Data-Efficient Characterization of the Global Dynamics of Robot Controllers with Confidence Guarantees

Ewerton R. Vieira[3,5], Aravind Sivaramakrishnan[1], Yao Song[6], Edgar Granados[1],
Marcio Gameiro[2], Konstantin Mischaikow[2], Ying Hung[6], and Kostas E. Bekris[1]

Abstract— This paper proposes an integration of surrogate modeling and topology to significantly reduce the amount of data required to describe the underlying global dynamics of robot controllers, including closed-box ones. A Gaussian Process (GP), trained with randomized short trajectories over the state-space, acts as a surrogate model for the underlying dynamical system. Then, a combinatorial representation is built and used to describe the dynamics in the form of a directed acyclic graph, known as *Morse graph*. The Morse graph is able to describe the system's attractors and their corresponding regions of attraction (`RoA`). Furthermore, a pointwise confidence level of the global dynamics estimation over the entire state space is provided. In contrast to alternatives, the framework does not require estimation of Lyapunov functions, alleviating the need for high prediction accuracy of the GP. The framework is suitable for data-driven controllers that do not expose an analytical model as long as Lipschitz-continuity is satisfied. The method is compared against established analytical and recent machine learning alternatives for estimating `RoA`s, outperforming them in data efficiency without sacrificing accuracy. Link to code: **https://go.rutgers.edu/49hy35en**

I. INTRODUCTION

Multiple tools have been developed to estimate the region of attraction (`RoA`) of a dynamical system [1]–[3]. These tools are useful for understanding the conditions under which a controller can be safely applied to solve a task. Finding the *true* `RoA` of a controlled system is challenging. Thus, many efforts try to estimate the largest possible set contained in the true `RoA`. For closed-box systems, such as learned controllers that do not provide an analytical expression, it is impractical to apply Lyapunov methods directly. Many non-Lyapunov methods often have significant data requirements so as to estimate `RoA`s effectively.

This paper addresses the problem of finding `RoA`s of controllers with unknown dynamics by proposing an efficient way to use data. It explores surrogate modeling together with topological tools not only to identify the `RoA` for a specific goal region but also to describe the global dynamics. This also includes data-driven controllers, where a key challenge in their application is *verification*, i.e. explaining when the controller works and when it fails. To achieve this objective, this work uses Gaussian Processes (GPs) as surrogate models
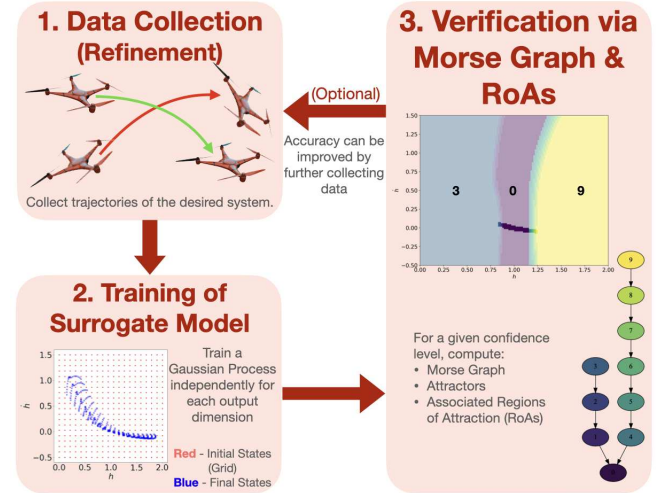

Fig. 1. Overview of the proposed framework: 1) initial data collection; 2) a Gaussian Process (GP) is trained as a surrogate model; 3) computation of Morse Graph and the Region of Attraction (`RoA`) for verification. For an optional refinement, steps 1-3 can be repeated as necessary.

to compute a *Morse graph*, which constructs a finite, combinatorial representation of the state space given access to a discrete-time representation of the dynamics. It achieves data efficiency and improved accuracy relative to alternatives that are either analytical tools (and can only be used for analytical systems) or learning-based frameworks. Fig. 1 highlights the iterative nature of the approach.

In particular, the key contribution of this work is the use of GPs as a statistical surrogate model of the underlying controlled system, alongside the Morse Graphs framework to compactly describe the global dynamics. The integration results in data efficiency: significantly fewer samples of the underlying dynamics are necessary for an informative representation of the global dynamics. Data efficiency in surrogate modeling is achieved by leveraging the effectiveness of Morse Graphs, alleviating the high prediction accuracy requirements typically required for this purpose.

Furthermore, this integration allows working with trajectories that may not uniformly cover the state space of the underlying system. Prior efforts with topological tools and combinatorial decompositions of the underlying state space required sampling the dynamics uniformly over a grid-based discretization of the state space. One of the benefits of GP based learning approach is that it can characterize uncertainty. So the sampling points can be carefully selected to minimize the uncertainty about the global dynamics and the region of attractions. Additionally, GPs provide confidence

[1] Dept. of Computer Science, Rutgers, NJ, USA. [2] Dept. of Mathematics, Rutgers, NJ, USA. [3] DIMACS, Rutgers, NJ, USA. [4] ICMC, Universidade de São Paulo, São Carlos, São Paulo, Brazil. [5] IME, Universidade Federal de Goiás, Goiânia, GO, Brazil. [6] Dept. of Statistics, Rutgers University, NJ, USA. E-mail: {er691,kb572}@rutgers.edu.

levels on the accuracy of the results at each subset of the state space.

II. Related Work

Numerical methods that estimate the RoA given a closed-form expression of the system dynamics include maximal Lyapunov functions (LFs) and linear matrix inequalities (LMIs). Ellipsoidal RoA approximation via LMIs [4], [5] has been used for mobile robots [6], [7], and LMI relaxations can also approximate the RoA of polynomial systems [8]. LFs constructed by restricting them to be sum-of-squares (SoS) polynomials [9] have been used in building randomized trees with LQR feedback [10], funnel libraries [11] and stability certificates for rigid bodies [12].

