Learning on Manifolds: Universal Approximations Properties using Geometric Controllability Conditions for Neural ODEs

Karthik Elamvazhuthi KELAMVAZHUTHI@ENGR.UCR.EDU
Xuechen Zhang XZHAN394@UCR.EDU
Samet Oymak OYMAK@ECE.UCR.EDU
Fabio Pasqualetti FABIOPAS@ENGR.UCR.EDU
900 University Avenue, Riverside, CA, USA, 92521

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

In numerous robotics and mechanical engineering applications, among others, data is often constrained on smooth manifolds due to the presence of rotational degrees of freedom. Common data-driven and learning-based methods such as neural ordinary differential equations (ODEs), however, typically fail to satisfy these manifold constraints and perform poorly for these applications. To address this shortcoming, in this paper we study a class of neural ordinary differential equations that, by design, leave a given manifold invariant, and characterize their properties by leveraging the controllability properties of control affine systems. In particular, using a result due to Agrachev and Caponigro on approximating diffeomorphisms with flows of feedback control systems, we show that any map that can be represented as the flow of a manifold-constrained dynamical system can also be approximated using the flow of manifold-constrained neural ODE, whenever a certain controllability condition is satisfied. Additionally, we show that this universal approximation property holds when the neural ODE has limited width in each layer, thus leveraging the depth of network instead for approximation. We verify our theoretical findings using numerical experiments on PyTorch for the manifolds S^2 and the 3-dimensional orthogonal group $SO(3)$, which are model manifolds for mechanical systems such as spacecrafts and satellites. We also compare the performance of the manifold invariant neural ODE with classical neural ODEs that ignore the manifold invariant properties and show the superiority of our approach in terms of accuracy and sample complexity.

Keywords: Neural Ordinary Differential Equations, Universal Approximation, Manifold-Valued Data, Controllability

1. Introduction

Due to complexity of robotics systems, data based modeling has become an important technique for predictive and control tasks in many robotics applications Brunke et al. (2022). One important consideration when using data based methods in such applications is the presence of rotational degrees of freedom in mechanical systems Choset et al. (2005); Bullo and Lewis (2019). These rotational degrees of freedom require that most robotics systems' states be constrained to on a lower dimensional surface, or manifold, embedded in a higher dimensional Euclidean space. Therefore, it is essential that any black box model that is used in these applications respect the manifold constraints arising from the physics of the system, without which the model can provide nonphysical outputs, affecting their applicability. Beyond the concern of nonphysical nature of outputs generated by classical black box models, another reason to take this information of lower dimensionality of the state space into account is that it may also reduce the number of parameters required to fit the model to data, and thus mitigate the curse of dimensionality.

A class of neural networks that has now become standard in machine learning applications is residual neural networks (ResNets) He et al. (2016). However, ResNets do not necessarily satisfy manifold constraints as desired in robotics applications. For this reason, there has been some effort recently on generalizing ResNets to manifold-valued data. One way to generalize ResNets on manifolds is by considering them as numerical discretizations of parameterized ordinary differential equations (ODEs) called neural ODEs E (2017); Haber and Ruthotto (2017); Chen et al. (2018). For example, Falorsi and Forré (2020); Lou et al. (2020) present variations of neural ODEs on manifolds, from which one can derive a ResNet respecting geometric constraints, by appropriately discretizing the neural ODE. This point of view, of ResNets as numerical discretizations of neural ODEs, was originally developed to open tools from dynamical systems and control theory to design and train more stable versions of ResNets. This perspective has also been fruitful in understanding the approximation properties of deep neural networks. For example, in Tabuada and Gharesifard (2020) investigated the universal approximation properties of deep ResNets using geometric control techniques. In Agrachev and Sarychev (2021), the authors construct a new class control systems that, similar to neural ODEs, have the capability to approximate diffeomorphisms.

