A DYNAMICAL SYSTEMS BASED FRAMEWORK FOR DIMENSION REDUCTION

RYEONGKYUNG YOON AND BRAXTON OSTING

Abstract. We propose a novel framework for learning a low-dimensional representation of data based on nonlinear dynamical systems, which we call *dynamical dimension reduction* (DDR). In the DDR model, each point is evolved via a nonlinear flow towards a lower-dimensional subspace; the projection onto the subspace gives the low-dimensional embedding. Training the model involves identifying the nonlinear flow and the subspace. Following the equation discovery method, we represent the vector field that defines the flow using a linear combination of dictionary elements, where each element is a pre-specified linear/nonlinear candidate function. A regularization term for the average total kinetic energy is also introduced and motivated by optimal transport theory. We prove that the resulting optimization problem is well-posed and establish several properties of the DDR method. We also show how the DDR method can be trained using a gradient-based optimization method, where the gradients are computed using the adjoint method from optimal control theory. The DDR method is implemented and compared on synthetic and example datasets to other dimension reductions methods, including PCA, t-SNE, and Umap.

1. Introduction

There has been a growing effort to develop dimension reduction techniques, which find an embedding of high-dimensional data into a meaningful representation space of smaller dimension. Such methods can be applied to a variety of machine learning tasks such as data visualization, outlier detection, and clustering. The most traditional approach is the principal component analysis (PCA) [16], which determines the linear subspace of a fixed dimension that captures the most variance in the data. PCA is a very practical method for extracting characteristic features in massive datasets and has a relatively small computational cost. However, as a linear method, PCA may not perform well in learning complex or nonlinear structures in data. In particular, since PCA equally weights all pairwise distances within the data, it favors preserving global structure over local structure and it can lose local information within a dataset.

To overcome these limitations, a variety of nonlinear methods have been proposed, including t-distributed stochastic neighbor embedding (t-SNE) [19], Uniform manifold approximation and projection (Umap) [20], kernel PCA, spectral embeddings, autoencoders [1, 17]. In particular, an autoencoder learns an *encoder* $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^k$ as well as a *decoder* $\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$ so that the composition $\mathcal{D} \circ \mathcal{E}$ approximates the identity when applied to the data. The encoding step can be viewed as a nonlinear dimension reduction mapping and the reduced-dimension space is referred to as the *latent space*; see section 2.1 for more details.

Our *goal* in this paper will be do develop a dimension reduction method based on nonlinear dynamical systems. The data is evolved via a nonlinear flow towards a lower-dimensional subspace,

Department of Mathematics, University of Utah, Salt Lake City, UT
E-mail addresses: {rkyoon,osting}@math.utah.edu.
Date: April 19, 2022.
2020 *Mathematics Subject Classification.* 34H05 and 68T07.
Key words and phrases. dimension reduction, equation discovery, dynamical systems, adjoint method, optimal transportation.
R. Yoon and B. Osting acknowledge partial support from NSF DMS 17-52202.

the latent space; the projection onto the latent space gives the low-dimensional embedding of the data. Our loss function for training the model is a modification of the loss function for an autoencoder; it penalizes the projection residual for the latent space. In the past few years, there has been significant research on the connections between dynamical systems and (residual) neural networks [8, 9, 12, 14]; we will discuss these related works and describe how these ideas differ from our model in section 2.2.

We begin with a motivating example that helps illustrate (and prompts many questions) how dynamical systems might be used to develop a low-dimensional representation of data.

Motivating example. Suppose we have data $X = [x_1 \mid \cdots \mid x_N] \in \mathbb{R}^{d\times N}$ with $N > d$ with singular value decomposition, $X = U\Sigma V^*$, where the singular values are arranged in decreasing order, *i.e.*, $\sigma_1 \geq \cdots \geq \sigma_d$. Let $U_k \in \mathbb{R}^{d\times k}$ be the first k columns of U and $U_{-k} \in \mathbb{R}^{d\times d-k}$ be the remaining $d-k$ columns of U, *i.e.*, $U = [U_k, U_{-k}]$. The low dimensional representation of this data using PCA would be $\{U_k^* x_i\}_{i\in[N]} \subset \mathbb{R}^k$ with mean squared residual error $\frac{1}{N}\sum_{i\in[N]} \|x_i - U_k U_k^* x_i\|_2^2 = \frac{1}{N}\sum_{j=k+1}^{d} \sigma_j^2(X) = MSE_{PCA}$. Alternatively, we can construct a linear dynamical system that approximately gives this low dimensional representation. Define the matrix $A_\varepsilon \in \mathbb{R}^{d\times d}$ by $A_\varepsilon = \frac{1}{T} U \operatorname{diag}(0,\ldots,0,\log\varepsilon,\ldots,\log\varepsilon) U^*$, where 0 is repeated k times and $\log\varepsilon$ is repeated $d-k$ times. We then consider the initial value problem for each $i \in [N]$,

$$\begin{aligned} \dot{h}_i(t) &= A_\varepsilon h_i(t) \\ h_i(0) &= x_i, \end{aligned}$$

where x_i denotes the i-th column of X. The solution is given by $h_i(t) = e^{A_\varepsilon t} x_i$, $i \in [N]$, so that at time $t = T$, we have

$$h_i(T) = e^{A_\varepsilon T} x_i = U \operatorname{diag}(1,\ldots,1,\varepsilon,\ldots,\varepsilon) U^* x_i = \left(U_k U_k^* + \varepsilon U_{-k} U_{-k}^*\right) x_i$$

Each data point x_i evolves in $\mathbb{R}^d$ towards a low dimensional subspace $h_i(T)$, with

$$\|h_i(T) - U_k U_k^* x_i\|_2^2 = \varepsilon^2 \|U_{-k} U_{-k}^* x_i\|_2^2$$

This implies that

$$\frac{1}{N}\sum_{i\in[N]} \|h_i(T) - U_k U_k^* x_i\|_2^2 = \frac{\varepsilon^2}{N}\sum_{j=k+1}^{d} \sigma_j^2(X) = \varepsilon^2 MSE_{PCA}$$

In other words, the mean squared distance between the solution at time $t = T$ and the best k-dimensional representation of the data in $\mathbb{R}^d$ is $O(\varepsilon^2)$.

Our framework. In this paper, we formulate a method, which we call the *dynamical dimension reduction* (DDR) model, that generalizes the above example in several ways: (i) We allow the right-hand side (RHS) of the dynamical system to be a nonlinear vector field. (ii) We formulate an optimization problem that finds a RHS which evolves the data towards a low dimensional representation. (iii) We also introduce a regularization term in the objective function based on the mean total kinetic energy of the trajectories, which preserves the local and global structure of the data.

For each data point $x_i \in \mathbb{R}^d$, $i \in [N]$, we introduce a time-dependent hidden variable $h_i \in \mathbb{R}^d$ that is governed by the dynamical system

$$\begin{aligned} \frac{dh_i}{dt} &= \Phi(h_i;\beta) \\ h_i(0) &= x_i, \end{aligned}$$

where the vector field, $\Phi(\cdot;\beta)$, is parametrized by β. A description of the parameterization of $\beta \mapsto \Phi(\cdot;\beta)$ using a dictionary of linear and nonlinear terms will be given in section 3. For fixed final time $T > 0$, the low-dimensional embedding $x_i \overset{\mathcal{E}}{\mapsto} y_i$ is defined using the solution at time $t = T$; the data point $x_i = h_i(0)$ is encoded in a lower ($k < d$) dimensional space via $y_i = Qh_i(T)$, where $Q \in \mathbb{R}^{k\times d}$ is a matrix with orthonormal rows. Training the network then involves learning the parameters β and Q. To achieve this goal, we introduce an objective function of the form

$$J(\beta, Q) := \frac{1}{N}\sum_{i\in[N]} \|h_i(T) - Q^*Qh_i(T)\|^2 + \mu R(h_i;\beta)$$

and minimize J over an appropriate set of parameters. The first term in the objective is the mean squared projection residual and encourages the dynamical system to flatten the data as time evolves. The second term is a regularization term that will be used to enforce smoothness on the vector field Φ. In particular, we choose a regularization term of the form

$$R(h_i;\beta) = \int_0^T \|\Phi(h_i;\beta)\|^2 \, dt,$$

which is a measure of the the kinetic energy of the trajectory $h_i(t)$, $t \in [0, T]$. This regularization term can also be interpreted in terms of optimal transport theory and the Wasserstein distance between the data distribution at initial and final times [21, 2]; see further discussion in section 2.4.

Overview of results. In section 3, we formulate the optimization problem in more detail, including the proof of several theoretical results about the DDR method. We prove the existence of a minimizer of the proposed optimization problem (see section 3.3). We also show that the gradient of the objective function $J(\beta, Q)$ with respect to the parameters can be efficiently computed using the adjoint method from optimal control theory (theorem 3.9). We introduce an alternating optimization method, described in section 3.5, that alternatively updates β and Q. We show that the Q-subproblem can be explicitly solved in terms of the singular value decomposition. In section 4, we present a few properties of the DDR model. We prove the stability/generalizability of the embedding $x_i \overset{\mathcal{E}}{\mapsto} y_i$ (theorem 4.1). We also revisit the motivating linear example discussed above and reproduce the result of PCA based on the DDR framework (lemma 4.4).

In section 3.2, we extend the DDR method as a generative model by approximating the decoder $y_i \overset{\mathcal{D}}{\mapsto} x_i$ based on the time-reversal of the learned dynamical system (see also theorem 4.3).

Finally, in section 5, we describe the results of several numerical experiments that examine the performance of the DDR method on a variety of synthetic and example datasets. In these experiments, the DDR method achieves a competitive lower dimensional embedding with respect to other methods; PCA, t-SNE, and Umap. We illustrate that nonlinearity in the vector field of the dynamical system governing the time-evolution of a given data increases the representability/expressibility of the dimension reduction mapping. We also exhibit how stable the encoder is to the noise in the dataset and illustrate the DDR-based generative model.

We conclude in section 6 with a discussion of the DDR method and ideas for several future directions.

2. Background and related work

In this section, we review some related work that motivates the framework of the DDR method: autoencoders, neural ODEs, equation discovery, and optimal transportation.

2.1. **Autoencoders.** An *autoencoder* [1] is comprised of two neural networks: an *encoder* $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^k$ and a *decoder* $\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$. The networks are trained so that the composition, $\mathcal{D} \circ \mathcal{E}$, approximates the identity on the data in terms of the mean residual error, $\frac{1}{N}\sum_{i\in[N]} \|x_i - \mathcal{D}(\mathcal{E}(x_i))\|^2$. Since $k \ll d$, we can interpret an autoencoder passing the data through a bottleneck structure while preserving as much information as possible. The encoder can be viewed as a nonlinear dimension reduction mapping into the latent space, $\mathbb{R}^k$.

However, if the capacity of the model is very large (*i.e.*, there is a large degree of freedom in the autoencoder), it could fail to learn meaningful features in the data manifold and to achieve the generative purpose [11]. To prevent this from happening, there are several ways to regularize an autoencoder, including (i) a penalizing regularity term can be introduced to promote sparsity in the model weights and reduce the sensitivity of the model with respect to given data, or (ii) reinterpreting the model based on variational inference, referred to as variational autoencoders (VAEs) [17]. VAEs estimate a posterior conditional probability of the encoder from a known prior distribution on the latent vector.

2.2. **Neural ODEs.** In [14], the connection between residual neural networks with infinite depth and their continuum limit—a dynamical system—was developed. This idea was extended by [8] and the framework was named Neural ODE (NODE). Here, for an input datapoint $x \in \mathbb{R}^d$, we introduce a time-dependent hidden variable $h(t) \in \mathbb{R}^d$ that is governed by the dynamical system

$$\frac{dh(t)}{dt} = \Phi(h(t), t; \Theta), \tag{1a}$$

$$h(0) = x. \tag{1b}$$

The underlying vector field Φ is represented using a feedforward neural network [8, 14]. Instead of backpropagation, the NODE is trained using the adjoint method from optimal control theory. Recently, the NODE framework has been further developed and extended in a variety of ways, including (i) demonstrating the NODE architecture improves accuracy and stability of the model [7, 8], (ii) generalizing the network by allowing time dependence in the parameters [6], and (iii) modifying the mathematical framework of NODE via statistical process [15] or partial differential equation [18].

Recently, NODE models have also been used to study unsupervised learning problems, particularly density estimation. [8, 9, 12] have developed a novel and easily computed framework for a continuous normalizing flow that minimizes the difference in log densities for the data x and hidden variable h. In particular, [9] introduces a well-conditioned ODE-based model by imposing regularity via optimal transportation theory. However, this framework is not applicable for dimension reduction because the dimension of the latent space should have the same dimension as the data. Motivated by VAEs, [8, 10] proposes time-invariant generative models for time series. However, it doesn't completely rely on the NODE model because the data is encoded by a recurrent neural network (RNN) whereas latent vectors are decoded by NODE.

