

Scalable Data-Driven Reachability Analysis and Control via Koopman Operators with Conformal Coverage Guarantees

Devesh Nath*¹

Haoran Yin*¹

Glen Chou²

DNATH7@GATECH.EDU

HYIN95@GATECH.EDU

CHOU@GATECH.EDU

Georgia Institute of Technology. ¹ School of ECE, ² Schools of Cybersecurity & Privacy & Aerospace Eng.

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

We propose a scalable reachability-based framework for probabilistic, data-driven safety verification of unknown nonlinear dynamics. We use Koopman theory with a neural network (NN) lifting function to learn an approximate linear representation of the dynamics, enabling fast and scalable closed-loop reachability analysis in the lifted space. We design a linear tracking controller in this space for a given reference trajectory, and map the resulting reachable set back to the original state space via NN verification tools. To capture model mismatch between the Koopman dynamics and the true system, we apply conformal prediction to produce statistically-valid error bounds that inflate the reachable sets to ensure the true trajectories are contained with a user-specified probability. These bounds generalize across references, enabling reuse without recomputation. Results on high-dimensional MuJoCo tasks (11D Hopper, 28D Swimmer) and 12D quadcopters show improved reachable set coverage rate, computational efficiency, and conservativeness over existing methods.

Keywords: safety-critical control, reachability, Koopman operators, conformal prediction

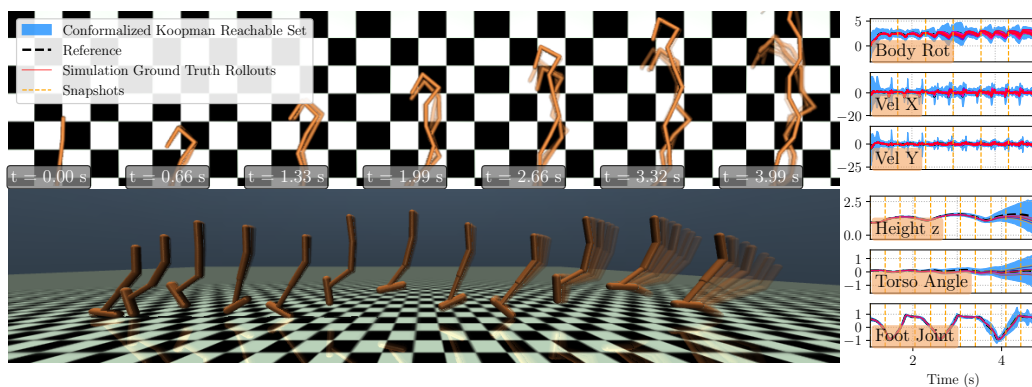


Figure 1: Executed trajectories / reachable sets for a 28D MuJoCo swimmer (a); 11D hopper (b).

1. Introduction

Computing reachable sets, i.e., the set of states reachable over a time horizon, is key to robot safety. However, reachability analysis for robots is difficult as they often (i) lack analytical models, (ii) have high-dimensional nonlinear dynamics, and (iii) require reasoning over long horizons. Even if analytical dynamics are available, challenges (ii) and (iii) cause nonlinear reachability tools (Bansal et al., 2017; Althoff, 2015) to become prohibitive or to suffer from excessive overapproximation over long horizons (Rober and How, 2024). Data-driven reachability (Park et al., 2024; Bak et al., 2025) can be used without analytical models, but can be data-inefficient or lack guarantees that the estimated sets contain all possible trajectories. In contrast, linear reachability scales to long horizons and high dimensions (Bak et al., 2019). The Koopman framework learns linear dynamics in a lifted state space, e.g., via neural networks (NNs), to approximate the nonlinear system (Shi et al.,

2024; Mezić, 2020). While reachability analysis for the Koopman linearization can be more scalable than for the nonlinear dynamics, existing methods only provide guarantees for the approximate lifted dynamics. Thus, *safety may not hold for the true system* (Bak et al., 2021, 2025). Moreover, existing Koopman reachability methods analyze a *single* dynamical system. However, robots often require multiple controllers to complete different tasks, forcing existing methods to recompute reachable sets from scratch for each new controller, which is inefficient. Reusing previously computed information would enable more efficient reachability analysis across controllers.