Reachability analysis [3], i.e., computing a backward reachable tube to obtain the RoA without shape imposition, for computing RoAs of dynamical walkers [13], has been combined with machine learning to maintain safety over a given horizon [14]. GPs can learn barrier functions for ensuring the safety of unknown dynamical systems [15]. Similarly, barrier certificates (BCs) can identify areas for exploration to expand the safe set [16].

Machine learning can learn LFs by alternating between a learner and a verifier [17], [18], or via stable data-driven Koopman operators [19]. Rectified Linear Unit (ReLU) activated neural networks can learn robust LFs for approximated dynamics [20]. The Lyapunov Neural Network [21] can incrementally adapt the RoA's shape given an initial safe set. As an alternative, GPs can obtain a Lyapunov-like function [22], or an LF can be synthesized to provide guarantee's on a controller's stability while training [23].

GPs are a popular choice to reduce data requirements while modeling dynamical systems [24]. Some of their applications in robotics include model-based policy search [25], modeling non-smooth dynamics of robots with contacts [26], and stabilizing controllers for control-affine systems [27]. For RoA estimation problems, given an initial safe set computed using a Lyapunov function, a GP can approximate the model uncertainties on a discrete set of sampling points from the safe region while expanding it [28].

Topology has multiple applications in robotics, such as deformable manipulation and others [29]–[34]. Morse theory can help incrementally build local minima trees for multi-robot planning [35] and finds paths to cover 2D or 3D spaces [36]. In recent work [37], Morse graphs are shown to be effective in compactly describing the global dynamics of a control system without an analytical expression of its dynamics. To the best of the authors' knowledge, the current work is the first to apply surrogate modeling with uncertainty quantification in conjunction with topological tools to identify the global dynamics of robot controllers.

III. Problem Setup

This work aims to provide a data-efficient framework for the analysis of global dynamics of robot controllers based on combinatorial dynamics and order theory [37]–[40]. Consider a non-linear, continuous-time control system:

$$\dot{x} = f(x, u), \qquad (1)$$

where $x(t) \in X \subseteq \mathbb{R}^M$ is the state at time t, X is a compact set, $u : X \mapsto \mathbb{U} \subseteq \mathbb{R}^M$ is a Lipschitz-continuous control as defined by a deterministic control policy $u(x)$, and $f : X \times \mathbb{U} \mapsto \mathbb{R}^M$ is a Lipschitz-continuous function. Neither $f(\cdot)$ nor $u = u(x)$ are necessarily known analytically. For a given time $\tau > 0$, let $\phi_\tau : X \to X$ denote the function obtained by solving Eq. (1) forward in time for duration τ from everywhere in X. A trajectory (or an *orbit*) is defined as a sequence of states obtained by integrating Eq. 1 forward in time.

The analysis of the global dynamics can reveal the system's attractors, which include fixed points, such as a state that the control law manages to bring the system to; or limit cycles, such as a periodic behavior of the system. It will also reveal a Region of Attraction (RoA) which is a subset of the basin of attraction of an attractor $\mathcal{A}$. The basin of attraction is the largest set of points whose forward orbits converge to $\mathcal{A}$, or more formally, the maximal set $\mathcal{B}$ that has the property:

$$\mathcal{A} = \omega(\mathcal{B}) := \bigcap_{n \in \mathbb{Z}^+} \text{cl}\left(\bigcup_{k=n}^{\infty} \phi_\tau^k(\mathcal{B})\right)$$

where ϕ_τ^k is the composition $\phi_\tau \circ \cdots \circ \phi_\tau$ (k times) and cl is topological closure.

Since f and u are Lipschitz-continuous, ϕ_τ is too; furthermore any RoA of Eq. (1) is an RoA under ϕ_τ. Hence, it is possible to study Eq. (1) by analyzing the behavior of the dynamics according to ϕ_τ, which is not assumed, however, to be computable and available.

IV. Topological Framework and Uncertainty Quantification via GPs

There are two key components for capturing meaningful conditions of the dynamics according to ϕ_τ. First, identifying effective combinatorial representations of the attractors and maximal RoAs. And second, to achieve data efficiency by employing GP-based surrogate modeling with uncertainty quantification.

Morse Graphs for Understanding Global Dynamics: Fig. 2(left) is used as a running 1-dim. example. The function ϕ_τ is first approximated by decomposing the state space X into a collection of regions $\mathcal{X}$, for instance, by defining a *grid*. Fig. 2(left) shows a grid on the interval $[-3, 3]$ decomposed into sub-intervals **a** through **e**. Given a region $\xi \in \mathcal{X}$ (a cell), the system is forward propagated for multiple initial states within ξ for a time τ to identify regions reachable from ξ. Consider, for example, the sub-interval **b** as such a cell ξ in Fig. 2(left). The arrows from the boundary of **b** depict the forward propagation of the dynamics where **b** maps to itself given the underlying dynamics

