Geometry Regularized Autoencoders

Andres F. Duque*, Sacha Morin*, Guy Wolf**, Kevin R. Moon**, *Member, IEEE*

Abstract—A fundamental task in data exploration is to extract low dimensional representations that capture intrinsic geometry in data, especially for faithfully visualizing data in two or three dimensions. Common approaches use kernel methods for manifold learning. However, these methods typically only provide an embedding of the input data and cannot extend naturally to new data points. Autoencoders have also become popular for representation learning. While they naturally compute feature extractors that are extendable to new data and invertible (i.e., reconstructing original features from latent representation), they often fail at representing the intrinsic data geometry compared to kernel-based manifold learning. We present a new method for integrating both approaches by incorporating a geometric regularization term in the bottleneck of the autoencoder. This regularization encourages the learned latent representation to follow the intrinsic data geometry, similar to manifold learning algorithms, while still enabling faithful extension to new data and preserving invertibility. We compare our approach to autoencoder models for manifold learning to provide qualitative and quantitative evidence of our advantages in preserving intrinsic structure, out of sample extension, and reconstruction. Our method is easily implemented for big-data applications, whereas other methods are limited in this regard.

Index Terms—Autoencoders, dimensionality reduction, manifold learning, semi-supervised learning

1 Introduction

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. 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.

Kernel methods for manifold learning include some of the most popular nonlinear dimensionality reduction methods, dating back to the introduction of Isomap [3] and Locally Linear Embedding (LLE) [4]. These two methods proposed the notion of data manifolds as a model for intrinsic low-dimensional geometry in high dimensional data. The manifold construction in both cases is approximated by a local neighborhood graph, which is then leveraged to form a low-dimensional representation that preserves either pairwise geodesic distances (in the case of Isomap) or local linearity of neighborhoods (in LLE). The construction of neighborhood graphs to approximate manifold structures was further advanced by Laplacian eigenmaps [5] and diffusion maps [6], together with a theoretical framework for relating the captured geometry to Riemannian manifolds via the Laplace-Beltrami operators and heat kernels. These approaches that, until recently, dominated manifold learning can collectively be considered as spectral methods, since the embedding provided by them is based on the spectral decomposition of a suitable kernel matrix that encodes (potentially multiscale) neighborhood structure from the data. They are also known as kernel PCA methods, as they conceptually extend the spectral decomposition of covariance matrices used in PCA, or that of a Gram (inner product) matrix used in classic MDS.

Recent work in dimensionality reduction has focused on visualization for data exploration [7], [8], [9], [10], [11], [12], [13]. Spectral methods are generally unsuitable for such tasks because, while their learned representation has lower dimension than the original data, they tend to embed data geometry in more dimensions than can be conveniently visualized (i.e., $\gg$ 2 or 3). This is typically due to the orthogonality constraint and linearity of spectral decompositions with respect to the initial dimensionality expansion of kernel constructions [7], [14]. This led to the invention of methods like t-SNE (t-Distributed Stochastic Neighbor Embedding) [15], UMAP (Uniform Manifold Approximation and Projection) [8], and PHATE (Potential of Heat-diffusion for Affinity-based Transition Embedding) [7]. These methods embed the data by preserving pairwise relationships between points and can be viewed as generalizations of metric and non-metric MDS. These methods and their extensions have been used in applications such as single cell genomics [7], [10], [16], [17], [18], [19], [20], [21], visualizing time series [9], visualizing music for a recommendation system [22], and analyzing the internal representations in neu-

- *Andres F. Duque and Kevin R. Moon are with the department of Mathematics & Statistics, Utah State University, Logan, UT 84322 US.*
- *Sacha Morin and Guy Wolf are with the departments of Computer Science & Operations Research and Mathematics & Statistics (correspondingly), Université de Montréal, Montréal, Quebec, H3T 1J4, Canada, and with Mila - Quebec AI Institute, Montreal, Quebec, H2S 3H1, Canada .*
- *(*) The first two authors contributed equally. (**) The last two authors jointly supervised the work. Corresponding author: Kevin R. Moon (kevin.moon@usu.edu). Code and data can be found at https://github.com/KevinMoonLab/GRAE.*

ral networks [23], [24]. However, these and spectral methods typically provide fixed latent coordinates for the input data. Thus, they do not provide a natural embedding function to perform out-of-sample extension (OOSE). This shortcoming is usually tackled by employing geometric harmonics [25], Nyström extension [26], or a landmark approach [27].

In contrast, Autoencoders (AEs) are a different paradigm for non-linear dimensionality reduction. First introduced in [28] and extended in many ways such as variational autoencoders (VAE) [29], this non-convex and parametric approach has gained more attention in recent years, especially due to the computational and mathematical advances in the field enabling neural networks to be efficiently implemented and trained. In contrast with kernel methods, AEs learn a parametric function and are thus equipped with a natural way to perform OOSE, as well as an inverse mapping from the latent to the input space. Despite these properties, AEs usually fail to accurately recover the geometry present in the data. This not only limits their utility in exploratory data analysis, e.g. via low-dimensional visualization, but can also lead to poor reconstructions over certain regions of the data, as we show in this work.

Motivated by the complementary advantages provided by AE and kernel methods, we introduce Geometry regularized autoencoders (GRAE), 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. To achieve this, GRAE introduces a regularization on the latent representation of the autoencoder, guiding it towards a representation previously learned by a kernel method. In this paper we focus our presentation using PHATE [7] as our preferred method for finding a sensible geometry. Nevertheless, we also show how GRAE performs when using UMAP [8] for computing the reference embedding.

Our main contributions are as follows. First, we present the geometric regularization, a general approach to leverage the geometry-preserving properties of kernel-based dimensionality reduction and manifold learning methods while providing them with a natural OOSE and an invertible mapping. Secondly, we show that including the geometric regularization leads to qualitatively improved visualizations relative to competing methods and decreases the reconstruction error of the autoencoder in many cases, suggesting that the geometry learned from the kernel method leads to a better representation for reconstruction. Further, we propose an approach to implement GRAE in a scalable fashion using mini-batch embeddings and combining them in a sensible way. This alleviates the computational cost, which is a typical limitation for kernel methods. Additionally, we demonstrate that the better preservation of the geometric properties of the data induced by GRAE allows for a consistent generation of new data by following geodesic trajectories discovered in the latent space. Finally , we show how our geometric regularization can be used to perform semi-supervised learning using a multi-task learning approach, acting as an inductive bias.

The outline of this paper is as follows. Section 2 summarizes the most relevant work related to ours. Section 3 describes GRAE. In Section 4, experimental comparisons with other methods are provided. Some applications of GRAE are also given in Section 4, in particular, semi-supervised learning as well as data generation from the latent space. Section 5 provides a discussion some of the limitations of the approach while Section 6 concludes the paper.

2 Related Work

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 $d \ll D$) such that the Euclidean relations between pairs (y_i, y_j) will reflect intrinsic nonlinear relations between the corresponding (x_i, x_j). Recent manifold learning kernel methods typically follow the framework introduced in [30] and further extended by t-SNE [15], which are themselves generalizations 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.

UMAP [8] was introduced as an improvement of t-SNE, with claims of improved preservation of global features and better run times. Specifically, the cost function of t-SNE is replaced with cross-entropy and similarities between objects in the input space are computed based on the smooth nearest neighbor distances, that is:

$$v_{j|i} = e^{\frac{-d(x_i, x_j) - p_i}{\sigma_i}}, \tag{1}$$

where p_i is the distance between x_i and its nearest neighbor, σ_i is the bandwidth, and d is a distance, not necessarily Euclidean. In contrast with t-SNE, UMAP does not normalize similarities and relies on an approximation of the neighborhoods using the Nearest-Neighbor-Descent algorithm of [31]. UMAP further distinguishes itself from t-SNE by not restricting the embedded space to two or three dimensions.

Recently, the claim that UMAP is superior to t-SNE in preserving global structure has been challenged in [32], in which the authors attribute the better global structure commonly obtained by UMAP to the differences between both methods in the initialization procedure. Typically t-SNE uses a random initialization, whereas UMAP uses Laplacian eigenmaps as its starting point. They showed that nearly identical results can be obtained by also initializing t-SNE with Laplacian eigenmaps. At any rate, as kernel methods, neither t-SNE nor UMAP provide a natural OOSE.

PCA naturally provides an extendable and (approximately) invertible embedding function of a given dimension by finding the optimal linear transformation in terms of the reconstruction loss. To generalize this approach to nonlinear embedding functions over a data manifold $\mathcal{M}$, autoencoders (AEs) define an encoder function $f : \mathcal{M} \to \mathbb{R}^d$ and 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

$$\arg\min_{f, f^\dagger} \mathcal{L}(f, f^\dagger) = \mathcal{L}_r(X, f^\dagger(f(X))), \tag{2}$$

where $f, f^\dagger$ are 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) [28]. 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 [33].

Manifold learning algorithms are typically based on the eigendecomposition of a kernel matrix (Diffusion Maps) or a stochastic optimization of the latent coordinates (metric MDS, UMAP). Therefore, in constrast to neural networks, they do not provide a general embedding function that operates on, or provides a representation of, the entire manifold $\mathcal{M}$. Thus these methods are not applicable to arbitrary input points in $\mathbb{R}^{\mathcal{D}}$, which we would ideally want to project onto the learned manifold. To address this shortcoming, a parametric version of t-SNE using a neural network approach was proposed in [34] where a multi-step training procedure is used to optimize the t-SNE objective with a neural network.

Another well-known solution to the lack of OOSE is the Nyström extension [35] and its improvements, such as geometric harmonics [25], which approximate an empirical function over new points using linear combinations of the eigenvectors of a kernel matrix computed on the training set. Let $X = \{x_1, ..., x_n\}$ be the training set used to compute the initial kernel matrix K with kernel function $k(\cdot, \cdot)$. Then a new point x' can be extended to the learned latent space using the eigenvectors ψ_i of K with eigenvalues λ_i as follows: $\hat{\psi}_i(x') \approx \frac{1}{\lambda_i} \sum_{j=1}^{n} k(x_j, x')\psi_j(x_j)$.

One can thus project a function on the eigenvectors of K and then extend to new points using the new approximated eigenvectors. However, this approach has several drawbacks [36]. Given n training points (resulting in K being $n \times n$), extending a function (e.g. a dimensionality reduction function such as PHATE or UMAP) to m new points requires us to compute m new kernel rows leading to a time complexity of $\mathcal{O}(nm)$. Furthermore, the empirical function must be within the interpolation range of a given kernel, which may require hyperparameter tuning or trying different kernel functions.

Other methods perform OOSE by a linear combination of training points (the "landmarks") close to the new points in the input space [27], as in PHATE [7] and landmark MDS [37]. UMAP takes a similar approach to OOSE by initializing latent coordinates of new points in the latent space based on their affinities with training points in the input space. The new layout is then optimized by gradient descent using the embeddings of training points as reference.

All of these approaches require the training points, or a subset, to be stored in memory with their embeddings as a proxy for the target function, which can quickly become inconvenient for a large dataset or lead to a loss in embedding quality due to subsampling or the use of landmarks. Moreover, they do not provide a straightforward approximation of the inverse function, which is critical in assessing how well the information is preserved in the embedded space. As such, they are not directly comparable to GRAE and other AE based models, which present a native approximation of the inverse and only need to store the weights and biases of the network to perform OOSE. Thus their memory requirements are independent of the training set size.

The vanilla AE formulation in (2) has been extended for many purposes by adding regularization as a prior on the function space of f and $f^\dagger$. Denoising AEs (DAE) [38] are widely used to find good latent representations and perform feature extraction, exploiting the flexibility provided by neural networks (e.g. [39], [40]). Contractive autoencoders (CAE) [41] penalize the Frobenius norm of the Jacobian of the encoder f, encouraging a robust representation to small perturbations in the training data. When using a high dimensional latent space (e.g., the *overcomplete* case), sparse AEs [42] are particularly useful, introducing a sparsity constraint that forces the network to learn significant features in the data. Extensions to produce generative models, such as variational autoencoders (VAE) [29], regularize the latent representation to match a tractable probability distribution (e.g., isotropic multivariate Gaussian), from which it is possible to sample over a continuous domain to generate new points. β-VAE [43] extends the VAE framework by adding a hyperparameter β to modulate the prior regularization.