2.3. **Equation Discovery.** Another method to parameterize a vector field Φ is to use the equation discovery method introduced by [4]. In contrast to NODE, the equation discovery method writes the vector field Φ as a linear combination of dictionary functions,

$$\Phi = \beta\Xi(h(t)),$$

where Ξ consists of pre-specified candidate functions and β is a matrix of coefficients to be determined. Equation discovery has primarily been applied to learn the underlying equations that describe a physical system from measured data. To encourage sparsity on the representation of Φ in these applications, [4] proposes the Sparse Identification of Nonlinear Dynamics (SINDy) method, which uses iterative thresholds least-squares methods. This method was proved to be convergent in [25]. We recently employed equation discovery methods to develop a non-autonomous equation discovery method (NAED) for the time signal classification problem [24].

2.4. **Optimal transportation theory and the Wasserstein metric.** Here, we briefly recall some concepts from optimal transportation theory that help motivate our choice of regularization function. For simplicity, we ignore technical details and refer to [21] for a more rigorous discussion. The squared *2-Wasserstein distance* between probability measures $\mu_0, \mu_T \in \mathcal{P}(\mathbb{R}^d)$ can be written

$$d_W^2(\mu_0, \mu_T) = \inf_{P_\# \mu_0 = \mu_T} \int \|x - P(x)\|^2 d\mu_0(x), \tag{2}$$

Here, $P\colon \mathbb{R}^d \to \mathbb{R}^d$ is a *transportation plan* and the pushfoward constraint ($P_\#\mu_0 = \mu_T$) means that $\mu_T(A) = \mu_0(P^{-1}(A))$ for any set $A \subset \mathbb{R}^n$. This constraint can be interpreted that a transportation plan P rearranges the density corresponding to the measure μ_0 into the density corresponding to measure μ_T. Eq. (2) is known as the *Monge formulation* of the 2-Wasserstein distance.

There is also an equivalent dynamical formulation of the Wasserstein metric due to Benamou and Brenier [2]. Here we think about *continuously* transporting mass from μ_0 to μ_T. We introduce a family of measures $\mu_t \in \mathcal{P}(\mathbb{R}^d)$, $t \in [0, T]$ and abuse notation by also denoting their densities by μ_t, $t \in [0, T]$. The Benamou-Brenier formulation is then to find the time-dependent velocity field $\Phi\colon \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ so that when the density evolves according to the *continuity equation*, the *action* is minimized:

$$d_W^2(\mu_0, \mu_T) = \inf_{\mu_t, \Phi} \quad T \cdot \int_0^T \int \|\Phi(x(t))\|^2 d\mu_t(x) dt \tag{3a}$$

$$\text{s.t.} \quad \frac{\partial \mu_t}{\partial t} + \nabla \cdot (\mu_t \Phi) = 0 \tag{3b}$$

$$\mu_{t=0} = \mu_0, \quad \mu_{t=T} = \mu_T. \tag{3c}$$

In particular, if P is the optimal transportation map in the Monge formulation (2) and we define $P_t = \frac{T-t}{T} I + \frac{t}{T} P$, the optimal solution to (3) is given by $\mu_t = (P_t)_\# \mu_0$.

We now consider two pointsets $\{x_i\}_{i \in [N]} \subset \mathbb{R}^d$ and $\{\tilde{x}_i\}_{i \in [N]} \subset \mathbb{R}^d$ with the same cardinality and their corresponding empirical distributions

$$\mu_0(x) = \frac{1}{N} \sum_{i \in [N]} \delta(x - x_i) \quad \text{and} \quad \mu_T(x) = \frac{1}{N} \sum_{i \in [N]} \delta(x - \tilde{x}_i).$$

In this case, the Benamou-Brenier formulation reduces to finding trajectories $\{x_i(t)\} \subset \mathbb{R}^d$, $i \in [N]$, $t \in [0, T]$ and the time-dependent velocity field $\Phi\colon \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ satisfying

$$d_W^2(\mu_0, \mu_T) = \inf_{\Phi, x_i(t)} \quad \frac{T}{N} \sum_{i \in [N]} \int_0^T \|\Phi(x_i(t))\|^2 dt \tag{4a}$$

$$\text{s.t.} \quad \frac{dx_i}{dt} = \Phi(x_i(t)), \quad i \in [N], \ t \in [0, T] \tag{4b}$$

$$x_i(t = 0) = x_i, \ x_i(t = T) = \tilde{x}_i. \tag{4c}$$

This can be viewed as a *Lagrangian perspective* for the Benamou-Brenier formulation while (3) is the *Eulerian perspective*. If we re-enumerate the points so that $\tilde{x}_i = P(x_i)$, where P is the optimal transportation plan in the Monge formulation, then the optimal trajectories are simply given by

$x_i(t) = \left(1 - \frac{t}{T}\right) x_i + \frac{t}{T}\tilde{x}_i$ and the optimal cost is $d_W^2(\mu_0, \mu_T) = \frac{1}{N}\sum_{i\in[N]} \|x_i - \tilde{x}_i\|^2$. That is, the velocity field with the smallest action simply linearly transports each point from its initial to final position at a constant speed.

3. Dynamical Dimension Reduction

In this section, we formulate our proposed dynamical dimension reduction model (section 3.1), prove the well-posedness of the model (section 3.3), and describe a gradient-based optimization method for training (sections 3.4 to 3.6).

3.1. Dynamical Dimension Reduction Model.

Let the data $x_i \in \mathbb{R}^d$, $\forall i \in [N]$ be given. We propagate the data to a lower dimensional subspace using the solution to a dynamical system, where the solution is initialized at the data and at a fixed terminal time T, the solution lies in (or very near) the low dimensional subspace. To this end, we define hidden variables $h_i \colon [0, T] \to \mathbb{R}^d$ for $i \in [N]$ that describe the trajectories of each data point and satisfy the dynamical system,

$$\frac{d}{dt} h_i(t) = \Phi\left(h_i(t); \beta\right), \quad \forall i \in [N] \tag{5a}$$

$$h_i(0) = x_i. \tag{5b}$$

Here $\beta \in \mathbb{R}^{d\times d_n}$ is matrix used to parameterize the vector field $\Phi \colon \mathbb{R}^d \to \mathbb{R}^d$. The solution to (5) at time T is used to define a low dimensional representation $y_i \in \mathbb{R}^k$ with $k \ll d$ as

$$y_i = Q h_i(T), \qquad \forall i \in [N], \tag{6}$$

where

$$Q \in \mathcal{O}_k := \{Q \in \mathbb{R}^{k\times d} \colon QQ^* = I_k\} \tag{7}$$

is a matrix with orthonormal rows. As described further below, the parameters in this model, $\beta \in \mathbb{R}^{d\times d_n}$ and $Q \in \mathcal{O}_k$, will be optimized (a.k.a. trained) as to obtain a low-dimensional embedding of the data. We will refer to this mapping $\mathcal{E} \colon \mathbb{R}^d \to \mathbb{R}^k$ that assigns $x_i \overset{\mathcal{E}}{\mapsto} y_i$ as the *dynamical dimension reduction (DDR) embedding*.

We have chosen an autonomous vector field, Φ, on the right hand side of (5a) in this work for simplicity, however a non-autonomous vector field could also be used. We represent the vector field Φ using a dictionary of functions, as in the equation discovery method described in section 2.3. In our model, Φ is parameterized by

$$\Phi(h_i; \beta) = \beta \Xi(h_i), \qquad \forall i \in [N], \tag{8}$$

for a pre-specified dictionary $\Xi(h_i) = [\xi_1(h_i), \cdots, \xi_{d_n}(h_i)] \in \mathbb{R}^{d_n}$ that consists of candidate functions $\xi_\ell \colon \mathbb{R}^d \to \mathbb{R}$, for $\ell \in [d_n]$. There is tremendous freedom in the choice of dictionary which, in turn, determines the *representability* or *expressiveness* of our model. A key attribute of our method will be to choose dictionary elements which are *nonlinear*; if only linear dictionary elements are chosen, the DDR embedding residual error can only be as good as the PCA embedding. For example, in the implementation discussed in Section 5, we utilize multivariate polynomials with degree ≤ 3 as dictionary elements. However, with the introduction of nonlinear dictionary elements, we must consider whether, for each data point x_i, there exists a unique solution to the governing ODEs (5) on the time interval $[0, T]$. The following theorem recalls sufficient conditions to guarantee the existence and uniqueness of a solution to (5) depending on a choice of a dictionary. Its proof relies on a standard existence/uniqueness argument in the theory of ordinary differential equations (see, *e.g.*, [22, Theorem 3.2]).

Theorem 3.1. *Let $K \subset \mathbb{R}^d$ be a compact set that contains every data point x_i, $i \in [N]$ and define*

$$r_K := \sup_{y \in K} \|y - x_i\|_2, \qquad \forall i \in [N].$$

Suppose every $\xi_\ell \colon \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous on K with Lipschitz constant $\mathcal{L}$, i.e., for every $h_1, h_2 \in K$,

$$|\xi_\ell(h_1) - \xi_\ell(h_2)| \le \mathcal{L}\|h_1 - h_2\|_2, \qquad \forall \ell \in [d_n].$$

Furthermore, assume $\max_{x \in K} |\xi_\ell(x)| \le M$. Then the initial value problem (5) with the right hand side (8) has the unique solution on interval $[-s, s]$, where $s = \frac{1}{\sqrt{d_n}\|\beta\|_2} \cdot \min\{\frac{r_K}{M}, \frac{1}{\mathcal{L}}\}$.

Note that theorem 3.1 only guarantees the existence/uniqueness of the initial value problem (5) on a time interval $[-s, s]$, whereas, for our method, we require the existence/uniqueness on the time interval $[0, T]$. Note that we could accomplish this by constraining $\|\beta\|_2$ to be sufficiently small. However, this is too restrictive and we alternatively define the set

$$\mathcal{B}_0 := \{\beta \in \mathbb{R}^{d \times d_n} \colon \text{ for each } i \in [N], \text{ the solution to (5) uniquely exists on } [0, T]\}. \tag{9}$$

Note that $\mathcal{B}_0$ contains a ball around the origin (by theorem 3.1) and is star-shaped with respect to the origin. Later, it will be useful (for compactness) to additionally assume that there exists a constant $b > 0$ such that $\|\beta\|_2 \le b$, so we define the subset $\mathcal{B}_1 = \mathcal{B}_1(b)$

$$\mathcal{B}_1 := \{\beta \in \mathcal{B}_0 \colon \|\beta\|_2 \le b\}. \tag{10}$$

We collect the assumptions on the data, dictionary functions, and parameters β, Q in the following.

Assumption 3.2.

(1) The N samples of data $x_i \in \mathbb{R}^d$ lie in the compact set $K \subset \mathbb{R}^d$.
(2) The dictionary Ξ consists of fixed d_n candidate functions $\xi_\ell \colon \mathbb{R}^d \to \mathbb{R}$, which are Lipschitz continuous on K with Lipschitz constant $\mathcal{L}$.
(3) For some fixed (large) $b > 0$, we assume $\beta \in \mathcal{B}_1(b)$, defined in (10).
(4) For fixed $k \ll d$, $Q \in \mathcal{O}_k$, defined in (7)

Loss function. To train the model and obtain the DDR embedding, we introduce the loss function

$$J(\beta, Q) = \frac{1}{N} \sum_{i \in [N]} \underbrace{\|h_i(T) - Q^* Q h_i(T)\|^2}_{i\text{-th sample residual error}} + \mu \cdot \underbrace{R(h_i; \beta)}_{\text{regularization}}. \tag{11}$$

Here, $\beta \in \mathcal{B}_1$ and $Q \in \mathcal{O}_k$ are model parameters and μ is a model hyperparameter that gives a trade-off between the two terms in the objective (11). The first term is seen to be the mean squared residual error; it encourages the solutions to the ODE (5) at time T to lie in a lower dimension subspace. The second term is a regularization term, which we will discuss next.

We introduce a regularization term in the objective (11) since there are many flows $\Phi = \beta\Xi$ which give the same final-time hidden variables $\{h_i(T)\}_{i \in [N]}$. We would like to choose a regularization term so that the resulting vector field $x \mapsto \beta\Xi(x)$ has very regular, smooth trajectories. We choose the regularization function

$$R(h; \beta) = \frac{1}{N} \sum_{i=1}^{N} \int_0^T \|\beta\Xi(h_i(t))\|^2 \, dt. \tag{12}$$

Since we can trivially rewrite $R(h; \beta) = \frac{1}{N} \sum_{i=1}^N \int_0^T \|\dot{h}_i(t)\|^2 dt$, we can interpret $R(h; \beta)$ as the *mean total kinetic energy* of the trajectories. We can also interpret the regularization $R(h; \beta)$ in terms of the Lagrangian perspective for the Benamou-Brenier formulation of the Wasserstein metric (see

section 2.4). Namely, the regularization term $R(h;\beta)$ is the *action* for the vector field which advects the time-parameterized probability measure $\mu_t(x) = \frac{1}{N}\sum_{i=1}^N \delta(x - h_i(t))$, $t \in [0,T]$,

$$R(h;\beta) = \int_0^T \int_{\mathbb{R}^d} \|\beta \Xi(x(t))\|^2 \, d\mu_t(x) \, dt.$$

Thus, as in the definition of the 2-Wasserstein distance, the regularization term penalizes the deviation of the trajectories from the constant-speed linear path between initial and final positions. Similar ideas were used in [9] where the speed up in training is emphasized resulting from better-conditioned ODEs.