Then, a directed graph representation $\mathcal{F}$ stores each region in $\xi \in \mathcal{X}$ as a vertex and edges pointing from ξ to each region reachable from ξ. In Fig. 2(left), $\mathcal{F}$ is the graph containing nodes given by the grid cells **a** to **e**. Each edge represents a pair given by a cell and its image according to the dynamics. For instance, $(\mathbf{b}, \mathbf{b})$ and $(\mathbf{b}, \mathbf{c})$ are two edges added since **b** both maps to itself and also maps to **c**.
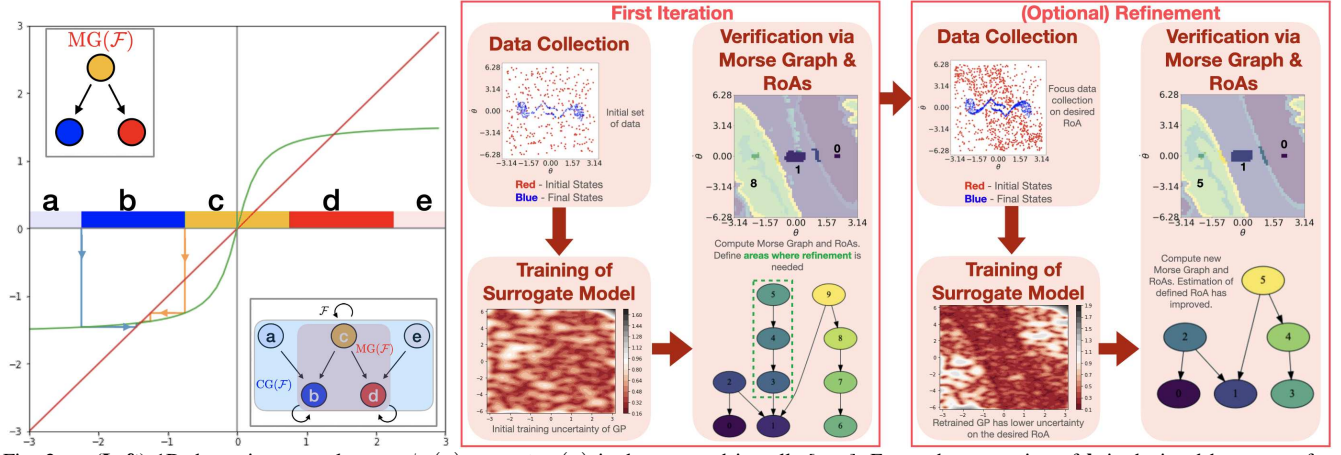
Fig. 2. **(Left)** 1D dynamics example $y = \phi_\tau(x) = \arctan(x)$ is decomposed in cells [**a**, **e**]. Forward propagation of **b** is depicted by arrows from its boundary. $\mathcal{F}$ is a directed graph capturing reachable vertices (regions) from other vertices. Strongly connected components of $\mathcal{F}$ result in $\mathrm{CG}(\mathcal{F})$. Finally, the Morse Graph $\mathrm{MG}(\mathcal{F})$ (nodes $\{\mathbf{b}, \mathbf{c}, \mathbf{d}\}$) contains the attractors of interest. **(Center)** Proposed Method (initial step) for Pendulum (LQR). Data collected; heat map of initial GP; MG and RoAs of the initial GP. **(Right)** Proposed method (refinement step) for Pendulum (LQR). Few new samples focused on the desired RoA are collected; retrained GP has lower uncertainty; MG and RoAs of the retrained GP.

Condensing all the nodes belonging to a strongly connected components (SCCs) of $\mathcal{F}$ into a single node, results in the condensation graph $\mathrm{CG}(\mathcal{F})$. Then, edges on $\mathrm{CG}(\mathcal{F})$ reflect reachability according to a topological sorting of $\mathcal{F}$. In Fig. 2(left), $\mathrm{CG}(\mathcal{F})$ is the subgraph with nodes **a** to **e** and all non-self edges, i.e., $\mathrm{CG}(\mathcal{F})$ has no cycles.

Since $\mathrm{CG}(\mathcal{F})$ is a directed acyclic graph, it is also a partially ordered set (i.e., a poset). A recurrent set is an SCC that contains at least one edge. Finally, the Morse graph of $\mathcal{F}$, denoted by $\mathrm{MG}(\mathcal{F})$, is the subposet of recurrent set of $\mathrm{CG}(\mathcal{F})$ (excluding single-node SCCs). In Fig. 2(left), $\mathrm{MG}(\mathcal{F})$ is the graph with nodes **b**, **c** and **d** and the corresponding edges between them. The Morse graph $\mathrm{MG}(\mathcal{F})$ captures the recurrent and non-recurrent dynamics by representing the recurrent sets of $\mathcal{F}$ as vertices and whose edges reflect reachability between these sets. The nodes of Morse graphs can contain attractors of interest.

In summary, Morse graphs and RoAs are obtained by a four step procedure. **1)** State space decomposition and generation of input to represent ϕ_τ. **2)** Construction of the combinatorial representation $\mathcal{F}$ of the dynamics given an outer approximation of ϕ_τ. **3)** Compute the *Condensation Graph* $\mathrm{CG}(\mathcal{F})$ and *Morse Graph* $\mathrm{MG}(\mathcal{F})$ by identifying recurrent sets/SCCs of $\mathcal{F}$ and topological sort. **4)** Derive RoAs for the recurrent sets given the reachability of $\mathrm{CG}(\mathcal{F})$.

The state space decomposition is an orthotope $X = \prod_{i=1}^{n}[a_i, b_i]$ (i.e., generalization of a rectangle for high-dim.), allowing for periodic boundary conditions. More specifically, a uniform discretization of X is applied based on 2^{k_i} subdivisions in the i-th component resulting in a decomposition of the state space into $\prod_{i=1}^{n} 2^{k_i}$ cubes of dimension n. $\mathcal{X}$ denotes the collection of these cubes.

The input representation of ϕ_τ is generated by the set of values of ϕ_τ at the corner points of cubes in $\mathcal{X}$. More precisely, let $V(\mathcal{X})$ denote the set of all corner points of cubes in $\mathcal{X}$. The method computes the set of ordered pairs $\Phi_\tau(\mathcal{X}) := \{(v, \phi_\tau(v)) \mid v \in V(\mathcal{X})\}$, by forward propagating the dynamics for time τ from all $V(\mathcal{X})$. Note that, no analytical version of ϕ_τ is required, allowing a surrogate

model to generate data $\Phi_\tau(\mathcal{X})$, as proposed in this work.

The combinatorial representation of the dynamics is approximated by a combinatorial multivalued map $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{X}$, where vertices are n-cubes $\xi \in \mathcal{X}$. The map $\mathcal{F}$ contains directed edges $\xi \to \xi', \forall\, \xi, \xi' \in \mathcal{X}$ such that $\xi' \cap \Phi_\tau(\xi) \neq \emptyset$. The set of cubes identified by $\mathcal{F}(\xi)$ are meant to capture the possible states of $\phi_\tau(\xi)$. Then, $\forall \xi \in \mathcal{X}$ a multivalued map $\mathcal{F}$ that satisfies:

$$\mathcal{F}_{min}(\xi) := \{\xi' \in \mathcal{X} \mid \xi' \cap \phi_\tau(\xi) \neq \emptyset\} \subset \mathcal{F}(\xi) \quad (2)$$

is called an outer approximation of ϕ_τ. Computation of $\mathcal{F}_{min}$ is typically prohibitively expensive. But it is sufficient to find an outer approximation $\mathcal{F}$, which still leads to mathematically rigorous results. The flexibility in defining an outer approximation provides versatility in its construction, which allows integration with a surrogate model.

