Proceedings of Machine Learning Research vol 120:1–10, 2020 2nd Annual Conference on Learning for Dynamics and Control

Euclideanizing Flows: Diffeomorphic Reduction for Learning Stable Dynamical Systems

M. Asif Rana[1,4], **Anqi Li**[2,4], **Dieter Fox**[2,4], **Byron Boots**[2,4], **Fabio Ramos**[3,4], **Nathan Ratliff**[4]

[1]*Georgia Institute of Technology,* [2]*University of Washington,* [3]*University of Sydney,* [4]*NVIDIA*

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

Robotic tasks often require motions with complex geometric structures. We present an approach to learn such motions from a limited number of human demonstrations by exploiting the regularity properties of human motions e.g. stability, smoothness, and boundedness. The complex motions are encoded as rollouts of a stable dynamical system, which, under a change of coordinates defined by a diffeomorphism, is equivalent to a simple, hand-specified dynamical system. As an immediate result of using diffeomorphisms, the stability property of the hand-specified dynamical system directly carry over to the learned dynamical system. Inspired by recent works in density estimation, we propose to represent the diffeomorphism as a composition of simple parameterized diffeomorphisms. Additional structure is imposed to provide guarantees on the smoothness of the generated motions. The efficacy of this approach is demonstrated through validation on an established benchmark as well demonstrations collected on a real-world robotic system.

Keywords: Learning from demonstration, Robot learning, Learning of dynamical systems

1. Introduction

In many applications, robots are required to execute complex motions in potentially dynamic and unstructured environments. Since hand-coding such motions can be cumbersome or even infeasible, learning from demonstration (LfD) [1] enables robots to, instead, acquire new motion skills by observing humans. In this work, we focus on learning goal-directed motions from human demonstrations, i.e. motions that stop at a given target location. This is without loss of generality since complex tasks can often be achieved by an ordered execution of goal-directed motions [2].

Human motions naturally preserve regularity properties, e.g. continuity, smoothness and boundedness [3]. Since we assume goal-directed motions, human demonstrations should also naturally be stable. Exploiting these regularity properties can significantly improve the sample efficiency of the learning algorithms, which is critical since the number of human demonstrations are often limited. Thus, we model human motions as rollouts from a stable dynamical system. Motions governed by stable dynamical systems exhibit some additional desirable properties. First, such motions can react to temporal and spatial perturbations, which is necessary in dynamic task settings. Second, the stability property formally guarantees that the motions converge to the goal region.

The challenge, however, is to encode stable human motions into dynamical systems that are stable by construction. A number of approaches have been proposed, which address this problem by explicitly parameterizing the class of stable dynamical systems and the notions of stability [4; 5; 6]. In this work, we take a fundamentally different approach than the aforementioned methods. Instead of explicitly learning a stable dynamical system, we view demonstrations as motions on a Riemannian manifold which is linked, under a smooth bijective map, i.e. a *diffeomorphism*, to a

Corresponding author: M. Asif Rana, asif.rana@gatech.edu

latent Euclidean space. This diffeomorphism implicitly gives rise to an inherently stable dynamical system. The learning problem thus involves finding a diffeormorphism which explains the observed demonstrations. Compared to existing diffeomorphism learning approaches to encoding motions [7; 8], our formulation, **S**table **D**ynamical **S**ystem learning using **E**uclideanizing **F**lows (SDS-EF), is based on a more expressive formulation of diffeomorphisms.

We present an approach for learning a time-invariant continuous-time dynamical system (or reactive motion policy), which is globally asymptotically stable. Our dynamics formulation allows encoding severely curved goal-directed motions, and can be learned from a few demonstrations with minimal parameter tuning. Our specific contributions include: (i) a formulation of stable dynamics through warping curves into simple motions on a latent space using diffeomorphisms, and (ii) an expressive class of diffeomorphisms suitable for learning stable and smooth dynamical systems. We demonstrate the effectiveness of our approach on a standard handwriting dataset [9], and data collected on a robot manipulator [10].

2. Related Work

Over the past decade, a number of approaches have been proposed towards learning goal-directed time-invariant dynamical systems from human demonstrations [11]. An early approach is SEDS [4]. SEDS assumes the demonstrations comply with a Lyapunov (or Energy) function given by the squared-distance to the goal, and thus is restricted to motions which monotonically converge to the goal over time. A relaxation to the stability criterion was proposed in CLF-DM [5], which instead assumes Lyapunov functions in the form of a weighted sum of asymmetric quadratic functions (WSAQF). The parameters of the Lyapunov function are learned independently from the (unstable) dynamics, and then used in an online fashion to generate stabilizing controls. The online correction scheme may interfere significantly with the learned dynamics [7]. A more recent approach, CDSP [12] instead enforces incremental stability, a notion concerned with relative displacement of motions. Instead of learning a Lyapunov function, CDSP proposed learning a positive-definite contraction metric. However, CDSP restricts the class of contraction metrics to (potentially limiting) sum of squared polynomials.