Some attempts to impose geometrically driven regularizations on the latent space have been proposed over the past two decades, which are more closely related to our work. Stacked Similarity-Aware Autoencoders, for instance, enforce a cluster prior based on pseudo-class centroids to increase subsequent classification performance [44].

A relatively new implementation called Diffusion Nets [36] encourages the AE embedding to learn the geometry from Diffusion Maps (DM) [6], a manifold learning algorithm. This approach combines an MSE loss in the embedding coordinates with the so-called eigenvector constraint to learn the diffusion geometry. Diffusion Nets inherits some of the inherent issues of Diffusion Maps. Perhaps most importantly, they inherit its inability as a spectral method to ensure significant representation of the data on a fixed lower dimension, due to the natural orthogonality imposed among the diffusion coordinates [7]. Therefore,

effective use of Diffusion Nets may require the network architecture to be determined from the numerical rank of the diffusion affinity kernel used in DM. This, in turn, would limit the capabilities of this approach in data exploration (e.g., visualization), while in contrast PHATE (and UMAP) can specifically optimize a chosen dimension (e.g., 2D or 3D). Moreover, as a spectral method, DM itself[1] tends to be more computationally expensive than PHATE (which we use in this work, see Sec. 3) or UMAP [7], [8].

The formulation of Diffusion Nets is closely related to Laplacian autoencoders (LAE) [45] and Embedding with Autoencoder Regularization (EAER) [46]. Both of these methods include a regularization term that penalizes inaccurate preservation of neighborhood relationships:

$$\begin{aligned}\arg\min_{f,f^\dagger} \mathcal{L}(f,f^\dagger) = \mathcal{L}_r(X, f^\dagger(f(X))) \\ + \lambda \sum_{i<j}^{n} \mathcal{L}'(f(x_i), f(x_j), \phi_{ij}),\end{aligned} \tag{3}$$

where $\mathcal{L}_r$ is the AE reconstruction loss presented in (2) and $\mathcal{L}'$ is a specific regularization applied to the encoder. The values ϕ_{ij} are given by some pairwise distance or similarity measure. For instance, in EAER the regularization term $\mathcal{L}'$ is the classical MDS objective:

$$\mathcal{L}'(f(x_i), f(x_j), \phi_{ij}) = (\|f(x_i) - f(x_j)\| - \phi_{ij})^2, \tag{4}$$

where ϕ_{ij} is a given distance computed in the input space. It can also be margin-based, where embedding distances between neighbors are penalized if not 0, while distances between non-neighbors are penalized if not above a certain margin. Finally, $\mathcal{L}'$ can take the form $\sum_{i<j}^{n} \|f(x_i) - f(x_j)\|^2 \phi_{ij}$. In this case, the ϕ_{ij} values are computed using the weighted edges in an adjacency graph. This gives the objective function of Laplacian eigenmaps. LAE employs a similar loss term, but goes further by adding a second-order term involving the Hessian of f.

Another approach derived from manifold learning methods can be found in [47], in which the authors aimed to replicate Isomap's objective function by using a Siamese network architecture trained over the pairwise geodesic distances. Their method, however, scales quadratically in the number of landmarks selected for training.

Recently, topological autoencoders (TAE) [48] were proposed which include a regularization based on a topological signature of the input data. The topology of a manifold can be characterized by *homology groups*, which, depending on their dimension, represent various topological features such as the number of disconnected components or the number of cycles on the manifold that cannot be deformed into each another. When data are sampled from a manifold, such topological features can be approximately derived from an ε-ball neighborhood graph of the data. Persistent homology [49], [50] was introduced as a means to identify the topological signature of manifolds based on how long topological features persist when progressively increasing the ε-ball of the neighborhood graph. Topological features with a short ε lifespan are attributed to noise. TAE thus penalizes discrepancies between the topological signatures of the input space and the latent space.

1. We note that the DM runtime (relative to UMAP) is equivalent to that of Laplacian eigenmaps reported in [8], as the algorithmic difference between these spectral methods is negligible [6].

3 Geometry-Regularized Autoencoder

3.1 Learning embedding functions

In this work, we aim to learn a data manifold geometry to find an appropriate embedding function $f : \mathcal{M} \to \mathbb{R}^d$, rather than just a fixed point-cloud embedding. This contrast can be seen, for example, by considering the classic PCA and MDS methods. While both of these methods can be shown analytically to extract equivalent (or even the same) linear embeddings from data, MDS only assigns coordinates to fixed input points (similar to the described manifold learning methods), while PCA provides an embedding function (albeit linear) defined by a projection operator on principal components. Here, we aim to establish a similar equivalence in nonlinear settings by providing an alternative to popular manifold learning approaches that constructs an embedding function (as a nonlinear successor of PCA) and also yields representations that capture an intrinsic geometry similar to that of established kernel methods (seen as successors of MDS).

3.2 Extendable and invertible embedding with autoencoders

The AE formulation presented in (2) departs from manifold learning approaches as it lacks an explicit condition to recover geometric interactions between observations. To fill that gap, we propose a general framework called GRAE (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:

$$\arg\min_{f,f^\dagger} \mathcal{L}(f,f^\dagger) = \mathcal{L}_r(X, f^\dagger(f(X))) + \lambda \mathcal{L}_g(f(X), \mathcal{E}). \tag{5}$$

The $\mathcal{L}_g$ term in (5) 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 for $\lambda > 0$, the network will implicitly force the latent space of the autoencoder to preserve the relationships learned by the manifold learning technique, resulting in a nonlinear embedding function f and its inverse $f^\dagger$ that are consistent with sensible geometric properties. Assuming a neural network architecture of sufficient capacity, a high λ will ensure the latent space almost perfectly matches $\mathcal{E}$. Conversely, a low λ will yield an embedding indistinguishable from a standard AE embedding. In general, λ can be tuned by a visual assessment of the bottleneck or, as we do in this work, by cross-validation of the reconstruction term $\mathcal{L}_r$. We observed empirically that the latter approach will select a value of λ that improves both the geometry of the latent space and the reconstruction quality of the decoder in comparison to standard AE training.

3.3 Computing the geometric reference $\mathcal{E}$

Geometric regularization can rely on any manifold learning approach to compute $\mathcal{E}$, e.g. UMAP, 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 suggest using PHATE [7] 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 [6], [51]. PHATE first computes an $\alpha-$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.

Subsequently PHATE computes the potential distances D'_t, which have proven to be informative 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. Figure 1 shows an overview of GRAE using the PHATE embedding.

As with any kernel-based method, PHATE's computational complexity will be dominated by the need to compute $\mathcal{O}(n^2)$ pairwise distances for n observations. In practice, the implementation will approximate the full algorithm using m landmarks with $m << n$ [7]. Additionally, only the nearest neighbors are actually connected in the pairwise distance matrix, leading to a sparse structure than can be leveraged to save memory and computations.

3.4 Embedding-based optimization

Most of the regularized AE methods discussed in Section 2 rely on jointly minimizing the reconstruction loss and the regularizing loss during training. We instead choose to precompute $\mathcal{E}$ for three reasons. First, typical optimization objectives based on manifold learning, such as those used by Diffusion Nets, EAER, and parametric t-SNE, require a pairwise affinity or distance matrix which needs to be accessible during training and entails a memory cost of scale $\mathcal{O}(n^2)$ for n observations. While GRAE may require a pairwise matrix to precompute $\mathcal{E}$, only $\mathcal{E}$ needs to be retained for training the AE architecture, with a memory requirement of $\mathcal{O}(nd)$, where d is the latent space dimension and typically $d \ll n$. Other methods, such as TAE, do not require a pairwise distance matrix from the whole dataset, but instead compute distance matrices within mini-batches. Given a mini-batch B of size p, this approach leads to $\mathcal{O}(p^2)$ additional operations for every gradient step beyond typical AE training. The GRAE loss, for its part, only requires computing the Euclidean distance between $f(x_i)$ and its target $e_i \in \mathcal{E}$, which is $\mathcal{O}(p)$. Thus, our optimization approach is better for big data applications given the computational complexity of competing methods. See Appendix B.5 for a runtime comparison.

Second, the optimization techniques employed for some dimensionality reduction methods can be superior in performance than common approaches for neural network optimization. For instance, PHATE uses the SMACOF algorithm to perform MDS over the potential distances matrix. In practice, we find that this obtains a better optimum than stochastic gradient descent or its variants.

Finally, we depart from previous methods by providing a more general approach that is applicable to many manifold learning techniques. In many applications, there may not be strong reasons for imposing a particular relationship in the geometric loss that resembles a loss function from a specific kernel method. Any approach employed to find $\mathcal{E}$, whether it be PHATE, Isomap, t-SNE, LLE, etc., is already performing an optimization to its particular loss function, imposing the preservation of its desired geometric relationships in the data. Thus, GRAE implicitly enforces such a relationship.

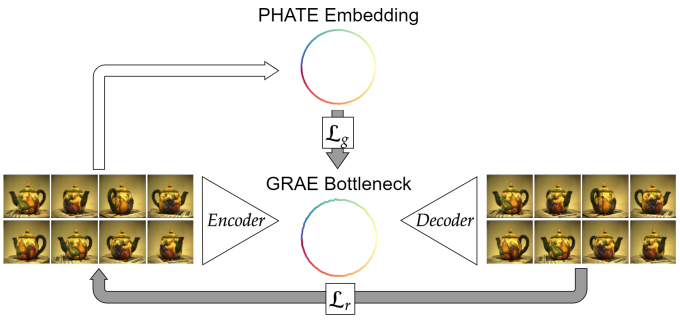

Fig. 1. **Overview of GRAE on the Teapot dataset [52]**. The geometric regularization is applied to the bottleneck to enforce similarity between the GRAE and PHATE embeddings. The resulting embedding captures the rotational geometry of the data whereas the vanilla autoencoder fails (see Figure 3).

3.5 Scalable GRAE

Kernel methods typically have computational limitations. Even when using speed-up variants, such as landmarks or the Barnes-Hut algorithm for t-SNE, their application to large data sets is not as scalable as an AE. Motivated by such limitations, we show how the introduction of the autoencoder structure in GRAE also allows us to create a visualization of a very large dataset (see Figure 2).