Surrogate Modeling and Uncertainty Quantification by GPs: Assume that there is access to data of the form $\mathcal{D} = \{(x^n, y^n) \in X \times X \mid y^n = \phi_\tau(x^n) \text{ and } n = 1, \ldots, N\}$, which may have Gaussian noise. In the pair $(x^n, y^n) \in \mathcal{D}$, x^n is an initial state of the system, and y^n is the end state after forward propagating the dynamics (1) from x^n for time τ. In Figs 1 and 2(center and right), (x^n, y^n) are denoted as red and blue points respectively. Let $\phi_{\tau,\ell}$ denote the ℓ-th component of ϕ_τ, for $\ell = 1, \ldots, M$ and assume that $\phi_{\tau,\ell}$ is the realization of ϕ_τ from GP:

$$\phi_{\tau,\ell}(x) \sim GP(\beta_\ell, \sigma_\ell^2 k(x, x'; \theta_\ell)), \quad (3)$$

where β_ℓ and σ_ℓ^2 are the unknown mean and variance, and the correlation is defined by the kernel $k(x, x'; \theta_\ell) = Corr(\phi_{\tau,\ell}(x), \phi_{\tau,\ell}(x'); \theta_\ell)$ with $k(x, x; \theta_\ell) = 1$, $k(x, x'; \theta_\ell) = k(x', x; \theta_\ell)$ for $x, x' \in X$ and θ_ℓ is a set of parameters associated with k.

The prediction for an untried $x \in X$ can be obtained by a d-dimensional multivariate normal distribution, $MN(\mu(x), \Sigma(x))$, where $\mu = (\mu_1, \cdots, \mu_d)$, $\mu_\ell(x) = E(\phi_{\tau,\ell}(x)|\mathcal{D}) = \hat{\beta}_\ell + k(x; \hat{\theta}_\ell)^T K^{-1}(\hat{\theta}_\ell)(y_\ell^T - \hat{\beta}_\ell)$, and the covariance matrix $\Sigma(x)$ is a diagonal matrix with elements $\hat{\sigma}_\ell^2 \left(1 - k(x; \hat{\theta}_\ell)^T K^{-1}(\hat{\theta}_\ell) k(x; \hat{\theta}_\ell)\right)$ assuming the M-dimensional outputs are independent, $\hat{\beta}_\ell$, $\hat{\sigma}_\ell$, and $\hat{\theta}_\ell$ are the

maximum likelihood estimators, $k(x; \hat{\theta}) = [k(x, x_n; \hat{\theta}), n = 1, \cdots, N]$, and $K(\hat{\theta})$ is an $N \times N$ matrix with elements $k(x_i, x_j; \hat{\theta})$ for $1 \leq i, j \leq N$.

V. Proposed Integrated Solution

The proposed framework brings together topological tools for combinatorial dynamics and GPs. GPs are used as surrogate models to identify the global dynamics and RoAs of controllers, including data-driven ones without access to an analytical model. Fig. 2 (center and right) summarizes the method's application for a pendulum controlled by a linear quadratic regulator (LQR). Overall, the method can be divided into the following steps:

1. Collect data from the system.
2. Apply GP regression to get an initial surrogate model with predictive mean μ and covariance function Σ.
3. Compute the Morse Graph of the trained GP to obtain, with a given confidence level, the information about the global dynamics and the RoAs.
4. (Optional) To increase the confidence level, select state space points to collect more data as in step 1 to improve the accuracy of the representation and return to step 2.

Step 1: Data Collection Two procedures for data collection are explored. The first one collects short trajectories from random initial points in X with a fixed duration of time. The second option collects time series data in the form of long trajectories, breaking them into smaller ones. Denote by $\mathcal{D} = \{(x^n, y^n) \in \mathbb{R}^M \times \mathbb{R}^M \mid y^n = \phi_\tau(x^n) \text{ for } n = 1, \ldots, N\}$ the robot trajectory data collected.

Step 2: GP regression Given the training data $\mathcal{D}$ obtained in the previous step, a GP model is trained independently for each output dimension using a zero mean prior and a Matérn kernel with $\nu > 1$. Learned controllers typically give Lipschitz-continuous functions, yet, not necessarily smooth. Hence kernels requiring less smoothness assumptions are ideal. As such, Matérn kernels provide better performance than a more common radial basis function kernel.

Let μ and Σ denote the predictive mean and covariance functions of the GP, respectively. For a confidence level $1 - \delta$ and an $x \in \mathbb{R}^M$, since the GP model is trained independently for each output dimension, the confidence ellipsoid is an M-dimensional hypercube $E_{\Sigma(x)}^\delta := \prod_{n=1}^M I_{n,\Sigma(x)}^\alpha$, where $\alpha = 1 - (1 - \delta)^{1/M}$, $I_{n,\Sigma(x)}^\alpha = \{u \in \mathbb{R} \mid \|u - \mu_n(x)\|/\sigma(x) < z_{\alpha/2}\}$ is the confidence interval for the n-th output at x, and $z_{\alpha/2}$ is the corresponding critical value of the standard normal distribution.

Step 3: Confidence level of Morse Graph and RoAs Let $\mathcal{X}$ be a discretization of X into cubes, $V(\mathcal{X})$ and $C(\mathcal{X})$ be the collections of corner (vertices) and the center points of the cubes in $\mathcal{X}$, respectively. From a trained GP with predictive

Fig. 3. Pointwise confidence multi-valued map $\mathcal{F}_\mu$.

mean μ and covariance functions Σ generate the set $M(\mathcal{X}) := \{(v, \mu(v)) \mid v \in V(\mathcal{X}) \cup C(\mathcal{X})\}$ and let $E_{\Sigma(v)}^\delta$ be the confidence ellipsoid centered at $\mu(v)$ for $v \in C(\mathcal{X})$. Note the dependence of $E_{\Sigma(v)}^\delta$ on parameter δ, which can be conveniently selected to maximize the confidence level to provide an accurate representation of the global dynamics.