Optimization formulation. To train the DDR model, we formulate the ODE-constrained optimization problem,

$$J^\star = \arg\min_{\beta, Q} J(\beta, Q), \quad J(\beta,Q) := \frac{1}{N}\sum_{i\in[N]} \left(\|h_i(T) - Q^* Q h_i(T)\|^2 + \mu \int_0^T \|\beta \Xi(h_i(t))\|^2 \, dt \right) \tag{13a}$$

$$\text{s.t. } \beta \in \mathcal{B}_1(b), \quad Q \in \mathcal{O}_k \tag{13b}$$

$$\frac{d}{dt} h_i(t) = \beta \Xi(h_i(t)), \quad h_i(0) = x_i, \tag{13c}$$

We will show that this ODE-constrained optimization problem is well-posed (section 3.3) and derive a gradient-based optimization method for solving it in section 3.4.

Remark 3.3. *It is be useful to consider the problem when $\beta = 0$. In this case, the ODE (13c) is trivial and $h_i(T) = h_i(0) = x_i$. We obtain*

$$J(\beta = 0, Q) = \frac{1}{N}\|X - Q^* Q X\|_F^2,$$

where $X = [x_1 \mid \cdots \mid x_N] \in \mathbb{R}^{d\times N}$. Assuming the singular value decomposition, $X = U\Sigma V^*$, where the singular values are arranged in decreasing order, i.e., $\sigma_1 \geq \cdots \geq \sigma_d$, the Eckart–Young theorem gives that for all $Q \in \mathcal{O}_k$,

$$J(\beta = 0, Q) \geq \sum_{i=k+1}^{d} \sigma_i^2(X) \equiv J_0,$$

with equality attained by $Q = U_k^*$ where $U_k \in \mathbb{R}^{d\times k}$ are the first k columns of U. It follows that $J^\star \leq J_0$. We interpret this computation as follows. By allowing β to vary, the method finds a nonlinear transformation $x_i \mapsto h_i(T)$ so that the PCA of the transformed data has a smaller objective value than the original PCA objective value.

3.2. **A DDR-based generative model.** An interesting property of the DDR framework is that an (approximate) decoding can be obtained by the time-reversal of the learned dynamical system and thus, the model can be extended as a generative model [17]. More precisely, the encoder $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^k$ can be written $y = \mathcal{E}(x) = Qh(T)$, where $h(T)$ is the the solution to (5) at time $t = T$ with initial condition x. *Assuming zero residual training error*, we have that $h(T) = Q^* y$. In this case, the decoder, $\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$, is exactly obtained by solving the time-reversed dynamical system,

$$\frac{d}{dt} h(t) = \beta \Xi(h(t)) \tag{14a}$$

$$h(T) = Q^* y \tag{14b}$$

backwards in time from $t = T$ to $t = 0$ and setting $\mathcal{D}(y) = h(0)$. In the case of zero training error, we have $\mathcal{D} \circ \mathcal{E}(x_i) = x_i$ for all training data $x_i \in \mathbb{R}^d$. In general, we thus define the *decoder*

$\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$ to be $\mathcal{D}(y) = h(0)$, where $h(t)$, $t \in [0,T]$ satisfies (14) with final condition given by $h(T) = Q^* y$.

Moreover, if the data has distribution ρ_0, the low-dimensional representation has distribution $\rho_T = (\mathcal{E})_{\#}\rho_0$. Thus, using a density estimate ν of ρ_T (*e.g.*, kernel density estimate), the decoder could be used to generate new data via $(\mathcal{D})_{\#}\nu$. Finally, if we assume that the distribution ρ_0 is supported on a low-dimensional manifold embedded in $\mathbb{R}^d$, then $\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$ is a parameterization of the manifold, so the DDR framework can be used in the context of manifold learning.

3.3. **Existence of a minimizer.** In this section we show that the constrained optimization problem in (13) is well-defined in the setting of assumption 3.2. We will employ the direct method in the calculus of variations.

The following Lemma shows that for $\beta \in \mathcal{B}_0$, if the kinetic energy of the solution is bounded then so is the solution.

Lemma 3.4. *Let $\beta \in \mathcal{B}_0$. If the solution $h(t)$ to the Cauchy problem*

$$\begin{aligned} \dot{h}(t) &= \beta \Xi(h(t)) \\ h(0) &= x. \end{aligned}$$

satisfies $\int_0^T \|\dot{h}(\tau)\|^2 d\tau \le C$ for some constant C, then there exists an $H > 0$ such that

$$\|h(t)\| \le H, \qquad \forall t \in [0,T].$$

Proof. We compute

$$C \ge \int_0^T \|\dot{h}(\tau)\|^2 d\tau \ge \int_0^t \|\dot{h}(\tau)\|^2 d\tau \ge \left\| \int_0^t \dot{h}(\tau) d\tau \right\|^2 = \|h(t) - h(0)\|^2$$

This implies that $\|h(t)\| - \|h(0)\| \le \|h(t) - h(0)\| \le \sqrt{C}$ so that $\|h(t)\| \le |K| + \sqrt{C} =: H$. □

Recalling remark 3.3, we need only consider (β, Q) such that $J(\beta, Q) \le J_0$. Observing that

$$J(\beta, Q) \ge \frac{\mu}{N} \sum_{i \in [N]} \int_0^T \|\dot{h}_i(\tau)\|^2 \, d\tau,$$

lemma 3.4 shows that there exists an $H > 0$ such that we need only consider (β, Q) such that the corresponding hidden solutions, $h_i(t)$ for $i \in [N]$, are bounded by H, *i.e.*, $\|h_i(t)\| \le H$. For this fixed $H > 0$, we define $\mathcal{B}_2 = \mathcal{B}_2(H)$ by

$$\mathcal{B}_2 := \{\beta \in \mathcal{B}_1(b) \colon \text{ the solutions } h_i(t),\ i \in [N] \text{ to (5) are bounded with } \|h_i(t)\| \le H,\ t \in [0,T]\}.$$

We next prove the Lipschitz continuity of the solution h to (5) at time t with respect to the parameter β.

Lemma 3.5. *For β and $\tilde{\beta}$ in $\mathcal{B}_2$, denote the solutions to (5) as h and $\tilde{h}$ respectively. Then we have*

$$\|h(t) - \tilde{h}(t)\| \le C \|\beta - \tilde{\beta}\|_2, \qquad \forall t \in [0,T] \tag{15}$$

for some constant $C > 0$.

Proof. Since h and $\tilde{h}$ are solution to the ODEs (13c), we have

$$\frac{d}{dt}\left(h - \tilde{h}\right)(t) = \beta \Xi(h) - \tilde{\beta} \Xi(\tilde{h}),$$

with the initial condition $h(0) - \tilde{h}(0) = 0$. Then we estimate

$$\begin{aligned}\|h(t) - \tilde{h}(t)\| &\leq \int_0^t \|\beta\Xi(h(\tau)) - \tilde{\beta}\Xi(\tilde{h}(\tau))\| d\tau \\ &\leq \int_0^t \|\beta - \tilde{\beta}\|_2 \|\Xi(h(\tau)\| \ d\tau + \int_0^t \|\tilde{\beta}\|_2 \|\Xi(h(\tau)) - \Xi(\tilde{h}(\tau))\| \ d\tau \\ &\leq \int_0^t \|\beta - \tilde{\beta}\|_2 \|\Xi(h(\tau)\| \ d\tau + \int_0^t \sqrt{d_n}\mathcal{L}M\|h(\tau) - \tilde{h}(\tau)\| \ d\tau.\end{aligned}$$

By Gronwall's inequality, $\|h(t) - \tilde{h}(t)\| \leq \|\beta - \tilde{\beta}\|_2 \int_0^t \|\Xi(h(\tau))\| \ d\tau e^{\sqrt{d_n}\mathcal{L}Mt}$. Since $\beta \in \mathcal{B}_2$, $h(t) \leq H$, which implies that the norm of the finite number of dictionary terms at $h(t)$ is also bounded, *i.e.*, $\|\Xi(h(t))\| \leq \tilde{H}$, for some constant $\tilde{H} > 0$ over interval $t \in [0, T]$. Therefore, (15) holds with $C = T\tilde{H}e^{\sqrt{d_n}\mathcal{L}MT}$. □

Using lemma 3.5, we prove the following theorems establishing continuity and compactness of the objective function over $\mathcal{B}_2 \times \mathcal{O}_k$.

Theorem 3.6. *The objective function J is 2−Hölder continuous over $\mathcal{B}_2$ and $\mathcal{O}_k$ respectively, i.e. for all $\beta, \tilde{\beta} \in \mathcal{B}_2$ and $Q, P \in \mathcal{O}_k$,*

$$|J(\beta, Q) - J(\tilde{\beta}, P)| \leq C_1\|Q - P\|^2 + C_2\|\beta - \tilde{\beta}\|^2, \tag{16}$$

for some positive constants C_1 and C_2.

Proof. For any $Q, P \in \mathcal{O}_k$,

$$\|Q^*Q - P^*P\| \leq \|Q^*Q - Q^*P\| + \|Q^*P - P^*P\| \leq (\|Q\| + \|P\|)\|Q - P\| = 2\|Q - P\|, \tag{17}$$

where orthonormality of Q implies $\|Q\| = \sqrt{\lambda_{max}(QQ^*)} = 1$. Then for fixed $\beta \in \mathcal{B}_2$,

$$\begin{aligned}|J(\beta, Q) - J(\beta, P)| &= \frac{1}{N}\sum_{i\in[N]} \left|\|(I - Q^*Q)h_i(T)\|^2 - \|(I - P^*P)h_i(T)\|^2\right| \\ &\leq \frac{1}{N}\sum_{i\in[N]} \|(I - Q^*Q)h_i(T) - (I - P^*P)h_i(T)\|^2 \leq \frac{1}{N}\sum_{i\in[N]} \|h_i(T)\|^2\|Q^*Q - P^*P\|^2 \\ &\leq C_1\|Q - P\|^2,\end{aligned}$$

where $C_1 = 2H^2$ and the last inequality is obtained by (17). Therefore, the objective is Hölder continuous w.r.t Q. Moreover, for $\beta, \tilde{\beta} \in \mathcal{B}_2$,

$$\begin{aligned}|J(\beta, P) - J(\tilde{\beta}, P)| &\leq \frac{1}{N}\sum_{i\in[N]} \left|\|(I - Q^*Q)h_i(T)\|^2 - \|(I - Q^*Q)\tilde{h}_i(T)\|^2\right| + \mu\int_0^T \left|\|\beta\Xi(h_i)\|^2 - \|\tilde{\beta}\Xi(\tilde{h_i})\|^2\right| \ d\tau \\ &\leq \frac{1}{N}\sum_{i\in[N]} \|(I - Q^*Q)(h_i(T) - \tilde{h}_i(T))\|^2 + \mu\int_0^T \|\beta\Xi(h_i) - \tilde{\beta}\Xi(\tilde{h_i})\|^2 \ d\tau \\ &\leq \frac{1}{N}\sum_{i\in[N]} \|I - Q^*Q\|^2\|h_i - \tilde{h}_i\|^2 + \mu\int_0^T \|\beta - \tilde{\beta}\|^2\|\Xi(h_i(\tau)\|^2 + \|\tilde{\beta}\|^2\|\Xi(h_i) - \Xi(\tilde{h}_i)\|^2 \ d\tau \\ &\leq \frac{1}{N}\sum_{i\in[N]} \|h_i - \tilde{h}_i\|^2 + \mu\int_0^T \|\beta - \tilde{\beta}\|^2\tilde{H}^2 + d_n\mathcal{L}^2M^2\|h_i - \tilde{h}_i\|^2 \ d\tau \\ &\leq (1 + \mu T d_n\mathcal{L}^2M^2)\|h - \tilde{h}\|^2 + \mu T\tilde{H}^2\|\beta - \tilde{\beta}\|^2.\end{aligned}$$

Using the inequality in lemma 3.5,

$$|J(\beta, P) - J(\tilde{\beta}, P)| \leq C_2\|\beta - \tilde{\beta}\|^2,$$

where $C_2 = C^2 + \mu T(\tilde{H^2} + d_n \mathcal{L}^2 C^2 M^2)$. Therefore, we have

$$|J(\beta, Q) - J(\tilde{\beta}, P)| \le |J(\beta, Q) - J(\beta, P)| + |J(\beta, P) - J(\tilde{\beta}, P)| \le C_1 \|Q - P\|^2 + C_2 \|\beta - \tilde{\beta}\|^2.$$

□

Theorem 3.7. *The feasible set $\mathcal{B}_2 \times \mathcal{O}_k$ is compact.*

Proof. Since $Q \in \mathcal{O}_k \subset \mathbb{R}^{k\times d}$ has orthonormal rows, $\|Q\|_2 \le 1$ and $\mathcal{O}_k$ is bounded. To show $\mathcal{O}_k$ is closed, define a mapping $f\colon \mathbb{R}^{k\times d} \mapsto \mathbb{R}^{k\times k}$ such that $f\colon A \mapsto AA^*$. Arguing as in (17), for $Q, P \in \mathcal{O}_k$, we have

$$\|f(Q) - f(P)\| = \|QQ^* - PP^*\| \le 2\|Q - P\|,$$

so f is continuous. Since $\mathcal{O}_k = f^{-1}(\{I_k\})$ and the singleton is closed, $\mathcal{O}_k$ is also closed.