To close these gaps in scalability, speed, and reusability, we propose **ScaRe-Kro** (Scalable Reachability via Koopman Operators), which uses Koopman theory with an NN lifting function to perform reachability analysis and control. Unlike prior Koopman reachability methods for autonomous systems, we also design a linear controller in this lifted space to track a distribution of reference trajectories, each completing a different task. ScaRe-Kro efficiently computes closed-loop reachable sets under the induced *linear tracking error dynamics* in the lifted Koopman space and maps them to the original state space via NN verification tools. To account for Koopman model error, we apply conformal prediction (CP) to derive statistically-valid error bounds. By inflating the mapped reachable sets with these CP bounds, they are guaranteed to contain the *true system trajectories* with a user-specified probability (e.g., 97.5%). Moreover, as the CP bounds are calibrated across the trajectory distribution, they can be reused across different references and thus closed-loop systems, unlike prior Koopman approaches that certify only a single autonomous system. Overall, using linear reachability and CP, our method enables scalable data-driven probabilistic verification and control, improving both efficiency and conservativeness over prior work. Our contributions are:

1. A scalable Koopman-based reachability framework for unknown nonlinear dynamics, using NN lifting functions and NN verification to estimate reachable sets in the original state space.
2. An approach for inflating the resulting Koopman reachable sets using CP, to obtain probabilistic guarantees of overapproximation for the *true* dynamics.
3. Control design in the lifted space for closed-loop tracking of a reference trajectory distribution, with CP-based inflation bounds calibrated to be reusable across the distribution.
4. Extensive evaluation, showing reliable verification for high-dimensional robotic systems (up to 28D), outperforming baselines in safety rate, computation speed, and conservativeness.

2. Related Work

Reachability via Linear Relaxation While exact reachable set computation for nonlinear systems is typically intractable, reachable set overapproximations (RSOAs) can often be efficiently computed to certify safety (Althoff et al., 2021). When the closed-loop system involves NN components (e.g., learned dynamics or controller), RSOAs can be computed via linear relaxation-based NN verification (NNV) tools (e.g., CROWN (Zhang et al., 2018)) (Everett et al., 2021; Jafarpour et al., 2024). Here, conservativeness grows with the number of nonlinearities (Nath et al., 2025) and can be excessive for nonlinear robot dynamics. Symbolic and one-shot methods mitigate this but raise runtime and memory costs (Chen et al., 2023; Rober and How, 2024; Everett et al., 2021). We instead model the robot with Koopman-linearized dynamics, reducing the number of nonlinearities in the computational graph (CG) and improving scalability (i.e., computation time, conservativeness).

Koopman Reachability Koopman (Koopman and Neumann, 1932) and Carleman (Amini et al., 2021; Bayer and Leine, 2023) linearizations accelerate control (Strässer et al., 2026) and reachability but introduce approximation error. Error bounds exist but are limited to quadratic or polynomial dynamics (Liu et al., 2021; Forets and Pouly, 2017; Amini et al., 2021), which do not apply to many

robot dynamics models. Prior Koopman reachability methods also lack reachable set coverage calibration (Thapliyal and Hwang, 2022) or guarantee containment only for a finite set of observed trajectories (Kochdumper and Bak, 2022), failing to ensure safety for previously-unseen trajectories of the *true* system. Prior methods also do not support NN liftings and consider autonomous systems, which corresponds to fixing a single controller. Changing controllers, as often done in robotics, requires repeating the calibration (Forets and Schilling, 2021; Bak et al., 2021, 2025), which is inefficient. In contrast, we support NN liftings, provide probabilistic coverage guarantees for the *true* dynamics, and integrate control design, enabling data-efficient reachability across multiple controllers.

Data-Driven Reachability Data-driven reachability uses black-box trajectory data. Some methods adapt Hamilton-Jacobi analysis (Chilakamarri et al., 2024), but this is costly for high-dimensional systems. Others learn reachability functions (Sun and Mitra, 2022) but require dense data to accurately generalize, or yield ellipsoidal RSOs (Park et al., 2024) assuming privileged system data (e.g., Lipschitz constants). Sampling-based methods (Lew et al., 2022) are limited to short horizons. Scenario optimization provides probabilistic guarantees (Dietrich et al., 2024) but is slower than Koopman-based methods (Bak et al., 2025). Moreover, these methods assume a fixed controller, yet robots often switch controllers to achieve different goals, altering the closed-loop dynamics and forcing reachable sets to be recomputed. Trajectory-tracking reachability provides reusable tracking-error sets across reference trajectories (Herbert et al., 2017; Singh et al., 2018; Knuth et al., 2021). Data-driven variants learn contraction metrics (Chou et al., 2021, 2022; Knuth et al., 2023) for tracking-based reachability of unknown systems, but finding such metrics is generally difficult. We build on this work, yielding Koopman reachable sets that are fast to compute, scalable to long horizons, and reusable for various closed-loop dynamics induced by tracking different trajectories.