There are a few approaches that learn stable dynamical systems via learned diffeomorphisms. However, by definition, a diffeomorphism must be invertible, making the learning problem nontrivial. One approach [8], realizes a diffeomorphism as a composition of locally weighted translations. The authors only apply this approach to the problem of learning a single (or average) demonstration. Another strategy, τ-SEDS [7], learns diffeomorphisms from multiple demonstrations. However, τ-SEDS defines a diffeomorphism by the square-root of a WSAQF, thus restricting the hypothesis class. In contrast to this prior work, we learn diffeomorphisms by using flexible function approximators including kernel methods [13] and neural networks [14].

Our diffeomorphism learning approach builds on normalizing flows [15; 16], which have recently been successfully used for density estimation. The goal of normalizing flows is to map a simple base distribution into a complicated probability distribution over observed data by applying the change of variable theorem sequentially. Our problem is similar to the normalizing flows problem as we seek to map simple straight-line motions to complicated motions captured from human demonstrations. However, our problem is fundamentally different in two ways. First, normalizing flows only require the mapping to be bijective, a property less strict than diffeomorphism (see Section 3). Second, normalizing flows are concerned primarily with mapping scalar functions to scalar functions (i.e. probability densities). In our method, we seek to map vector fields to vector fields.

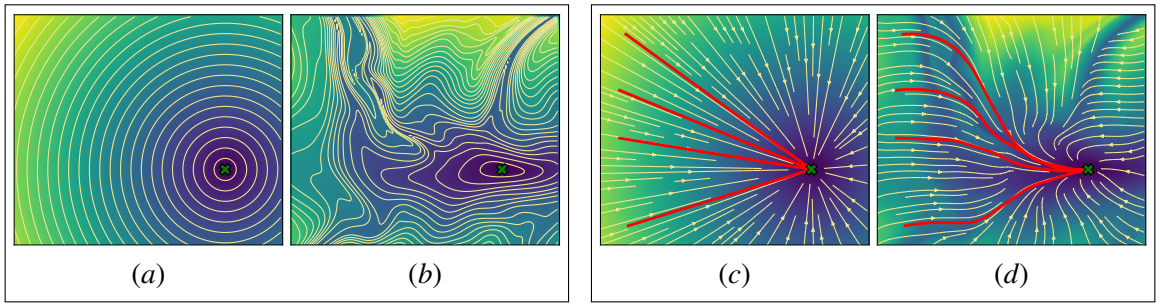

Figure 1: *(a)-(b)*: Isocontours showing equidistant points from the origin (green cross) in a Euclidean space and a Riemannian manifold respectively. *(c)-(d)*: Velocity fields governing shortest distance paths to the origin in the two spaces; rollouts from specific locations are overlayed in red.

3. Background: Stability, Diffeomorphism, and Riemannian Manifolds

We briefly summarize the theoretical background for this paper. First, we introduce the concept of global asymptotic stability and how it can be shown through an auxiliary function, i.e. a Lyapunov function. Next, we discuss how one dynamical system can be described in different ways through a change of coordinates using a diffeomorphism. Finally, we shed light into the geometric interpretation of diffeomorphisms, especially when applied to gradient descent dynamical systems.

Notation: To define mappings, we use symbol $\rightarrow$ to specify domains and co-domians, e.g. $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$, and $\mapsto$ for how individual elements of the domains are mapped, e.g. $\psi : \mathbf{x} \rightarrow \mathbf{y}$. We use both symbols ∇ and ∂ to denote derivatives, with a transposed relationship: consider $\mathbf{x} \in \mathbb{R}^n$ and a differentiable map $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$, then $\nabla_{\mathbf{x}}\psi(\mathbf{x}) = \frac{\partial \psi}{\partial \mathbf{x}}^\top$. To evaluate functions which involve derivatives denoted by ∂, we use subscripts of brackets, e.g., $\left[\frac{\partial \psi}{\partial \mathbf{x}}\right]_{\mathbf{x}=0} = \nabla_{\mathbf{x}}\psi(0)^\top$. We use the notation $\circ$ for function composition, i.e., for $f : \mathbb{R}^m \rightarrow \mathbb{R}^p$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $(f \circ g)(\mathbf{x}) = f(g(\mathbf{x}))$.