For a given confidence level $1 - \delta$, define a multivalued map based on the trained GP as follows:

$$\mathcal{F}_\mu(\xi) := \left\{ \xi' \mid \xi' \cap \left(E_{\Sigma(v)}^\delta \cup R(\xi) \right) \neq \emptyset \text{ for } v \in C(\xi) \right\},$$

where $R(\xi)$ is the smallest box containing $\{\mu(v) \mid v \in V(\xi)\}$ and $V(\xi)$ is the set of corner points (vertices) of $\xi \in \mathcal{X}$, as in Fig. 3. $\mathcal{F}_\mu$ is refer as the *pointwise confidence multi-valued map*. Note that, for every $v \in C(\mathcal{X})$, the map $\mathcal{F}_\mu$ contains the confidence ellipsoid centered at $\mu(v)$. Therefore, all cubes in $\mathcal{X}$ have a pointwise confidence level of $1 - \delta$ at their centers.

Finally, use the multivalued map $\mathcal{F}_\mu$ to compute the Morse graph, the associated attractors and their RoAs. The CMGDB library [41] and RoA implement topological computations of MG and RoAs. In fact, assuming that the unknown ϕ_τ dynamics is a realization of the GP, the multivalued map $\mathcal{F}_\mu$ captures any given realization of GP with a pointwise confidence level of 1-δ at the center points of the discretization.

Step 4: Incremental Update After computing the RoAs and the attractors, the accuracy in estimating the RoAs can be further improved by iteratively collecting more data and performing steps 2 and 3 again. The new data can be randomly selected either in the whole state space or in the RoA of the interested attractor and it has to be consistent with the inital choice of the forward time τ. The former results in a more accurate description of the global dynamics since it decreases the overall uncertainty. The latter focuses on increasing the accuracy of the desired RoA.

Properties and Contribution: The theoretical foundations for this line of work can be found in prior publications [37]–[40], where it is shown that a Morse Graph reflects the global dynamics of any continuous system under assumptions aligned with those of Section III. When the system dynamics are generated by the (continuous) predicted mean μ of a GP, it is possible to incorporate the uncertainty estimate of the GP to obtain confidence levels on the global dynamics obtained by the Morse Graph [42]. Furthermore, under the assumption of sufficient data, $\mathcal{F}_{min}(\xi) \subset \mathcal{F}_\mu(\xi)$ for all $\xi \in \mathcal{X}$. Thus, the global dynamics obtained via Morse Graph has a pointwise confidence level of $1 - \delta$ for identifying the unknown dynamics. Relative to previous theoretical efforts, this work contributes: a) greater efficiency by using pointwise confidence guarantees as opposed to a global confidence level; b) an adaptive strategy for selecting samples guided by the Morse Graph and the GP; c) an effective solution for 2 to 4-dim. systems (relatively to 1-dim. examples in [42]); and d) implementation and experiments on models of robotic systems.

Discussion on Computational Cost: The cost of training the GP model is $O(M^4 N^3)$, N the size of the dataset and M the dimension of X. Computing the Morse Graph and RoAs is $O(V + E + G^2)$ with V the number of vertices in
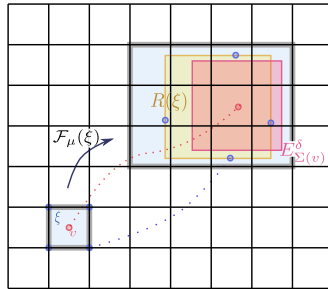
the directed graph $\mathcal{F}_\mu$, E the number of edges, and G the size of the Morse graph (typically smaller than 32) [1]. Thus, the total computational cost is $O((k+1)(M^4N^3+V+E))$, with k the number of incremental updates performed (Step 4). The memory requirement is $O(V+2(MN)^2)$, where the predicted mean and variance of the GP model store two matrices with total entries $(MN)^2$ and the grid size is V.

VI. Experimental Evaluation

The proposed framework, GPMG, is compared against alternatives from the literature for different dynamical systems and controllers (Table I). Section VI-A reports the following metrics for each benchmark: (a) Accuracy of RoA estimation, and (b) data efficiency, i.e., number of forward propagations of the true dynamics needed. Section VI-B describes the global dynamics and RoA discovered by GPMG.

Systems and Controllers: The **1D Quadrotor** (Quad) [45] is stabilized at a given height, generating trajectories rolled out in the simulator. The **Pendulum** (Pend) is governed by $m\ell^2\ddot{\theta} = mG\ell\sin\theta - \beta\dot{\theta} + u$, given mass m, gravity G, pole length l, and friction coefficient β. The **Mountain Car** (Car) is the continuous version of the popular Reinforcement Learning (RL) benchmark [46]. **Ackermann** (Ack) is a forward-only car-like first order system. **Lunar Lander** (Land) [47] is governed by $\ddot{x} = -\frac{k\dot{m}}{m} - g$, with $k > 0$ the velocity of exhaust gasses, mass flow rate $\dot{m} \leq 0$, and $g = 1.62$ moon's gravity. The **Two-link Acrobot** (Acro) [48] is controlled by a single torque between the links.

System	X	$\mathbb{U}$	Controllers
1D-Quadrotor[45]	$(z,\dot{z})$	T	Learned
Pendulum	$(\theta,\dot{\theta})$	τ	Learned, LQR
Mountain-Car	$(x,\dot{x})$	τ	Learned
Ackermann [49]	(x,y,θ)	(γ,V)	Learned, LQR, Corke
Lunar-Lander	$(h,\dot{h},m)$	$\dot{m}$	TOC
Acrobot [48]	$(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2)$	τ_2	Hybrid, LQR

TABLE I
SYSTEMS AND CONTROLLERS CONSIDERED IN THE EVALUATION.