3. Preliminaries

We consider unknown, black-box nonlinear systems, either in open-loop (1a) or closed-loop (1b):

$$x_{t+1} = f(x_t, u_t), \quad (1a) \quad x_{t+1} = f(x_t, \pi(x_t, t)) \doteq \tilde{f}(x_t), \quad (1b)$$

with timestep $t \in \{0, \dots, T\} \doteq \mathcal{T}$, state $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$, control $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$, and feedback control law $\pi : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{U}$. The initial state x_0 lies in set $\mathcal{X}_0 \subseteq \mathcal{X}$. Denote $\text{Int}(\underline{a}, \bar{a}) = \{a \mid \underline{a} \leq a \leq \bar{a}\}$, as an interval, where $\underline{a}, \bar{a} \in \mathbb{R}^A$ and inequalities hold element-wise, i.e., $\underline{a}_i \leq a_i \leq \bar{a}_i, 1 \leq i \leq A$. Denote the a -ball as $\mathcal{B}_a(c) \doteq \{x \mid \|x - c\|_\infty \leq a\}$ and sequence $q_{1:Q} \doteq \{q_1, q_2, \dots, q_Q\} = \{q_i\}_{i=1}^Q$. We use \oplus and \ominus to denote the Minkowski sum and difference, respectively.

Koopman Operators As our work relies on Koopman operators, we discuss the basic concepts here. Consider autonomous discrete-time nonlinear dynamics $x_{t+1} = f(x_t)$. A Koopman operator \mathcal{K} is an infinite-dimensional linear operator that evolves the observables ϕ^∞ of the state $\mathcal{K}\phi^\infty(x) = \phi^\infty \circ f(x)$ according to $\phi^\infty(x_{t+1}) = \mathcal{K}\phi^\infty(x_t)$, where \circ denotes function composition. As infinite-dimensional Koopman operators are intractable to obtain in practice, we define an approximation to \mathcal{K} using a finite-dimensional lifting function $\phi : \mathcal{X} \rightarrow \mathcal{Z} \subseteq \mathbb{R}^l$ and matrix $K_A \in \mathbb{R}^{l \times l}$, which define lifted dynamics $z_{t+1} = K_A z_t$, where $z_t = \phi(x_t)$ is the lifted state and $z_t \in \mathcal{Z} \subseteq \mathbb{R}^l$. Here, l can be arbitrarily selected by the choice of the lifting function ϕ . For systems with control input (1a), the lifted dynamics can be similarly written as $z_{t+1} = K_A z_t + K_B u_t$, where $K_B \in \mathbb{R}^{l \times m}$. In practice, ϕ , K_A , and K_B can be learned from data, with the lifting function ϕ parameterized using polynomials or NNs (Shi et al., 2024). In this work, we utilize an NN-based autoencoder architecture, where the encoder $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ defines the lifting function and the decoder $\psi : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$ defines its learned inverse, i.e., $\psi(\phi(x)) \approx x$; if ψ is a perfect inverse, $\psi(\phi(x)) = x$ and $\hat{\mathcal{X}} = \mathcal{X}$.

Reachability via NN Verification (NNV) We use NNV to compute reachable sets for closed-loop systems involving NN-based lifting and inverse functions. NNV tools represent an NN as a

CG – a directed acyclic graph encoding the sequence of operations applied to an input (Rober and How, 2024). For some CG G with input set $\mathcal{S} \subseteq \mathbb{R}^{n_i}$ and output $G(\mathcal{S}) \subseteq \mathbb{R}^{n_o}$, we can compute a guaranteed overapproximation of its image $G(\mathcal{S})$ using CROWN-based (Zhang et al., 2018) tools like `auto_LiRPA` (Xu et al., 2020). These tools provide guaranteed affine lower and upper bounds, \underline{G} and \overline{G} , on the output $G(\mathcal{S})$ for any interval \mathcal{S} , formalized in this proposition:

Proposition 1 (CG Robustness (Xu et al., 2020)) *For some CG G and interval $\mathcal{S} \doteq \{s \in \mathbb{R}^{n_i} \mid \underline{s} \leq s \leq \overline{s}\}$, there are affine functions $\underline{G}, \overline{G}$ such that $\forall s \in \mathcal{S}, \underline{G}(s) \leq G(s) \leq \overline{G}(s)$. The inequalities hold element-wise and $\underline{G}(s) = \Psi s + \alpha$, $\overline{G}(s) = \Phi s + \beta$, with $\Psi, \Phi \in \mathbb{R}^{n_o \times n_i}$ and $\alpha, \beta \in \mathbb{R}^{n_o}$.*

Given controller π , dynamics (1a), and initial set \mathcal{X}_0 , the T -timestep exact reachable set is denoted as $\mathcal{X}_{0:T} \doteq \{\mathcal{X}_0, \{f(\mathcal{X}_t, \pi(\mathcal{X}_t, t))\}_{t=0}^{T-1}\}$. While exact reachability analysis is generally intractable for nonlinear systems (Althoff et al., 2021), it is often feasible to compute a reachable set overapproximation (RSOA) $\overline{\mathcal{X}}_{0:T}$, which satisfies $\mathcal{X}_t \subseteq \overline{\mathcal{X}}_t$ for all $t \in \{0, \dots, T\}$. To obtain $\overline{\mathcal{X}}_{0:T}$, Prop. 1 can be applied to the CG F of the T -step composition $\tilde{f}^T(\cdot) \doteq \tilde{f} \circ \dots \circ \tilde{f}(\cdot)$ of the closed-loop dynamics \tilde{f} (1b), i.e., $F(x_0) \doteq \{\tilde{f}(x_t) \doteq f(x_t, \pi(x_t, t))\}_{t=0}^T$, with `auto_LiRPA` computing the RSOA.