3.1. Global Asymptotic Stability

Consider an n-dimensional differentiable manifold $\mathcal{M}$ with a global coordinate $\mathbf{x} : \mathcal{M} \rightarrow \mathbb{R}^n$, which maps the manifold $\mathcal{M}$ to a point in $\mathbb{R}^n$. With a slight abuse of notation, we will use the coordinate $\mathbf{x}$ to both represent the map $\mathbf{x} : \mathcal{M} \rightarrow \mathbb{R}^n$ and the point $\mathbf{x} \in \mathbb{R}^n$. A dynamical system on the manifold $\mathcal{M}$ can be described as a dynamical system on $\mathbb{R}^n$ under coordinate $\mathbf{x}$. We consider the dynamical system in the following form,

$$\dot{\mathbf{x}} = f(\mathbf{x}), \quad \text{where } f : \mathbb{R}^n \rightarrow \mathbb{R}^n \text{ is locally Lipschitz continuous.} \tag{1}$$

While there are multiple definitions of stability, we are specifically concerned with whether the system can converge to a point of interest $\mathbf{x}^* \in \mathbb{R}^n$ from an arbitrary initial state on $\mathbb{R}^n$. Assume that $\mathbf{x}^*$ is an equilibrium point for the system, i.e., $f(\mathbf{x}^*) = 0$, this desired convergent property can be characterized by *global asymptotic stability*[1] of the equilibrium point $\mathbf{x}^*$. The global asymptotic stability can be shown through a continuously differentiable, positive-definite, and radially unbounded function called *Lyapunov function* [17]. A Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is a scalar valued function which satisfies that (i) $\dot{V}(\mathbf{x}) = \nabla_{\mathbf{x}}V(\mathbf{x})^\top f(\mathbf{x}) < 0$ for all $\mathbf{x} \neq \mathbf{x}^*$ and (ii) $\dot{V}(\mathbf{x}^*) = 0$ at the equilibrium $\mathbf{x}^*$. If there exists such a Lyapunov function V, the equilibrium point $\mathbf{x}^*$ is globally asymptotically stable: all trajectories converge to the point $\mathbf{x}^*$.

3.2. Change of Coordinates for Dynamical Systems

Consider a bijective map $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$. The bijective map ψ is a *diffeomorphism* if both the map ψ and its inverse map ψ^{-1} are continuously differentiable. We further assume that the diffeomorphism

1. We refer the reader to [17] for a more rigorous and thorough introduction of stability properties of dynamical system.

ψ is bounded, namely that it maps bounded vectors to bounded vectors. A diffeomorphism can be used to describe a *change of coordinates* for differentiable manifolds. Consider again the manifold $\mathcal{M}$ with global coordinates $\mathbf{x} \in \mathbb{R}^n$. A diffeomorphism $\psi : \mathbf{x} \mapsto \mathbf{y}$ creates another global coordinate $\mathbf{y} : \mathcal{M} \to \mathbb{R}^n$ for the manifold $\mathcal{M}$ through the map $\mathbf{y} = \psi(\mathbf{x})$. As such, the dynamical system (1) can be described under the other coordinate $\mathbf{y}$ as,

$$\dot{\mathbf{y}} = \left[\frac{\partial \psi}{\partial \mathbf{x}} f(\mathbf{x})\right]_{\mathbf{x}=\psi^{-1}(\mathbf{y})} = \mathbf{J}_\psi(\psi^{-1}(\mathbf{y}))\, f(\psi^{-1}(\mathbf{y})) \quad := \tilde{f}(\mathbf{y}), \tag{2}$$

where $\mathbf{J}_\psi(\mathbf{x}) = \frac{\partial \psi}{\partial \mathbf{x}}$ is the *Jacobian matrix* of ψ. In the special case where both the system dynamics f and the diffeomorphism ψ are linear maps, this change of coordinates reduces to *change of basis* for linear dynamical systems [18].

The two dynamical systems (1) and (2) are descriptions of *the same internal dynamical system* evolving on the manifold $\mathcal{M}$. Therefore, the two systems share stability properties. Assume the existence of a Lyapunov function $V(\mathbf{x})$ showing that the equilibrium point $\mathbf{x}^*$ is globally asymptotically stable. Then, the diffeomorphism ψ defines a Lyapunov function $\tilde{V} : \mathbf{y} \mapsto V(\psi^{-1}(\mathbf{y}))$,