LQR linearizes the system to compute a gain k used in the control law $u(x_t) = -k\cdot x_t$. A time-optimal controller (TOC) for the lunar lander [47] achieves soft landing by having a free-fall period and then switching to full-thrust until touchdown. The learned controllers are Soft Actor-Critic policy networks [50] trained to maximize the expected return $\mathbb{E}_{x_0\sim X}[\sum_{t=0}^\tau \mathcal{R}(x_t)]$, where the reward function is $\mathcal{R}: X \to \{0,-1\}$. $\mathcal{R}(x_t) = 0$ iff x_t is within an ϵ distance from the goal state and -1 otherwise. A hybrid controller takes two (analytical or learned) controllers $u_1(x), u_2(x), x \in X$, and applies one controller in predetermined subsets of the state space $X_1, X_2 \subset X$, i.e.: $\dot{x}_{t+1} = f(x_t,u_1)$ if $x_t \in X_1$, and $f(x_t,u_2)$ if $x_t \in X_2$.

A. Quantitative Results

Comparison Methods: Two lyapunov-based analytical methods (L-LQR and L-SoS) are used as comparison (as in [21], [51]). Both use a linearized unconstrained

---

[1]A detailed discussion of the computational costs can be found in [37], [43], [44].

---

form of the dynamics [52] to obtain a Lyapunov function (LF). L-LQR uses the solution of the Lyapunov equation $v_{LQR}(x) = x^T P x$ while L-SoS computes the LF as $v_{sos}(x) = m(x)^T Q m(x)$ where $m(x)$ are monomials on x and Q is a positive semidefinite matrix. L-SoS is implemented with SOSTOOLS [52] and SeDuMi [53]. These methods cannot be used with data-driven controllers (like Learned) since closed-form expression is required. The Lyapunov Neural Network (L-NN) [21] is a machine learning tool for identifying RoAs. The Morse graph (TopMG) employs the topological tools described but without any use of a surrogate model, instead it queries the true dynamics at every vertex of the discretization [37].

RoA Estimations: An approximation of the ground truth RoA for the goal is computed by considering a very high-resolution grid over X, and forward propagating for a long time horizon, or until the goal is reached. Table II presents the ratio of the ground truth RoA volume identified by each method for all benchmarks. With the exception of the Land (TOC) benchmark, GPMG consistently estimates a larger ratio of the RoA volume compared to alternatives.

Benchmark	L-NN	L-LQR/SOS	TopMG	GPMG
Quad (Learned)	-	N.A.	-	**1.0**
Pend (LQR)	**0.98**	0.7 / 0.03	0.97	0.91
Car (Learned)	-	N.A.	**1.0**	**1.0**
Land (TOC)	-	N.A.	**1.0**	0.79
Ack (Learned)	0.91	N.A.	**1.0**	**1.0**
Acro (LQR)	0.89	0.27 / 0.26	0.96	**1.0**
Acro (Hybrid)	0.14	N.A.	0.99	**1.0**

TABLE II
RoA RATIOS FOR THE METHODS. BEST VALUES PER ROW IN BOLD.

Estimating larger ratios of the RoA, however, may also lead to False Positives (FP) – incorrectly identifying a volume of the state space as being in RoA. The optional fourth step (Section V) of the proposed framework is crucial to mitigate FPs. All other methods (L-NN, L-LQR, SOS and TopMG), which require access to the true dynamics model, have zero FP. GPMG falsely labels $0\% - 2\%$ of X (Pend (LQR)) and $1\% - 22\%$ of X (Land (TOC)) as part of the RoA. These cases are further discussed in Section VI-B.

Benchmark	L-NN	TopMG	Ours: GPMG	Dim
Quad (Learned)	-	-	**25,000**	2
Pend (LQR)	667.1M	6.6M	**120,000**	2
Car (Learned)	-	6.6M	**3,000**	3
Land (TOC)	-	1M	**300,000**	3
Ack (Learned)	704.6M	520M	**10,000**	3
Acro (LQR)	5.7B	1.1B	**100,000**	4
Acro (Hybrid)	533M	2.1B	**2.5M**	4

TABLE III
PROPAGATIONS REQUIRED. BEST VALUES PER ROW IN BOLD.

Data Efficiency: The data efficiency of methods requiring access to the underlying dynamical system (L-NN, TopMG and GPMG) is measured using the total propagation steps required to estimate the RoA (Table III). The data requirements for GPMG are $2-4$ orders of magnitude less than TopMG and $3-5$ orders of magnitude less than L-NN. The learned controllers benefit the most from GPMG, as it provides, in all cases, a good coverage of the RoA with significantly fewer propagations and without FP (false positives).
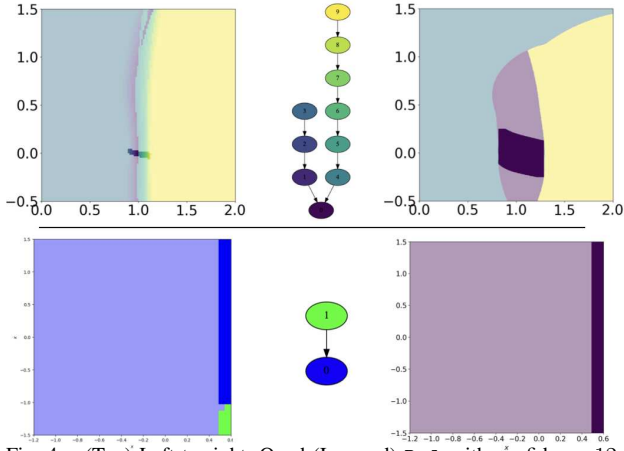
Fig. 4. (Top) Left to right: Quad (Learned) RoA with confidence 12.5%, associated Morse Graph, and RoA with confidence 95%. (Bottom) Left to right: RoA of trained GP for Car, associated Morse graph, and RoA of the true dynamics.

B. Qualitative case studies

1D Quadrotor: 100 random trajectories (average length 2.5s) that successfully reach the goal are used. To train the GP, trajectories are decomposed into short segments of $\tau = 0.3s$, giving a dataset $\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^M \times \mathbb{R}^M \mid y_i = \phi_\tau(x_i)\}_{i=1}^{800}$. For 12.5% and 95% confidence levels, GPMG outputs an MG with 9 nodes. The attractor discovered by GPMG shows the whole state space divided into two regions represented by the left (nodes 1-3) and right (nodes 4-9) parts of the MG. The trajectories of one region do not visit the other. These regions represent the system approaching the goal from above or below it. Node 0 (without leafs) of MG represents the region (Fig 4, darkest color) where all trajectories need to stabilize before reaching the goal region.