Conformal Prediction (CP) Our RSOA computations also rely on CP. In this work, we adopt split CP, which partitions i.i.d. input-output pairs $(v^{(i)}, y^{(i)}) \in \mathcal{V} \times \mathcal{Y}$ from a dataset $\mathcal{D} \doteq \{(v^{(i)}, y^{(i)})\}_{i=1}^{L+K}$ into a size- L training set \mathcal{D}_T and a size- K calibration set \mathcal{D}_C . For a given prediction function $\mu : \mathcal{V} \rightarrow \mathcal{Y}$, we compute a nonconformity score $R^{(i)} \doteq s(y^{(i)}, \mu(v^{(i)}))$ for each of the K calibration pairs in \mathcal{D}_C , where $s : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ is a chosen nonconformity score function. As an example, we can express the ℓ_2 prediction error of a learned dynamics model \hat{f} in this framework by defining $v = (x_t, u_t)$, $y = x_{t+1}$, $\mu(v) = \hat{f}(v)$, and $s(y, \mu(v)) = \|y - \mu(v)\|_2$. Given a miscoverage level $\delta \in (0, 1)$, a threshold C is defined as the $(1 - \delta)$ -quantile of the calibration scores, $C \doteq \text{Quantile}_{1-\delta}(R^{(1)}, \dots, R^{(K)}, \infty)$. This threshold defines the prediction set $\mathcal{Y}(v) \doteq \{y \in \mathcal{Y} \mid s(y, \mu(v)) \leq C\}$, which satisfies the marginal coverage guarantee $\mathbb{P}(y^{(0)} \in \mathcal{Y}(v^{(0)})) \geq 1 - \delta$ for an unseen data point $(y^{(0)}, v^{(0)})$ that is exchangeable with \mathcal{D} (Lindemann et al., 2024).

4. Problem Statement

We are given black-box access to f (1a), i.e., we can query (1a) but do not know its analytical form and its output can only be estimated via data. We have a planner $P : \mathcal{X} \rightarrow \mathcal{X}^{T+1} \times \mathcal{U}^T$ that generates open-loop reference trajectories $(x_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ satisfying (1a), starting from an initial condition $x_0^{\text{ref}} \in \mathcal{X}_0 \ominus \mathcal{B}_\epsilon(0)$ that is uniformly distributed over $\mathcal{X}_0 \ominus \mathcal{B}_\epsilon(0)$. We assume nothing further about P ; it can be a learned or traditional planner. At runtime, the true initial state x_0 may be perturbed from x_0^{ref} , i.e., $x_0 \in \mathcal{B}_\epsilon(x_0^{\text{ref}})$, where x_0 is uniformly distributed over $\mathcal{B}_\epsilon(x_0^{\text{ref}})$. To be robust to this error, we design a controller $\pi : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{U}$ to track references from P and compute high-probability RSOAs $\overline{\mathcal{X}}_{0:T}$ for the resulting closed-loop dynamics (1b), such that $\mathbb{P}(\bigwedge_{t=0}^T (x_t \in \overline{\mathcal{X}}_t)) \geq 1 - \delta$. This process should scale to high-dimensional nonlinear systems over long horizons (e.g., $n \geq 25$, $T \geq 200$). We solve:

Problem 1 (Model learning & control design) *Given N length- $(T + 1)$ open-loop, dynamically-feasible trajectories generated by P via black-box queries of (1a), forming a training dataset $\mathcal{D} = \{(x_{0:T}^{(i)}, u_{0:T-1}^{(i)})\}_{i=1}^N$, (a) train a Koopman model (ϕ, ψ, K_A, K_B) and (b) use the Koopman model to design a time-varying trajectory-tracking controller $\pi : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{U}$ to stabilize to references generated by P .*

Problem 2 ($(1 - \delta)$ -confident RSOA computation) *Given M length- $(T + 1)$ closed-loop trajectories generated by tracking references from P with the controller π from Prob. 1, compute a $(1 - \delta)$ -confident RSOA $\overline{\mathcal{X}}_{0:T}^{1-\delta}$ for the closed-loop dynamics (1b), which guarantees for any random reference $(x_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ produced by the planner P with $x_0^{\text{ref}} \in \mathcal{X}_0$ and initial state $x_0 \in \mathcal{B}_\epsilon(x_0^{\text{ref}})$, the closed-loop trajectory remains in $\overline{\mathcal{X}}_{0:T}^{1-\delta}$ with probability at least $1 - \delta$, i.e., $\mathbb{P}(\bigwedge_{t=0}^T (x_t \in \overline{\mathcal{X}}_t)) \geq 1 - \delta$.*

5. Methodology

Our method (Fig. 2, Alg. 1) learns a Koopman operator (Sec. 5.1), uses it for control design (Sec. 5.2), performs fast linear reachability analysis on the closed-loop lifted Koopman dynamics (Sec. 5.3), and inflates these reachable sets via CP to obtain probabilistic coverage guarantees (Sec. 5.4).

