

Topological Priors for Learning Stable Dynamical Systems from Demonstrations

Lucas Schwarz and Florian Röhrbein

Abstract—We present an approach to learn a topology-informed, first-order stable dynamical system to reproduce robot motions from expert demonstrations. Prototypes are extracted from trajectories and organized into a directed spatial graph, from which representative paths are obtained to fit splines. A Lyapunov function defined through these splines provides stabilizing constraints and a global notion of progress. To ensure predictable behavior, we decompose the dynamics into a conservative component active outside the data support and a locally accurate component that reproduces demonstrated velocity profiles within the data support, achieved through a covariance-based gating mechanism and local modulation matrices. The proposed method can faithfully represent unimodal and multimodal goal-directed motions, multi-attractor landscapes, and limit cycles, while supporting real-time learning. We further present promising initial results on benchmark datasets.

I. INTRODUCTION & RELATED WORK

Learning structured, stable dynamical systems from demonstrations for reactive robot motions has received significant attention [1]. While many approaches rely on time-dependent second-order systems such as dynamical movement primitives [2], others learn stable autonomous first-order systems offering improved robustness and adaptability. Stability is enforced via Lyapunov functions [3]–[7], contraction theory [8], [9], or diffeomorphic mappings from simple stable systems [10]–[12]. Recent work also considers dynamics on Riemannian manifolds [13], [14] and real-time adaptation on constrained hardware [15]. A central challenge is the *accuracy-stability dilemma* [10], where strong structural priors restrict the solution space. Existing approaches include quadratic Lyapunov functions [4], sums-of-squares polynomials [5], and spline-based constructions [7]. Although topological properties of the data have been explored [16], [17], they have not been leveraged to construct Lyapunov functions in this setting. In this work, we learn a topological summary of demonstrations using the Neural Gas algorithm [18] and a tailored topology construction, yielding prototypes and a directed spatial graph. From this, we extract data-supported paths or cycles to fit splines defining a flexible, piecewise Lyapunov-like function. Based on the prototypes, we further learn modulation matrices and a covariance-based gating mechanism, enabling accurate in-distribution dynamics and conservative behavior outside the data, with promising results on benchmark tasks.

We gracefully acknowledge funding from the BMFTR project Nr. 16SV9458. Lucas Schwarz and Florian Röhrbein are with the Department of Computer Science, Chemnitz University of Technology, D-09111 Chemnitz, Germany. {lucas.schwarz, florian.rohrbein}@informatik.tu-chemnitz.de

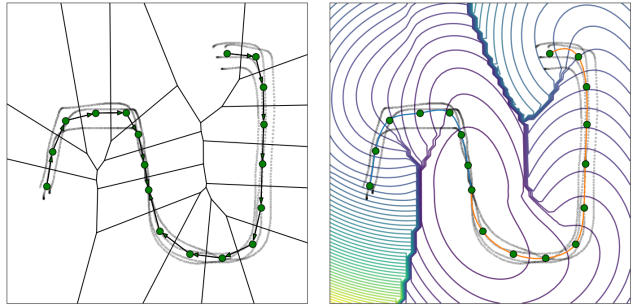


Fig. 1: Data, learned prototypes, topology and Voronoï cells (l). Splines and Lyapunov function values as contours (r).

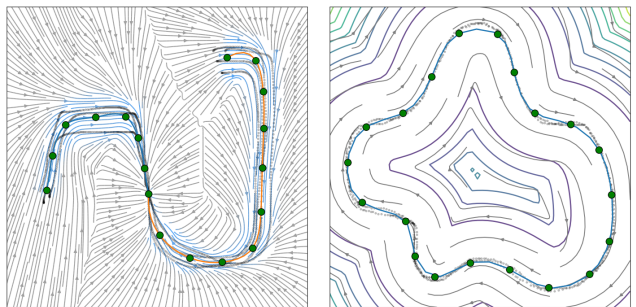


Fig. 2: Learned vector field with color indicating velocity (l). Limit cycle field with Lyapunov function contours (r).