While generalization of neural ODEs to manifolds have been presented in Falorsi and Forré (2020); Lou et al. (2020), there has been no work on the approximation capabilities of such neural ODEs, for learning manifold-valued maps. In this paper, we study the approximation property of a class of neural ODEs of manifolds from the point of view of controllability properties of control-affine systems, a popular class of systems studied in the control theory literature. For the purpose of machine learning problems, the controllability problem of interest is whether the weights of the control system can be used to control the flow of the ODE in such way that the flow, at a given time instant, is close to the map that is required to be approximated. This non-traditional nature of control problem makes the question of universal approximation challenging to address, since one is not just required to transfer a given initial condition of the control system to another, but simultaneously transfer a collection of initial conditions to a target collection of final values using the same control inputs. We show that a controllability condition on the vector-fields, well known in the form of *bracket generating condition* in geometric control theory literature, suffices to guarantee universal approximation of a sufficiently rich class of diffeomorphisms, including those that can be represented using the flow of a manifold-constrained dynamical system. We also use numerical simulations to verify the approximation properties of the neural ODEs on manifolds and compare their performance with classical neural ODEs that do not take into account the manifold valued nature of the data.

2. Notation

In this section, we introduce some notation that will be used throughout the paper. The open ball around $x \in \mathbb{R}^d$ will be denoted by $B_R(x) := \{r \in \mathbb{R}^d; |x - y| < R\}$. Let $\mathcal{M} \subseteq \mathbb{R}^d$ denote a compact smooth manifold without boundary Lee (2013). We will denote by $\mathrm{Diff}_0(\mathcal{M})$ the set of diffeomorphisms on $\mathcal{M}$ that are *isotopic* to the identity map I on $\mathcal{M}$. By *isotopic*, we mean that, $X \in \mathrm{Diff}_0(\mathcal{M})$ if there exist a smooth map $Y : [0,1] \times \mathcal{M} \to \mathcal{M}$ such that $Y(0,\cdot) = I$ and $Y(1,\cdot) = X$. A function $f : \mathbb{R}^d \to \mathbb{R}^d$ is *locally Lipschitz*, if for each $R > 0$, there exists $L_R > 0$ such that $|f(x) - f(y)| \leq L_R|x - y|$ for all $x, y \in B_R(x)$. Given two vector-fields $f : \mathbb{R}^d \to \mathbb{R}^d$

and $g : \mathbb{R}^d \to \mathbb{R}^d$, the *Lie-Bracket* between the two vector-fields is denoted by $[f, g]$ and given by

$$[f,g]^i(x) = \sum_{j=1}^{d} f^j(x)\frac{\partial g^i}{\partial x^i}(x) - g^j(x)\frac{\partial f^i}{\partial x^i}(x) \tag{1}$$

for all $x \in \mathbb{R}^d$, where $f^i(x)$ refers to the i^{th} element of the vector $f(x)$. Let $\mathcal{V} = \{f_1, ..., f_m\}$ be a finite number of vector fields. Setting $\mathcal{V}_0 = \mathcal{V}$, for each $i \in \mathbb{Z}_+$, we define in an iterative manner the set of vector-fields $\mathcal{V}^i = \{[g, h]; g, h \in \mathcal{V}^j,\ j = 0, .., i-1\}$. We will denote by $\mathrm{Lie}_x\mathcal{V}$ the set of vectors defined by,

$$\mathrm{Lie}_x\mathcal{V} = \mathrm{span}\{g(x); g \in \mathcal{V}^i,\ i = 0, 1, ...\}. \tag{2}$$

We will say that the collection of vector fields $\mathcal{V}$ is *bracket generating* on $\mathcal{M}$ if $\mathrm{Lie}_x\mathcal{V} = T_x\mathcal{M}$ for all $x \in \mathcal{M}$, where $T_x\mathcal{M}$ is the tangent space of the manifold at $x \in \mathcal{M}$.