$$\begin{aligned}\dot{\tilde{V}}(\mathbf{y}) &= \left[\frac{\partial V}{\partial \mathbf{x}} \frac{\partial \psi^{-1}}{\partial \mathbf{y}} \dot{\mathbf{y}}\right]_{\mathbf{x}=\psi^{-1}(\mathbf{y})} = \left[\frac{\partial V}{\partial \mathbf{x}} \left(\mathbf{J}_\psi(\mathbf{x})\right)^{-1} \tilde{f}(\mathbf{y})\right]_{\mathbf{x}=\psi^{-1}(\mathbf{y})} \\ &= \left[\frac{\partial V}{\partial \mathbf{x}} \left(\mathbf{J}_\psi(\mathbf{x})\right)^{-1} \mathbf{J}_\psi(\mathbf{x})\, f(\mathbf{x})\right]_{\mathbf{x}=\psi^{-1}(\mathbf{y})} = \left[\frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}}\right]_{x=\psi^{-1}(\mathbf{y})} = \dot{V}(\psi^{-1}(\mathbf{y})),\end{aligned} \tag{3}$$

where the second equality follows from the implicit function theorem. Therefore, the system after the change of coordinate (2) has a globally asymptotically stable equilibrium point $\mathbf{y}^* = \psi(\mathbf{x}^*)$. Moreover, since the diffeomorphism is bijective, the converse is also true: if there exists a Lyapunov function $\tilde{V}$ under coordinate $\mathbf{y}$, equilibrium point $\mathbf{x}^*$ is globally asymptotically stable.

3.3. Riemannian Manifolds and Natural Gradient Descent

From a geometric standpoint, a diffeomorphism $\psi : \mathbf{x} \mapsto \mathbf{y}$ can also help us understand the geometry of a Riemannian manifold in relation to Euclidean geometry. The geometry on an n-dimensional Riemmanian manifold can be defined by a Riemannian metric $\mathbf{G}_\psi : \mathbb{R}^n \to \mathbb{R}^{n\times n}_{++}$, which gives a notion of distance on the manifold [19].

To elaborate, let us define Euclidean goemetry on the co-domain, with coordinates $\mathbf{y}$. Consider a gradient descent dynamical system $\dot{\mathbf{y}} = \tilde{f}(\mathbf{y}) = -\nabla_{\mathbf{y}}\Phi(\mathbf{y})$ with a potential function $\Phi : \mathbb{R}^n \to \mathbb{R}$. Then, the change of coordinates defined by the diffeomorphism ψ provides a description of the same dynamics under $\mathbf{x}$ coordinate,

$$\dot{\mathbf{x}} = -\,\mathbf{G}_\psi(\mathbf{x})^{-1}\, \nabla_{\mathbf{x}}\Phi(\psi(\mathbf{x})) \quad := f_\psi(\mathbf{x}). \tag{4}$$

where the induced Riemannian metric in the domain is given by $\mathbf{G}_\psi(\mathbf{x}) = \mathbf{J}_\psi(\mathbf{x})^\top \mathbf{J}_\psi(\mathbf{x}) \in \mathbb{R}^{n\times n}_{++}$. The aforementioned dynamics is known as *natural gradient descent*, which is steepest descent on a Riemannian manifold [20], with respect to the potential function $\Phi \circ \psi$. The system (4) can generate sophisticated trajectories although the potential function Φ may only take a simple form. Moreover, if the potential function Φ is also positive definite,[2] convex, continuously differentiable, and radially unbounded, then the potential function Φ is a valid Lyapunov function for the gradient descent system. Therefore, both the gradient descent system and natural gradient descent system (4) admit a globally asymptotically stable equilibrium point.

2. Φ is strictly positive everywhere except $\mathbf{y}^*$, and $\Phi(\mathbf{y}^*) = 0$

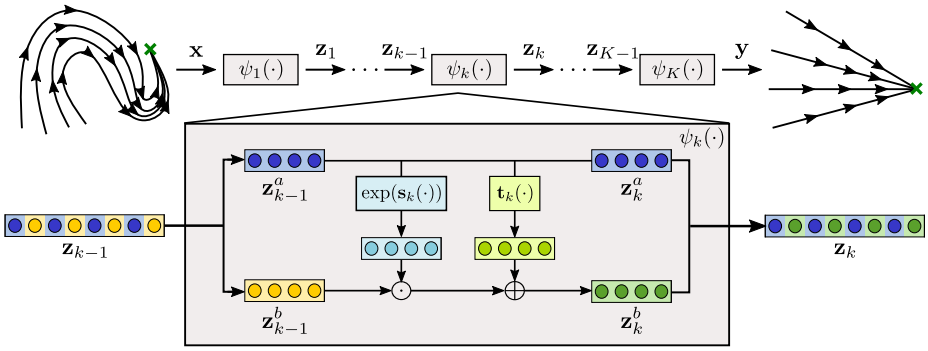

Figure 2: Architecture of the diffeomorphic mapping network.

