperation of representation will be more clear.

6. Conclusion

Geometry regularized autoencoder, a general parametric framework to enhance autoencoders' latent representation by taking advantage of well-established manifold learning methods. By imposing a geometrical soft constraint on the bottleneck of the autoencoder, we demonstrated empirically how GRAE can achieve good visualizations and good latent representations on several performance metrics compared to AE and other methods motivated by geometry. Furthermore, GRAE is equipped with an inverse mapping that often produces a better reconstruction than AE. While the primary focus of this work is on using PHATE embeddings to regularize the bottleneck, future work will focus on the study of other manifold learning algorithms as constraints for learning AE representations with better geometry and the benefits they bring in terms of visualizations, reconstruction, and data generation.

Reference

[1] K. 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, vol. 2, no. 11, pp. 559–572, 1901.
[2] M. A. Cox and T. F. Cox, "Multidimensional scaling," in Handbook of data visualization. Springer, 2008, pp. 315–347.
[3] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," NeurIPS, vol. 15, no. Nov, p. 833–840, 2008.
[4] L. v. d. Maaten and G. E. Hinton, "Visualizing data using t-SNE," Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.
[5] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," Neural networks, vol. 2, no. 1, pp. 53–58, 1989.
[6] K. R. Moon et al., "Visualizing structure and transitions in highdimensional biological data," Nature Biotechnology, vol. 37, no. 12, pp. 1482–1492, 2019.
[7] B. Nadler, S. Lafon, I. Kevrekidis, and R. R. Coifman, "Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators," in NeurIPS, 2006, pp. 955–962.