3. Problem Formulation

In this section, we formulate the main problem being addressed in the paper. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a given *activation function*. We define $\Sigma : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\Sigma(x) = [\sigma(x_1), ..., \sigma(x_d)]^T \tag{3}$$

An example of the class of activation functions that we consider is *sigmoidal functions* with globally bounded derivatives. An activation function σ is said to *sigmoidal* if its range lies in $[0, 1]$,

$$\lim_{x\to-\infty} \sigma(x) = 0 \text{ and } \lim_{x\to\infty} \sigma(x) = 1. \tag{4}$$

One such sigmoidal function is

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{5}$$

Let $\mathcal{M} \subseteq \mathbb{R}^d$ be a smooth compact manifold that is known. The learning problem that we consider is the following. Suppose there is an unknown map $\Psi : \mathcal{M} \to \mathcal{M}$. We wish to learn the map Ψ using samples of input output data $\{(x_1, y_1), ..., (x_n, y_n)\}$ where x_i are distributed according to some density function $\rho(x)$. One way construct an approximation is using time-one map $X : \mathbb{R}^d \to \mathbb{R}^d$of the neural ODE,

$$\dot{x}(t) = A(t)\Sigma(W(t)x + b(t)) \quad x(0) = x_0 \tag{6}$$

where $A : [0, T] \to \mathbb{R}^{d\times d}$, $W : [0, T] \to \mathbb{R}^{d\times d}$ and $b : [0, T] \to \mathbb{R}^d$ are the weights for the neural network. The time-one map X is defined by setting $X(x_0) = x(1)$ for each $x_0 \in \mathbb{R}^d$. The feasibility of approximating Ψ using X has been shown in Tabuada and Gharesifard (2020), for a large class of activation functions σ.

A drawback of the previous formulation is that the learning problem does not use prior knowledge of the manifold $\mathcal{M}$. Hence, $x_0 \in \mathcal{M}$ does not necessarily imply that $X(x_0) \in \mathcal{M}$. This is critical in applications such as modeling of mechanical systems where the physical state of the system is known to be confined to a given manifold $\mathcal{M}$. To use this knowledge we propose a generalization of neural ODE inspired by control affine systems in control theory.

To ensure that the output data generated by neural ODE stays on $\mathcal{M}$ we choose some m vector-fields $g_i : \mathbb{R}^d \to \mathbb{R}^d$ that are smooth and satisfy $g_i(x) \in T_x\mathcal{M}$ for all x, where $T_x\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at x.

Then we consider the following manifold invariant neural ODE

$$\dot{x}(t) = \sum_{i=1}^{m} a_i(t) g_i\Big(x(t)\Big)\sigma\Big(w_i^T(t)x(t) + b_i(t)\Big) \tag{7}$$

$$x(0) = x_0 \tag{8}$$

where $a_i : [0,1] \to \mathbb{R}$, $w_i : [0,1] \to \mathbb{R}^d$ and $b : [0,1] \to \mathbb{R}$ are the weight parameters of the generalized neural ODE. Let $X : \mathcal{M} \to \mathcal{M}$ be the flow map generated by the neural ODE (8), defined by setting $X(x_0) = x(1)$ and all $x_0 \in \mathcal{M}$.

Given this generalized neural ODE, the approximation problem that we consider is the following.

Problem 1 *Given $\Psi : \mathcal{M} \to \mathcal{M}$ and $\epsilon > 0$ do there exist weight parameters (a_i, w_i, b_i) such that the time-one map X of the neural ODE on manifold (8) satisfies.*

$$|X(x) - \Psi(x)| \leq \epsilon \tag{9}$$

for all $x \in \mathcal{M}$.

In order to address the above problem, we will need some additional assumptions. The first assumption that we will make is on the vector fields $\{g_1, ..., g_m\}$

Assumption 2 *We make the following assumptions on the vector field g_i*

1. *The vector fields $g_i : \mathbb{R}^d \to \mathbb{R}^d$ are smooth.*
2. *The vector fields g_i satisfy $g_i(x) \in T_x\mathcal{M}$ for all $x \in \mathcal{M}$.*