II. METHODOLOGY

A. Learning a Topological Summary

To obtain an informative but compressed representation of the demonstrated expert trajectories in the form of position-velocity pairs $\{x_t, \dot{x}_t\}_{t=1}^T$, $x, \dot{x} \in \mathbb{R}^n$, we apply the so-called Neural Gas (NG) algorithm [18] on the position-samples first. The NG learns a number of representatives $W = \{w_1, \dots, w_N\} \in \mathbb{R}^n$, also called prototypes, in an unsupervised manner using the Euclidean distance $d_i := d(x, w_i)$ and stochastic gradient descent on the energy function over the data $X \subseteq \mathbb{R}^n$:

$$E = \sum_{i=1}^N \int_X P(x) h_\lambda(k_i(x, W)) d(x, w_i)^2 dx \quad (1)$$

using:

$$\Delta w_i = h_\lambda(k_i(x, W))(x - w_i) \quad i = 1, 2, \dots, N \quad (2)$$

such that they represent the data density $P(x)$. The prototypes divide the data space into a set of Voronoï cells, i.e. sets \mathcal{V}_i where the prototype w_i is the closest to any point in

that set than to any other:

$$\mathcal{V}_i = \{x \in X : d(x, w_i) \leq d(x, w_j), \forall j\} \quad (3)$$

Compared to methods such as the Self-Organizing Map [19] and K-Means, the NG imposes no global topological prior and updates all prototypes according to the ranking function $k_i(x, W) = \#\{l : d_l^2 < d_i^2\} = \sum_l \theta(d_l^2 - d_i^2)$, where $\theta(\cdot)$ is the Heaviside function. The neighborhood cooperativity is controlled by $h_\lambda(x, W) = \exp(-k_i(x, W)/\lambda)$ with $\lambda > 0$ [18]. Although extensions learn prototype topologies [20], the resulting graph edges may not reflect the data density [21]. Therefore, we propose a trajectory-informed method to construct an accurate topology from the demonstrations.

Algorithm 1: Topology Construction

Input: Prototypes $W = \{w_i\}_{i=1}^N$ with Voronoï cells $\{\mathcal{V}_i\}_{i=1}^N$, data $\{x_t\}_{t=1}^T$, threshold η

Output: Directed graph $G = (W, E)$, selected paths or cycles Π

$C \leftarrow 0 \in \mathbb{R}^{N \times N}$;

$a_t := i$ s.t. $x_t \in \mathcal{V}_i \quad \forall t$;

for $t = 1$ **to** $T - 1$ **do**

$(i, j) \leftarrow (a_t, a_{t+1})$;
 $C_{ij} \leftarrow C_{ij} + \mathbb{I}[i \neq j]$;

$P_{ij} \leftarrow C_{ij} / \sum_k C_{ik}$ if $\sum_k C_{ik} > 0$, else 0 ;

$E \leftarrow \{(i, j) \mid i \neq j, P_{ij} \geq \eta\}$;

$d_{ij} \leftarrow \|w_i - w_j\|_2 \quad \forall (i, j) \in E$;

$\Pi \leftarrow \emptyset$;

foreach *connected component* $C \subseteq G$ **do**

 let $\{w_s\} \subseteq W_C, w_g \in W_C$;
 if $\{w_s\} \neq \emptyset$ **and** w_g *exists* **then**
 $\Pi \leftarrow \Pi \cup \text{Unique}\{\text{SP}(\{w_s\}, w_g; d)\}$;
 else
 $\Pi \leftarrow \Pi \cup \text{Unique}\{\text{CYCLES}(C)\}$;

return $G = (W, E), \Pi$

The procedure is summarized in Algorithm 1. We first count transitions between adjacent Voronoï cells along the demonstrations, where a transition $\mathcal{V}_i \rightarrow \mathcal{V}_j$ occurs if $x_t \in \mathcal{V}_i$ and $x_{t+1} \in \mathcal{V}_j$. The counts are normalized to represent transition probabilities, and edges are created for probabilities above a low threshold $\eta \in [0, 1]$. For specific prototypes we obtain sinks w_g with only incoming edges and sources w_s with only outgoing edges. Assuming goal-directed motions or limit cycles, we extract unique distance-weighted shortest paths between w_s and w_g , or unique cycles within each connected component. These paths resemble principal curves [22], as they follow high data density regions while minimizing the quantization error, thereby approximating the underlying data manifold [23].