We first partition the data in mini-batches $\mathcal{B} = \{B_1, B_2, \ldots, B_N\}$ each containing common observations $\boldsymbol{X_c}$ and unique observations $\boldsymbol{X_i}$. Thus, $B_i = \{\boldsymbol{X_c} \cup \boldsymbol{X_i}\}$. Then, we apply the manifold learning method (in this case, PHATE) to each mini-batch, which produces an embedding $\mathcal{E}_i = \{\mathcal{E}_{\boldsymbol{X_c}} \cup \mathcal{E}_{\boldsymbol{X_i}}\}$ containing embedding coordinates $\mathcal{E}_{\boldsymbol{X_i}}$ for the unique observation and coordinates $\mathcal{E}_{\boldsymbol{X_c}}$ for the common observations (Figure 2**A**). Since each embedding might vary in orientation, scale, and reflection with respect to the others, we apply the Procrustes method [53] among the common points to extract a linear transformation, which is then applied to the whole mini-batch embedding. This allows us to consistently combine all embeddings. This final embedding is not as refined as computing PHATE for the whole dataset, and some local information is lost (Figure 2**B**). Fortunately, GRAE is able to refine the discrepancies (Figure 2**C**), generating near similar embeddings whether $\mathcal{E}$ is generated by computing PHATE over the whole data

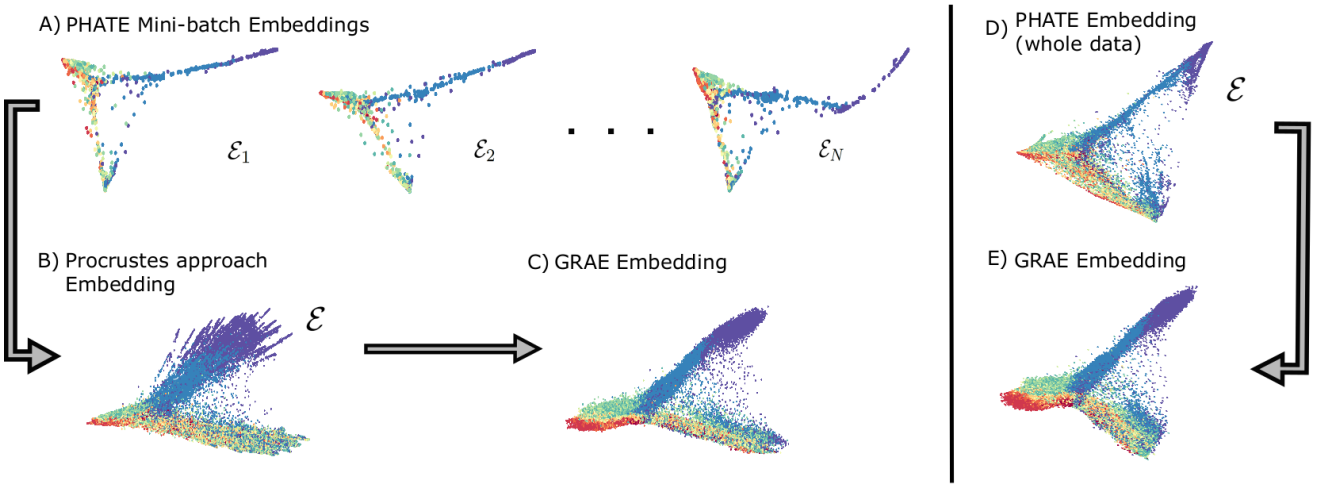

Fig. 2. **Scalable GRAE.** Overview of GRAE applied to 200,000 observations of iPSC data (see Section 4.1). Data points are colored by observation time. **A)** Mini-batch PHATE embeddings, each of which share some common observations. **B)** Combined embedding using the Procrustes method to align the mini-batch embeddings. **C)** GRAE embedding using (**B**) as $\mathcal{E}$ in the geometric loss. **D)** PHATE embedding computed over the whole data set. **E)** GRAE embedding using (**D**) as $\mathcal{E}$ in the geometric regularization. Although both approaches produce near identical embeddings, the mini-batch approach takes around 850 seconds to compute and scales linearly. In contrast, PHATE applied to the whole data takes around 3894 seconds and scales quadratically.

(Figure 2**E**), or by the Procrustes transformations over mini-batches. This makes the computational complexity linear with respect to the number of mini-batches.

4 Experiments

In this section, we experimentally compare GRAE with a standard AE, Diffusion Nets [36], TAE [48], EAER-Margin [46], DAE [38], CAE [41], and β-VAE [43] on 9 different datasets using a two-dimensional latent space. The motivation behind such a low-dimensional latent space is two-fold: to visualize the geometry of the latent space, and to offer a challenging dimensionality reduction task. We provide results for higher dimensional bottlenecks on the image datasets in Appendix B.4. Training is unsupervised and the ground truth is only used for visualizing and scoring embeddings. To compare different ways of computing $\mathcal{E}$, we benchmark GRAE with both PHATE and UMAP.

4.1 Experimental setup

4.1.1 Compared methods

We compared with a standard AE to measure improvements in latent space geometry and reconstruction quality. We selected additional regularized autoencoders (EAER-Margin[2], TAE, and Diffusion Nets) as they also include regularizations to force the AE to introduce more structure in the latent space via a prior driven by geometry or topology. While of limited use for data exploration and visualization, other non-geometric auto-encoders (DAE, CAE, and β-VAE) were added for a complete evaluation of the metrics pertaining to the prediction of latent factors or classes. We also included comparisons with PCA to show the relevance of non-linear dimensionality reduction techniques on our problem set.

2. While EAER also introduces loss terms based on Laplacian Eigenmaps and MDS, the margin-based loss performed best according to their benchmarks.

We did not benchmark parametric t-SNE [34] given the absence of an invertible mapping and the fact that UMAP is similar to t-SNE from an algorithmic standpoint and provides similar embeddings at a lower computational cost.

4.1.2 Datasets

We perform comparisons on 9 datasets, which are illustrated in Figure 3 and described in depth in Appendix A. They include two 3D manifold problems (Swiss Roll, Toroidal Helices), five images datasets (Teapot, Rotated Digits, Object Tracking, UMIST Faces, COIL-100), and two single-cell datasets (induced pluripotent stem cell (iPSC) mass cytometry data [54] and single-cell RNA-sequencing measurements of embryoid bodies (EB) [7]). The latter two datasets consist of genetic markers of cultured cells sampled at various time points. In both cases, cells are known to specialize as time goes on, leading to distinct manifold branches.

We focus our analysis on datasets where we can reasonably expect local Euclidean distances to reflect true pairwise similarities; i.e. the manifold assumption is valid. This is motivated by the importance of the distance matrix in GRAE (to compute $\mathcal{E}$) and in other compared methods driven by geometry or topology. If such an assumption on the Euclidean distance does not hold (e.g. as in the case of high-dimensional heterogeneous images such as CIFAR-10), methods based on manifold learning and topological data analysis are unlikely to yield useful representations unless an alternative means of computing similarities is developed, which is tangential to this work.

4.1.3 Architecture, Training & Tuning

All autoencoder-based models in the experiments use the same network architecture. The encoder and decoder are typical fully-connected networks in the case of the synthetic manifolds and the biological data, with additional convolution layers for the image datasets. Furthermore, we used a random search over the parameter space to tune each

model on each dataset. Details regarding the architecture, the training procedure and the hyperparameter search can be found in Appendix B.

4.1.4 Implementations

Some compared methods were limited computationally while running our benchmarks, especially on larger problems like the iPSC data. For example, the algorithm presented in EAER had to be improved to support mini-batch training. Such an improvement was not as straightforward for Diffusion Nets because of the eigenvector constraint, which requires the full dataset and the affinity matrix to be held in GPU memory for training. In this case, subsampling 30,000 observations from the iPSC problem was necessary to run experiments on the same hardware used to fit other methods. We were unable to obtain a satisfactory embedding with Diffusion Nets on the COIL-100 problem due to memory constraints and thus chose not to report it. In Appendix B.5, we compare the training times of GRAE and the other methods on Swiss Roll, COIL-100 and iPSC.

4.1.5 Evaluation and Metrics

All experiments use a train-validation-test ratio of 70%-15%-15%, except on the Swiss Roll problem, where a thin middle slice of 500 points is removed and set aside for testing to study how various methods would behave when required to generalize to out-of-distribution data. The validation split is used for tuning hyperparameters and early stopping (see Appendix B). To account for the inherent stochasticity of manifold learning algorithms and neural network training, we report the average of each metric over 10 runs using different seeds and different validation splits. All runs are benchmarked on the same test split.

We score models using three measures depending on the dataset: i) reconstruction as measured by the MSE between the reconstructed samples and the original samples, ii) disentanglement of ground truth factors by reporting the R^2 of a linear regression predicting said factors using the latent representation as input[3], and iii) classification performance by reporting the accuracy of a logistic regression on the latent representation. For both R^2 and classification accuracy, we use the train split to fit the linear models and report the metric on the test split (as embedded by the main model).

More specifically, the ground truth factors of interest for the R^2 metric are the manifold coordinate along the test slice (Swiss Roll), the angle on the manifold (Toroidal Helices, Teapot, Rotated Digits) and the x and y coordinates of the character (Object Tracking). Additional information is provided in Appendix C for the computation of the R^2 metric. While the iPSC and EB Differentiation problems do have a known factor of variation (i.e. time), we do not expect methods to recover it in a linear fashion since cells of the same age may express different genes and thus appear at distinct locations in the latent space. We still use time for coloring these embeddings in Figure 3.

4.2 Qualitative results

We qualitatively evaluate GRAE and the other methods by visualizing the embedding layer after training as shown in Figure 3. We first notice that GRAE recovers a sensible geometry for all problems while other methods fail at basic tasks such as uncoiling the Swiss Roll, disentangling the Rotated Digits, showing the coordinate plane on Object Tracking, recovering a circular structure on Teapot, or showing rings on COIL-100 (to reflect rotations of distinct objects). Diffusion Nets outputs decent embeddings, except for Rotated Digits and COIL-100. Interestingly, EAER-Margin recovers good global structure on Object Tracking, but appears to be collapsing neighborhoods and does not display the grid-like texture of the GRAE and Diffusion Nets embeddings, which could be related to the higher MSE witnessed in Table 3. DAE and β-VAE embeddings do not display any appreciable structural benefits over vanilla AE, whereas the CAE embeddings of the Rotated Digits and the Teapot datasets do show smoother curves — a likely result of the Jacobian regularization.

Only GRAE (PHATE) and Diffusion Nets show the two expected branches of the iPSC manifold as well as some branches on the EB Differentiation problem. Other methods fail to output any useful structure that can be leveraged for data analysis (see Section 4.5). While GRAE (UMAP) comes close to forming branches, we can see the built-in uniform assumption of the UMAP algorithm favors spreading out samples as opposed to concentrating them in branches.

TABLE 1
Average performance metrics for all considered methods on 3D manifold problems. Mean squared error ("MSE") benchmarks reconstruction quality. The R^2 metric quantifies how some ground truth factors of variation can be predicted from the 2D embeddings using a linear regression. Acc. stands for the accuracy of a supervised linear classifier trained on the 2D embedding to predict some ground truth class labels. All metrics are averaged over 10 runs on the test data. The MSE of each run can be further visualized in Figure 4.

Dataset	Model	Metrics		
		MSE	***R^2***	***Acc.***
Swiss Roll	GRAE (PHATE)	0.0061 (3)	**0.81** (2)	n/a
	GRAE (UMAP)	**0.0026** (1)	**0.87** (1)	n/a
	AE	0.0202 (7)	0.08 (7)	n/a
	EAER-Margin	0.0188 (6)	0.12 (6)	n/a
	TAE	0.0144 (4)	0.07 (9)	n/a
	Diffusion Nets	**0.0039** (2)	0.68 (3)	n/a
	DAE	0.0288 (9)	0.08 (7)	n/a
	CAE	0.0169 (5)	0.23 (4)	n/a
	β-VAE	0.0208 (8)	0.16 (5)	n/a
	PCA	0.3235 (10)	0.05 (10)	n/a
Toroidal Helices	GRAE (PHATE)	**0.0002** (1)	0.98 (3)	0.98 (3)
	GRAE (UMAP)	**0.0002** (1)	0.75 (10)	**1.00** (1)
	AE	0.0013 (3)	0.82 (8)	0.51 (5)
	EAER-Margin	0.0031 (7)	0.92 (5)	0.52 (4)
	TAE	0.0023 (5)	**1.00** (1)	0.50 (7)
	Diffusion Nets	0.0028 (6)	0.79 (9)	**1.00** (1)
	DAE	0.0165 (9)	0.90 (6)	0.49 (10)
	CAE	0.0040 (8)	0.97 (4)	0.51 (5)
	β-VAE	0.0014 (4)	0.85 (7)	0.50 (7)
	PCA	0.1660 (10)	**1.00** (1)	0.50 (7)

4.3 Quantitative results

We report the quantitative results of our experiments in Tables 1 (3D manifold problems), 2 (biological data), and 3 (image datasets). We see from these results that GRAE outperforms the vanilla AE with respect to the reconstruction MSE on nearly all benchmarks using either PHATE or UMAP. This suggests that the geometric regularization generally guides the AE to a better region in the optimization

3. If more than one factor exists (e.g. Object Tracking), an R^2 score is computed for each factor and the average is shown.