Let $\mathcal{U}$ be the subset of functions defined by

$$\mathcal{U} = \big\{u : [0,T] \times \mathbb{R}^d \to \mathbb{R} : u \text{ is piecewise constant in time and } u(t,\cdot) \text{ is locally Lipschitz for each } t \in [0,1]\big\}$$

In addition to this, we will need some mild assumptions on the activation function $\sigma : \mathbb{R} \to \mathbb{R}$. For this purpose, let us define the set of functions

$$\mathcal{F} = \bigcup_{m \in \mathbb{Z}_+} \{\sum_{i=1}^{m} \alpha_i \sigma(w_i^T x + b_i) \mid \alpha_i \in \mathbb{R}, w_i \in \mathbb{R}^d, b_i \in \mathbb{R}\}.$$

Note that the set $\mathcal{F}$ is the set of arbitrarily wide single-hidden layer neural networks. Using this set we define the subset of $\mathcal{U}$ that take values over $\mathcal{F}$,

$$\mathcal{N} = \big\{u \in \mathcal{U} : u(t,\cdot) \in \mathcal{F} \text{ for each } \ t \in [0,1]\big\}$$

Similarly, we define the set of functions

$$\begin{aligned}\mathcal{A} = \big\{ & u \in \mathcal{U} : \text{for some piecewise constant functions } \alpha : [0,1] \to \mathbb{R}, \\ & w : [0,1] \to \mathbb{R}^d, b : [0,1] \to \mathbb{R}, \\ & u(t,\cdot) = \alpha(t)\sigma(w^T(t)x + b(t)) \in \mathcal{F} \text{ for each } \ t \in [0,1]\big\}\end{aligned}$$

We will make the following assumptions on the activation function σ.

Assumption 3 *We make the following assumptions:*

1. ***(Regularity)*** *The activation function σ is globally Lipschitz. That is, there exists $K > 0$ such that*
$$|\sigma(x) - \sigma(y)| \leq K|x - y|, \tag{10}$$
for all $x, y \in \mathbb{R}^d$.
2. ***(Density of superpositions)*** *Given a function $f \in C(\mathbb{R}^d; \mathbb{R})$, and $R > 0$, for every $\delta > 0$, there exists a function $h \in \mathcal{F}$ such that*
$$|f(x) - h(x)| + \sum_{i=1}^{d} |\frac{\partial f}{\partial x_i}(x) - \frac{\partial h}{\partial x_i}(x)| < \delta.$$
for (Lebesgue) almost every $x \in B_R(x)$.

The above assumption is satisfied by both, the Sigmoid activation function, as well as the Rectified Linear unity (ReLu) activation function.

4. Analysis

Before we begin our analysis we make the standard observation that due to Assumption 2, given $u_i \in \mathcal{U}$ for each $i = 1, ..., m$, for the following ODE

$$\begin{aligned}\dot{x} = \sum_{i=1}^{m} u_i(t, x(t)) g_i\Big(x(t)\Big) \\ x(0) = x_0,\end{aligned} \tag{11}$$

the solution $x(t)$ is in fact global and lies in $\mathcal{M}$. This result follows from Lee (2013).

Theorem 4 *Given Assumption 2, and functions $u_i \in \mathcal{U}$, suppose that $x(0) \in \mathcal{M}$, then a unique solution $x : [0,1] \to \mathbb{R}^d$ to the differential equation (11) exists. Moreover, $x(t) \in \mathcal{M}$ for all $t \in [0,1]$.*

The general idea behind the main result, stated in Theorem 5, is the following. There exist a large class of maps that can be approximated using the flow of the ODE (11) using an appropriate choice of functions u_i for $i = 1, ..., m$, as long as the vector-fields $\mathcal{V} = \{g_1, ..., g_m\}$ satisfy a controllability condition. This result is due to Agrachev and Caponigro (2009). However, given functions u_i, it is not in general possible to approximate the vector field of (11) using the the vector field of the neural ODE (8) in any classical sense (for example, in the uniform norm or L_2 norm), due to the fact that $\mathcal{A}$ is a much smaller set that $\mathcal{U}$. Despite this, using weak convergence arguments, we can use a sequence of vector-fields of the form of the the neural ODE (8) to *weakly converge* to functions u_i. This, in turn, results in uniform convergence of the flow of neural ODE (8) to that of the differential equation (11), thus allowing us to extend the approximation result of Agrachev and Caponigro (2009) to flows of the neural ODE (8).