Following the definition of $\mathcal{B}_2$, it is bounded. Suppose a sequence $\{\beta_j\}_{j\in\mathbb{N}} \subset \mathcal{B}_2$ converges to $\tilde{\beta}$. Then we can define a sequence $\{h_j\}_{j\in\mathbb{N}}$ of solution to (5) corresponding to $\{\beta_j\}_{j\in\mathbb{N}}$, which is equivalent to $h_j\colon [0,T] \mapsto \mathbb{R}^d$ solves the integral equation

$$h_j(t) = x + \int_0^t \beta_j \Xi(h_j(\tau))\, d\tau, \qquad \forall t \in [0,T]. \tag{18}$$

By the definition of $\beta_j \in \mathcal{B}_2$, a sequence $\{h_j\}_{j\in\mathbb{N}}$ is uniformly bounded, that is,

$$\|h_j(t)\| \le H, \qquad \forall j \in \mathbb{N}, t \in [0,T].$$

Moreover, for arbitrary $t, s \in [0,T]$, we have

$$\|h_j(t) - h_j(s)\| \le \int_s^t \|\beta_j \Xi(h_j(\tau))\|\, d\tau \le \int_s^t \|\beta_j\|_2 \|\Xi(h_j(\tau))\|\, d\tau \le M\tilde{H}|t - s|, \quad \forall j \in \mathbb{N}.$$

Hence a sequence $\{h_j\}_{j\in\mathbb{N}}$ is uniformly equicontinuous. By the Arzela-Ascoli theorem, there exists subsequence $\{h_j\}_{j\in\mathbb{N}}$, denoted with same index, such that it converges uniformly, say $h_j \to \tilde{h}$. By the continuity of dictionary Ξ,

$$\tilde{h}(t) = \lim_{j\to\infty} h_j(t) = x + \int_0^t \left(\lim_{j\to\infty} \beta_j\right)\left(\lim_{j\to\infty} \Xi(h_j(\tau))\right) d\tau = x + \int_0^t \tilde{\beta}\Xi(\tilde{h}(\tau))\, d\tau.$$

Hence, the ODE (5) with $\tilde{\beta}$ is uniquely solved in $[0,T]$. Also, by the continuity of the norm,

$$\|\tilde{\beta}\| = \|\lim_{j\to\infty} \beta_j\| = \lim_{j\to\infty} \|\beta_j\| \le M, \quad \text{and} \quad \|\tilde{h}(t)\| = \|\lim_{j\to\infty} h_j(t)\| = \lim_{j\to\infty} \|h_j(t)\| \le H.$$

Therefore, $\tilde{\beta} \in \mathcal{B}_2$ and thus $\mathcal{B}_2$ is compact. □

Finally, we use theorem 3.6 and theorem 3.7 to prove the following result that the constrained minimization problem (13) is well-defined.

Theorem 3.8. *There exists a $(\beta_\star, Q_\star) \in \mathcal{B}_2 \times \mathcal{O}_k$ that attains the infimum value*

$$\inf_{(\beta,Q)\in\mathcal{B}_1\times\mathcal{O}_k} J(\beta, Q),$$

where the ODE constraints (13c) are implicit.

Proof. We argue via the direct method in the Calculus of Variations. We know $J(\beta, Q) \ge 0$ for all $(\beta, Q) \in \mathcal{B}_1 \times \mathcal{O}_k$. We take a minimizing sequence $(\beta_j, Q_j) \subset \mathcal{B}_1 \times \mathcal{O}_k$. Since this is a minimizing sequence, by lemma 3.4, we know that there exists a constant $H > 0$, such that $(\beta_j, Q_j) \subset \mathcal{B}_2(H) \times \mathcal{O}_k$. By compactness (theorem 3.7), we can extract a convergent subsequence, which we again index (β_j, Q_j), such that $\lim_{j\to\infty}(\beta_j, Q_j) = (\beta_\star, Q_\star)$. Now using the continuity of J (theorem 3.6), we have

$$J^\star = \inf_{(\beta,Q)\in\mathcal{B}_2\times\mathcal{O}_k} J(\beta, Q) = \lim_{j\to\infty} J(\beta_j, Q_j) = J(\beta_\star, Q_\star).$$

So, $(\beta_\star, Q_\star) \in \mathcal{B}_2 \times \mathcal{O}_k$ attains the infimum value. $\square$

Although theorem 3.8 gives the existence of a solution, we do not necessarily have a unique solution. Of course, this is also the case for PCA if the singular values have a multiplicity greater than one.

We also remark that our method is not identifiable. We illustrate this in section 5.2, where we train our model for a synthetic dataset which is formed using a known vector field and orthonormal subspace. We find that the learned parameters can differ from the ground truth.

3.4. **Gradient computations.** We use a gradient-based optimization method to solve (13) and learn the parameters for the DDR model. To compute the gradient of the loss with respect to each parameter, we apply the adjoint method.

Theorem 3.9. *The gradients of the objective function (13a) with respect to the parameters $\beta \in \mathcal{B}$ and $Q \in \mathcal{Q}_k$ are given by*

$$d_\beta J = \frac{1}{N} \sum_{i\in[N]} \int_0^T \left[\frac{2\mu}{T} \beta \Xi(h_i) \Xi(h_i)^* - \lambda_i \Xi(h_i)^* \right] dt \tag{19a}$$

$$d_Q J = \frac{1}{N} \sum_{i\in[N]} 2Qh_i(T)h_i(T)^* Q^* Q - 2Qh_i(T)h_i(T)^*, \tag{19b}$$

where $\lambda_i(t) : [0,T] \to \mathbb{R}^d$ is a solution to the adjoint equation, for all $i \in [N]$

$$\frac{d\lambda_i}{dt} = -\nabla \Xi(h_i)^* \beta^* \lambda_i + \frac{2\mu}{T} \nabla \Xi(h_i)^* \beta^* \beta \Xi(h_i) \tag{20a}$$

$$\lambda_i(T) = -2(I_d - Q^*Q)h_i(T). \tag{20b}$$

Proof. Let the Lagrangian multipliers $\lambda_i(t)[0,T] \to \mathbb{R}^d$ be given. Then the Lagrangian is defined as

$$\mathcal{L}(\lambda_i) = \frac{1}{N} \sum_{i\in[N]} \int_0^T \left(\frac{\mu}{T} |\beta \Xi(h_i)|^2 + \lambda_i^* \dot{h_i} - \lambda_i^* \beta \Xi(h_i) \right) dt + \|h_i(T) - Q^*Qh_i(T)\|^2.$$

Using the integration by parts, $\int_0^T \lambda_i^* \dot{h_i} dt = [\lambda_i^* h_i]_0^T - \int_0^T \dot{\lambda_i}^* h_i dt$, the Lagrangian can be rewritten as

$$\begin{aligned}\mathcal{L}(\lambda_i) = \frac{1}{N} \sum_{i\in[N]} \int_0^T \left(\frac{\mu}{T} |\beta \Xi(h_i)|^2 - \dot{\lambda_i}^* h_i - \lambda_i^* \beta \Xi(h_i) \right) dt \\ + \|h_i(T) - Q^*Qh_i(T)\|^2 + \lambda_i(T)^* h_i(T) - \lambda_i(0)^* h_i(0).\end{aligned}$$

Taking the total derivative of $\mathcal{L}$ w.r.t β, Q, we obtain

$$\begin{aligned}d_\beta \mathcal{L} = \frac{1}{N} \sum_{i\in[N]} \int_0^T \left[\left(\frac{2\mu}{T} \Xi^* \beta^* \beta \Xi - \lambda_i^* \beta \nabla \Xi - \dot{\lambda_i}^* \right) d_\beta h_i + \frac{2\mu}{T} \beta \Xi \Xi^* - \lambda_i^* \Xi \right] dt \\ + d_\beta h_i(T)(\lambda_i(T)^* + 2(I_d - Q^*Q)h_i(T)), \\ d_Q \mathcal{L} = \frac{1}{N} \sum_{i\in[N]} \int_0^T \left(\frac{2\mu}{T} \Xi^* \beta^* \beta \Xi - \lambda_i^* \beta \nabla \Xi - \dot{\lambda_i}^* \right) d_Q h_i \, dt \\ + 2Q(Q^*Qh_i(T)h_i(T)^* + h_i(T)h_i(T)^*Q^*Q - 2h_i(T)h_i(T)^*) \\ + d_Q h_i(T)(\lambda_i(T)^* + 2(I_d - Q^*Q)h_i(T)).\end{aligned}$$

Since $d_\beta h_i$ and $d_Q h_i$ are expensive to compute, we solve the adjoint equation alternatively. By setting $d_\beta \mathcal{L}$ and $d_Q \mathcal{L}$ to be zero, we derive the adjoint equations (20) and the gradients of the objective with respect to β and Q are then formulated as (19). □

3.5. **Solution to the Q-subproblem.** Let $H_T = [h_1(T) \mid \cdots \mid h_N(T)] \in \mathbb{R}^{d\times N}$ be a matrix of hidden variables $h_i(T)$ at the final time T. To seek the optimal Q^* of the problem (13), we consider the *Q-subproblem* minimizing a mean squared residual error

$$Q^\star = \arg\min_{Q\in\mathcal{O}_k} \frac{1}{N}\sum_{i\in[N]} \|h_i(T) - Q^*Qh_i(T)\|^2 = \arg\min_{Q\in\mathcal{O}_k} \frac{1}{N}\|(I - Q^*Q)H_T\|_F^2. \tag{21}$$

The following lemma provides the solution to the Q-subproblem.

Lemma 3.10. *For $Q \in \mathbb{R}^{k\times d}$ with orthonormal rows and $H_T \in \mathbb{R}^{d\times N}$ with SVD $H_T = U\Sigma V^*$, we have that*

$$\|(I - Q^*Q)H_T\|_F^2 \geq \sum_{i=k+1}^{d} \sigma_i^2(H_T)$$

with equality attained by $Q = U_k^*$, where U_k are the first k columns of U, corresponding to the largest singular values, $\{\sigma_i\}_{i=1}^k$. Thus, the solution to the Q-subproblem is explicitly given by $Q^\star = U_k^*$.

Proof. First note that since Q^*Q is a projection matrix,

$$\begin{aligned}\|(I - Q^*Q)H_T\|_F^2 &= \|H_T\|_F^2 - \|Q^*QH_T\|_F^2 \\ &= \sum_{i=1}^{d} \sigma_i^2(H_T) - \langle Q^*Q, H_T H_T^*\rangle\end{aligned}$$

Fan's inequality states that for any symmetric matrices X and Y, we have that $\langle X, Y\rangle \leq \lambda(X)\lambda(Y)$, where λ denotes the eigenvalues listed in non-increasing order. Furthermore, equality holds if and only if X and Y have a simultaneous ordered spectral decomposition [3]. Since the eigenvalues of the projection matrix Q^*Q are $\lambda = 1$ with multiplicity k and $\lambda = 0$ with multiplicity $d-k$, we have that

$$\langle Q^*Q, H_T H_T^*\rangle \leq \sum_{i=1}^{k} \lambda_i(H_T H_T^*) = \sum_{i=1}^{k} \sigma_i^2(H_T)$$

with equality if and only if there exists an orthogonal $\tilde{U}$ such that $Q^*Q = \tilde{U}\mathrm{diag}\lambda(Q^*Q)\tilde{U}^*$ and $H_T H_T^* = \tilde{U}\mathrm{diag}\lambda(H_T H_T^*)\tilde{U}^*$. Clearly, we can pick $\tilde{U} = U$ and $Q^*Q = U_k U_k^*$. □

3.6. **An algorithm for the solution of the DDR model.** There are several different approaches to solve the optimization problem for DDR model (13). One approach would be a projected gradient-based method. Here, the gradients in theorem 3.9 would be use to take a gradient-based step (*e.g.*, a stochastic gradient descent step) and then the updated Q would be projected onto the constraint set $\mathcal{O}_k$. Instead, we use an alternating method, summarized in algorithm 1, which uses the exact solution for the Q-subproblem (see section 3.5).

Algorithm 1 Dynamical Dimension Reduction

Input: initial parameters, β, Q.
for epoch $= 1, \ldots, N_{epoch}$: **do**
  Shuffle data and create batches of size N_{batch}
  for each batch: **do**
    (Solve the forward ODE for h_i) For the current parameters β, solve the forward ODE (5), *i.e.*, for each example $i \in [N_{batch}]$ and discrete times t_m, $m \in [T_m]$, find $h_i(t_m)$.

    (Solve Q-subproblem) Using the hidden state at the final time $h_i(T)$, solve the Q-*subproblem* in (21) *i.e.* $H_T = U\Sigma V^*$ and update $Q \leftarrow U_k^*$.