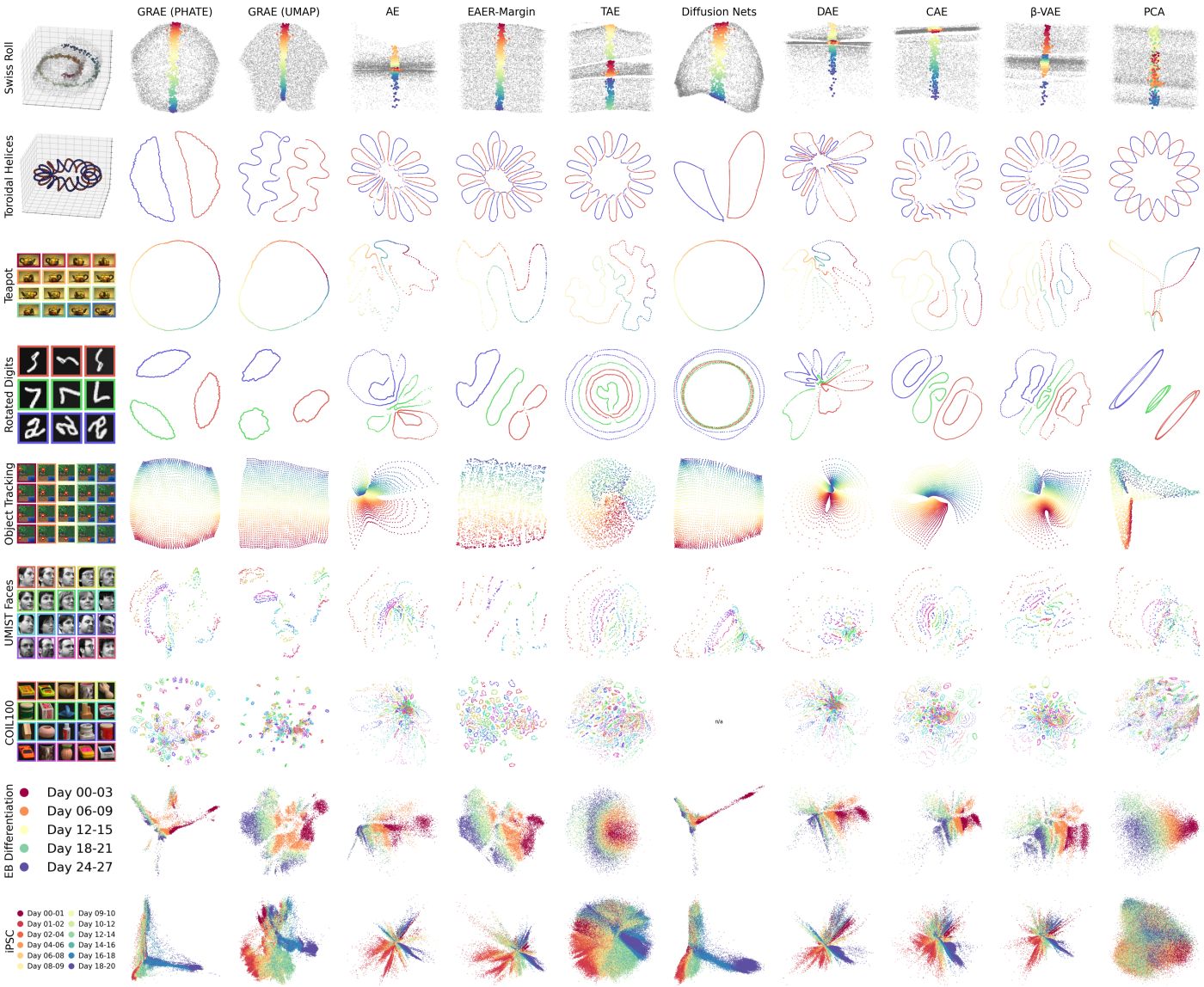

Fig. 3. **Latent space visualizations.** Latent representations learned in a fully unsupervised manner by all considered methods on 9 datasets. On the Swiss Roll plots, training points are grayscale and test points are colored. We show here the run with the best test MSE, as assessed in Sec. 4.3, without any additional hand tuning. GRAE recovers a sensible geometry for all problems while other methods fail on multiple datasets (e.g. unrolling the Swiss Roll and disentangling the Rotated Digits).

space than is typically achieved without the regularization. The reduction in MSE by GRAE is achieved while still preserving ground truth information in the data as measured by the R^2 and accuracy metrics, being top three on all problems with the exception of GRAE (UMAP)'s R^2 score on Toroidal Helices. These results show that explicitly preserving geometry in the encoder loss gives GRAE a superior ability to reveal latent structure in data, while enhancing the decoder's ability to invert the latent space.

Importantly, no stable competitor to GRAE arises from the comparisons. Diffusion Nets perform well on reconstructing the Swiss Roll and the Teapot samples, but is less successful at reconstructing Toroidal Helices and Rotated Digits, in addition to struggling at providing a separable embedding for the latter (see Figure 3). EAER-Margin achieves good classification performance on most of the relevant problems. This is unsurprising since its margin regularization explicitly seeks to pull apart points belonging to different neighborhoods and is therefore expected to separate classes relatively well. This was not universal, however, as EAR-Margin had poor accuracy on Toroidal Helices. Additionally, EAER-Margin reduces reconstruction quality on many benchmarks, performing worse than vanilla AE.

The MSE comparison over the 10 runs are further visualized in Figure 4. Not only does GRAE achieve lower MSE, but it does so with less variance between runs on many problems, most notably on Toroidal Helices, Teapot, Rotated Digits, and Object Tracking. This could be explained by the non-parametric optimization of UMAP and PHATE, which yield stable reference embeddings $\mathcal{E}$ throughout runs.

For its part, β-VAE only managed to extract useful representations with low values of β. The poor performance on the Object Tracking dataset on both the MSE and R^2 metrics can be linked to the empirical observation that, in

TABLE 2
Average performance metrics on the biological datasets. See Table 1 caption and the text for descriptions of the metrics.

Dataset	Model	Metrics		
		MSE	R^2	*Acc.*
EB Differentiation	GRAE (PHATE)	**0.1766** (2)	n/a	n/a
	GRAE (UMAP)	**0.1765** (1)	n/a	n/a
	AE	0.1773 (5)	n/a	n/a
	EAER-Margin	0.1773 (5)	n/a	n/a
	TAE	0.1806 (9)	n/a	n/a
	Diffusion Nets	0.1777 (8)	n/a	n/a
	DAE	0.1776 (7)	n/a	n/a
	CAE	0.1770 (4)	n/a	n/a
	β-VAE	0.1768 (3)	n/a	n/a
	PCA	0.1814 (10)	n/a	n/a
iPSC	GRAE (PHATE)	0.7173 (6)	n/a	n/a
	GRAE (UMAP)	0.7130 (5)	n/a	n/a
	AE	0.7055 (4)	n/a	n/a
	EAER-Margin	0.7658 (7)	n/a	n/a
	TAE	0.8928 (9)	n/a	n/a
	Diffusion Nets	0.8774 (8)	n/a	n/a
	DAE	**0.6919** (2)	n/a	n/a
	CAE	**0.6910** (1)	n/a	n/a
	β-VAE	0.6929 (3)	n/a	n/a
	PCA	1.3669 (10)	n/a	n/a

TABLE 3
Average performance metrics on the images datasets. See Table 1 caption and the text for descriptions of the metrics.

Dataset	Model	Metrics		
		MSE	R^2	*Acc.*
Teapot	GRAE (PHATE)	0.0004 (3)	**1.00** (1)	n/a
	GRAE (UMAP)	**0.0003** (2)	**1.00** (1)	n/a
	AE	0.0016 (6)	0.21 (9)	n/a
	EAER-Margin	0.0026 (8)	0.27 (8)	n/a
	TAE	0.0015 (5)	0.40 (5)	n/a
	Diffusion Nets	**0.0002** (1)	**1.00** (1)	n/a
	DAE	0.0010 (4)	0.35 (7)	n/a
	CAE	0.0016 (6)	0.43 (4)	n/a
	β-VAE	0.0028 (9)	0.19 (10)	n/a
	PCA	0.0112 (10)	0.40 (5)	n/a
Rotated Digits	GRAE (PHATE)	**0.0002** (1)	**0.94** (1)	**1.00** (1)
	GRAE (UMAP)	**0.0004** (2)	**0.87** (2)	**1.00** (1)
	AE	0.0045 (8)	0.20 (9)	0.90 (7)
	EAER-Margin	0.0024 (3)	0.45 (3)	**1.00** (1)
	TAE	0.0038 (5)	0.23 (6)	0.34 (9)
	Diffusion Nets	0.0043 (6)	0.23 (6)	0.32 (10)
	DAE	0.0049 (9)	0.22 (8)	0.77 (8)
	CAE	0.0044 (7)	0.26 (4)	0.91 (6)
	β-VAE	0.0032 (4)	0.24 (5)	0.96 (5)
	PCA	0.0619 (10)	0.19 (10)	**1.00** (1)
Object Tracking	GRAE (PHATE)	**0.0031** (1)	**0.98** (2)	n/a
	GRAE (UMAP)	**0.0035** (2)	**1.00** (1)	n/a
	AE	0.0051 (6)	0.39 (7)	n/a
	EAER-Margin	0.0061 (7)	**0.98** (2)	n/a
	TAE	0.0084 (8)	0.37 (8)	n/a
	Diffusion Nets	0.0048 (4)	0.97 (4)	n/a
	DAE	0.0049 (5)	0.29 (9)	n/a
	CAE	0.0040 (3)	0.51 (5)	n/a
	β-VAE	0.0093 (9)	0.15 (10)	n/a
	PCA	0.0100 (10)	0.51 (5)	n/a
UMIST Faces	GRAE (PHATE)	**0.0078** (2)	n/a	0.50 (3)
	GRAE (UMAP)	**0.0076** (1)	n/a	**0.56** (2)
	AE	0.0090 (6)	n/a	0.29 (9)
	EAER-Margin	0.0102 (8)	n/a	**0.64** (1)
	TAE	0.0093 (7)	n/a	0.36 (6)
	Diffusion Nets	0.0081 (5)	n/a	0.39 (4)
	DAE	0.0104 (9)	n/a	0.27 (10)
	CAE	0.0080 (4)	n/a	0.35 (7)
	β-VAE	0.0079 (3)	n/a	0.33 (8)
	PCA	0.0181 (10)	n/a	0.39 (4)
COIL100	GRAE (PHATE)	**0.0098** (2)	n/a	0.60 (3)
	GRAE (UMAP)	**0.0093** (1)	n/a	**0.65** (2)
	AE	0.0113 (6)	n/a	0.47 (7)
	EAER-Margin	0.0114 (7)	n/a	**0.66** (1)
	TAE	0.0145 (8)	n/a	0.48 (6)
	Diffusion Nets	n/a	n/a	n/a
	DAE	0.0112 (4)	n/a	0.46 (8)
	CAE	0.0110 (3)	n/a	0.53 (5)
	β-VAE	0.0112 (4)	n/a	0.55 (4)
	PCA	0.0269 (9)	n/a	0.35 (9)

many runs, the decoder only reconstructed the background without the character. We further discuss the selection of the β parameter in the β-VAE in Appendix B.2.

Furthermore, no distance-based method managed to improve beyond the AE reconstruction performance on the iPSC data, although GRAE fared considerably better than its counterparts in that regard (Figure 4). Other regularization methods (DAE, CAE, β-VAE) do show a small improvement

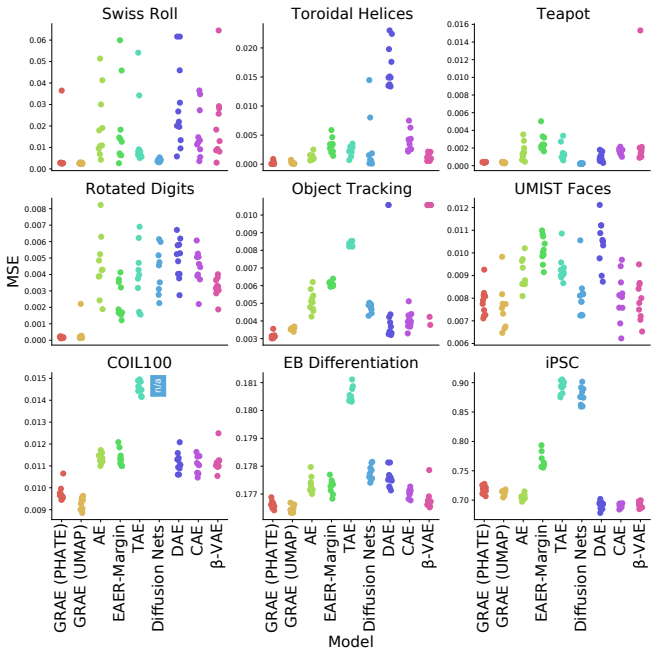

Fig. 4. MSE between the input and the reconstructions. Lower is better. Each point is a given run of a model on the data with the same hyperparameters, but using a different seed, with 10 runs total. PCA's MSE was not included to improve plot scaling. Not only does GRAE achieve lower MSE, but it is also more stable with a lower variance across runs on most datasets.

over the AE. However we show that geometry-based regularizations still help to gather useful insights and explore the GRAE iPSC embedding in more depth in Section 4.5.