Theorem 5 *Given that Assumption 3 holds, suppose that $Y \in \text{Diff}_0(\mathcal{M})$ and $\epsilon > 0$. Additionally, suppose that $\mathcal{V}$ satisfies the Bracket generating property on $\mathcal{M}$. That is, $Lie_x\mathcal{V} = T_x\mathcal{M}$ for all $x \in \mathcal{M}$. Let $X : \mathcal{M} \to \mathcal{M}$ be the flow map generated by the neural ODE (8), defined by setting $X(x_0) = x(1)$ and all $x_0 \in \mathcal{M}$. Then there exist piecewise constant parameters $a_i \in \mathbb{R}$, $w_i : [0,1] \to \mathbb{R}^d$ and $b_i : [0,1] \to \mathbb{R}^d$ such that*

$$\sup_{x \in \Omega} |X(x) - Y(x)| \leq \frac{\epsilon}{2}.$$

for all $x \in \mathcal{M}$.

Proof Since from $Y \in \text{Diff}_0(\mathcal{M})$ and due the bracket generating assumption that $Lie_x\mathcal{V} = T_x\mathcal{M}$, it follows from (Agrachev and Caponigro, 2009, Corollary 3.1), that there exists $u \in \mathcal{U}$ such that the corresponding time one map X satisfies,

$$\sup_{x \in \Omega} |X(x) - Y(x)| \leq \frac{\epsilon}{2}.$$

The result then follows by Proposition 6 (stated in the Appendix) by approximating u weakly using an element in $\mathcal{A}$. ■

Formally, the above approximation result can be understood as follows. The class of maps isotopic to the identity include flows generated by dynamical systems. Suppose the map Y is the time one map of the ODE,

$$\dot{x}(t) = f(t, x(t)) \tag{12}$$

for some time-dependent smooth vector field $f : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. Additionally, suppose that that $x(0) \in \mathcal{M}$ implies $x(t) \in \mathcal{M}$ for all $t \in \mathcal{M}$, then Y is isotopic to the identity and hence, can be approximated by the flow of the neural ODE (8). The bracket generating assumption can be interpreted as requiring that the following version of the controllability problem,

$$\dot{x} = \sum_{i=1}^{m} u_i(t) g_i(x(t))$$
$$x(0) = x_0, \quad x(1) = x_1$$

is well-posed for all $x_0, x_1 \in \mathcal{M}$. That is, if for every $x_0, x_1 \in \mathcal{M}$, there exist time-dependent functions $u_i(t)$ such that $x(0) = x_0,\ \ x(1) = x_1$, then the flow of the manifold invariant neural ODE (8) can be used to approximate any element of $\text{Diff}_0(\mathcal{M})$.

5. Numerical Results

In this section, we numerically test the approximation properties of the neural ODE on manifold, using two examples. For each example, we will compare the performance of our manifold constrained neural ode with the performance of the classical neural ODE (4). For this purpose, we discretize the problem in time to solve the problem in Pytorch. For this purpose, we introduce the time discretized version of the problem for the classical neural ODE and then for each case, the manifold invariant neural ODE.

Suppose we are given input-output data $\{(x_0^1, y^1), ..., (x_0^p, y^p))\} \subset \mathbb{R}^d \times \mathbb{R}^d$ we wish to solve the problem,

$$\min_{\Theta} \frac{1}{P} \sum_{j=1}^{P} \|x_M^j - y^j\|_2^2 + \frac{\lambda \Delta t}{2} \sum_{N=1}^{M} \|\Theta_N\|_2^2 \tag{13}$$

with $\Theta_N = \{A_N, W_N, b_N\} \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d} \times \mathbb{R}^d$ subject to constraints arising from the forward Euler discretization of (6), which is the M-layer deep Resnet,

$$x_{N+1}^j = x_N^j + \Delta t A_N \Sigma(W_N x_N^j + b_N) \text{ for } N = 0, ..., M, \;\; j = 1, ..., P. \tag{14}$$

Here Σ is as defined in (3), $A_N \in \mathbb{R}^{d\times d}$, $W_N \in \mathbb{R}^{d\times d}$ and $b_N \in \mathbb{R}^d$, and $\Delta t = \frac{1}{M}$.