Figure 1(a)–(b) shows the iso-contours of a potential function Φ which generate straight-line motions on the Euclidean space, and the corresponding potential function $\Phi \circ \psi$ on the Riemannian manifold. The isocontours show equidistant points to the goal in the Riemannian manifold, thus revealing the underlying geometry. Figure 1(c)–(d) shows the trajectories generated by the same underlying system observed in the corresponding coordinate systems.

4. Learning Stable Dynamics Using Diffeomorphisms

We view human demonstrations as goal-directed motions on an n-dimensional Riemannian manifold, governed by a stable dynamical system of the form (4). We view this dynamical system to be equivalent, under a change of coordinates, to another system defined on a *latent* space. Our key insight here is: *a diffeomorphism can warp a simple potential function into a more complicated one, and hence transform straight-line trajectories into severely curved motions.* As a result, the problem of learning stable dynamical systems reduces to a diffeomorphism learning problem.

4.1. Problem Statement

Assume the availability of N human demonstrations, each composed of T_i position-velocity pairs, denoted by $\{\{(\tilde{\mathbf{x}}_{i,t}, \dot{\tilde{\mathbf{x}}}_{i,t})\}_{t=1}^{T_i}\}_{i=1}^{N}$. We seek to find a dynamical system $\dot{\mathbf{x}} = f_\psi(\mathbf{x})$ that reproduces the demonstrations while ensuring stability. As illustrated in Section 3.3, we represent the system as the gradient descent system $\dot{\mathbf{y}} = -\nabla_{\mathbf{y}}\Phi(\mathbf{y})$ after a change of coordinates defined by ψ.

Although both the diffeomorphism ψ and the potential function Φ can shape the dynamical system (4), we view them as playing fundamentally different roles: the potential function Φ dictates the theoretical property of the dynamics, e.g. stability guarantees, while the diffeomorphism ψ provides expressivity to our hypothesis class. As a result, we specify a simple potential function $\Phi(\mathbf{y}) = \|\mathbf{y} - \mathbf{y}^*\|$, with $\mathbf{y}^* = \psi(\mathbf{x}^*)$. This potential function generates unit-velocity straight-line motions to the globally asymptotically stable equilibrium point $\mathbf{y}^*$. The diffeomorphism acts to deform these straight lines to arbitrarily curved motions converging to $\mathbf{x}^*$, where the demonstrations converge. With a parameterized diffeomorphism ψ_θ, the learning problem reduces to solving,

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \left\| \dot{\tilde{\mathbf{x}}}_{i,t} - f_{\psi_\theta}(\tilde{\mathbf{x}}_{i,t}) \right\|_2^2 \tag{5}$$

To avoid notation complexity, we will drop the subscript θ in the remainder of this paper.

4.2. A Class of Expressive Diffeormorphisms

By definition, a diffeomorphism is required to be both, *bijective* and *continuously differentiable*. To achieve this, we impose structure in the learning problem by realizing a diffeomorphism by a composition of K diffeomorphisms $\psi = \psi_1 \circ \psi_2 \circ \cdots \circ \psi_K$, with each diffeomorphism $\psi_k : \mathbb{R}^n \to \mathbb{R}^n$

given by a *coupling layer* [21]. Let $\mathbf{z}_k \in \mathbb{R}^n$ denote the output from the kth coupling layer ψ_{k-1}, i.e., $\mathbf{z}_k = \psi_k(\mathbf{z}_k)$, $\mathbf{z}_0 = \mathbf{x}$, and $\mathbf{z}_K = \mathbf{y}$. Each coupling layer, ψ_k involves splitting the input $\mathbf{z}_{k-1}$ followed by scaling and translating one of the parts. The `split` operation divides inputs into two parts, $\mathbf{z}^a_{k-1} \in \mathbb{R}^{\lfloor n/2 \rfloor}$ and $\mathbf{z}^b_{k-1} \in \mathbb{R}^{\lceil n/2 \rceil}$, constituting alternate input dimensions, whereby the pattern of alternation is reversed after each layer. Formally,

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}^a_k \\ \mathbf{z}^b_k \end{bmatrix} = \begin{bmatrix} \mathbf{z}^a_{k-1} \\ \mathbf{z}^b_{k-1} \odot \exp\left(s_k(\mathbf{z}^a_{k-1})\right) + t_k(\mathbf{z}^a_{k-1}) \end{bmatrix} := \psi_k(\mathbf{z}_{k-1}), \tag{6}$$

where $\odot$ denotes pointwise product and $\exp$ denotes pointwise exponential. The functions $s_k : \mathbb{R}^{\lfloor n/2 \rfloor} \to \mathbb{R}^{\lceil n/2 \rceil}$ and $t_k : \mathbb{R}^{\lfloor n/2 \rfloor} \to \mathbb{R}^{\lceil n/2 \rceil}$ are the parameterized scaling and translation functions, respectively. The coupling layer is a bijective affine mapping by construction. Since each mapping is bijective, the composed mapping ψ is also guaranteed to be bijective. Further, the composed mapping ψ is continuously differentiable as long as the scaling and translation functions in each coupling layer are continuously differentiable. This formulation of a bijection has been previously employed for density estimation by Dinh et al. [21], where the scaling and translation functions are given by convolutional residual networks.