5.1. Koopman Operator Training

To solve Prob. 1, we train a Koopman operator model (ϕ, ψ, K_A, K_B) on the training dataset $\mathcal{D} \doteq \{(x_{0:T}^{(i)}, u_{0:T-1}^{(i)})\}_{i=1}^N$ of open-loop trajectories from a planner P (details in Sec. 6). We use a composite loss $\mathcal{L} = \sum_{i=1}^N (\lambda_1 \mathcal{L}_1^{(i)} + \lambda_2 \mathcal{L}_2^{(i)})$. The first term, $\mathcal{L}_1^{(i)} = \sum_{j=0}^T [\|x_j^{(i)} - \psi(\phi(x_j^{(i)}))\|^2]$, enforces autoencoder accuracy via state reconstruction error. The second term, $\mathcal{L}_2^{(i)} = \sum_{j=t+1}^{t+H} [\|x_j^{(i)} - \psi(\tilde{z}_j^{(i)})\|^2 + \|\phi(x_j^{(i)}) - \tilde{z}_j^{(i)}\|^2]$, where $\tilde{z}_j^{(i)} \doteq K_A \tilde{z}_{j-1}^{(i)} + K_B u_{j-1}^{(i)}$ and $\tilde{z}_0^{(i)} = \phi(x_0^{(i)})$, is a multi-step (horizon H) dynamics loss penalizing prediction errors from the linear model (K_A, K_B) in both latent and state space. The latent consistency term is crucial for long-horizon prediction accuracy. We initialize K_A as identity and K_B with Xavier uniform, train via learning rate annealing and weight decay, and set (ϕ, ψ) to have ReLU or GELU activations.

5.2. Koopman Tracking LQR Design

To reliably imitate state/control reference trajectories generated by the planner P in the lifted state space \mathcal{Z} , a tracking controller π is required to stabilize against initial condition perturbations $x_0 - x_0^{\text{ref}}$ and to reject disturbances due to model error in the learned lifted dynamics $z_{t+1} = K_A z_t + K_B u_t$. To obtain π , we use the linearity of the lifted dynamics to design a linear quadratic regulator (LQR) that tracks trajectories in \mathcal{Z} . Given a reference state-space trajectory $x_{0:T}^{\text{ref}}$ produced by P , we map it to the lifted state space as $z_{0:T}^{\text{ref}} \doteq \{\phi(x_t^{\text{ref}})\}_{t=0}^T$. The corresponding feedforward control trajectory $u_{0:T-1}^{\text{ref}}$ which minimizes the lifted-state imitation error is then computed as:

$$u_t^{\text{ref}} = K_B^\dagger (z_{t+1}^{\text{ref}} - K_A z_t^{\text{ref}}), \quad (2)$$

where K_B^\dagger is the pseudo-inverse of K_B . The error states $\delta z_t = z_t - z_t^{\text{ref}}$ and controls $\delta u_t = u_t - u_t^{\text{ref}}$ follow linear error dynamics $\delta z_{t+1} = K_A \delta z_t + K_B \delta u_t$, starting from an initial error $\delta z_0 = \phi(x_0) - z_0^{\text{ref}}$, where $x_0 \in \mathcal{B}_\epsilon(x_0^{\text{ref}})$. We solve an LQR problem, finding the feedback controls $\delta u_{0:T-1}$ that minimize the finite-horizon quadratic cost $\delta z_T^\top Q_T \delta z_T + \sum_{t=0}^{T-1} (\delta z_t^\top Q \delta z_t + \delta u_t^\top R \delta u_t)$, subject to the linear dynamics $\delta z_{t+1} = K_A \delta z_t + K_B \delta u_t$, where $Q \succ 0$, $R \succ 0$, and $Q_T \succ 0$. In particular, we solve a backward Riccati recursion to obtain the optimal feedback gains $G_{0:T-1}$. This yields the tracking control law in the lifted state space (3), which can be applied on the nonlinear system (1a) as (4) by measuring and encoding the state x_t :

$$u_t = u_t^{\text{ref}} - G_t (z_t - z_t^{\text{ref}}), \quad (3) \quad u_t = u_t^{\text{ref}} - G_t (\phi(x_t) - z_t^{\text{ref}}). \quad (4)$$