Example 1: Supervised Learning on the Sphere In this example we take the manifold $\mathcal{M} = \mathbb{S}^2 = \{x \in \mathbb{R}^3; x^T x = 1\} \subseteq \mathbb{R}^3$, the two dimensional sphere. We define the matrices,

$$B_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \tag{15}$$

Let $g_i(x) = B_i x$ for all $x \in \mathbb{R}^d$ for $i = 1, 2$. It can be verified that

$$B_1 B_2 - B_2 B_1 = B_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{16}$$

and hence that $[g_1, g_2](x) = g_3(x) = B_3 x$ for all $x \in \mathbb{S}^2$. We consider the following neural ODE on the sphere,

$$\dot{x}(t) = \sum_{i=1}^{2} a_i(t) B_i x(t) \sigma\Big(w_i^T(t) x(t) + b_i(t)\Big) \tag{17}$$

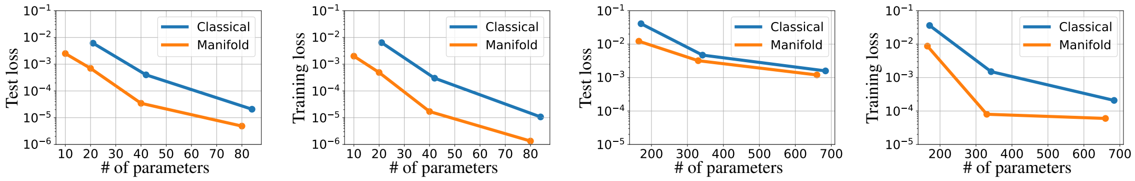
$$x(0) = x_0 \tag{18}$$

By construction $g_1(x) = B_1 x, g_2(x) = B_2 x \in T_x\mathbb{S}^2$ for all $x \in M$. Define

$$f_i(t, x, \Theta^i(t)) = a_i(t)\sigma(w_i^T(t)x + b_i(t))$$

for $x \in \mathbb{R}^3$ and $\Theta^i(t) = \{a_i(t), w_i(t), b_i(t)\} \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}$, for each t. The discretized problem that we solve is the following,

$$\min_{\Theta} \frac{1}{P} \sum_{j=1}^{P} \|x_M^j - y^j\|_2^2 + \frac{\lambda \Delta T}{2N} \sum_{i=1}^{2} \sum_{j=1}^{N} \|\Theta_N\|_2^2 \tag{19}$$



(a) exp1: test loss (b) exp1: training loss (c) exp2: test loss (d) exp2: training loss

Figure 1: (a) and (b) show the results of example 1. (c) and (d) show the results of example 2.

subject to the geometric Euler discretization Hairer et al. (2006) of the ODE (23), a M-layer manifold invariant Resnet,

$$x^j_{N+1} = e^{\Delta t f_1(t,x^j_N,\Theta^1_N)B_1 + f_2(t,x^j_N,\Theta^2_N)B_2} x^j_N$$

for $N = 0, ..., M$. In this example, data is generated using the following ODE,

$$\begin{gathered} \dot{x}(t) = x_2(t) g_1(x(t)) + x_3(t) g_3(x(t)); \\ x(0) = x^j_0 \end{gathered} \tag{20}$$

The geometric discretization using matrix exponentials ensures that the solution of the discrete system remains on the manifold for every time step N Hairer et al. (2006).