In contrast to [21], for the scaling and translation functions, we use single-layer neural networks with the layer resembling an approximated kernel machine [22]. This special network structure leverage the advantages of both kernel machines and neural networks: (i) the kernel functions act as regularizers to enforce desired properties on the learned function, e.g. smoothness, which can further improve the sample efficiency of the learning algorithm, and (ii) as a parameterized model, the composed diffeomorphism can be efficiently trained using learning techniques for neural networks.

We use the Gaussian kernel given by $k(\mathbf{z}, \mathbf{z}') = \exp(-\frac{\|\mathbf{z}-\mathbf{z}'\|^2}{2l^2})$, with length-scale l. The Gaussian kernel restricts the hypothesis class to the class of C^∞ functions, imposing a stricter smoothness constraint than continuous differentiability, i.e. C^1. This is desirable since human motions are known to be maximizing smoothness (or minimizing jerk) [3]. We employ a Fourier feature approximation [22] to the vector-valued kernel machine with Gaussian kernel, given by,

$$g(\mathbf{z}) = \varphi(\mathbf{z})^\top \mathbf{w}, \qquad \text{with } \varphi(\mathbf{z}) = \sqrt{\frac{2}{m}} \begin{bmatrix} \cos(\mathbf{a}_1^\top \mathbf{z} + \mathbf{b}_1) \\ \cos(\mathbf{a}_2^\top \mathbf{z} + \mathbf{b}_2) \\ \vdots \\ \cos(\mathbf{a}_m^\top \mathbf{z} + \mathbf{b}_m) \end{bmatrix} \otimes \mathbf{I}_{n \times n}, \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^{m \times n}$ constitutes the learnable parameters. The projection vector $\varphi(\mathbf{z})$ is composed of m randomly sampled Fourier features such that the kernel matrix is given by $K(\mathbf{z}, \mathbf{z}') \approx \varphi(\mathbf{z})^\top \varphi(\mathbf{z}')$. Concretely, the coefficients $\{\mathbf{a}_i\}_{i=1}^m$ are sampled from a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}, l^{-2}\mathbf{I})$, and the bias terms $\{\mathbf{b}_i\}_{i=1}^m$ sampled from a uniform distribution $\mathrm{U}(0, 2\pi)$. Under the formulation in (7), we define $s_k(\cdot) = \varphi(\cdot)^\top \mathbf{w}_{s_k}$ and $t_k(\cdot) = \varphi(\cdot)^\top \mathbf{w}_{t_k}$. The set of parameters in the diffeomorphism learning problem in (5) is therefore given by $\theta := \{\mathbf{w}_{s_k}, \mathbf{w}_{t_k}\}_{k=1}^K$.

4.3. Practical Considerations

Regarding the choice of potential functions, there are a few alternatives, including various soft versions of ℓ_2-norm, and quadratic potential functions, and other norms. In this paper, we view the potential function as dictating the stability properties while the diffeomorphisms provides expressivity to our hypothesis class. We tested the proposed ℓ_2-norm, a soft version of ℓ_2-norm, and also a quadratic potential function, and we did not observe significant differences in performance. This observation also shows that our parameterization of diffeomorphsim is expressive enough so that the choice of potential function is not significant as long as it provides desirable stability guarantees.

To learn (5) through back-propogation, the Jacobian of the diffeomorphism can be calculated analytically from (6) and (7), or through auto-differentiation packages, e.g. pytorch [23].

5. Experimental Results

We evaluate our approach on the LASA dataset [9] as well data collected for multiple tasks on a Franka Emika robot [10]. For all our experiments, we start the learning procedure by normalizing the demonstrations to stay in the range $[-0.5, 0.5]$. This allows us to fix the hyperparameters of our model irrespective of the scale of the data. In our experiments, we use $K = 10$ coupling layers constituting of $m = 200$ random Fourier features with length-scale $l = 0.45$. For consistency in results, the diffeomorphism is always initialized with an identity map. We optimize using ADAM [24] with default hyperparameters, alongside a learning rate of 1×10^{-4} and an ℓ_2-regularization on the weights θ with coefficient 1×10^{-8}.