5.3. Reachability Analysis for Closed-Loop Koopman Dynamics

The high computational cost of nonlinear reachability motivates our use of Koopman operators. Standard RSOA computation repeatedly overapproximates nonlinearities at each timestep in both f and π (which typically must be nonlinear to stabilize f), leading to high conservativeness. A T -step RSOA requires $2T$ sequential nonlinear function evaluations, causing reachable sets to become vacuously loose as T increases. To reduce conservativeness, partition-based methods split the input set $\bar{\mathcal{X}}_t$, propagate each partition element, and union the result; however, this incurs computational and memory costs that increases with the cardinality of the partition. In contrast, Koopman-based reachability improves scalability and reduces conservativeness without partitioning overhead by reducing the number of nonlinearities. Mapping to the lifted space \mathcal{Z} creates linear dynamics, eliminating

the need to bound nonlinearities during propagation. The linear dynamics also permit a linear tracking controller. Consequently, computing an RSOA $\bar{\mathcal{X}}_t$ for any *single* timestep t requires bounding only a single application of the nonlinear encoder (ϕ) and decoder (ψ), reducing the number of nonlinear bounding operations from $2T$ to just 2. Moreover, the RSOA for all timesteps $\bar{\mathcal{X}}_{1:T}$ can be computed efficiently via T parallel decoder bounding operations. To apply this insight formally for RSOA computation, we define the Tracking Koopman Feedback Loop (TKFL), denoted Γ , as a CG that describes the closed-loop dynamics for tracking $z_{0:T}^{\text{ref}} \doteq \{\phi(x_t^{\text{ref}})\}_{t=0}^T$ in the lifted state space using the feedback control law (4). Specifically, the TKFL $\Gamma : \mathcal{X}_0 \times \mathcal{U}^T \times \mathcal{Z}^{T+1} \rightarrow \hat{\mathcal{X}}^{T+1}$ maps the initial state $x_0 \in \mathcal{B}_\epsilon(x_0^{\text{ref}})$, the feedforward control trajectory $u_{0:T-1}^{\text{ref}}$ (2), and the lifted reference $z_{0:T}^{\text{ref}}$ to the decoded trajectory $\hat{x}_{0:T} \doteq \{\psi(z_t)\}_{t=0}^T$ representing the predicted closed-loop trajectory (5) in the original state space, where the lifted states z_t are updated via the closed-loop lifted dynamics (6):

$$\hat{x}_{0:T} = \{x_0, \{\psi(\tilde{f}_z(z_t))\}_{t=0}^{T-1}\}, \quad (5) \quad z_{t+1} = K_A z_t + K_B (u_t^{\text{ref}} - G_t(z_t - z_t^{\text{ref}})) \doteq \tilde{f}_z(z_t). \quad (6)$$

To summarize, the CG is constructed as follows (see Fig. 2 for visualization). First, we map the initial state via ϕ into the latent space, computing $z_0 = \phi(x_0)$. Second, we unroll the T -fold recursive application of the single-step latent dynamics, $z_{t+1} = K_A z_t + K_B u_t$, where u_t is the closed-loop tracking control computed using (3), to obtain the latent trajectory $z_{0:T} = \{z_0, z_2, \dots, z_T\}$. Third, $z_{0:T}$ is mapped back to the original state space via the decoder ψ to obtain $\hat{x}_{0:T}$. The CG Γ is passed to the `auto_LiRPA` library (Xu et al., 2020), which enables set-based bound propagation through Γ using CROWN (Zhang et al., 2018). `auto_LiRPA` computes vector-valued affine bounding functions, $\underline{\Gamma}$ and $\bar{\Gamma}$, which enables us to compute an RSOA $\hat{\mathcal{X}}_{0:T}$ for the decoded closed-loop lifted dynamics (5) by considering specific components of Γ , $\underline{\Gamma}_t$ and $\bar{\Gamma}_t$, that bound the states reached under (5) at timestep t , starting from any initial condition $x_0 \in \mathcal{X}_0$. This resulting hyper-rectangular set $\hat{\mathcal{X}}_{0:T}$, which we refer to as the Koopman reachable set (KRS), is found by solving the optimization (7), which is computed efficiently by `auto_LiRPA`:

$$\hat{\mathcal{X}}_{0:T} \doteq \left\{ \text{Int} \left(\min_{x_0 \in \mathcal{X}_0} \underline{\Gamma}_t(x_0, u_{0:T-1}^{\text{ref}}, z_{0:T}^{\text{ref}}), \max_{x_0 \in \mathcal{X}_0} \bar{\Gamma}_t(x_0, u_{0:T-1}^{\text{ref}}, z_{0:T}^{\text{ref}}) \right) \right\}_{t=0}^T \quad (7)$$

The KRS $\hat{\mathcal{X}}_{0:T}$ overapproximates $\hat{x}_{0:T}$ so that $\hat{x}_t \in \hat{\mathcal{X}}_t$, for all $t \in \mathcal{T}$, ensuring the decoded closed-loop lifted dynamics (5) remain in $\hat{\mathcal{X}}_{0:T}$. Formally, we state the following result (proof in App. 8.1):

Lemma 1 (KRS overapproximation) *Let Γ denote the closed-loop CG of TKFL, and let \mathcal{X}_0 be the initial set. Suppose the KRS $\hat{\mathcal{X}}_{0:T}$ is defined as in (7). Then, for any initial state $x_0 \in \mathcal{X}_0$, the decoded closed-loop trajectory (5), i.e., $\hat{x}_{0:T} = \Gamma(x_0, z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ satisfies $\hat{x}_t \in \hat{\mathcal{X}}_t$ for all $t \in \mathcal{T}$.*

Crucially, the KRS overapproximates the reachable set for the decoded Koopman linearized dynamics (5), *not the true nonlinear dynamics (1b)*, due to Koopman model error. This motivates Sec. 5.4.