    (Solve the adjoint equation for λ_i) Using updated Q and $h_i(t_m)$, compute the terminal condition and solve the backward ODE in (20) *i.e.*, for each example $i \in [N_{batch}]$ and discrete times t_m, $m \in [T_m]$, find $\lambda_i(t_m)$.

    (Compute gradients) Using $h_i(t_m)$ and $\lambda_i(t_m)$, evaluate the gradient of the objective function with respect to the parameters $\nabla_\beta J$ as in (19).

    (Update β) Use a gradient-based optimization method, *e.g.*, gradient descent or ADAM method, to update the parameters, β.

  end for
end for

4. Properties of the Dynamical Dimension Reduction Model

Here we present a few properties of the DDR model. In section 4.1 we describe stability/generalizability of the forward model. In section 4.2 we describe the reduction to PCA for a linear dictionary.

4.1. Stability/generalizability of the forward model.

In this section, we prove that the dynamical dimension embedding $x_i \overset{\mathcal{E}}{\mapsto} y_i$ is stable under the perturbation in given data. Denote the optimal parameter of (13) by $(\beta^\star, Q^\star)$ and $h_i(T)$ the solution to (5) with parameters $\beta^\star$, so that $y_i = \mathcal{E}(x_i) = Q^\star h_i(T)$.

Theorem 4.1. *Consider the dimension reduction embedding $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^k$ with dictionary Ξ satisfying assumption 3.2. Then mapping $\mathcal{E}$ is Lipshcitz continuous, i.e. for $x_1, x_2 \in \mathbb{R}^d$,*

$$\|\mathcal{E}(x_1) - \mathcal{E}(x_2)\| \le C\|x_1 - x_2\|,$$

$C > 0$ is a constant described in the proof.

Proof. Consider ODEs of hidden variable h_1 and h_2 with unperturbed and perturbed initial condition respectively,

$$\begin{cases} \dot{h_1} = \beta\Xi(h_1) \\ h_1(0) = x_1 \end{cases} \quad \text{and} \quad \begin{cases} \dot{h_2} = \beta\Xi(h_2) \\ h_2(0) = x_2. \end{cases}$$

By subtracting these equations, we estimate

$$\|h_1(t)-h_2(t)\| \le \|x_1-x_2\| + \int_0^t \|\beta\|\|\Xi(h_1(\tau))-\Xi(h_2(\tau))\|\, d\tau \le \|x_1-x_2\| + \int_0^t M\mathcal{L}\sqrt{d_n}\|h_1(\tau)-h_2(\tau)\|\, d\tau.$$

Gronwall's inequality yields

$$\|h_1(t) - h_2(t)\| \le \|x_1 - x_2\| e^{tM\mathcal{L}\sqrt{d_n}} \le \tilde{C}\|x_1 - x_2\|,$$

where $\tilde{C} = e^{TML\sqrt{d_n}}$ is a constant. Then low dimensional representation of each data from the mapping provides

$$\|\mathcal{E}(x_1) - \mathcal{E}(x_2)\| = \|Qh_1(T) - Qh_2(T)\| \leq \tilde{C}\|Q\|\|x_1 - x_2\|,$$

as desired. □

The theorem 4.1 can be interpreted as the generalizability of our model. Suppose new data x_η is in the η-ball of the original data x used for the training mapping. Then output $\mathcal{E}(x_\eta)$ of embedding doesn't move far from $\mathcal{E}(x)$. Thus we could obtain a reliable lower dimensional representation of new data without retraining the model. Later, we will illustrate the stability of our embedding model under the noise in a given data through numerical experiments in section 5.2.

Theorem 4.2. *Suppose the residual training error is zero. Suppose x_1, x_2 are two points in the training dataset and $y_1 = \mathcal{E}(x_1)$ and $y_2 = \mathcal{E}(x_2)$ are the embedded points. Then*

$$\|y_1 - y_2\| \geq C^{-1}\|x_1 - x_2\|,$$

where $C > 0$ is the same constant as in theorem 4.1. In particular, this implies that the embedding $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^k$ is a quasi-isometry on the training data, i.e., it satisfies

$$C^{-1}\|x_1 - x_2\| \leq \|\mathcal{E}(x_1) - \mathcal{E}(x_2)\| \leq C\|x_1 - x_2\|.$$

Proof. Since there is zero training error, we have $h_i(T) = Q^*Qh_i(T)$. So,

$$\|y_1 - y_2\| = \|Qh_1(T) - Qh_2(T)\| = \|Q^*Qh_1(T) - Q^*Qh_2(T)\| = \|h_1(T) - h_2(T)\|.$$

But now using Gronwall's inequality for the time-reversed dynamical system, we obtain

$$\|x_1 - x_2\| \leq C\|h_1(T) - h_2(T)\| = C\|y_1 - y_2\|,$$

which proves the first claim.

The second claim now follows from theorem 4.1. □

The next theorem gives a result of for the decoder, discussed in section 3.2.

Theorem 4.3. *Suppose the residual training error is zero and let $\{y_i\}_{i\in[N]} \subset \mathbb{R}^k$ be the DDR embedded points. There exists an open set $Y \supset \{y_i\}_{i\in[N]}$ such that for every $y \in Y$, $\mathcal{D}(y)$ is finite.*

Proof. Fix $i \in [N]$. There exists a neighborhood $Y_i \ni y_i$ such that for every $y \in Y_i$ there exists a unique solution to (14) at time $t = 0$ with final condition given by $h(T) = Q^*y$. We simply take $Y = \cup_{i\in[N]} Y_i$. □

4.2. **Connections with principal component analysis.** In this section, we consider again the motivating PCA-based example described in section 1. Suppose we have data $X = [x_1 \mid \cdots \mid x_N] \in \mathbb{R}^{d\times N}$ with $N > d$ with singular value decomposition, $X = U\Sigma V^*$, where the singular values are arranged in decreasing order, *i.e.*, $\sigma_1 \geq \cdots \geq \sigma_d$. We consider the application of the DDR method with a *linear dictionary. i.e.* $d_n = d$ and $\Xi(h) = [h_1, \ldots, h_d]$. We consider

$$\beta = A_\varepsilon = \frac{1}{T}U\operatorname{diag}[0, \ldots, 0, \log\varepsilon, \ldots, \log\varepsilon]U^* \quad \text{and} \quad Q = U_k^*. \tag{22}$$

where $\varepsilon > 0$ and $U_k \in \mathbb{R}^{d\times k}$ are the first k columns of U. The following lemma shows that the proposed solution (22) is a stationary point for (13) for a particular choice of ε.

Lemma 4.4. *For fixed $\mu > 0$, we consider the DDR method with a linear dictionary with the notation introduced above. We consider $(\beta, Q) = (A_\varepsilon, U_k^*)$ given in (22). For any $\varepsilon > 0$, $d_Q J(A_\varepsilon, U_k^*) = 0$. For $\mu > 0$, there exists a unique $\varepsilon^\star = \varepsilon^\star(\mu) \in [e^{-1}, 1)$, such that $d_\beta J(A_{\varepsilon^\star}, U_k^*) = 0$. In particular, $(\beta, Q) = (A_{\varepsilon^\star}, U_k^*)$ is a stationary point for (13).*

Proof. For this choice of (β, Q), the solution to ODE in (13c) is simply

$$h_i(t) = UD(1, \varepsilon^{\frac{t}{T}})U^* x_i.$$

where we use the notaiton $D(a,b) = \text{diag}(a,\dots,a,b,\dots,b)$ where the a is repeated k times and b is repeated $d-k$ times. We evaluate $d_Q J$ in (19b) at $Q = U_k^*$ to obtain $d_Q J = 0$.

The adjoint equation (20) is then written

$$\begin{aligned} \frac{d\lambda_i}{dt} &= -\frac{1}{T} UD(0, \log\varepsilon)U^*\lambda_i + \frac{2\mu}{T^3} UD\left(0, \varepsilon^{\frac{t}{T}} \log^2 \varepsilon\right) U^* x_i \\ \lambda_i(T) &= -2\varepsilon UD(0,1)U^* x_i. \end{aligned}$$

The solution to the adjoint equation, which can be derived using variation of parameters, is given by

$$\lambda_i(t) = -\left[2\varepsilon^{2-t/T} + \frac{\mu\varepsilon\log\varepsilon}{T^2}\left(\varepsilon^{1-t/T} - \varepsilon^{-(1-t/T)}\right)\right] UD\,(0,1)\,U^* x_i.$$

We now compute the derivative of the objective function with respect to β using (19a) and the explict solutions for $h_i(t)$ and $\lambda_i(t)$ derived above. We obtain

$$d_\beta J = f(\varepsilon,\mu) U_{-k} D(0,1) \Sigma U_{-k}^*, \qquad \text{where } \; f(\varepsilon,\mu) := \frac{T}{N}\left[-\frac{\mu}{2T^2} - \left(-\frac{2\varepsilon^2\log\varepsilon}{T^2}\right) + \varepsilon^2\left(2 + \frac{\mu}{2T^2}\right)\right].$$

This gives

$$\|d_\beta J\|_F^2 = f^2(\varepsilon,\mu) \sum_{i=k+1}^{d} \sigma_i^2.$$

We claim that for fixed $\mu > 0$ there exists a unique $\varepsilon_\star = \varepsilon_\star(\mu) > 0$ such that $f(\mu, \varepsilon_\star(\mu)) = 0$. First, we solve the equation for μ

$$\mu(\varepsilon) = \frac{4\varepsilon^2\log\varepsilon + 4\varepsilon^2 T^2}{1-\varepsilon^2}$$

This function of ε is a mapping from $[e^{-1}, 1)$ onto $[0,\infty)$ such that for $T \geq 1$ it is monotonically increasing with strictly positive derivative on $[e^{-1}, 1)$. By the inverse function theorem, $\mu(\varepsilon)$ is invertible and we attain a unique $\varepsilon^\star(\mu)$ for any $\mu \geq 0$. □

We can approximate the function $\varepsilon^\star(\mu)$ guaranteed to exist in lemma 4.4 by expanding μ in a power series about $\varepsilon = e^{-1}$,

$$\mu(\varepsilon) = c_1\left(\varepsilon - e^{-1}\right) + c_2\left(\varepsilon - e^{-1}\right)^2 + \mathcal{O}\left(\left(\varepsilon - e^{-1}\right)^3\right),$$

where $c_1 = \frac{4e}{e^2-1}$ and $c_2 = \frac{2(e^2+3e^4)}{(e^2-1)^2}$. Solving for ϵ, we obtain the approximation

$$\varepsilon^\star(\mu) \approx e^{-1} + \frac{-c_1 + \sqrt{c_1^2 + 4c_2\mu}}{4c_2}. \tag{23}$$

We will use the approximate value (23) in the initialization of our method.

5. Model implementation and numerical experiments

In this section, we describe an implementation of the Dynamical Dimension Reduction (DDR) method and describe its performance on a variety of datasets, including an S-shaped synthetically generated dataset (section 5.2), the iris dataset (section 5.3), and handwritten digit dataset (section 5.4). We demonstrate that the DDR method attains a desirable dimension reduction and compares with embeddings generated by PCA, t-SNE and Umap. Source code for our implementation is available at https://github.com/rkyoon12/DDR.

5.1. **Model implementation.** We implemented the DDR method summarized in algorithm 1 and described in section 3.1. The optimization problem (13) was solved using the ADAM gradient-based method, implemented via the JAX library `jax.example_libraries.optimizers.adam` with the learning rates decaying from $lr = 0.01$ to $lr = 0.001$ as the iterations progress. The gradients of the objective function with respect to the parameters are computed by the formula in (19). The solutions to both the forward ODE (13c) of the hidden variable $h\colon [0,T] \to \mathbb{R}^d$ and the adjoint equation (20) of the Lagrange multiplier $\lambda\colon [0,T] \to \mathbb{R}^d$ are used to compute the gradients. We use the given data as the initial condition in (13c) and set the terminal time $T = 1$. The forward Euler ODE solver is used with discretized interval with time step $dt = 0.01$. To avoid blow-up for the solutions to ODEs in the given time interval and ensure the convergence of algorithms, we threshold the values of the state variable h and adjoint variable λ pointwise to be less than 100.

Dictionary choice and initialization. Since we employ the solution of nonlinear ODEs to define the objective function, the initialization for β is important in the convergence of our model. As shown in theorem 3.1, we initialize β to satisfy the assumption in (10) for pre-specified dictionaries. There is a lot of freedom in choosing the dictionary elements; one could use, *e.g.*, polynomials, multinomials, trigonometric functions, *etc.* For example, we choose candidate functions that are polynomials of h up to degree 3 such as

$$\Xi(h) = [\mathcal{P}_0(h), \mathcal{P}_1(h), \mathcal{P}_2(h), \mathcal{P}_3(h)] \in \mathbb{R}^{3d+1}, \tag{24}$$

where $\mathcal{P}_k(h) = \{h_1^k, \dots, h_d^k\}$ contains k-th degree of polynomials. Denote the coefficients in β corresponding to the dictionary functions $\mathcal{P}_k$ as β_k where $k = 0, 1, 2, 3$. As described in section 4.2, the DDR method using only the linear dictionary $\mathcal{P}_1(h)$ reproduces the result of PCA with parameters β_1 as described in lemma 4.4. Building on this result, to initialize parameters β for an extended dictionary, we set

$$\beta_1 = UD(0, \log(\varepsilon^\star))U^*, \qquad Q = U_k^*,$$

where $\varepsilon^\star$ is chosen according to (23) for a given fixed μ. The remaining entries of β are randomly initialized via the normal distribution, $[\beta]_{ij} \sim \mathcal{N}(0, s^2)$, where we take $s = 0.5$ so that its values are relatively similar to values of β_1.