By comparing Figure 3 and the quantitative results, we notice that embeddings with low MSE also tend to show good global structure. That is, the manifold is unfolded in the latent space with little to no self-intersections (e.g. the GRAE embeddings of Rotated Digits, Object Tracking and Teapot). We further discuss this connection in Section 4.4.

4.4 Impact of geometric regularization on reconstruction quality

Based on GRAE's reconstruction errors (Tables 1, 2 and 3 and Figure 4), we observe that GRAE generally improves the MSE of the decoder, despite adding a regularization term that deteriorates the reconstruction error global minimum. A possible explanation is that some latent space shifts

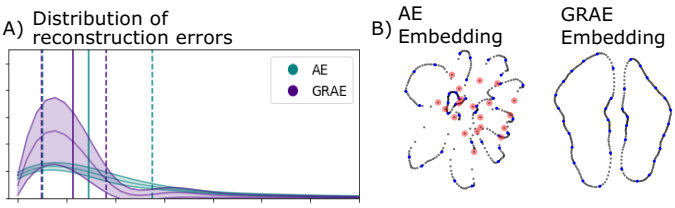

Fig. 5. **Reconstructing latent space interpolations with GRAE. A)** Distributions of errors of two rotated digits averaged over ten runs for AE vs GRAE. Dashed lines represent the 1st and 3rd quartiles, and solid lines represent the median. We notice that AE is more unstable than GRAE, having a heavier tail, since it fails completely to reconstruct certain images, while GRAE typically presents lighter tails. **B)** Typical embeddings produced for AE and GRAE. Blue points represent a subsample of the training data (subsampling only done for visualization purposes). Black points are the generated points on the latent space via interpolation. Red colored points in the AE embedding represent the 20 interpolated points with the highest reconstruction error. We observe that bad reconstruction typically occurs in sparse regions or crossing lines, i.e., in regions with poorly learned geometry.

caused by geometric regularization (e.g. forcing a circle on Teapot, uncoiling the Swiss Roll) actually drive gradient descent out of the local MSE minima in which vanilla AE falls. Indeed, most AE embeddings in Figure 3 show structural overlaps where points from different regions of the original data manifold share the same latent space coordinates, meaning the encoder function f is not injective and hence not invertible. By enforcing better geometry, GRAE appears to favor injective encoder mappings which, assuming bijectivity on the manifold, should facilitate learning of the inverse function $f^{\dagger}$, leading to the better reconstructions.

Additionally, the regularization generates a more stable reconstruction over the whole dataset, especially in those regions where the autoencoder produces inaccurate geometry in the latent space. To support these claims, we conduct an experiment on two rotated MNIST digits (Figure 5), generating a full rotation for each of the digits and removing in-between observations as the test set. After training GRAE (PHATE) on the remaining observations, we interpolate over consecutive observations in the embedding space i.e., consecutive angle rotations in the training set. Then we compute the reconstruction error between the generated points via interpolation with the previously removed in-between observations. The results show that the distribution of the AE errors has a much heavier tail than the distribution of the GRAE errors, suggesting that the GRAE embedding is more stable.

This experiment also shows that learning accurate geometry from the data via GRAE can be useful for generating new points on the latent space via interpolation over adjacent observations. Such points can then be fed to the decoder, generating a more faithfully reconstructed geodesic trajectory between points on the manifold in the ambient space in comparison to AE. We further explore this in Section 4.5.

4.5 Geometry consistent interpolations in the latent space

Uncovering a consistent geometric structure in the latent space, allows us to study the behavior of the data following discovered trajectories which are otherwise unidentifiable in the vanilla autoencoder scheme. We present a case study showing the capabilities of GRAE (PHATE) in this regard. We aim to recover marker interactions in the iPSC mass cytometry data [54] across time-evolving trajectories discovered in the latent representation. The experimental outline is shown in Figure 6.

Mass cytometry data is noisy and marker expression interactions are difficult to extract from the raw data. This issue can be tackled with methods such as MAGIC [57], a powerful diffusion-based approach that performs data imputation and denoising, which has been shown to be particularly useful for recovering gene-gene interactions in complicated single-cell RNA-sequencing datasets. Thus, we chose to compare GRAE's reconstructed ambient space with the MAGIC transformation of the raw data.

Figure 6 shows a comparison of GRAE (PHATE) with a vanilla autoencoder in recovering gene-gene interactions. We can observe how the latent space of GRAE preserves a geometric structure consistent with the known biology of the data [7], whereas the autoencoder generates a cloud of points making it difficult to understand clear patterns. After we have identified a trajectory, we observe the marker-to-marker interactions in the ambient space following it. Thus, we build a path in the latent space across an identified trajectory and feed these new generated points through the decoder. We use the method presented in [55] to compute density-based geodesic trajectories estimated from finite data. To draw a comparison with the vanilla autoencoder, we compute a geodesic trajectory in both latent representations between the same start and end points. From the results in Figure 6, it is clear that obtaining a latent space consistent with the geometry of the data enables accurate reconstruction of newly generated points. In general, geodesic paths drawn in GRAE's latent space do not suffer from the same ambiguities as the vanilla autoencoder, e.g. crossing lines or lack of structure in the latent factors. Decoding the trajectories back to the ambient space will follow the true structure of the data. This however has its limitations when the latent dimension is higher than two or three dimensions, which complicates the identification and computation of geodesic paths.

4.6 Semi-supervised learning using geometric regularization

Lastly, we show how to leverage the geometric regularization to perform semi-supervised learning. Following the multi-task learning strategy implemented in [58], we build a fully connected network where the output layer focuses on classification over the labeled data. Auxiliary tasks are imposed on the inner layers seeking to minimize the geometric loss. A schematic overview of our approach is displayed in Figure 7.

Our method relies on the *manifold assumption* for semi-supervised learning, i.e. observations that are close to each other on a low dimensional manifold share the same label. This enables us to leverage the geometry learned from the unlabeled data via the geometric regularization. By encouraging common representations, useful for classification as well as for learning the geometry of the data, the neural network will tend to force close points in the geometric

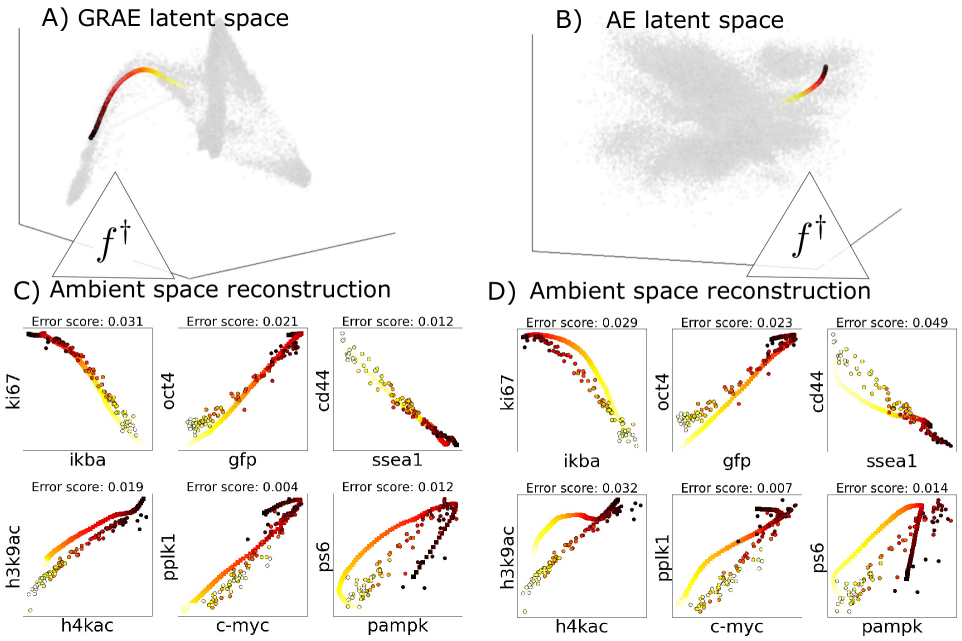

Fig. 6. **Geodesic paths in the regularized latent space of iPSC data**. **A,B)** Latent representations in three dimensions of GRAE (PHATE) and a vanilla autoencoder respectively. The grey points are the training data representation. The colored paths are the geodesic paths found using the method presented in [55] between the same pair of points in both cases. The construction of such path clearly follows an identified branch in GRAE's embedding. In contrast, the autoencoder produces a cloud, from which it is difficult to determine if the computed geodesic follows a reasonable trajectory consistent with the data. **C,D)** Reconstructed ambient space for the newly generated points (paths) compared with MAGIC's recovered interactions (solid points). We plot pairs of markers with the highest mutual information score computed by DREMI [56]. GRAE is able to denoise the data and recover gene marker interactions from the raw data more consistently than the autoencoder. The geodesic found in the autoencoder manifests a higher deviation from the MAGIC recovered interactions when decoded back to the ambient space, showing how the trajectories drawn in the latent space do not reflect real paths in the data. To compute an error score that measures the discrepancy between the created path and MAGIC's trajectories, we compute the MSE between each generated point and its 3 nearest neighbors in the MAGIC reconstruction, as well as the MSE between each MAGIC point and its 3 nearest neighbors in the generated points. The final obtained error scores indicate that the GRAE latent space can be used to generate new points more accurately than the vanilla AE as GRAE outperforms the AE in most settings.

embedding to share the same label for classification. Thus, the geometric loss acts as an auxiliary task, creating an inductive bias that drives the model to prefer hypotheses where the *manifold assumption* is included.

As a base classification model we implemented a neural network with 3 fully connected hidden layers with dimensions 100-100-100. For these experiments we used a two-dimensional UMAP embedding as the target $\mathcal{E}$ in the geometric loss. Our approach, called GRNN (Geometrically Regularized Neural Network), is compared against three state of the art graph-based semi-supervised learning methods: Laplace learning [59], p-Laplace learning [60], and Poisson learning [61]. We also compared to a vanilla neural network equipped with the same architecture as GRNN, but lacking the geometric regularization.

TABLE 4
Datasets used for the semi-supervised learning experiments.

Dataset	Number of classes	Observations	Features	Reference
HumanPancreas	14	8569	1000	[62]
PBMC	6	1919	50	[63]
MNIST	10	5000	784	
Xin	4	1449	1000	[64]
Zheng Sorted	10	20000	1000	[65]
Zheng68k	11	10000	1000	[65]

We performed comparisons using the datasets summarized in Table 4 which include four single-cell RNA sequencing datasets (HumanPancreas, Xin, Zheng Sorted, and Zheng 68k) previously tested for single-cell classification in [66]. The datasets are publicly available at https://zenodo.org/record/3357167#.YdX7q2jMKUk. We applied the same preprocessing steps as in [66]. The PBMC dataset consists of single-cell ATACseq measurements of peripheral blood mononuclear cells [67]. For the MNIST dataset, experiments where conducted on a subsampled version of 5000 observations as in [68] to maintain a small quantity of labeled observations.

The results are summarized in Table 5. We considered different label rates and reported the average accuracy on the unlabeled data over 10 randomized runs. We found that the geometric regularization helps to perform better learning, obtaining the best results in most of the cases, especially in the low sample regime (1%-10%). In contrast, the vanilla NN often, but not always, has the worst performance. The other three approaches vary in their performance but perform worse than GRNN in nearly all cases.