5.4. Conformalizing Koopman Reachable Sets

We now explain how to inflate the KRS to guarantee it contains the true system (1b) trajectories with a user-specified probability $1 - \delta$. We use CP to compute a high-probability upper bound on

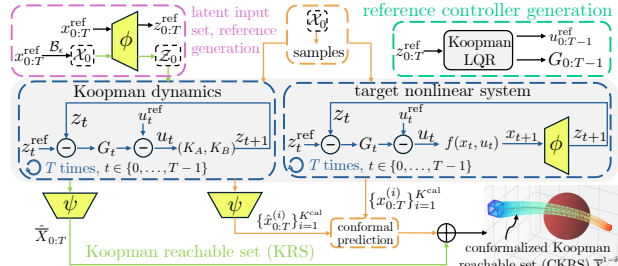


Figure 2: ScaRe-Kro. Green and orange arrows represent KRS and CP computations, respectively.

The CG Γ is passed to the `auto_LiRPA` library (Xu et al., 2020), which enables set-based bound propagation through Γ using CROWN (Zhang et al., 2018). `auto_LiRPA` computes vector-valued affine bounding functions, $\underline{\Gamma}$ and $\bar{\Gamma}$, which enables us to compute an RSOA $\hat{\mathcal{X}}_{0:T}$ for the decoded closed-loop lifted dynamics (5) by considering specific components of Γ , $\underline{\Gamma}_t$ and $\bar{\Gamma}_t$, that bound the states reached under (5) at timestep t , starting from any initial condition $x_0 \in \mathcal{X}_0$. This resulting hyper-rectangular set $\hat{\mathcal{X}}_{0:T}$, which we refer to as the Koopman reachable set (KRS), is found by solving the optimization (7), which is computed efficiently by `auto_LiRPA`:

the state-space prediction error of the decoded closed-loop lifted dynamics (5). Specifically, we adapt the single nonconformity score approach (SNSA) (Lindemann et al., 2024) to compute state- and dimension-dependent error bounds using a calibration dataset $\mathcal{D}_C = \{(x_{0:T}^{(i)}, \hat{x}_{0:T}^{(i)})\}_{i=1}^{K^{\text{cal}}}$ of trajectories of the true closed-loop dynamics $x_{0:T}$ (1b) and the decoded closed-loop lifted dynamics $\hat{x}_{0:T}$ (5). To construct each data tuple in \mathcal{D}_C , we sample a reference trajectory $x_{0:T}^{\text{ref},(i)}$ from planner P and an initial state $x_0^{(i)}$ uniformly from $\mathcal{B}_\epsilon(x_0^{\text{ref}})$. We then compute the latent reference trajectory $z_{0:T}^{\text{ref},(i)} = \{\phi(x_t^{\text{ref},(i)})\}_{t=0}^T$ and the feedforward control sequence $u_{0:T-1}^{\text{ref},(i)}$ using (2). We use $u_{0:T-1}^{\text{ref},(i)}$ to calculate the trajectory returned by 1) tracking $z_{0:T}^{\text{ref}}$ in the lifted state space (8) and decoding to the state space via ψ and 2) executing the Koopman controller on the true *nonlinear dynamics* (9):

$$\hat{x}_{1:T}^{(i)} = \left\{ \psi(K_A z_t + K_B(u_t^{\text{ref}} - G_t(z_t - z_t^{\text{ref}}))) \right\}_{t=0}^{T-1} \quad (8) \quad x_{1:T}^{(i)} = \left\{ f(x_t, u_t^{\text{ref}} - G_t(\phi(x_t) - z_t^{\text{ref}})) \right\}_{t=0}^{T-1} \quad (9)$$