The LASA dataset [9] consists of a library of 30 two-dimensional handwritten letters, each with 7 demonstrations. The scale of the dataset is $100mm \times 100mm$. For each letter, we find a diffeomorphism using (5). Fig. 5 shows the vector fields as governed by (4) on a subset of letters. In all the plots, the rollouts (in red) closely match the demonstrations (in white), coming to rest at the goal. Furthermore, due to the structure imposed in learning, the dynamical system generalizes smooth and stable motions throughout the state-space. Also shown in Fig. 4 are isocontours of the potential function $\Phi(\psi(x))$. For quantitative evaluations, we employ three error metrics: *root mean squared error (RMSE)*, *dynamic time warping distance (DTWD)* [25], and *Frechet disance (FD)* [26]. These metrics evaluate performance of our approach in terms of its capability to reproduce the demonstrated motions. Fig. 3 reports these metrics, in millimeters, evaluated over 210 demonstrations (7 letters $\times$ 30 demonstrations). Each aforementioned metric focuses on different aspects of the motions. RMSE penalizes both spatial and temporal misalignment between demonstrated and reproduced motions. On the other hand, DTWD and FD disregard time misalignment, and instead focus solely on the spatial misalignment between motions. Since DTWD between any two trajectories is a time-aggregated measure of error, we report *average* DTWD, found by dividing the DTWD by the number of points T_i in a trajectory. In Fig. 3, the median and mean errors are observed to be small relative to the scale of the data, signifying that the learned dynamical systems are able to accurately reproduce most motions. However, higher errors are occasionally observed, accounting for outliers. This is mostly due to intersecting demonstrations which can not be modeled by a first-order dynamical system.

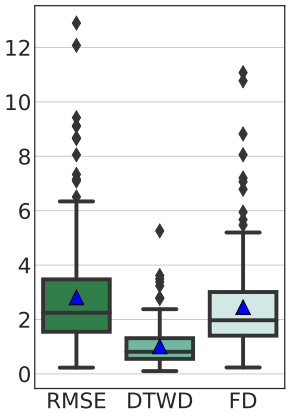

Figure 3: Box plots for RMSE, average DTWD, and FD, in millimeters, evaluated over the LASA dataset. The overlayed blue triangles are the means for each metric.

For demonstrations collected on a Franka Emika robot, we evaluate on two tasks: *door reaching*, and *drawer closing* [10]. Each task dataset consists of 6 three-dimensional end-effector motions collected by physically guiding the robot. The *door reaching* task required the robot to start from inside a cabinet and reach the door handle, while the *drawer closing* task required reaching a drawer handle and pushing the drawer close. Fig. 6 shows the tasks, demonstrated motions (blue), as well as the reproduced motions (red). Our learning approach is observed to accurately reproduce three-dimensional motions collected on a real robot.

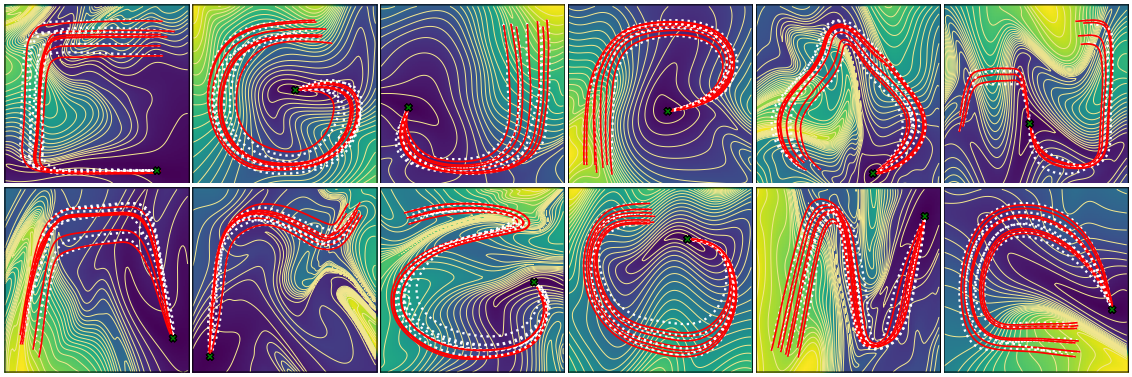

Figure 4: Isocontours of the potential function on LASA dataset. Overlayed are the demonstrations (white) and the reproductions (red). The reproductions cut across the isocontours.