Finally, we present an ablation study showing the differences in performance between the vanilla network and our approach for networks of different sizes. The results are shown in Figure 10 of Appendix C.1. The geometric regularization in our approach improves the robustness of the network as it achieves fairly consistent accuracy for different numbers of layers. In contrast, the vanilla neural network's performance degrades when more layers are included, suggesting it is more vulnerable to overfitting.

TABLE 5
Semi-supervised classification accuracy results. Average testing accuracy is given over 10 runs for various levels of available labeled data. Comparisons are between our approach (GRNN), a Vanilla Neural Network, and three graph-based methods for semi-supervised learning.

Dataset	Model	Labeled percentage				
		1.00%	5.00%	10.00%	20.00%	40.00%
Human Pancreas	GRNN (Ours)	**0.909** (1)	**0.934** (1)	**0.936** (1)	**0.940** (1)	**0.942** (2)
	Vanilla NN	0.495 (5)	0.693 (5)	0.707 (5)	0.866 (4)	**0.943** (1)
	Laplace	0.765 (3)	0.818 (3)	0.854 (3)	0.884 (3)	0.909 (4)
	p-Laplace	**0.787** (2)	**0.837** (2)	**0.868** (2)	**0.892** (2)	0.913 (3)
	Poisson	0.693 (4)	0.765 (4)	0.796 (4)	0.821 (5)	0.838 (5)
PBMC	GRNN (Ours)	**0.867** (1)	**0.919** (1)	**0.922** (1)	**0.936** (1)	**0.941** (1)
	Vanilla NN	0.389 (5)	0.650 (5)	0.775 (5)	0.865 (5)	0.923 (4)
	Laplace	0.514 (4)	0.809 (4)	0.908 (3)	**0.930** (2)	**0.940** (2)
	p-Laplace	0.628 (3)	0.880 (3)	**0.911** (2)	0.925 (3)	0.934 (3)
	Poisson	**0.857** (2)	**0.899** (2)	0.905 (4)	0.911 (4)	0.919 (5)
MNIST	GRNN (Ours)	**0.884** (1)	**0.929** (1)	**0.947** (1)	0.954 (3)	0.955 (3)
	Vanilla NN	0.563 (5)	0.750 (5)	0.836 (5)	0.894 (5)	0.930 (4)
	Laplace	0.721 (4)	0.917 (3)	**0.943** (2)	**0.957** (1)	**0.966** (1)
	p-Laplace	**0.776** (2)	**0.919** (2)	0.942 (3)	**0.955** (2)	**0.963** (2)
	Poisson	0.775 (3)	0.886 (4)	0.891 (4)	0.902 (4)	0.903 (5)
Xin	GRNN (Ours)	**0.721** (2)	0.831 (3)	0.856 (3)	0.869 (4)	0.885 (3)
	Vanilla NN	0.459 (5)	0.562 (5)	0.590 (5)	0.565 (5)	0.470 (5)
	Laplace	0.645 (3)	**0.857** (2)	**0.876** (2)	**0.896** (2)	**0.903** (2)
	p-Laplace	**0.764** (1)	**0.872** (1)	**0.882** (1)	**0.900** (1)	**0.905** (1)
	Poisson	0.624 (4)	0.816 (4)	0.853 (4)	0.871 (3)	0.885 (3)
Zheng	GRNN (Ours)	**0.732** (1)	**0.784** (1)	**0.807** (1)	**0.812** (2)	**0.823** (2)
	Vanilla NN	0.528 (3)	**0.720** (2)	**0.783** (2)	**0.824** (1)	**0.855** (1)
	Laplace	0.446 (5)	0.545 (5)	0.604 (5)	0.656 (5)	0.691 (4)
	p-Laplace	0.498 (4)	0.565 (4)	0.617 (4)	0.660 (4)	0.691 (4)
	Poisson	**0.635** (2)	0.663 (3)	0.675 (3)	0.688 (3)	0.700 (3)
Zheng 68k	GRNN (Ours)	**0.555** (1)	**0.596** (1)	**0.604** (1)	**0.608** (1)	**0.618** (1)
	Vanilla NN	0.337 (4)	0.455 (4)	0.506 (4)	0.549 (4)	**0.590** (2)
	Laplace	0.390 (3)	0.541 (3)	**0.573** (2)	**0.588** (2)	0.576 (3)
	p-Laplace	**0.460** (2)	**0.549** (2)	0.568 (3)	0.581 (3)	0.572 (4)
	Poisson	0.334 (5)	0.380 (5)	0.390 (5)	0.411 (5)	0.443 (5)

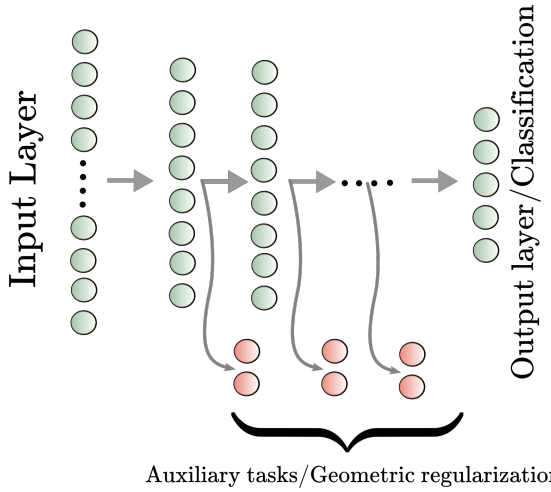

Fig. 7. Semi-supervised model architecture using the geometry regularization. The geometry regularization is imposed as an auxiliary task in all the hidden layers forcing the hidden representations to be consistent with the geometrical structure of the data. In this way, our model takes into account labeled and unlabeled observations.

5 Discussion

The improved performance of the geometric regularization for learning unsupervised two-dimensional representations (Section 4.3) and semi-supervised learning (Section 4.6) can be largely attributed to PHATE and UMAP's ability to capture important aspects of the high-dimensional geometry in the two-dimensional latent space we considered. This observation gives rise to two potential concerns regarding our proposed approach. The first one is that our regularization will be sensitive to the manifold learning algorithm used to compute the reference embedding $\mathcal{E}$ that we use as a prior. Indeed, geometry-regularized embeddings might inherit some of the less desirable properties of $\mathcal{E}$, such as an absence of global structure caused by a mistuned or poorly designed manifold learning algorithm [69]. Similar issues can arise if geometry regularization with any distance-based reference algorithm is applied to data where the chosen distance is not an adequate dissimilarity measure between observations.

The second concern relates to the dimensionality of the latent space. While in Section 4.3 we focused on the two-dimensional case to enable qualitative assessments and data exploration —the main purpose of PHATE and UMAP— other autoencoder applications may require a higher dimensional latent space. We test GRAE (PHATE) and some competing methods with a bottleneck size of 2, 4, 6 and 8 in Appendix B.4. The results show that GRAE typically outperforms other approaches in representing the latent factors and classes in a convincing way for all bottleneck sizes on most problems, even though the benefit in the reconstruction error seems to disappear for higher dimensions. Furthermore, the semi-supervised results of Section 4.6 indicate that the geometric regularization can be successfully used to learn higher-dimensional intermediary representations in neural networks. Additionally, most exploratory data analysis focuses on a low-dimensional (i.e. single digit) representation of the data for practical reasons as higher dimensions are difficult for humans to explore. Nevertheless, the use of geometric regularization and more generally manifold learning for learning all-purpose high-dimensional representations (e.g., 128D or 256D) remains an open problem.

6 Conclusion

We proposed the geometry regularized autoencoder (GRAE), a general parametric framework to enhance autoencoders' latent representation by taking advantage of 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. We also show that a similar regularization applied to a neural network results in superior performance for the semi-supervised problem. While the primary focus of this work is on using PHATE and UMAP embeddings to regularize the networks, we leave to future work 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.

Acknowledgments

This research was partially funded by an IVADO (l'Institut de valorisation des données)MSc. Scholarship and an FRQNT (Fonds de recherche du Québec Nature et technologies)B1XScholarship [S.M.], in part by an IVADO (l'Institut de valorisation des données) Undergraduate introduction to research scholarship (2020) [S.M.], in part by ISM (Institut des sciences mathematiques) Undergraduate Summer

Scholarship (2021) [S.M.], in part by Canada CIFAR AI Chair [G.W.], in part by IVADO Professor research funds [G.W.], in part by NSERC under Discovery Grant 03267 [G.W.], in part by the NIH under Grant [R01GM135929 [G.W.], and in part by the NSF under Grant 2212325 [K.M.]. The content provided here is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies. Moreover, this research was enabled in part by support provided by Calcul Québec (www.calculquebec.ca) and Compute Canada (www.computecanada.ca).

REFERENCES

[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] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[4] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NeurIPS*, 2002, pp. 585–591.

[6] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[7] K. R. Moon *et al.*, "Visualizing structure and transitions in high-dimensional biological data," *Nature Biotechnology*, vol. 37, no. 12, pp. 1482–1492, 2019.

[8] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv e-prints*, Feb. 2018.

[9] A. F. Duque, G. Wolf, and K. R. Moon, "Visualizing high dimensional dynamical processes," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2019.

[10] E.-a. D. Amir *et al.*, "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia," *Nature Biotechnology*, vol. 31, no. 6, p. 545, 2013.

[11] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap embeddings for representation and semisupervised learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, 2021.

[12] J. S. Rhodes, A. Cutler, G. Wolf, and K. R. Moon, "Random forest-based diffusion information geometry for supervised visualization and data exploration," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2021, pp. 331–335.

[13] K. R. Moon, J. S. Stanley III, D. Burkhardt, D. van Dijk, G. Wolf, and S. Krishnaswamy, "Manifold learning-based methods for analyzing single-cell rna-sequencing data," *Current Opinion in Systems Biology*, vol. 7, pp. 36–46, 2018.

[14] L. Haghverdi, M. Buettner, F. A. Wolf, F. Buettner, and F. J. Theis, "Diffusion pseudotime robustly reconstructs lineage branching," *Nature Methods*, vol. 13, no. 10, p. 845, 2016.

[15] 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.

[16] L. Sgier, R. Freimann, A. Zupanic, and A. Kroll, "Flow cytometry combined with viSNE for the analysis of microbial biofilms and detection of microplastics," *Nature Communications*, vol. 7, no. 1, pp. 1–10, 2016.

[17] E. Z. Macosko *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[18] E. Becht *et al.*, "Dimensionality reduction for visualizing single-cell data using umap," *Nature biotechnology*, vol. 37, no. 1, p. 38, 2019.

[19] Y. Zhao *et al.*, "Single cell immune profiling of dengue virus patients reveals intact immune responses to Zika virus with enrichment of innate immune signatures," *PLoS Neglected Tropical Diseases*, vol. 14, no. 3, p. e0008112, 2020.

[20] W. R. Karthaus *et al.*, "Regenerative potential of prostate luminal cells revealed by single-cell analysis," *Science*, vol. 368, no. 6490, pp. 497–505, 2020.

[21] C. Baccin *et al.*, "Combined single-cell and spatial transcriptomics reveal the molecular, cellular and spatial bone marrow niche organization," *Nature Cell Biology*, pp. 1–11, 2019.

[22] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *NeurIPS*, 2013, pp. 2643–2651.

[23] S. Gigante, A. S. Charles, S. Krishnaswamy, and G. Mishne, "Visualizing the phate of neural networks," in *NeurIPS*, 2019, pp. 1840–1851.

[24] S. Horoi, V. Geadah, G. Wolf, and G. Lajoie, "Low-dimensional dynamics of encoding and learning in recurrent neural networks," in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 276–282.

[25] R. R. Coifman and S. Lafon, "Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 31–52, 2006.

[26] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *NeurIPS*, 2001, pp. 682–688.

[27] M. Vladymyrov and M. Á. Carreira-Perpinán, "Locally linear landmarks for large-scale manifold learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 256–271.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[30] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *NeurIPS*, vol. 15, no. Nov, p. 833–840, 2008.

[31] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 577–586.

[32] D. Kobak and G. C. Linderman, "UMAP does not preserve global structure any better than t-SNE when using the same initialization," *bioRxiv*, 2019.

[33] 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.