Pendulum: Initially, 300 random trajectories are used to train the initial surrogate model. The initial corresponding MG, RoAs and the σ are computed with 12.5% confidence (Fig 5, left). The initial procedure results in False Postives (FPs) that correspond to 2% of X. To decrease the number of FPs, and to improve the accuracy of the estimated MG, Step 4 of the proposed framework is applied. 10 more samples are randomly selected at the boundary of and inside the RoA of the attractor of interest. When the procedure is repeated 90 times, a confidence level of 95% is obtained, and all FPs are completely removed (Fig 5, right).

Mountain Car: GP is trained with 300 randomly sampled trajectories, each with duration $\tau = 1s$, the predicted σ is small enough to use a confidence level of 95%. The resulting MG (Fig 4 bottom) has two nodes and describes the expected global dynamics given by TopMG.

Lander: GPMG relies on the assumption that the underlying dynamical system ϕ_τ can be realized by a GP that uses traditional stationary kernels (Matérn, Exponential, and Logistic for the sake of this discussion). The Lander benchmark violates this assumption since the goal region is a line $\{(0, 0, m) \mid m_1 \leq m \leq m_2\}$ and not a system attractor. When trained with a Matérn kernel, GPMG obtains FP. Even with additional data, the surrogate model does not satisfactorily capture the underlying dynamics of the system.
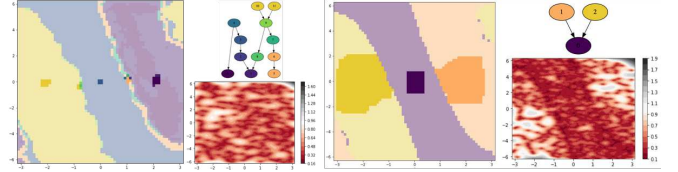

Fig. 5. RoAs, Morse Set and Standard Deviation (left) of the initial surrogate model with 12.5% confidence. After sampling 900 more trajectories: RoAs, Morse Set and Standard Deviation (right) with 95% confidence.

Ackermann: The GP is trained from 1000 randomly sampled trajectories, each of duration $\tau = 1s$. The predicted σ of the GP is small enough to consider the 95% confidence level. GPMG outputs an MG with a single node representing the system's attractor, which exhibits periodic behavior, agreeing with the global dynamics captured by TopMG. Hence, with significantly less data requirements, GPMG successfully captures the global dynamics information (a torus-like shaped attractor). If longer trajectories ($\tau = 40s$) are used, GPMG outputs a MG with a single node representing the attractor (0.008% of X by volume) without periodicity. This corresponds to the learned controller for Ackermann first performing periodic oscillations around the goal region before reaching it.

Acrobot: For both the Acro (LQR) and Acro (Hybrid) benchmarks, the initial data collected used 1000 random trajectories, each of duration 14s (LQR) and 25s (Hybrid). In both cases, the MG has a single node, and the identified attractor is a small set that contains the goal region. When additional data is provided, the uncertainty of GPMG decreases, but the size of the attractor does not change notably, and the ratio of its volume of X remains unchanged.

VII. DISCUSSION

This work integrates surrogate modeling via GPs with topology tools, achieving a data-efficient framework for identifying the global dynamics (attractors and RoAs), even for closed-box systems. Tests on different benchmarks show the proposed method consistently identifying attractors with larger RoA coverage and a significant reduction in data requirements. A confidence level is also assigned to the global dynamics representation output. This novel approach allows the user to either work with a sparse dataset sacrificing confidence level associated with the Morse Graph; or guide the process for additional collection, increasing confidence levels for the Morse Graph by training GPs with low overall uncertainty. For dynamical systems that cannot be realized via a GP with a traditional kernel (e.g., Lander), non-conventional kernels can be explored to accommodate irregular input domains and non-Gaussian outputs.

On the theoretical side, the rate of convergence of the Morse Graph to the true dynamical system as a function of incremental samples is unknown. The decreasing rate of the overall standard deviation might provide insights to estimate this rate of convergence. Finally, the application to large and high-dimensional state spaces may still be challenging although it is shown to be more data efficient than alternatives. Possible strategies to mitigate this are non-uniform state space discretizations, such as adaptive schemes.

References

[1] P. Giesl and S. Hafstein, "Review on computational methods for Lyapunov functions," Discrete & Continuous Dynamical Systems-B, vol. 20, no. 8, p. 2291, 2015.

[2] A. Vannelli and M. Vidyasagar, "Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems," Automatica, vol. 21, no. 1, pp. 69–80, 1985.

[3] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in CDC, 2017.

[4] A. V. Pesterev, "Attraction domain estimate for single-input affine systems with constrained control," Automation and Remote Control, vol. 78, no. 4, pp. 581–594, 2017.

[5] ——, "Attraction domain for affine systems with constrained vector control closed by linearized feedback," Autom. & Remote Control, vol. 80, no. 5, 2019.

[6] L. B. Rapoport and Y. V. Morozov, "Estimation of attraction domains in wheeled robot control using absolute stability approach," IFAC, vol. 41, no. 2, pp. 5903–5908, 2008.

[7] R. Pandita, A. Chakraborty, P. Seiler, and G. Balas, "Reachability and RoA analysis applied to GTM dynamic flight envelope assessment," in AIAA CNC, 2009.

[8] D. Henrion and M. Korda, "Convex computation of the RoA of polynomial control systems," IEEE Tran. on Automatic Control, vol. 59, no. 2, pp. 297–312, 2013.

[9] P. A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology, 2000.

[10] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," IJRR, vol. 29, no. 8, 2010.

[11] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," The International Journal of Robotics Research, vol. 36, no. 8, pp. 947–982, 2017.

[12] M. Posa, M. Tobenkin, and R. Tedrake, "Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares," in HSCC, 2013, pp. 63–72.