Example 2: Supervised Learning on the 3D rotation group In this example with consider the 3 dimensional orthogonal groups $\mathcal{M} = SO(3) = \{R \in \mathbb{R}^{3\times3} = R^T R = I, \ \det(R) = 1\}$

$$B_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ B_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \ B_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{21}$$

In this case we take $g_i(X) = B_i X$ for all $X \in \mathbb{R}^{3\times3}$ for $i = 1, 2, 3$. We define the neural ODE in the form of a matrix differential equation

$$\dot{X}(t) = \sum_{i=1}^{3} a_i(t) B_i X(t) \sigma\Big(\langle w_i(t), X(t) \rangle + b_i(t) \Big) \tag{22}$$

$$X(0) = X_0 \in \mathbb{R}^{3\times3} \tag{23}$$

By construction $g_i(X) = B_i X \in T_X SO(3)$ for all $X \in M$. Define

$$f_i(t, X, \Theta^i(t)) = a_i(t) \sigma(\langle w_i(t), X \rangle + b_i(t))$$

for $X \in \mathbb{R}^{3\times3}$ and $\Theta^i(t) = \{a_i(t), w_i(t), b_i(t)\}$. The discretized problem that we solve is the following,

$$\min_{\Theta} \frac{1}{P} \sum_{j=1}^{P} \|X^j_M - Y^j\|_2^2 + \frac{\lambda \Delta t}{2} \sum_{i=1}^{2} \sum_{j=1}^{M} \|\Theta_N\|_2^2 \tag{24}$$

subject to M layer manifold Resnet,

$$X_{N+1}^{j} = e^{\Delta t f_1(t, X_N^j, \Theta_N^1) B_1 + \Delta t f_2(t, X_N^j, \Theta_N^2) B_2 + \Delta t f_3(t, X_N^j, \Theta_N^3) B_3} X_N^j$$

for $N = 0, ..., M$.

In this example, data is generated using the following ODE,

$$\dot{X}(t) = \mathrm{Tr}(X^2 + 1)\Big(g_1(X) + g_2(X) + g_3(X(t))\Big); \tag{25}$$

$$X(0) = X_0^j \tag{26}$$

where Tr denotes the Trace of the matrix.

Implementation

For the experiments, we compare the manifold ODE with classical neural ODE. We use the sigmoid function as the activation function σ in all experiments. All the models were trained with standard stochastic gradient descent (SGD) using learning rate 10, and momentum 0.9. The learning rate decays at epochs 500,1000,2000,4000,6000 with a factor 0.8. We started training with a large learning rate to avoid getting stuck into local minima. The classical ODE in all cases needed much more parameters per layer. For Example 1, there are 10 parameters per layer for manifold invariant ODE while there are 21 for classical one. For Example 2, the value is 33 and 171. For a fair comparison, in example 1, we use $M = [1, 2, 4]$ layers for classical ODE and $M = [1, 2, 4, 8]$ layers for manifold one. In example 2, we use $M = [1, 2, 4, 8]$ for classical ODE and $M = [5, 10, 20]$ for manifold ODE.

Discussion

The training loss and test loss for the case over the sphere is shown in Figure (1) a-b, respectively. The training loss and test loss for the case over the $SO(3)$ is shown in Figure (1) c-d, respectively. Compared to the classical ODE (blue curve), the manifold ODE (orange curve) could always achieve low loss with smaller number of parameters. The result verifies that manifold ODE could truly reduce the number of parameters required to fit the model to data, and mitigate the curse of dimensionality. The greater difference can be observed in the test losses, indicated the superior capability of manifold-invariant neural ODE in terms of generalization.

6. Conclusion

We presented a class of neural ODEs for which a given manifold is invariant, and studied their universal approximation properties from the perspective of controllability analysis. Additionally, we use numerical simulations to verify that in terms, of sample complexity and accuracy the neural ODEs on manifolds show superior performance. Possible future directions of work include qualitatively establishing the formal sample complexity estimates of these neural ODEs, and extending them to situations when the manifold $\mathcal{M}$ is unknown.