B. Generating a Spline-based Lyapunov Function

A radially unbounded Lyapunov function $V(x)$ is a function such that $\lim_{\|x\| \rightarrow \infty} V(x) = \infty$ and:

$$V(x^*) = 0, V(x) > 0 \quad \forall x \neq x^*, \dot{V}(x) < 0 \quad \forall x \neq x^* \quad (4)$$

For a system $\dot{x} = f(x)$ with $\dot{V}(x) = \nabla V(x)^\top f(x)$, these conditions imply that the equilibrium x^* is globally asymptotically stable under standard smoothness assumptions. To include multiple equilibria, where trajectories converge to different states out of a set $\mathcal{S}^* := \{x \in \mathbb{R}^n \mid f(x) = 0\}$, or limit cycles which describe invariant sets $\mathcal{S}^+ := \{x(t) \mid x(t + \Psi) = x(t), \forall t, \Psi > 0\}$ for an orbital period Ψ , we utilize the definition of a *Lyapunov-like* function, that satisfies the usual Lyapunov conditions for the respective sets \mathcal{S}^* and \mathcal{S}^+ as well [24]. To generate a valid Lyapunov-like function, we fit an arc-length parametrized C^1 -differentiable cubic Hermite spline $\gamma_k : [0, 1] \rightarrow \mathbb{R}^n$ on each path or cycle present in Π , such that the spline tangents are aligned with the edge orientations. Boundary constraints, such as multiple paths meeting at shared endpoints or cyclic constraints for limit cycles, arise naturally from the topology. For each spline γ_k , we define the projection and distance:

$$s_k(x) = \arg \min_{s \in [0, 1]} \|x - \gamma_k(s)\|^2, \quad (5)$$

$$d_k(x) = \|x - \gamma_k(s_k(x))\|. \quad (6)$$

The closest active spline is selected using:

$$k^*(x) = \arg \min_k d_k(x). \quad (7)$$

For a single spline, we define:

$$u_k(x) = x - \gamma_{k(x)}(s_{k(x)}(x)), \quad (8)$$

$$\tau_k(x) = \gamma'_{k(x)}(s_{k(x)}(x)), \quad (9)$$

$$\kappa_k(x) = \gamma''_{k(x)}(s_{k(x)}(x)). \quad (10)$$

A Lyapunov function combining distance to the spline and progression along it is defined as:

$$V_k(x) = \frac{\alpha}{2} \|x - \gamma_k(s_k(x))\|^2 + \frac{(1 - \alpha)}{2} (1 - s_k(x))^2. \quad (11)$$

The parameter $\alpha \in [0, 1]$ is used to control the behavior of the energy gradient field: high values cause strong contraction toward the spline, while low values allow greater flexibility to accommodate perturbations. The gradient includes a second-order correction due to the dependence of the progression term on the projection and the implicit function theorem:

$$\nabla V_k(x) = \alpha u_k(x) - (1 - \alpha)(1 - s_k(x)) \cdot \frac{\tau_k(x)}{\|\tau_k(x)\|^2 - u_k(x)^\top \kappa_k(x)} \quad (12)$$

In the case of a limit cycle, the phase-progression related terms are dropped. We also define a term involving the normalized gradient for paths with small constant $\delta > 0$ to avoid division by zero:

$$\nabla \tilde{V}_k(x) = \frac{\nabla V_k(x)}{\|\nabla V_k(x)\| + \delta}, \quad (13)$$

where a tangential bias $-(1 - \alpha)\tau_k(x)$ from the spline is added in the case of limit-cycles. The global Lyapunov-like function is then defined by selecting the local active piece

through the closest spline:

$$V(x) = V_{k^*(x)}(x). \quad (14)$$

While this hard switching introduces discontinuities at the switching surfaces, one can show that the resulting function is still valid using Clarke’s generalized gradient [25]. We omit the proof here for brevity.

C. Covariance-based Local Activations and Gating

For the active path p , let $\{w_i\}_{i=1}^P$ denote its constituent prototypes and let Σ_i be the empirical local covariance calculated from the position data in \mathcal{V}_i . For a state-space position x , the squared Mahalanobis distance is:

$$d_{M_i}(x) = (x - w_i)^\top \Sigma_i^{-1} (x - w_i). \quad (15)$$

The local activations are given by a numerically stabilized partition of unity:

$$\phi_i(x) = \frac{\exp(-d_{M_i}(x) - \alpha(x))}{\sum_{j=1}^P \exp(-d_{M_j}(x) - \alpha(x))}, \quad (16)$$

$$\alpha(x) = \max_j (-d_{M_j}(x)) = -\min_j d_{M_j}(x), \quad (17)$$

The scalar gate $\Gamma(x) \in [0, 1]$ determines whether the dynamics are governed mainly by the conservative unit-norm Lyapunov descent field, $\Gamma(x) \approx 0$, or by the learned local model, $\Gamma(x) \approx 1$. It is computed from the covariances around the prototypes of the active spline:

$$d(x) = \min_j d_{M_j}(x) \quad (18)$$

$$\Gamma(x) = \sigma(-\beta(d(x) - \delta)) \quad (19)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (20)$$