[13] J. J. Choi, A. Agrawal, K. Sreenath, C. J. Tomlin, and S. Bansal, "Computation of RoAs for Hybrid Limit Cycles Using Reachability," arXiv:2201.08538, 2022.

[14] J. H. Gillulay and C. J. Tomlin, "Guaranteed safe online learning of a bounded system," in IROS, 2011.

[15] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," in CDC, 2014.

[16] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in ICRA, 2018.

[17] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning lyapunov functions for hybrid systems," in HSCC, 2021, pp. 1–11.

[18] A. Abate, D. Ahmed, A. Edwards, M. Giacobbe, and A. Peruffo, "FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks," in HSCC, 2021, pp. 1–11.

[19] G. Mamakoukas, I. Abraham, and T. D. Murphey, "Learning stable models for prediction and control," IEEE Trans Robot, 2020.

[20] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning region of attraction for nonlinear systems," in 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6477–6484.

[21] S. M. Richards, F. Berkenkamp, and A. Krause, "Lyapunov Neural Network: Adaptive stability certification for safe learning of dynamical systems," in CoRL, 2018.

[22] A. Lederer and S. Hirche, "Local Asymptotic Stability Analysis and Region of Attraction Estimation with Gaussian Processes," in CDC, 2019.

[23] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," arXiv preprint arXiv:2109.14152, 2021.

[24] J. Wang, A. Hertzmann, and D. J. Fleet, "Gaussian process dynamical models," Advances in neural information processing systems, vol. 18, 2005.

[25] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 2, pp. 408–423, 2013.

[26] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold gaussian processes for regression," in 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016, pp. 3338–3345.

[27] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics," in 2021 American Control Conference (ACC), 2021, pp. 3683–3690.

[28] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of RoAs for uncertain, nonlinear systems with Gaussian Processes," in CDC, 2016.

[29] S. Bhattacharya, S. Kim, H. Heidarsson, G. S. Sukhatme, and V. Kumar, "A topological approach to using cables to manipulate sets of objects," IJRR, vol. 34, no. 6, 2015.

[30] R. Antonova, A. Varava, P. Shi, J. F. Carvalho, and D. Kragic, "Sequential topological representations for predictive models of deformable objects," in L4DC, 2021.

[31] Q. Ge, T. Richmond, B. Zhong, T. M. Marchitto, and E. J. Lobaton, "Enhancing the morphological segmentation of microscopic fossils through localized topology-aware edge detection," Autonomous Robots, vol. 45, no. 5, pp. 709–723, 2021.

[32] A. Varava, K. Hang, D. Kragic, and F. T. Pokorny, "Herding by caging: a topological approach towards guiding moving agents via mobile robots." in R:SS, 2017.

[33] F. T. Pokorny, D. Kragic, L. E. Kavraki, and K. Goldberg, "High-dimensional winding-augmented motion planning with 2D topological task projections & persistent homology," in ICRA, 2016.

[34] J. F. Carvalho, M. Vejdemo-Johansson, F. T. Pokorny, and D. Kragic, "Long-term prediction of motion trajectories using path homology clusters," in IROS, 2019.

[35] A. Orthey and M. Toussaint, "Visualizing local minima in multi-robot motion planning using multilevel morse theory," in International Workshop on the Algorithmic Foundations of Robotics. Springer, 2020, pp. 502–517.

[36] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," IJRR, vol. 21, no. 4, pp. 331–344, 2002.

[37] E. R. Vieira, E. Granados, A. Sivaramakrishnan, M. Gameiro, K. Mischaikow, and K. E. Bekris, "Morse graphs: Topological tools for analyzing the global dynamics of robot controllers," in Algorithmic Foundations of Robotics XV. Springer International Publishing, 2023, pp. 436–453.

[38] W. D. Kalies, K. Mischaikow, and R. Vandervorst, "Lattice structures for attractors I," J. Comput. Dyn., vol. 1, no. 2, pp. 307–338, 2014.

[39] ——, "Lattice structures for attractors II," Foundations of Computational Mathematics, vol. 1, no. 2, pp. 1–41, 2015.

[40] ——, "Lattice Structures for Attractors (III)," Journal of Dynamics and Differential Equations, pp. 1572–9222, 2021.

[41] S. H. Marcio Gameiro, "CMGDB: Conley Morse Graph Database Software," 2022.

[42] B. Batko, M. Gameiro, Y. Hung, W. Kalies, K. Mischaikow, and E. Vieira, "Identifying nonlinear dynamics with high confidence from sparse data," arXiv preprint arXiv:2206.13779, 2022.

[43] R. Gramacy, Surrogates: Gaussian process modeling, design and optimization for the applied sciences. Chapman Hall/CRC, Boca Raton, FL., 2020.

[44] J. Bush, M. Gameiro, S. Harker, H. Kokubu, K. Mischaikow, I. Obayashi, and P. Pilarczyk, "Combinatorial-topological framework for the analysis of global dynamics," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 22, no. 4, 2012.

[45] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, "safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning," arXiv preprint arXiv:2109.06325, 2021.

[46] A. W. Moore, "Efficient memory-based learning for robot control," 1990.

[47] J. Meditch, "On the problem of optimal thrust programming for a lunar soft landing," IEEE Transactions on Automatic Control, vol. 9, no. 4, pp. 477–484, 1964.

[48] M. Spong, "The swing up control problem for the Acrobot," IEEE Control Systems Magazine, vol. 15, no. 1, pp. 49–55, 1995.

[49] P. I. Corke and O. Khatib, Robotics, vision and control: fundamental algorithms in MATLAB. Springer, 2011, vol. 73.

[50] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in ICML, 2018.

[51] M. Farsi, Y. Li, Y. Yuan, and J. Liu, "A piecewise learning framework for control of unknown nonlinear systems with stability guarantees," in Learning for Dynamics and Control Conference. PMLR, 2022, pp. 830–843.

[52] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing SOSTOOLS: A general purpose sum of squares programming solver," in CDC, 2002.

[53] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," Optimization methods and software, vol. 11, no. 1-4, pp. 625–653, 1999.