For each dataset, we train our model several times under the different conditions on entries of dictionary, initialization, hyper-parameters.

Other methods. We compare the DDR method with PCA, t-SNE and Umap by plotting the lower-dimensional representations of the data. Briefly, t-SNE is a nonlinear method that preserves local structures in the data by minimizing the discrepancy between pairwise similarity in the data and the pairwise similarity in the lower-dimensional embedded data (computed using the t-distribution) [19]. Umap learns a Riemannian manifold so that the data is likely to be sampled from a uniform distribution on the manifold [20]. We use the PCA and t-SNE implementations in the scikit-learn package `sklearn.decomposition.PCA` and `sklearn.manifold.TSNE`. We use the Umap implementation provided in [20]. We use the default training settings for the compared methods.

5.2. **S-shaped manifold.** We first test our method on a synthetically generated dataset, which is in the shape of an S-shaped surface embedded in three dimensions. The dataset is generated by the solution at time $T = 1$ to the ODE

$$\dot{z_1} = 2z_3^3 \tag{25a}$$

$$\dot{z_2} = 0 \tag{25b}$$

$$\dot{z_3} = -2z_1^3, \tag{25c}$$

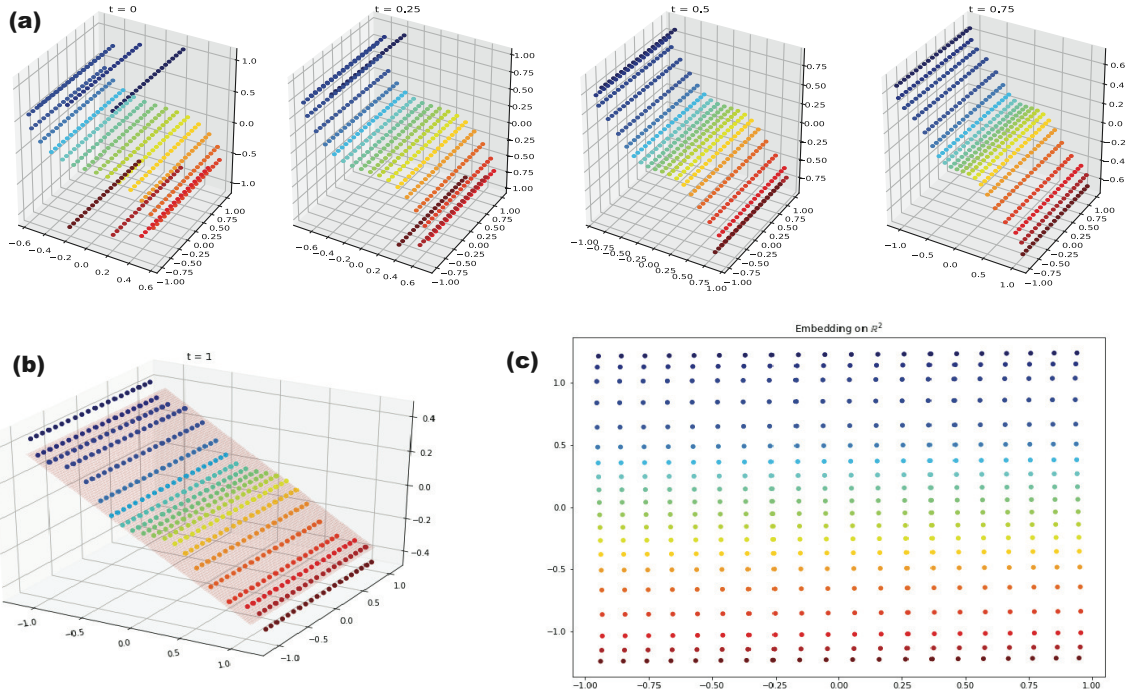

FIGURE 1. **(a)** We depict the positions of data over time evolving along the learned ODE by the DDR method with $\mathcal{P}_3$ in dictionary and $\mu = 0.001$. The four subfigures in (a) are the snapshots of the solution manifold at $t = 0, 0.25, 0.5$ and 0.75. **(b)** Solution to optimal ODE at terminal $T = 1$ and $Q^\star$ subplane which is shaded red. **(c)** We plot the learned lower representations of *S-data* in $\mathbb{R}^2$. See section 5.2.

with $N = 400$ initial conditions given by

$$[z_1(0), z_2(0), z_3(0)] \in [\text{meshgrid}([-1 : 20 : 1], [-1 : 20 : 1]), 0].$$

We refer to collection of points $x_i = [z_1(1), z_2(1), z_3(1)]_i$, for $i \in [400]$ as the *S-data*; a plot of the *S-data* is given in the first subplot of fig. 1(a) and colored by the first coordinate.

Since the *S-data* is created by the cubic polynomial vector fields, it is natural to employ $\mathcal{P}_3$ functions to build a dictionary. We visualize the DDR model by plotting the evolution of the learned dynamical system and projection space. As shown in fig. 1(a), the hidden variables are initially positioned in an S-shaped manifold and are gradually unfolded/flattened onto the $Q^\star$ space over time. In fig. 1(b), we draw both hidden variables at T (colored dots) and an orthonormal subspace spanned by row vectors of $Q^\star$ (red shaded surface). The DDR method maps the *S-data* to the low-dimensional representations shown in fig. 1(c).

Hyper-parameter tuning. The objective function of the DDR method contains two terms; a mean squared residual error ($J1$) and a kinetic energy of the data manifold traveling along the ODE ($J2$), where a regularization hyper-parameter μ balances between $J1$ and $J2$. In practice, selecting an appropriate μ is important to reasonably train the DDR model. We employ the $\mathbb{L}$-curve criterion proposed in [5] for the Tikhonov regularization hyper-parameter of the linear inverse problem. Denote β_μ and Q_μ as optimal solution to the problem

$$\beta_\mu, Q_\mu = \arg\min_{\beta, Q} J_1 + \mu J_2,$$

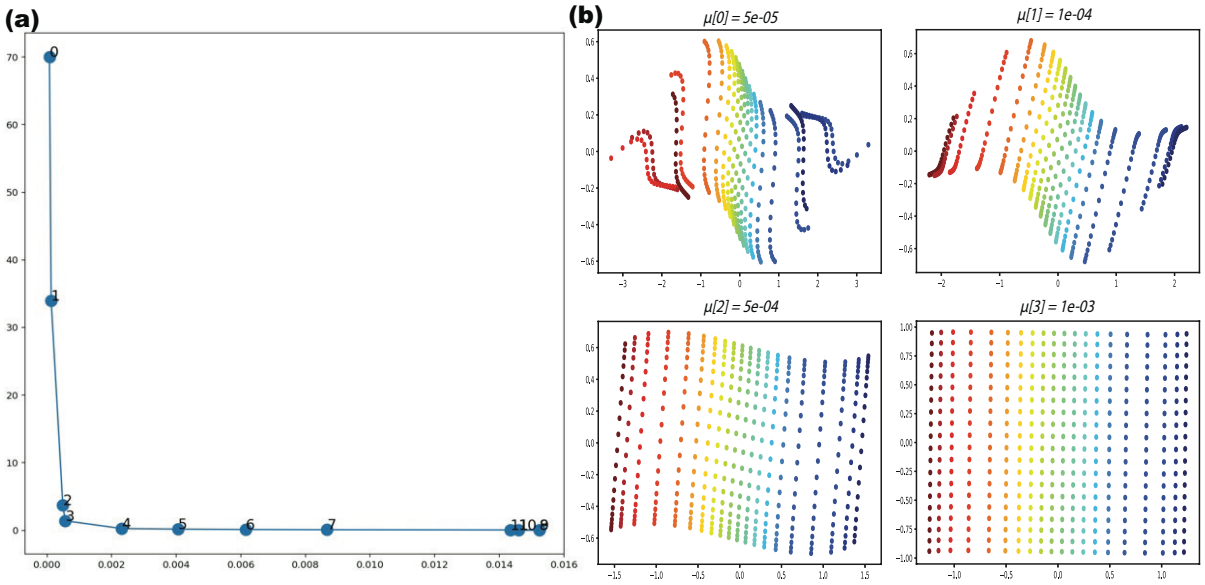

FIGURE 2. **(a)** Plot $\mathbb{L}$-curve defined as $\mathbb{L} = \{(J1, J2) : \mu \text{is in pre-listed set}\}$. **(b)** The lower representation of *S-data* by training the DDR method under the same conditions except hyper-parameter $\mu = \{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$. See section 5.2.

where we use the same definition of J_1 and J_2 in (12). We define the curve

$$\mathbb{L} = \{J_1(\beta_\mu, Q_\mu), J_2(\beta_\mu, Q_\mu) \colon \mu > 0\}.$$

As a function of μ, J_2 is monotonically decreasing whereas J_1 is monotonically increasing. Thus the $\mathbb{L}$-curve has a negative slope and, in practice, takes the shape of an "L". Moreover, both J_1 and J_2 are equitably minimized at the elbow of $\mathbb{L}$-curve. In practice, we tune the regularization parameters by training the model for μ in the set $\{5\times10^{-5}, 10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 0.05, 0.1, 0.5, 1, 1.5, 2\}$ and picking μ at the elbow of the resulting $\mathbb{L}$-curve.

We present the $\mathbb{L}$-curve from training the DDR method for *S-data* with each μ in the above set. Figure 2(a) shows that the vertex of L-curve is attained at the fourth element (numbered by 3) in the list of μ, which is 10^{-3}. In 2(b), we depict the learned lower representations of the DDR method with $\mu \in \{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$. Coincided with $\mathbb{L}$-curve criterion in (a), the most expected embedding is achieved with $\mu = 10^{-3}$.

Non-identifiablity. Next, we remark that the DDR method is *non-identifiable*. The *S-data* could be viewed as an initial condition for the reverse ODE to (25) so that its solution is lying onto $span\{e_1, e_2\}$ plane at $T = 1$, where $e_i \in \mathbb{R}^3$ denotes a canonical basis vector whose i-th entry is one. It implies that the ground-truth parameters of the DDR method are exactly a coefficient of time-reversed dynamical system such that

$$\beta_{true} = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}, \qquad Q_{true} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Remind that the goal of our method is finding a mapping $\mathcal{E} \colon x \mapsto y = Qh(T)$ so that only the last stage of the solution $h(T)$ should be as close as possible to Q space. Hence learned vector fields and subspace may not be uniquely determined and could differ from the ground-truth. Indeed, the

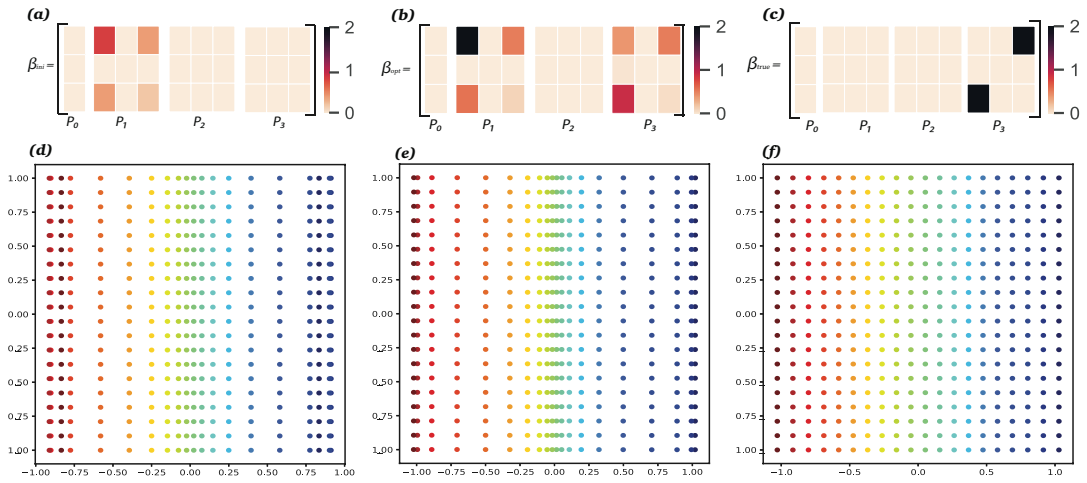

FIGURE 3. We represent the DDR method with a choice of dictionary with polynomials upto degree 3 in (24). **(a)-(c)** We draw heat-map about the magnitude of entries of initial β_{ini}, optimal β_{opt} and ground truth β_{true}. **(d)-(f)** Plot the embedding of *S-curve* in $\mathbb{R}^2$ generated with (β_{ini}, Q_{ini}), (β_{opt}, Q_{opt}) and (β_{true}, Q_{true}) respectively. See section 5.2.