where $d_{M_j}(x)$ is evaluated using the projection of x onto segment (w_j, w_{j+1}) and the corresponding interpolated covariance. Hence, $\phi_i(x)$ distributes responsibility *within* the data support among prototypes of the active spline, whereas $\Gamma(x)$ controls *whether* the learned local dynamics should be active at all. Further the second gating function $H(x)$ is defined as:

$$H(x) = 1 - \exp(-\beta_H \|x - w_g\|^2) \quad (21)$$

such that $H(x) = 1$, $H(w_g) = 0$ and $H(x) \rightarrow 0$ as $x \rightarrow w_g$ with β_H controlling the sharpness of the transition. This function scales down the dynamics as the state approaches any present goal prototype w_g .

D. Determining Modulation Matrices

The learned dynamical system is constructed using positive definite modulation matrices $G_i \in \mathbb{R}^{n \times n}$, $G_i \succ 0$ associated with each prototype, acting on the normalized gradient. For the active spline, they are combined as:

$$M_k(x) = \sum_{i=1}^P \phi_i(x) G_i, \quad (22)$$

$$\phi_i \geq 0, \quad \sum_{i=1}^P \phi_i = 1.$$

The resulting localized dynamics are:

$$f_{\text{local}}(x) = -M_k(x) \nabla \tilde{V}_k(x), \quad (23)$$

which enables anisotropic scaling and rotation of the gradient field while preserving its structure. Since $M_k(x)$ is a convex combination of positive definite matrices, it remains positive definite, ensuring that the descent direction of $\nabla \tilde{V}_k(x)$ is preserved. Consequently, the Lyapunov function remains non-increasing under the modulation such that stability is maintained. We omit a formal proof also for brevity. The final dynamics which incorporate the local dynamics, gating and normalized gradient are given by:

$$\dot{x} = f_k(x) = H(x) \left((1 - \Gamma(x)) (-\nabla \tilde{V}_k(x)) + \Gamma(x) (-M_k(x) \nabla \tilde{V}_k(x)) \right). \quad (24)$$

To determine the modulation matrices, we solve a convex least-squares optimization problem with linear matrix inequality constraints for the data assigned to the respective closest splines $\mathcal{T}_k := \{t \mid k^*(x_t) = k\}$ in parallel:

$$\begin{aligned} \min_{\{G_i^k\}_{i=1}^{P_k}} \quad & \sum_{t \in \mathcal{T}_k} \|\dot{x}_t - f_k(x_t)\|^2 \\ \text{s.t.} \quad & G_i^k \succeq \varepsilon I, \quad i = 1, \dots, P_k \end{aligned} \quad (25)$$

III. EXPERIMENTS & OUTLOOK

We report preliminary results on the LASA handwriting dataset, averaged over five runs with standard deviations. The metrics are the mean squared error (MSE), cosine error, dynamic time warping distance (DTWD) and swept error average (SEA) as described in [3], [4], calculated on normalized data. We trained the NG for 50 epochs with 18 prototypes and used the recommended parameters for the other approaches. Our model compares favourably while exhibiting an average learning time of 0.85 seconds on an AMD Ryzen Threadripper 3960X with 128GB of RAM for a single shape. Inference is possible at approximately 250Hz. An example visualizing all steps of the procedure is provided in Figure 1 and Figure 2. The proposed approach demonstrates, that topology-informed Lyapunov-like functions can capture complex motion patterns while preserving stability guarantees. Future directions will focus on validation on real robotic systems and incorporating task-adaptation mechanisms [26] to account for changes in the environment during inference. In addition, we aim to extend the framework from single motion trajectories to structured task sequences, enabling the representation and execution of multi-stage behaviors with transitions between distinct dynamical regimes.

TABLE I: Preliminary Results on LASA. Best **bold**.