Appendix A. Supplementary results

For the results stated in this section we will need $L^1(0, 1; \mathbb{R}^d)$, the set of vector-valued measurable functions that are integrable over $(0, 1)$. We will denote $L^1(0, 1; \mathbb{R}^d)$ by $L^1(0, 1)$ when $d = 1$. We will say that a sequence $(f^n)_{n=1}^{\infty}$ in $L^1(0,1;\mathbb{R}^d)$ weakly converges to $f \in L^1(0,1;\mathbb{R}^d)$ if

$$\lim_{n\to\infty} \int_0^1 \langle \phi(x), f^n(x) \rangle dx = \int_0^1 \langle \phi(x), f(x) \rangle dx$$

for all $\phi \in L^{\infty}(0,1;\mathbb{R}^d)$, where $\langle \cdot, \cdot \rangle$ denotes the usual inner product on $\mathbb{R}^d$. Given this definition, we have the following result.

Lemma 6 *Given Assumption 3, suppose that $(u_i^n)_{n=1}^{\infty}$ is a sequence of functions in $\mathcal{U}$ such that $(u_i^n(x,\cdot))_{n=1}^{\infty}$ is weakly converging to $u_i(x,\cdot)$ in $L^1(0,1)$ for each $x \in \mathcal{M}$ and each $i \in \{1,...,m\}$. Suppose additionally that the sequence of functions $(u_i^n)_{n=1}^{\infty}$ is bounded and Lipschitz with a common upper bound and common Lipschitz constant on $\mathcal{M}$. Let $X^n : \mathcal{M} \to \mathcal{M}$ be the map given by $X^n(x_0) = x^n(1)$ for all $x_0 \in \mathcal{M}$, where $x^n(t)$ is the solution of the differential*

$$\dot{x}^n(t) = \sum_{i=1}^{m} u_i^n(t, x^n(t)) g_i\Big(x^n(t)\Big) \tag{27}$$

$$x^n(0) = x_0^n \tag{28}$$

Then $(X^n)_{n=1}^{\infty}$ is uniformly converging to X on $\mathcal{M}$.

Proposition 7 *Given Assumption 3, suppose $u_i \in \mathcal{U}$. Then, for every $\epsilon > 0$, there exist control laws $u_i^{\epsilon} \in \mathcal{A}$ such that the time one map corresponding to the differential equation Let $X^{\epsilon} : \mathbb{R}^n \to \mathbb{R}^n$ be the map given by $X^{\epsilon}(x_0) = x^{\epsilon}(T)$ where $x^{\epsilon}(t)$ is the solution of the differential equation,*

$$\dot{x} = \sum_{i=1}^{m} u_i^{\epsilon}(t, x^{\epsilon}(t)) g_i\Big(x^{\epsilon}(t)\Big) \tag{29}$$

$$x(0) = x_0 \tag{30}$$

for each initial condition $x_0 \in \mathbb{R}^n$ and solution $x(t)$ of the differential equation (11) satisfies

$$\sup_{x \in K} |X(x) - X^{\epsilon}(x)| \leq \epsilon \tag{31}$$

References

Andrei Agrachev and Andrey Sarychev. Control on the manifolds of mappings with a view to the deep learning. *Journal of Dynamical and Control Systems*, pages 1–20, 2021.

Andrei A Agrachev and Marco Caponigro. Controllability on the group of diffeomorphisms. In *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, volume 26, pages 2503–2509. Elsevier, 2009.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*, volume 49. Springer, 2019.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

Howie Choset, Kevin M Lynch, Seth Hutchinson, George A Kantor, and Wolfram Burgard. *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Luca Falorsi and Patrick Forré. Neural ordinary differential equations on manifolds. *arXiv preprint arXiv:2006.06663*, 2020.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.

Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pages 1–31. Springer, 2013.

Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. *Advances in Neural Information Processing Systems*, 33:17548–17558, 2020.

Paulo Tabuada and Bahman Gharesifard. Universal approximation power of deep residual neural networks via nonlinear control theory. In *International Conference on Learning Representations*, 2020.