	(β_{ini}, Q_{ini})	(β_{opt}, Q_{opt})	(β_{true}, Q_{true})
$J1$ (residual)	0.01219	0.000285	**0.00053**
$J2$ (regularization)	**0.00019**	0.00069	0.00393
J (total loss)	0.01238	**0.00098**	0.00447

TABLE 1. A comparison of value of objective function evaluated at specific (β,Q). See section 5.2.

trained optimal parameters reported below do not agree with true parameters.

$$\beta^\star = \begin{bmatrix} -0.0680 & -0.0005 & -2.1890 \\ -0.0057 & -0.0546 & -0.0287 \\ 0.5832 & -0.0001 & -0.3004 \end{bmatrix}, \qquad Q^\star = \begin{bmatrix} -0.9482 & -0.0139 & 0.3173 \\ 0.0133 & -0.9999 & -0.0040 \end{bmatrix}$$

Dictionary comparison. We now consider the DDR framework with general choices of dictionaries. As formulated in (24), a dictionary Ξ consists of polynomial functions of h up to degree 3. We then derive embeddings $x \overset{\mathcal{E}}{\mapsto} y$ parametrized by three cases of parameters; an initializer (β_{ini}, Q_{ini}) described in section 5, the optimizer (β_{opt}, Q_{opt}) trained by the DDR method and the ground truth (β_{true}, Q_{true}) given in (5.2). In fig. 3, the subplots (a)-(c) visualize the magnitude of all entries of each β by varying the intensity of colors and the subplot (d)-(f) plot the resulting low representations.

As pointed out in 4.2, a framework of DDR model characterized by (β_{ini}, Q_{ini}) performs similarly to the PCA, where the embedding (d) formulated by an initializer is almost identical with the PCA projection. Such linear projection methods, however, couldn't capture nonlinearity in the data. As shown in Figure 3(d), the points located at the tail of *S-data* are not recovered by any linear vector fields and are folded/overwritten on the Q_{ini} space. In contrast, the DDR method encourages the underlying vector fields to be represented by nonlinear functions via the training process. In a comparison of heat maps Figure 3(a)-(b), the optimal coefficients corresponding to $\mathcal{P}_3$ being initialized by zero are activated, while the linear parts of components are still assisted. As plotted in Figure 3(e), the optimal lower dimensional representation perfectly rolled out $S-data$ than

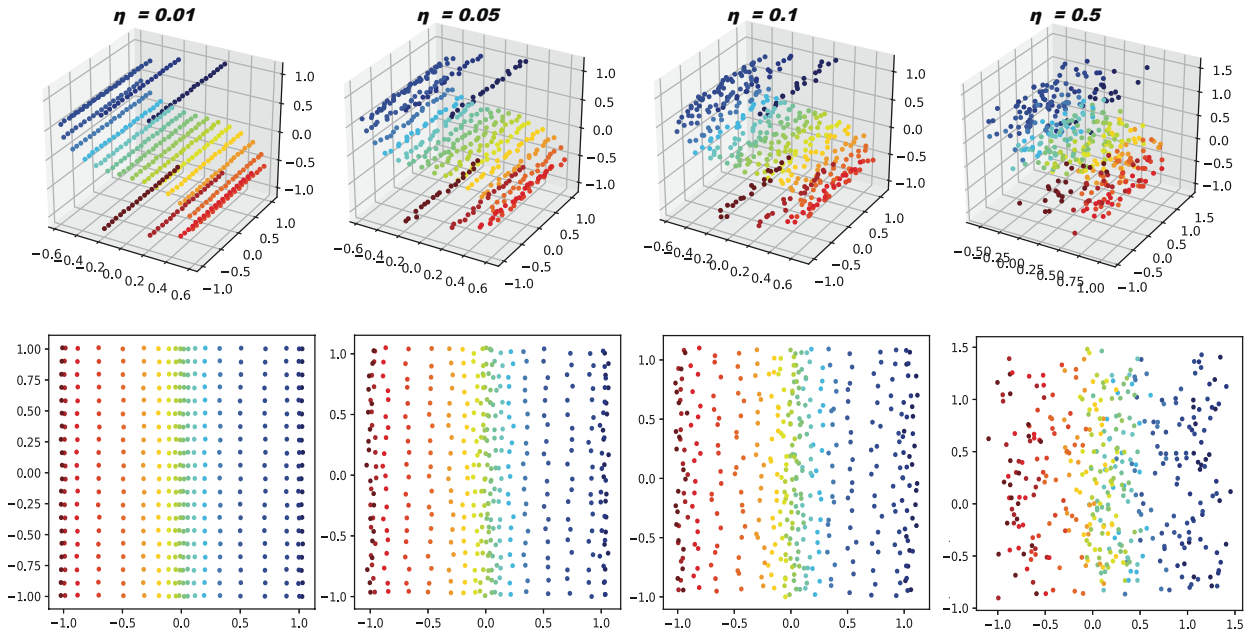

FIGURE 4. We apply the optimal DDR model parametrized by (β_{opt}, Q_{opt}) to the *S-data* that has been perturbed by random noise with standard deviation η; see section 5.2.

(d). Furthermore, we present the scores of objective functions evaluated at parameters in Table 1. By comparing the first two columns of the table, the residual error is mainly minimized, whereas the rise in regularization loss is relatively negligible. Therefore, the DDR method is established to reinforce complexity in dynamics and improve the performance of the dimension reduction mapping by minimizing a total objective function.

Next, we observe the influence of the regularization term in (11) on the learning of a data manifold. Both optimal and true embedding in fig. 3(e)-(f) could be considered as a good lower dimensional representation of *S-data* because the initial mesh grid is well retrieved. As tabulated in the last column of table 1, however, embedding (f) spends extensive kinetic energy of dynamics to transform the manifold. If a given manifold is forced to move by a higher speed of vector field, then inherent properties or key structure of data could be contaminated. Indeed, the minimum of total loss is attained at (β_{opt}, Q_{opt}). Therefore, we show that the DDR method is designed to balance between projecting onto reduced dimensional space and preserving the structures of the data.

Stability. In section 4.1, we prove that the DDR mapping $x_i \overset{\mathcal{E}}{\mapsto} y_i$ is stable under the noise in a given data. We numerically examine that the mapping learned with a given data is generalizable to perturbed data without retraining the model. In fig. 4, we depict *S-data* interrupted by the noise and its lower representation applied by the optimal embedding expressed by (β_{opt}, Q_{opt}). Note that four different perturbed data are created by adding a perturbation z element-wise, where $z \sim N(0, \eta^2)$ where the standard deviation of the noise η varies in $[0.01, 0.05, 0.1, 0.5]$. Since the magnitude of plane *S-data* is ranged in $[-1.2, 1.2]$, low dimensional representations of noisy data are reliable as long as η is relatively small.

Generative model for the S-shaped manifold. In section 3.2, we explained how the DDR method can be extended as a generative model. After training the DDS method $x \overset{\mathcal{E}}{\mapsto} y = Q^* h(T)$, the decoder $\mathcal{D}\colon \mathbb{R}^k \to \mathbb{R}^d$ is defined by $\mathcal{D}(y) = h(0)$, where $h(t)$, $t \in [0, T]$ satisfies the time-reversed dynamical

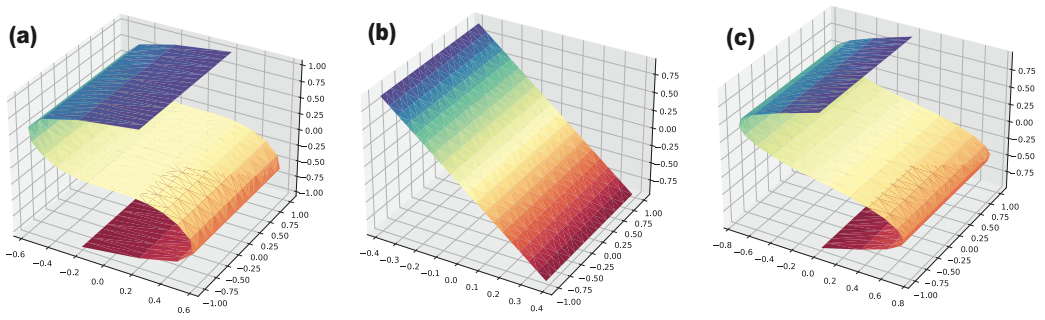

FIGURE 5. We extend the DDR framework for generative purposes. (a) A given S-shaped manifold is used for training an encoder $\mathcal{E}$. (b) We regularly sample points from the latent space and plot their image under the mapping $y \mapsto Q^{\star}y$. (c) Using the decoder, we plot the decoded manifold, which is an approximation to the original S-shaped manifold. Each of the surfaces are drawn using a triangular mesh.

system (14) with final condition $h(T) = Q^{*}y$. If there is zero training error, we have that $\mathcal{D} \circ \mathcal{E} = I$ on the training data. Further, in theorem 4.3, we showed that there exists a neighborhood about the embedded data, such that the decoder is well-defined.

We further illustrate this idea using the S-shaped dataset (see fig. 5(a)). We consider regularly sampled points in the latent space, $y \in \text{meshgrid}([-1:20:1],[-1:20:1])$. For each y, we solve the time-reversed dynamical system (14) with initial condition $Q^{*}y$; these initial conditions are plotted in fig. 5(b), using a triangular mesh. The decoded points $\mathcal{D}(y)$ are then plotted in fig. 5(c), again using a triangular mesh. We view the map $\mathcal{D}\colon \mathbb{R}^2 \to \mathbb{R}^3$ as a parameterization of an approximation to the data manifold in fig. 5(a). The approximation comes from the fact that the training error for the DDR method on this dataset is non-zero.

5.3. **Iris-data.** The *iris* dataset contains $N = 150$ instances where each data has $d = 4$ features and is classified into three types of iris. The data is downloaded via `sklearn.datases.load_iris()`. We consider embedding this $d = 4$ dimensional *iris* data onto $k = 2$ dimensional space. For the DDR method, we conduct a hyper-parameter search using the $\mathbb{L}$-curve test and choose $\mu = 0.005$. In fig. 6, we plot the embedded data which are colored by their classes along with the results obtained via PCA, t-SNE, and Umap. Comparing the four methods, we observe that the DDR method clusters the data as much as the other methods. In fact, the clustering boundary of the DDR method, especially the margin between group red and blue, is more distinct and noticeable than other methods. The nonlinear dynamics in the DDR method end up reducing the in-class variance slightly more than PCA but without collapsing the clusters as t-SNE and Umap do for this dataset. This shows that DDR method maintains both large-scaled structure and pairwise distances between dataset.

5.4. **Handwritten digits-data.** The *digits* data contains 8×8 images of handwritten digits $0-9$. We downloaded the data from the sklearn dataset dictionary using `sklearn.datasets.load_digits()`. Note that we only use a subset of the images, digits $0-3$, so we have $N = 364$ examples. We also normalized the data by changing the range of the pixel values from $[0, 16]$ to $[0, 1]$. To reduce computing time, we applied PCA to reduce the dimension from 64 to 10 dimensions. We examined the DDR method with extensive dictionary sweeps and hyper-parameter searching, and the best result is found with $\mathcal{P}_3$, using a random initialization, and $\mu = 0.01$.

The resulting two-dimensional embedding obtained via the DDR method is shown in Figure 7, as well as the embeddings obtained via PCA, t-SNE, and Umap. We observe that the DDR method clusters the digits but not as strongly as t-SNE and Umap. Compared to the PCA embedding,

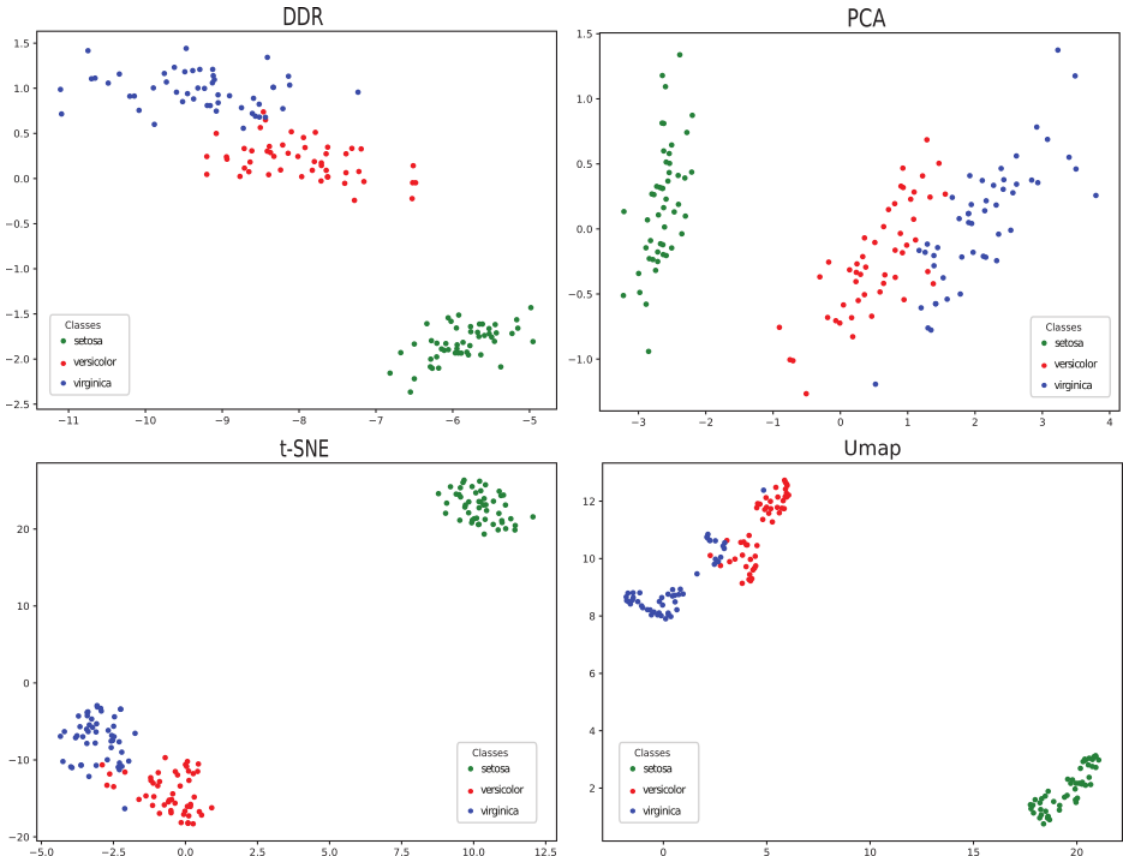

FIGURE 6. Comparison of projection of *iris*-data using DDR, PCA, t-SNE, and Umap. See section 5.3.

the boundary between classes 1, 2, and 3 (colored by sky blue, yellow, and brown, respectively) is better defined than PCA. This shows that the nonlinear mapping delivers more information than the linear one. However, we see a few misinterpreted instances by the DDR method (several brown dots in the yellow cloud), which may correspond to the brown island found in t-SNE and Umap subplots.