[34] L. Van Der Maaten, "Learning a parametric embedding by preserving local structure," in *Artificial Intelligence and Statistics*, 2009, pp. 384–391.

[35] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in *NeurIPS*, 2004, pp. 177–184.

[36] G. Mishne, U. Shaham, A. Cloninger, and I. Cohen, "Diffusion nets," *Applied and Computational Harmonic Analysis*, vol. 47, no. 2, pp. 259–285, 2019.

[37] V. d. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *NeurIPS*, 2003, pp. 721–728.

[38] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th ICML*, 2008, pp. 1096–1103.

[39] A. Supratak, L. Li, and Y. Guo, "Feature extraction with stacked autoencoders for epileptic seizure detection," in *2014 36th Annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2014, pp. 4184–4187.

[40] H. Liu and T. Taniguchi, "Feature extraction and pattern recognition for human motion by a deep sparse autoencoder," in *2014 IEEE International Conference on Computer and Information Technology*. IEEE, 2014, pp. 173–181.

[41] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th ICML*, 2011, pp. 833–840.

[42] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[43] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[44] W. Chu and D. Cai, "Stacked similarity-aware autoencoders." in *IJCAI*, 2017, pp. 1561–1567.

[45] K. Jia, L. Sun, S. Gao, Z. Song, and B. E. Shi, "Laplacian auto-encoders: An explicit learning of nonlinear data manifold," *Neurocomputing*, vol. 160, pp. 250–260, 2015.

[46] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He, and Z. Shi, "Embedding with autoencoder regularization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 208–223.
[47] G. Pai, R. Talmon, A. Bronstein, and R. Kimmel, "Dimal: Deep isometric manifold learning using sparse geodesic sampling," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 819–828.
[48] M. Moor, M. Horn, B. Rieck, and K. Borgwardt, "Topological autoencoders," *arXiv preprint arXiv:1906.00722*, 2019.
[49] H. Edelsbrunner and J. Harer, "Persistent homology-a Survey," *Contemporary mathematics*, vol. 453, pp. 257–282, 2008.
[50] S. Barannikov, "The framed morse complex and its invariants," *Singularities and Bifurcations*, 1994.
[51] 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.
[52] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings ICML*, 2004, p. 106.
[53] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1120–1127.
[54] E. R. Zunder, E. Lujan, Y. Goltsev, M. Wernig, and G. P. Nolan, "A continuous molecular roadmap to ipsc reprogramming through progression analysis of single-cell mass cytometry," *Cell stem cell*, vol. 16, no. 3, pp. 323–337, 2015.
[55] G. Arvanitidis, S. Hauberg, P. Hennig, and M. Schober, "Fast and robust shortest paths on manifolds learned from data," *arXiv preprint arXiv:1901.07229*, 2019.
[56] S. Krishnaswamy, M. H. Spitzer, M. Mingueneau, S. C. Bendall, O. Litvin, E. Stone, D. Pe'er, and G. P. Nolan, "Conditional density-based analysis of t cell signaling in single-cell data," *Science*, vol. 346, no. 6213, 2014.
[57] D. Van Dijk *et al.*, "Recovering gene interactions from single-cell data using data diffusion," *Cell*, vol. 174, no. 3, pp. 716–729, 2018.
[58] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 639–655.
[59] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.
[60] M. Flores, J. Calder, and G. Lerman, "Algorithms for lp-based semi-supervised learning on graphs," *arXiv preprint arXiv:1901.05031*, 2019.
[61] J. Calder, B. Cook, M. Thorpe, and D. Slepcev, "Poisson learning: Graph based semi-supervised learning at very low label rates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1306–1316.
[62] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein *et al.*, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure," *Cell systems*, vol. 3, no. 4, pp. 346–360, 2016.
[63] K. Cao, Y. Hong, and L. Wan, "Manifold alignment for heterogeneous single-cell multi-omics data integration using pamona," *Bioinformatics*, vol. 38, no. 1, pp. 211–219, 2022.
[64] Y. Xin, J. Kim, H. Okamoto, M. Ni, Y. Wei, C. Adler, A. J. Murphy, G. D. Yancopoulos, C. Lin, and J. Gromada, "Rna sequencing of single human islet cells reveals type 2 diabetes genes," *Cell metabolism*, vol. 24, no. 4, pp. 608–615, 2016.
[65] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu *et al.*, "Massively parallel digital transcriptional profiling of single cells," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.
[66] T. Wang, J. Bai, and S. Nabavi, "Single-cell classification using graph convolutional networks," *bioRxiv*, 2021.
[67] C. Wang, D. Sun, X. Huang, C. Wan, Z. Li, Y. Han, Q. Qin, J. Fan, X. Qiu, Y. Xie *et al.*, "Integrative analyses of single-cell transcriptome and regulome using maestro," *Genome biology*, vol. 21, no. 1, pp. 1–28, 2020.
[68] M. Budninskiy, A. Abdelaziz, Y. Tong, and M. Desbrun, "Laplacian-optimized diffusion for semi-supervised learning," *Computer Aided Geometric Design*, vol. 79, p. 101864, 2020.
[69] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, vol. 1, no. 10, p. e2, 2016.
[70] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," 2010.
[71] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-100)," 1996.
[72] D. B. Graham and N. M. Allinson, "Characterising virtual eigensignatures for general purpose face recognition," in *Face Recognition*. Springer, 1998, pp. 446–456.
[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Andres F. Duque obtained his B.S. degree in finance in Universidad de Medellin, in 2015, and his M.S. degree in applied mathematics from Universidad EAFIT, Medellin, Colombia, in 2017. He is currently pursuing his Ph.D. degree in Statistics at Utah State University.

Sacha Morin obtained his B.S. degree in mathematics and computer science at the Université de Montréal in 2021. He is currently pursuing his M.S. degree in computer science at the Université de Montréal and at Mila - Quebec AI Institute.

Guy Wolf is an associate professor in the Department of Mathematics and Statistics (DMS) at the Université de Montréal (UdeM), a core academic member of Mila (the Quebec AI institute), and holds a Canada CIFAR AI Chair. He is also affiliated with the CRM center of mathematical sciences and the IVADO institute of data valorization. He holds an M.Sc. and a Ph.D. in computer science from Tel Aviv University. Prior to joining UdeM in 2018, he was a postdoctoral researcher (2013-2015) in the Department of Computer Science at École Normale Supérieure in Paris (France), and a Gibbs Assistant Professor (2015-2018) in the Applied Mathematics Program at Yale University. His research focuses on manifold learning and geometric deep learning for exploratory data analysis, including methods for dimensionality reduction, visualization, denoising, data augmentation, and coarse graining. Further, he is particularly interested in biomedical data exploration applications of such methods, e.g., in single cell genomics/proteomics and neuroscience.

Kevin R. Moon is an assistant professor in the Department of Mathematics and Statistics at Utah State University (USU). He holds a B.S. and M.S. degree in electrical engineering from Brigham Young University and an M.S. degree in mathematics and a Ph.D. in electrical engineering from the University of Michigan. Prior to joining USU in 2018, he was a postdoctoral scholar (2016-2018) in the Genetics Department and the Applied Mathematics Program at Yale University. His research interests are in the development of theory and applications in machine learning, big data, information theory, deep learning, and manifold learning.

APPENDIX A
DATASETS

We provide here a detailed description of the nine datasets used in the experiments presented in Section 4. The first one is the classic manifold problem known as the "Swiss Roll" where data points lie on a two dimensional plane "rolled" in a three dimensional ambient space. Classical approaches such as PCA or MDS typically fail to recover the non linear geometry of the data, as they rely on pairwise Euclidean distances instead of the true geodesic distances along the curvature of the roll. We generated 10,000 points on the Swiss Roll using the `scikit-learn` library. The manifold is then stretched along the main "straight" axis (which facilitates uncoiling for all considered manifold learning methods) and Gaussian noise is added.

We generate an additional synthetic manifold benchmark dataset by uniformly sampling 8,000 points from two non-overlapping helices on a torus ("Toroidal Helices"). Conceptually, the data lies on two distinct one-dimensional closed curves in the input space and a good data embedding should disentangle them.

Three image datasets focus on full object rotations, where samples lie on one or multiple circular manifolds. One problem is derived from the MNIST dataset [70], where three digits are picked randomly and rotated 360 times at one-degree intervals, for a total of 1080 images. The second one is known as the Teapot problem [52], in which 400 RGB images of size 76 x 128 feature a rotating textured teapot. The final object rotation problem is the well-known COIL-100 benchmark [71].

We benchmark methods on two additional image datasets. The first one ("Object Tracking") was created with a 16 x 16 small character moving on a 64 x 64 background. Approximately 2000 RGB images were generated and Gaussian noise was added to the background. The intrinsic manifold consists of the plane spanned by the x and y character coordinates on the background. The second is known as the UMIST Faces dataset [72], where different views (over a 90° interval) of the faces of 20 different subjects are shown in 575 gray scale images of size 112 x 92.

The final datasets aim to assess the potential of the considered methods in analyzing biological data. One dataset consists of single-cell mass cytometry data measuring iPSC reprogramming of mouse embryonic fibroblasts (hereinafter, "iPSC") as introduced in [54]. The data show the expression of 33 markers in 220,450 embryonic cells at various stages of development. We know from [7] that the cells, while initially similar, eventually specialize into two different groups, leading to a two-branch Y-shaped manifold. The only known ground truth in this case is the age of the cell when measured. Another task is exploring single-cell RNA-sequencing data of human stem cells ("EB differentiation") [7]. The data was sequenced over 5 3-day intervals during a 27-day time course and includes approximately 17,000 cells after preprocessing. The goal here is to observe how cells specialize in different lineages.

APPENDIX B
TRAINING DETAILS

B.1 Architecture and Training

For the synthetic and biological datasets, the neural network architecture for all models consists of 3 fully-connected hidden layers in the encoder and in the decoder with a 2D latent space, producing the following sequence of neurons between the input and output layers: 800-400-200-2-200-400-800. For the image datasets, we use a similar architecture except the first layer is replaced by two convolutional layers with max pooling and the last layer of the decoder is replaced with two deconvolution layers. We apply ReLU activations on all of the layers except in the bottleneck and the output of the decoder. We used Adam [73] as the optimizer for all experiments and early stopping for regularization. Models were allowed to train for up to 1000 epochs on iPSC and EB Differentiation, 2000 epochs on COIL-100 and 4000 epochs on the other problems. Patience was set to 30 epochs for iPSC, 100 epochs for EB Differentiation, 200 epochs for COIL-100 and 400 epochs for the remaining datasets. For Diffusion Nets, the maximum number of epochs and patience were both multiplied by a factor of 10 for a fairer comparison as Diffusion Nets do not use mini-batch training (and therefore only compute one gradient update per "epoch").

B.2 Hyperparameter Tuning

Hyperparameters were optimized for each model using a random search over the parameter space. In addition to tuning the learning rate and the batch size for training the neural networks, most parameters specific to a given method were also sampled, such as the diffusion parameter for GRAE (PHATE) and the margin size for EAER-Margin. The range and sampling distribution for each parameter is presented in Table 6. In total, 30 hyperparameter combinations were sampled for each model on each dataset, and the best one was selected using a 3-fold cross-validation scheme (only 1-fold in the case of iPSC given the large number of observations). Specifically, MSE on the validation fold was used as the selection metric.

Parameter	Models	Distribution	Values
learning rate	All	log-uniform	[2e-4, 2e-3]
batch size	All	uniform	[32, 100]
λ	All but AE, DAE and β-VAE	log-uniform	[1e-2, 1e2]
n_neighbors/knn	All but AE and Diffusion Nets	uniform	{5, 10, 20}
n_neighbors	Diffusion Nets	uniform	{10, 20, 50}
t	GRAE(PHATE)	uniform	{10, 25, 50, 100, 250}
gamma	GRAE(PHATE)	uniform	{0, 1}
min_dist	GRAE(UMAP)	uniform	[0, .99]
epsilon	Diffusion Nets	uniform	[1, 70]
eta (EV constraint)	Diffusion Nets	log-uniform	[1e-2, 1e2]
margin	EAER-Margin	log-uniform	[.01, 10]
β	β-VAE	log-uniform	[1e-4, 10]
dropout probability	DAE (Images)	uniform	[.1, .7]
noise standard deviation	DAE (Manifolds, Biological data)	log-uniform	[1e-4, .1]

TABLE 6
Hyperparameter distributions for the random search procedure for all datasets. As an exception, for the iPSC data, lambda was restricted to the [.5, 1e2] range and t was restricted to {100, 250} in an effort to learn embeddings visually distinguishable from those produced by vanilla AE. Further, using 5 neighbors with Diffusion Nets led to a number of numerical difficulties with the underlying diffusion maps implementation. We chose to replace it with 50 neighbors, which gave better results.