Model	MSE	Cosine	DTWD	SEA
SEDS [3]	0.15 ± 0.04	0.11 ± 0.08	85.28 ± 36.80	0.39 ± 0.08
LPV-DS [4]	0.13 ± 0.03	0.07 ± 0.05	63.26 ± 13.78	0.42 ± 0.39
PLYDS [5]	0.12 ± 0.04	0.07 ± 0.05	129.65 ± 117.35	0.43 ± 0.13
DAMM [6]	0.14 ± 0.06	0.10 ± 0.09	46.02 ± 8.89	0.32 ± 0.05
TopoDS (ours)	0.08 ± 0.04	0.04 ± 0.02	12.01 ± 4.58	0.22 ± 0.05

REFERENCES

- [1] J. Fu, H. Huang, Z. Jin, A. Liu, W.-A. Zhang, L. Yu, W. Si, and C. Yang, "A survey on learning an autonomous dynamic system for human-robot skills transfer from demonstration," *Robotics and Computer-Integrated Manufacturing*, vol. 97, p. 103092, 2026.
- [2] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.
- [3] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [4] N. B. Figueroa Fernandez and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," *Proceedings of Machine Learning Research*, 2018.
- [5] A. Abyaneh and H.-C. Lin, "Learning lyapunov-stable polynomial dynamical systems through imitation," in *7th Annual Conference on Robot Learning*.
- [6] S. Sun, H. Gao, T. Li, and N. Figueroa, "Directionality-aware mixture model parallel sampling for efficient linear parameter varying dynamical system learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6248–6255, 2024.
- [7] Y. Li and S. Calinon, "From movement primitives to distance fields to dynamical systems," *IEEE Robotics and Automation Letters*, 2025.
- [8] H. C. Ravichandar and A. Dani, "Learning position and orientation dynamics from demonstrations via contraction analysis," *Autonomous Robots*, vol. 43, no. 4, pp. 897–912, 2019.
- [9] H. B. Mohammadi, S. Hauberg, G. Arvanitidis, N. Figueroa, G. Neumann, and L. Rozo, "Neural contractive dynamical systems," in *The Twelfth International Conference on Learning Representations*, 2023.
- [10] K. Neumann and J. J. Steil, "Learning robot motions with stable dynamical systems under diffeomorphic transformations," *Robotics and Autonomous Systems*, vol. 70, pp. 1–15, 2015.
- [11] M. A. Rana, A. Li, D. Fox, B. Boots, F. Ramos, and N. Ratliff, "Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems," in *Learning for Dynamics and Control*, pp. 630–639, PMLR, 2020.
- [12] S. Tesfazgi, L. Sprandl, and S. Hirche, "Learning geometrically-informed lyapunov functions with deep diffeomorphic rbf networks," in *The 28th International Conference on Artificial Intelligence and Statistics*.
- [13] S. Sun and N. Figueroa, "Se (3) linear parameter varying dynamical systems for globally asymptotically stable end-effector control," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5152–5159, IEEE, 2024.
- [14] S. Bakker, M. Schonger, T. Löw, J. Alonso-Mora, and S. Calinon, "Curve-induced dynamical systems on riemannian manifolds and lie groups," *arXiv e-prints*, pp. arXiv-2603, 2026.
- [15] T. Li, S. Sun, S. S. Aditya, and N. Figueroa, "Elastic motion policy: An adaptive dynamical system for robust and efficient one-shot imitation learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9846–9853, IEEE, 2025.
- [16] B. Fichera and A. Billard, "Linearization and identification of multiple-attractor dynamical systems through laplacian eigenmaps," *Journal of Machine Learning Research*, vol. 23, no. 294, pp. 1–35, 2022.
- [17] S. Gupta, A. Nayak, and A. Billard, "Compact oneshot modelling of high dimensional demonstrations using laplacian eigenmaps," *IEEE Transactions on Robotics*, 2026.
- [18] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'neural-gas' network for vector quantization and its application to time-series prediction," *IEEE transactions on neural networks*, vol. 4, no. 4, pp. 558–569, 1993.
- [19] T. Kohonen, *Self-organizing maps*, vol. 30. Springer Science & Business Media, 2012.
- [20] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.
- [21] M. Aupetit, "Robust topology representing networks.," in *ESANN 2003*, pp. 45–50, ESANN Proceedings, 2003.
- [22] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Learning and design of principal curves," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 3, pp. 281–297, 2000.
- [23] T. Villmann and J. C. Claussen, "Magnification control in self-organizing maps and neural gas," *Neural Computation*, vol. 18, no. 2, pp. 446–469, 2006.
- [24] J. Björnsson, P. Giesl, S. F. Hafstein, and C. M. Kellett, "Computation of lyapunov functions for systems with multiple local attractors," *Discrete and Continuous Dynamical Systems*, vol. 35, 2015.
- [25] F. H. Clarke, *Optimization and nonsmooth analysis*. SIAM, 1990.
- [26] T. Nierhoff, S. Hirche, and Y. Nakamura, "Spatial adaption of robot trajectories based on laplacian trajectory editing," *Autonomous Robots*, vol. 40, no. 1, pp. 159–173, 2016.