Computational time. The training of our model depends on the initial dimension d of the data and the size of the dictionary d_n. For, the *handwritten digits* dataset, we used $d = 10$ and $d_n = 30$. To train the DDR model on this dataset, we used 900 epochs taking an average of 2.2692 seconds per epoch. In comparison, t-SNE and Umap took less training time, 5.7168 seconds and 16.4622 seconds, respectively. Each of these methods used 1000 epochs and 500 epochs, respectively. Our implementation of the DDR method is slower than these other methods, which could be improved in future work.

6. Discussion

In this work, we proposed a framework for learning a low-dimensional representation of data based on nonlinear dynamical systems, called *dynamical dimension reduction* (DDR). In the DDR model, each point x is evolved via a nonlinear flow (5) towards a lower-dimensional subspace; the projection onto the subspace gives the low-dimensional embedding. Training the model involves identifying the nonlinear flow and the subspace. Following the equation discovery method, we represent the vector field that defines the flow using a linear combination of dictionary elements,

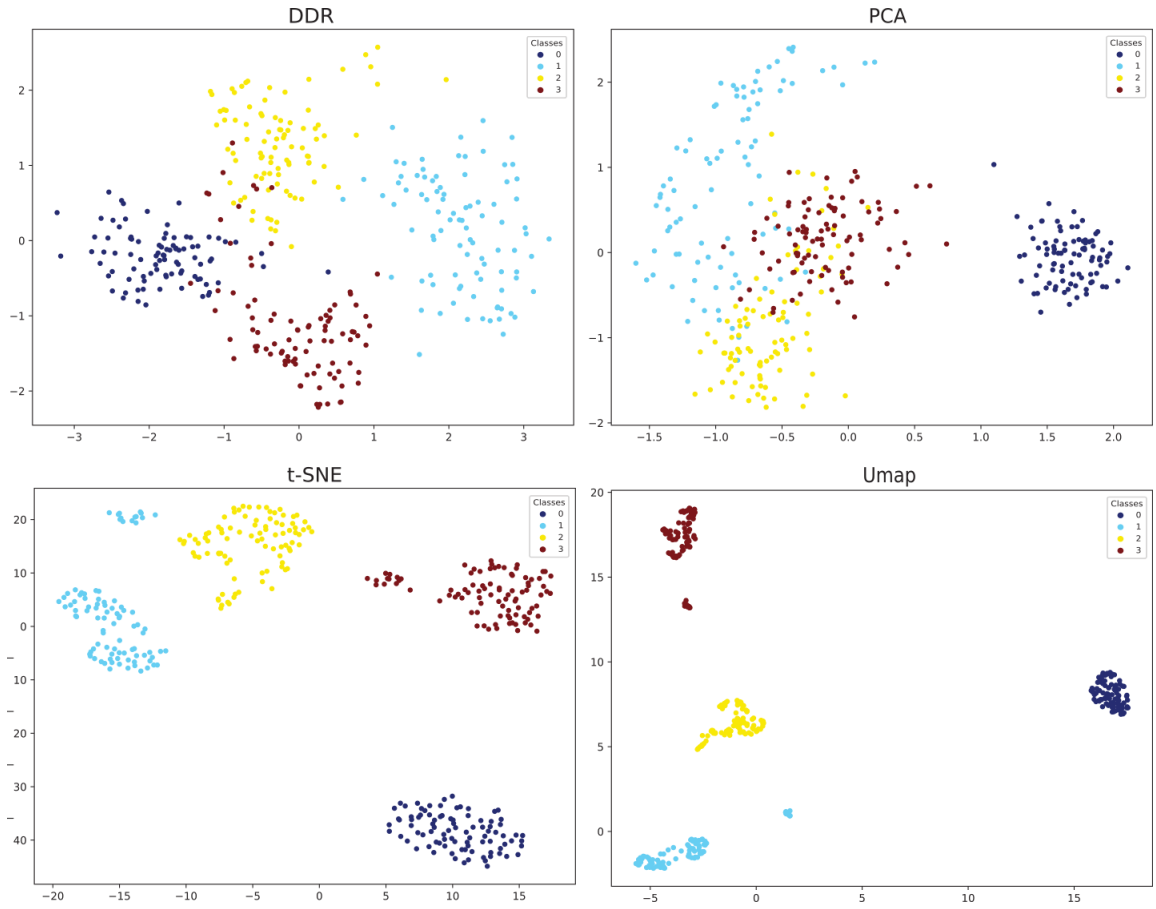

FIGURE 7. Comparison of embeddings for a subset of the *digit*-data which consists of digits $0-3$ using DDR, PCA, t-SNE, amd Umap. See section 5.4.

where each element is a pre-specified linear/nonlinear candidate function. A regularization term for the average total kinetic energy is also introduced and motivated by optimal transport theory. We prove that the resulting optimization problem (13) is well-posed (see theorem 3.8) and establish several properties of the DDR method (see section 4). We also show how the DDR method can be trained using a gradient-based optimization method, where the gradients are computed using the adjoint method from optimal control theory (see theorem 3.9). Implementing the DDR method via algorithm 1, we demonstrate that its performance is comparable to other dimension reduction methods including PCA, t-SNE and Umap (see in section 5). In examples, we observed that the representability/expressibility of the DDR method is improved over PCA due to the nonlinear functions in the governing vector field; to capture complex data structures, the parameters corresponding to the nonlinear dictionary elements are activated. The t-SNE and Umap methods solely rely on local distances and PCA focuses on the global structure of the data. In contrast, the DDR method balances these objectives, minimizing not only a residual error but also the kinetic energy of the trajectories (a rate of deformation of the data manifold along the flow).

We implemented the DDR method as a proof of concept. However, this method is slow to train because the solutions to the forward ODE for the hidden variable (5) and the adjoint ODE (20) are expensive to compute. A natural future direction for this work is to accelerate the algorithm by using multi-step ODE solvers and allowing the method to adaptively chose a coarser discretization. Furthermore, we could generalize the governing dynamical system to include non-autonomous

vector fields, Φ or respect additional structure, *e.g.*, Hamiltonian or symplectic [13, 26]. We could also modify the form of the dynamical system; for example, the second-order momentum equation might improve computational efficiency and long-term dependencies [23].

The theory of dynamical systems could be used to further prove analytical results for the DDR model. For example, while theorem 4.1 gives a stability result for a given DDR embedding in terms of the data, we view it as an interesting and challenging result to prove the stability of the training with respect to changes in the data as well as the consistency of the model. Further ideas from equation discovery could also be incorporated, such as looking for vector fields that have a sparse representation in terms of the dictionary.

Acknowledgements. We would like to thank Harish Bhat for helpful discussions in the early stages of this work.

References

[1] P. Baldi. "Autoencoders, Unsupervised Learning, and Deep Architectures". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. Ed. by I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver. Vol. 27. Proceedings of Machine Learning Research. Bellevue, Washington, USA: PMLR, 2012, pp. 37–49.

[2] J.-D. Benamou and Y. Brenier. "A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem". *Numerische Mathematik* 84.3 (2000), pp. 375–393. DOI: 10.1007/s002110050002.

[3] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer New York, 2000. DOI: 10.1007/978-1-4757-9859-3.

[4] S. L. Brunton, J. L. Proctor, and J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. DOI: 10.1073/pnas.1517384113.

[5] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari. "Tikhonov regularization and the L-curve for large discrete ill-posed problems". *Journal of computational and applied mathematics* 123.1-2 (2000), pp. 423–446. DOI: 10.1016/s0377-0427(00)00414-3.

[6] M. Chalvidal, M. Ricci, R. VanRullen, and T. Serre. "Go with the flow: Adaptive control for neural odes". *arXiv preprint arXiv:2006.09545* (2020). DOI: 10.48550/arXiv.2006.09545.

[7] B. Chang, M. Chen, E. Haber, and E. H. Chi. "AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks". In: *7th International Conference on Learning Representations, ICLR 2019*. 2019. DOI: 10.48550/arXiv.1902.09689.

[8] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. "Neural ordinary differential equations". *Advances in neural information processing systems* 31 (2018). DOI: 10.48550/arXiv.1806.07366.

[9] C. Finlay, J.-H. Jacobsen, L. Nurbekyan, and A. Oberman. "How to train your neural ODE: the world of Jacobian and kinetic regularization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3154–3164.

[10] M. Garsdal, V. Søgaard, and S. Sørensen. "Generative time series models using Neural ODE in Variational Autoencoders". *arXiv preprint arXiv:2201.04630* (2022). DOI: 10.48550/arXiv.2201.04630.

[11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. DOI: 10.1007/s10710-017-9314-z.

[12] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. "FFJORD: Free-form continuous dynamics for scalable reversible generative models". *arXiv preprint arXiv:1810.01367* (2018). DOI: 10.48550/arXiv.1810.01367.

[13] S. Greydanus, M. Dzamba, and J. Yosinski. "Hamiltonian neural networks". *Advances in Neural Information Processing Systems* 32 (2019). DOI: 10.48550/arXiv.1906.01563.

[14] E. Haber and L. Ruthotto. "Stable Architectures for Deep Neural Networks". *Inverse Problems* 34 (2017), p. 014004. DOI: 10.1088/1361-6420/aa9a90.

[15] M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki. "Learning unknown ODE models with Gaussian processes". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1959–1968. DOI: 10.1109/cdc45484.2021.9683426.

[16] H. Hotelling. "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology* 24.6 (1933), p. 417. DOI: 10.1037/h0071325.

[17] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114* (2013). DOI: 10.48550/arXiv.1312.6114.

[18] Z. Long, Y. Lu, and B. Dong. "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network". *Journal of Computational Physics* 399 (2019), p. 108925. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.108925.

[19] L. van der Maaten and G. Hinton. "Visualizing Data using t-SNE". *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605.

[20] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". *arXiv preprint arXiv:1802.03426* (2018). DOI: 10.48550/arXiv.1802.03426.

[21] F. Santambrogio. "Optimal transport for applied mathematicians". *Birkäuser, NY* 55.58-63 (2015), p. 94. DOI: 10.1007/978-3-319-20828-2.

[22] T. C. Sideris. *Ordinary Differential Equations and Dynamical Systems*. Springer, 2013. DOI: 10.2991/978-94-6239-021-8.

[23] H. Xia, V. Suliafu, H. Ji, T. Nguyen, A. Bertozzi, S. Osher, and B. Wang. "Heavy ball neural ordinary differential equations". *Advances in Neural Information Processing Systems* 34 (2021). DOI: 10.48550/arXiv.2110.04840.

[24] R. Yoon, H. S. Bhat, and B. Osting. "A Nonautonomous Equation Discovery Method for Time Signal Classification". *SIAM Journal on Applied Dynamical Systems* 21.1 (2022), pp. 33–59. DOI: 10.1137/21m1405216.

[25] L. Zhang and H. Schaeffer. "On the Convergence of the SINDy Algorithm". *Multiscale Modeling & Simulation* 17.3 (2019), pp. 948–972. DOI: 10.1137/18m1189828.

[26] Y. D. Zhong, B. Dey, and A. Chakraborty. "Symplectic ode-net: Learning hamiltonian dynamics with control". *arXiv preprint arXiv:1909.12077* (2019). DOI: 10.48550/arXiv.1909.12077.