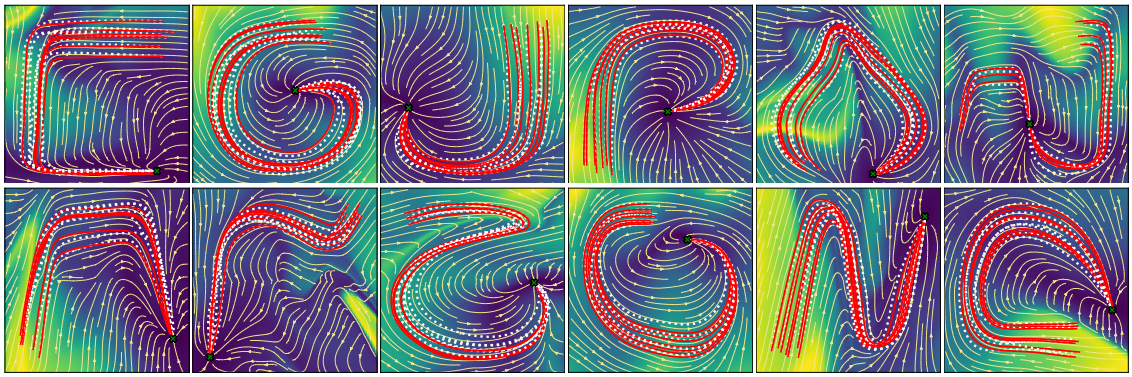

Figure 5: Vector fields of the dynamics learned on the LASA dataset, alongside demonstrations (white) and reproductions (red). The reproductions are governed by the natural gradient descent dynamics.

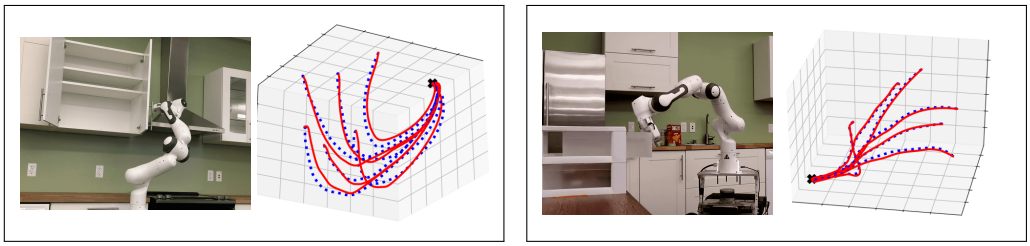

Figure 6: The *door reaching* (left) and *drawer closing tasks* (right). The demonstrations are plotted in blue while the reproductions are in red.

6. Conclusion

We have presented SDS-EF, an approach for learning motion skills from a few human demonstrations. SDS-EF encodes complex human motions as generated from a dynamical system, linked under a learnable diffeomorphism, to a simple gradient-descent dynamical system on a latent space. A class of parameterized diffeomorphisms is proposed to learn a wide range of motions and generalize to different tasks with minimal parameter tuning. Experimental validation on a handwriting dataset and data collected on a real robot is provided to show the efficacy of the proposed approach.

Acknowledgments

This work was supported in part by NVIDIA Research.

References

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[2] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[3] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[4] S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[5] S Mohammad Khansari-Zadeh and Aude Billard. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

[6] Harish Chaandar Ravichandar, Iman Salehi, and Ashwin P Dani. Learning partially contracting dynamical systems from demonstrations. In *Conference on Robot Learning (CoRL)*, pages 369–378, 2017.

[7] Klaus Neumann and Jochen J Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70:1–15, 2015.

[8] Nicolas Perrin and Philipp Schlehuber-Caissier. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems & Control Letters*, 96:51–59, 2016.

[9] Seyed Mohammad Khansari-Zadeh. Lasa handwriting dataset. https://bitbucket.org/khansari/lasahandwritingdataset.

[10] M Asif Rana, Anqi Li, Harish Ravichandar, Mustafa Mukadam, Sonia Chernova, Dieter Fox, Byron Boots, and Nathan Ratliff. Learning reactive motion policies in multiple task spaces from human demonstrations. In *Conference on Robot Learning (CoRL)*, 2019.

[11] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[12] Harish chaandar Ravichandar and Ashwin Dani. Learning position and orientation dynamics from demonstrations via contraction analysis. *Autonomous Robots*, 43(4):897–912, 2019.

[13] Bernhard Schölkopf, Christopher JC Burges, Alexander J Smola, et al. *Advances in kernel methods: support vector learning*. MIT press, 1999.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[16] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[17] Hassan K Khalil. Nonlinear systems. *Upper Saddle River*, 2002.

[18] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

[19] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[20] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[22] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[26] M Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.