ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning from Demonstrations over Riemannian Manifolds using Neural ODEs

Diana Cuervo Espinosa[1], Mahathi Anand[2] and Angela P. Schoellig[2]

***Abstract*— Learning from demonstratins (LfD) is usually performed over Euclidean spaces, while the robot state, e.g. orientation, naturally evolves over curved spaces. Therefore, to ensure natural, complex motion generation, we investigate learning from demonstrations over Riemannian manifolds that are capable of encoding both position and orientation data. Here, geodesic paths provide for natural motion between two arbitrary points within the manifold. We propose to numerically estimate geodesics via neural ordinary differential equations, mitigating large computational overhead of existing approaches. Finally, these geodesics can be decoded back into the original task space before deploying on the robot. In this extended abstract, we discuss the architecture of our framework, provide some initial insights from our simulation experiments, including comparison to other geodesic computation mechanisms, and discuss the challenges and prospects for future work.**

I. MOTIVATION

Traditional methods for robot motion planning are usually hard coded for specific tasks and require extensive coding expertise. This makes it difficult to adapt to new tasks, and significant time and effort is required to identify relevant goal locations or sequential waypoints. Learning from demonstrations seeks to enhance generalization by collecting demonstrations from human experts and directly converting them to robot motion in an adaptive manner without requiring significant programming experience [1], [2]. However, they are usually limited to generating motion in the Euclidean task space. On the other hand, robot orientations, which evolve over curved manifolds, cannot be represented accurately with a purely Euclidean framework, and any extension to capturing full end-effector data requires careful enforcement of geometric constraints [3].

Recently, a Riemannian perspective to LfD has emerged [4]. Here, demonstrations are encoded into a curved latent space, i.e. a Riemannian manifold, where full end-effector poses can be described naturally. Then, robot motion may be learned directly via geodesics, i.e., the shortest paths between two arbitrary points. However, finding geodesics is a challenging problem, as it requires solving complex, second order differential equations. Several numerical relaxations, including iterative schemes [5] and graph-based methods [4] are common. Nevertheless, they are either computationally intensive, or rely on discrete approximations making it tedious to interpolate motion between two unseen data points. In this work, we propose an adaptive motion generation pipeline over Riemannian manifolds by utilizing variational autoencoders for encoding demonstrations into manifolds and neural ordinary differential equations (NODE) for geodesic computation. The NODE can quickly generate a motion path between any two locations in the manifold at inference time, making our approach very attractive for adaptive generalization and for hybrid, sequential tasks.

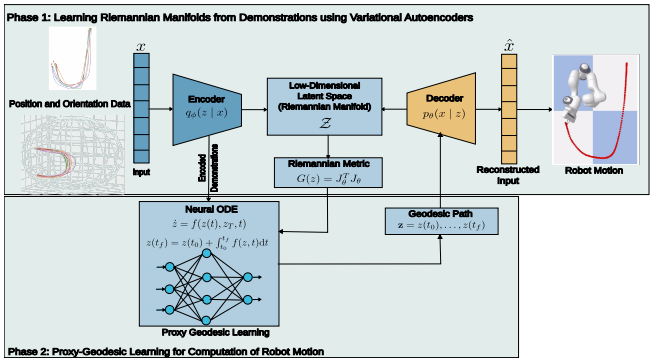

Fig. 1: The architecture for learning from demonstrations over Riemannian manifolds.

II. ARCHITECTURE

The proposed Riemannian motion generation framework is decoupled into two phases as described in Fig. 1 – the learning of robot's spatial constraints via the Riemannian manifold, and solving for the robot's temporal dynamics via geodesic paths in the learned manifold. The first phase is similar to the one proposed in [4], where a variational autoencoder (VAE) takes the demonstration data $x \in \mathcal{X}$ as input and encodes it into a lower-dimensional latent variable $z \in \mathcal{Z}$. In particular, the variational autoencoder consists of an encoder with parameters $\phi : \mathcal{X} \to \mathcal{Z}$, responsible for approximating the posterior distribution $p_\phi(z \mid x)$, and a decoder with parameters $\theta : \mathcal{Z} \to \mathcal{X}$ that approximates the generative distribution $p_\theta(x \mid z)$. It is trained under the standard evidence lower bound (ELBO) loss that consists of a regularization term as well as a reconstruction term. The resulting decoder function θ is then used to construct a local Riemannian metric given by $M(z) = J_\theta^T(z)J_\theta(z), \forall z \in \mathcal{Z}$, which characterizes the length and energy of a curve $\mathbf{z} : [0,1] \to \mathcal{Z}$ as $l_\mathbf{z} = \int_0^1 \sqrt{\dot{\mathbf{z}}(t)^T M(\mathbf{z}(t))\dot{\mathbf{z}}(t)}\mathrm{d}t$, and $E_\mathbf{z} = \frac{1}{2}\int_0^1 \dot{\mathbf{z}}(t)^T M(\mathbf{z}(t))\dot{\mathbf{z}}(t)\mathrm{d}t$, respectively. As a result, finding the geodesic, i.e., the shortest path between any two points in the manifold is reduced to finding a path that minimizes the energy (or length), respectively.