While the β-VAE authors recommend using $\beta > 1$ to favor disentanglement, we found such values to generally

lead to scrambled embeddings on our problem set. We therefore extended the range of the hyperparameter search to lower values of β to recover more interesting visualizations. For reference, we included examples of standard VAE embeddings ($\beta = 1$) in Figure 8.

As for the DAE, we use dropout noise (mask) on the images datasets and additive gaussian noise on the manifold and biological datasets.

B.3 Software

Experiments were executed using `Python` 3.8 and `Torch` 1.7.1 for the deep learning components. We used author implementations for UMAP (0.4.2) and PHATE (1.0.4). pyDiffMap (0.2.0.1) was used as the Diffusion Maps implementation behind Diffusion Nets. Other major utilities include `numpy` 1.19.1 and `scikit-learn` 0.23.0 [74]. We used our own implementations of EAER-Margin, Diffusion Nets, DAE, CAE and β-VAE. As for TAE, we reused the original source code for the topological soft constraint and adapted it to our AE architecture.

B.4 Increased bottleneck size

We show and discuss additional results with higher bottleneck dimensions in Figure 9 where we compare GRAE with the AE, β-VAE, CAE, and Diffusion Nets. We compare the results using the reconstruction error and either the R^2 or accuracy metrics across several datasets. The regression and classification metrics indicate that the manifold prior used by GRAE is still useful to recover latent factors in higher dimensional representations, most notably on the Teapot, Object Tracking and COIL-100 datasets. This advantage does not extend to reconstruction however, where GRAE usually performs comparably to other methods as dimensionality increases. We hypothesize that this observation could be related to the manifold structural overlaps we discussed in Section 4.4, which are less likely to occur in a latent space of increased dimensionality. As for other methods, CAE performs well on the UMIST dataset on both the reconstruction and classification benchmarks. Diffusion Nets recovers the latent factors well on Teapot and Object Tracking problems—as one would expect given a geometric loss term—but still does not manage to disentangle the Rotated Digits in 4D or 6D.

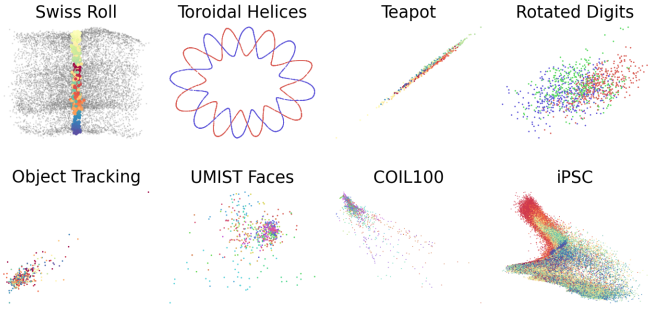

Fig. 8. VAE ($\beta = 1$) embeddings on 8 datasets. Higher values of β led to similarly noisy embeddings. We did not manage to train a VAE with $\beta = 1$ on the EB differentiation data.

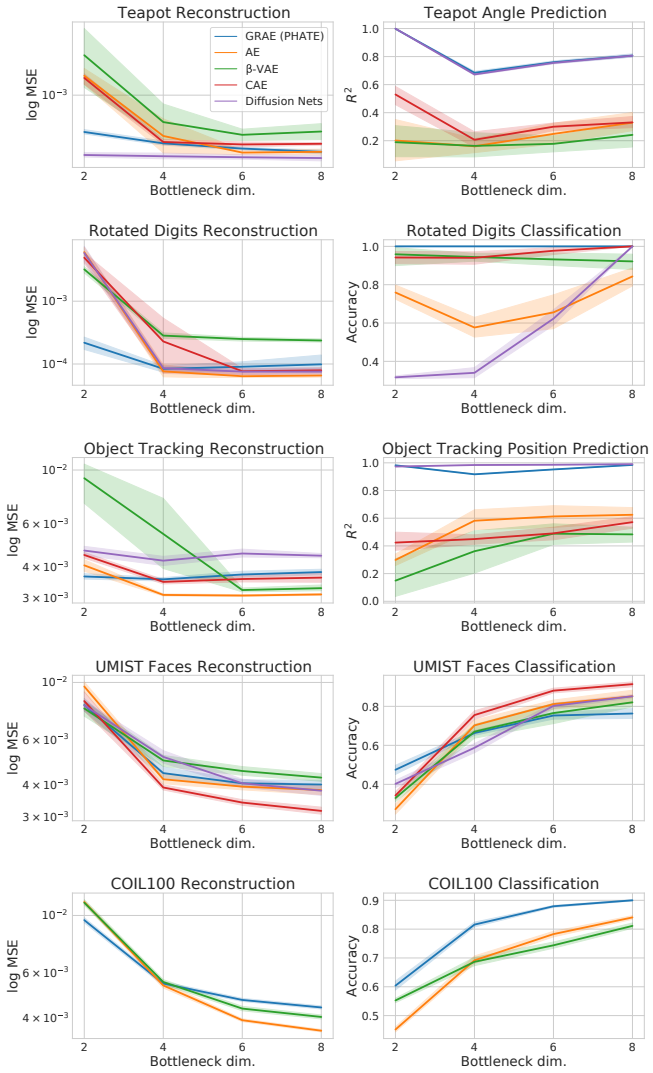

Fig. 9. Empirical metrics with various bottleneck sizes on the image datasets for some of the considered methods. We reused the best hyperparameters found in our main experiment and display the average of 10 runs and 95% CI. **Left column)** Log MSE (lower is better). The GRAE MSE advantage subsides as the bottleneck size increases. **Right column)** Accuracy or R^2 (higher is better) to assess representation quality. GRAE is consistently the best or second-best method in 4 of the datasets across all dimensions, showing that geometry regularized embeddings can better reflect latent factors or classes on some problems, even with a bottleneck size larger than 2. The higher R^2 performance of most models on Teapot with a two-dimensional bottleneck is a consequence of the polar coordinate conversion (see Appendix C for details) that we could not apply in higher dimensions. We could not compute results for CAE and Diffusion Nets on COIL100 due to computational limitations.

B.5 Training times

We present the training times for GRAE and some compared methods in Table 7. GRAE (PHATE) and GRAE (UMAP) compare favorably to other distance-based methods, such as TAE and EAER-Margin, while showing overall reasonable runtimes when compared to other methods.

B.6 Hardware

All experiments were run on nodes equipped with an *Intel Gold 6148 Skylake @ 2.4 GHz* CPU, 16 GB of available RAM,

Dataset	Model	Metrics	
		Time (min.)	AE-normalized
Swiss Roll	GRAE (PHATE)	1.01	2.04
	GRAE (UMAP)	0.74	1.48
	AE	0.50	1.00
	EAER-Margin	0.67	1.34
	TAE	5.50	11.03
	DAE	0.51	1.02
	CAE	0.69	1.38
	β-VAE	0.53	1.07
COIL100	GRAE (PHATE)	9.84	1.11
	GRAE (UMAP)	14.38	1.63
	AE	8.84	1.00
	EAER-Margin	13.83	1.56
	TAE	13.04	1.47
	DAE	13.39	1.51
	CAE	71.06	8.03
	β-VAE	9.26	1.05
iPSC	GRAE (PHATE)	27.29	2.80
	GRAE (UMAP)	15.18	1.56
	AE	9.76	1.00
	EAER-Margin	44.46	4.55
	TAE	106.13	10.87
	DAE	9.96	1.02
	CAE	14.04	1.44
	β-VAE	10.93	1.12

TABLE 7
Average training times over 3 runs for each model trained for 100 epochs with a batch size of 128. Training times include all required steps, such as precomputing $\mathcal{E}$ for GRAE and fitting the AE architecture. GRAE (PHATE) and GRAE (UMAP) use the scalable GRAE version presented in Figure 2. Diffusion Nets was not benchmarked due to the lack of mini-batch training. We further display the average runtimes normalized by the vanilla AE time.

and an *NVIDIA V100* GPU with 16 GB of memory.

APPENDIX C
R^2 METRIC DETAILS

Disentanglement of the latent factors of the various methods is assessed by fitting a linear regression to predict said factors using the embedding coordinates as regressors. High-quality embeddings should indeed be indicative of the data generating process and represent ground truth factors adequately, subject to a simple transformation. We report the resulting R^2 of the linear model to measure the strength of the relationship between the embeddings and a given ground truth factor. While in theory the R^2 metric can be negative, we elected to clip it to the range $[0, 1]$ for better plot scaling. This was mainly required by the occasional poor performance of $\beta-$VAE on the Object Tracking dataset.

Additionally, on datasets with class structure in addition to a ground truth specific to each class (e.g. Rotated Digits), we partition the embedding according to class labels, fit linear regressions independently on each part to predict the ground truth and report the average R^2 over all partitions.

For circular manifolds (e.g. Teapot, Rotated Digits), we center the manifolds before switching to polar coordinates and use the resulting angles as predictors. Furthermore, we align one angle of the ground truth and the embedded points to mitigate a "spin" in the embedding, which would break the linear relationship. While COIL-100 also has the aforementioned circular structure for each object, the relatively low density of samples for each "ring" (55 in the training set, 10 in the test set) prevented all models from reaching a satisfactory R^2 score using the angles. Thus we chose not to report it.

C.1 Semi-supervised learning ablation study

Here we show the results of a semi-supervised ablation study. We found significant differences in performance between the vanilla network and our approach for networks of different sizes. The results are shown in Figure 10. The geometric regularization in our approach improves the robustness of the network as it achieves fairly consistent accuracy for different numbers of layers. In contrast, the vanilla neural network's performance degrades when more layers are included, suggesting it is more vulnerable to overfitting.

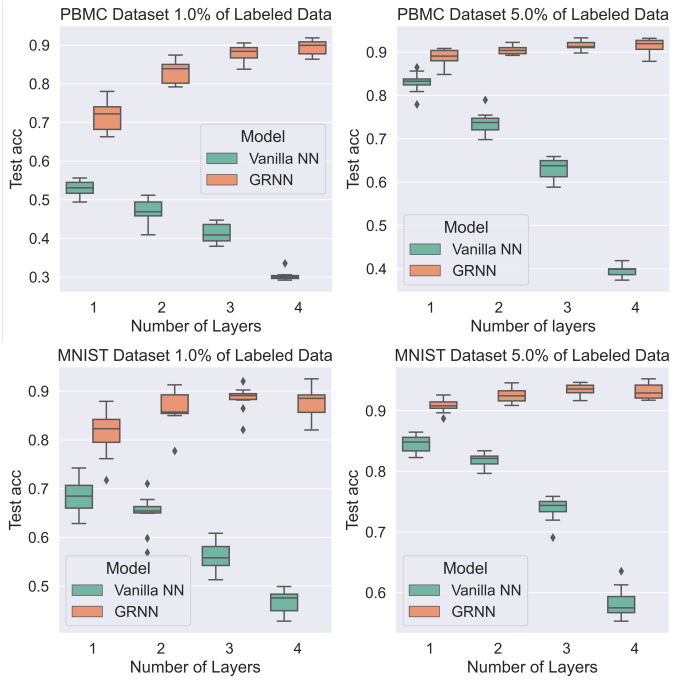

Fig. 10. Ablation study for the number of layers in the semi-supervised architecture. Here we give the test accuracy results for semi-supervised classification in MNIST and PBMC datasets with 1% and 5% of labeled data. We compare a vanilla NN against our regularized version (GRNN) for various numbers of hidden layers in the architecture. Our regularization makes the network resilient to overfitting, and achieves better accuracy in all cases.