*This work is supported in part by Robotics Institute Germany, funded by BMFTR grant 16ME0997K

[1]Chair of Robotics and System Intelligence, Technical University of Munich, Germany diana.cuervo@tum.de

[2]Learning Systems and Robotics Lab, Technical University of Munich, Germany {mahathi.anand, angela.schoellig}@tum.de

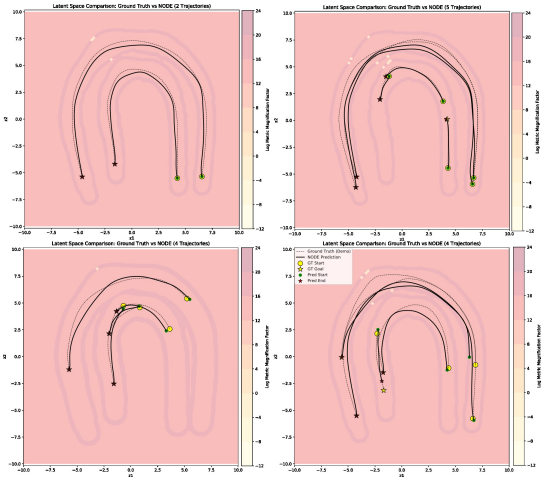

Fig. 2: Simulation plots illustrating the 3 scenarios - baseline (top left), generalization (top right), and generalization with perturbation (dist. 0.02 (bottom left) and 0.04 (bottom right)).

Having learned a suitable manifold capturing the demonstrations, one then needs to find a geodesic path between two arbitrary points. Knowing that the computation of geodesics naturally reduces to solving a differential equation, we utilize NODEs to compute proxy-geodesics, i.e., geodesic paths that are estimated by utilizing the learned manifold and the data from the demonstrations. Specifically, we use a goal-parameterized NODE given by $\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = f_\psi(\mathbf{z}(t), \mathbf{z}_g, t)$, where f_ψ is a standard feedforward neural network parameterized by ψ. To ensure smoothness of f_ψ, we use smooth neural activation functions. Then, the NODE is trained to minimize a loss function that takes into account (i) the energy of the path, (ii) imitation to demonstrations, and (iii) goal-reaching. Once trained, the NODE generates a solution geodesic path for any given start-goal pair (z_0, z_T) by solving a suitable initial value problem numerically. The obtained path is then decoded via the decoder function θ to obtain the full end-effector motion path in the original task space for the robot to track using a suitable low-level controller.

III. INSIGHTS

The results of our proposed approach were validated on a simple case study where the demonstration data [4] is restricted to $\mathbb{R}^2 \times \mathcal{S}^2$, with the position data resembling a J-shape and orientation data following a C-shape projected on a $3D$ sphere as in Fig. 1. Note that we used the same VAE as the one trained in [4]. The learned manifold can be visualized in Fig. 2. Here, the boundary of the manifold can also be distinctly observed (darker pink) due to the low data density in the region (measured via log-magnification factor $\log\sqrt{\det M}$). Then, we train a suitable NODE as described in Section II and use it during inference for the computation of geodesics. We perform several simulation case studies to test the following scenarios: (i) **baseline** case to test whether the geodesics can faithfully imitate the demonstrations encoded in the latent space. Here, the evaluation was performed by picking start and goal locations directly from the test dataset, and compared against the ground truth values, (ii) **generalization** case to study the adaptivity of the

scenario	baseline	gen.	gen. (std. 0.2)	gen. (std. 0.4)
tar. conv. error	0.01	0.08	0.17	0.32
inf. time (s)	0.27	8.50	6.54	6.49

TABLE I: The evaluation metrics (average) of our method obtained from 7000 proxy-geodesic computations

criteria	ours	graph-based [4]	stochman [6]
success rate (%)	100	100	0.25
tar. conv. error	0.10	6.125	0.0[1]
inf. time (s)	3.30	454.79	1999.35

TABLE II: Quantitative comparison when computing geodesics with our method vs. existing approaches (3500 trials).

paths when initialized with arbitrary start and goal locations from the test demonstrations, and (iii) **generalization with perturbation** (Gaussian) to investigate the closeness of paths from the ground truth when starting from locations slightly perturbed from the test data. The simulation results can be visualized in Fig. 2. One notices that the learned paths tend to stay within the manifold clusters, and respect the boundary of the manifold. In addition, we evaluate several quantitative metrics, including target convergence error w.r.t. ground truth data as well as time taken to generate the proxy geodesic path between arbitrary start and goal points at inference time. These metrics are also compared against existing geodesic computation methods ([4], [6]) as shown in Tables I and II. The results show that our method provides better target convergence and adaptability, while also being faster at inference.

IV. CHALLENGES AND FUTURE WORK

While we observed that the proxy-geodesic remains within the manifold empirically, the current framework provides no guarantees on the invariance within the manifold (particularly since the NODE is trained with local Euclidean loss functions and forward integration). There are also no convergence guarantees to the goal. Therefore, future work will focus on using Riemmannian geometry and formal methods to rigorously evaluate the correctness of the approach. More extensive experimentation on a real robot is also in order.

REFERENCES

[1] A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach*. MIT Press, 2022.

[2] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," vol. 42, no. 13, pp. 1133–1184, 2023.

[3] H. c. Ravichandar and A. Dani, "Learning position and orientation dynamics from demonstrations via contraction analysis," vol. 43, no. 4, pp. 897–912, 2019.

[4] H. Beik-Mohammadi, S. Hauberg, G. Arvanitidis, G. Neumann, and L. Rozo, "Learning riemannian manifolds for geodesic motion skills," vol. 17, 2021.

[5] G. Arvanitidis, S. Hauberg, P. Hennig, and M. Schober, "Fast and robust shortest paths on manifolds learned from data," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1506–1515.

[6] N. S. Detlefsen, A. Pouplin, C. W. Feldager, C. Geng, D. Kalatzis, H. Hauschultz, M. González-Duque, F. Warburg, M. Miani, and S. Hauberg, "Stochman," *GitHub. Note: https://github.com/MachineLearningLifeScience/stochman/*, 2021.

[1] average computed over successful trials.