The prediction error for each state dimension $j \in \{1, \dots, n\}$ at time t is denoted $e_{t,j}^{(i)} = x_{t,j}^{(i)} - \hat{x}_{t,j}^{(i)}$. This gives a calibration set of error trajectories $\mathcal{D}_E = \{e_t^{(i)}\}_{t=0}^T \}_{i=1}^{K^{\text{cal}}}$, where each $e_t^{(i)} \in \mathbb{R}^n$. Each error trajectory $e^{(i)}$ is normalized into a single non-conformity score $R^{(i)} = \max_{t,j}(\lambda_{t,j}|e_{t,j}^{(i)}|)$, which is the maximum weighted error over all time steps $t \in \{0, \dots, T\}$ and state dimensions $j \in \{1, \dots, n\}$. Here, the normalization constant $\lambda_{t,j} \doteq 1/(e_{t,j}^{\text{max}} + \sigma) \in \mathbb{R}_{>0}$ is computed using a separate normalization dataset \mathcal{D}_N . This dataset, \mathcal{D}_N , consists of M^λ additional error trajectories, $\{e^{(i)}\}_{i=K^{\text{cal}}+1}^{K^{\text{cal}}+M^\lambda}$, collected using the same process as the calibration data. Here, σ is a small positive constant to prevent division by zero. The term $e_{t,j}^{\text{max}}$ is the dimension-wise and time-wise maximum absolute error observed across \mathcal{D}_N , i.e., $e_{t,j}^{\text{max}} = \max_i |e_{t,j}^{(i)}|$ where $i \in \{K^{\text{cal}} + 1, \dots, K^{\text{cal}} + M\}$.

Now, given a user-defined miscoverage rate $\delta \in (0, 1)$, we compute the $(1 - \delta)$ quantile of the non-conformity scores $C = \text{Quantile}_{1-\delta}(\{R^{(i)}\}_{i=1}^{K^{\text{cal}}} \cup \{\infty\})$ by finding the p -th smallest value of the scores $\{R^{(i)}\}_{i=1}^{K^{\text{cal}}}$, where $p \equiv \lceil (K^{\text{cal}} + 1)(1 - \delta) \rceil$. The final, time- and dimension-dependent error bound $\bar{e}_{t,j} \in \mathbb{R}_{\geq 0}$ is obtained by reversing the normalization $\bar{e}_{t,j} = C/\lambda_{t,j}$. These bounds ensure that for an unseen test data point $(x_{0:T}^{(0)}, \hat{x}_{0:T}^{(0)})$, the true prediction error $|e_{t,j}^{(0)}|$ will be contained within the bound $\bar{e}_{t,j}$ for all t and j , with probability at least $1 - \delta$, i.e.,

$$\mathbb{P}(|e_{t,j}^{(0)}| \leq \bar{e}_{t,j}, \forall t \in \{0, \dots, T\}, \forall j \in \{1, \dots, n\}) \geq 1 - \delta$$

We use these bounds to inflate the KRS computed in (7) to compute RSOAs for the original closed-loop dynamics (1b) that are valid with probability at least $1 - \delta$, using (10) (proof in App. 8.1):

Theorem 1 (CKRS Coverage Guarantee) *Let $\hat{\mathcal{X}}_{0:T}$ be the KRS defined in (7) and let the time- and dimension-dependent error bound be $\bar{e}_t = [\bar{e}_{t,1}, \dots, \bar{e}_{t,n}]^\top \in \mathbb{R}^n$. Define the conformalized Koopman reachable set (CKRS) as*

$$\bar{\mathcal{X}}_{0:T}^{1-\delta} \equiv \left\{ \hat{\mathcal{X}}_t \oplus \text{diag}(\bar{e}_t) \mathcal{B}_1(0) \right\}_{t=0}^T, \quad (10)$$

where $\text{diag}(\bar{e}_t) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with \bar{e}_t on its diagonal. Then, for a new reference trajectory $x_{0:T}^{\text{ref},(0)}$ drawn from the same distribution, the CKRS contains the true closed-loop system trajectory generated by (1b) with probability at least $1 - \delta$, i.e., $\mathbb{P}(\bigwedge_{t=0}^T x_t^{(0)} \in \bar{\mathcal{X}}_t^{1-\delta}) \geq 1 - \delta$.

Offline Calibration As discussed so far, the method in Sec. 5.4 is reference-specific, as the calibration trajectories \mathcal{D}_C and error bounds \bar{e}_t hold specifically for a single reference $x_{0:T}^{\text{ref}}$. We propose an offline approach that pre-computes a single, global set of error bounds $\bar{e}_{t,j}$ valid for any reference $x_{0:T}^{\text{ref}}$ sampled from the planner P . This is done by building a larger, more diverse calibration set. We first sample N^{ref} reference trajectories $\{x_{0:T}^{\text{ref},(i)}\}_{i=1}^{N^{\text{ref}}} \sim P$. For each reference, we generate a calibration trajectory by sampling $x_0^{(i)} \sim \mathcal{B}_\epsilon(x_0^{\text{ref},(i)})$, yielding an error trajectory $e_t^{(i)} =$

$x_t^{(i)} - \hat{x}_t^{(i)}$, where $x_t^{(i)}$ and $\hat{x}_t^{(i)}$ are as defined in (8) and (9), respectively. The SNSA procedure (computing $R^{(i)}$, C , and $\bar{e}_{t,j}$ via $\lambda_{t,j}$) is then performed exactly once on this entire aggregated set.

At runtime, given a new reference $x_{0:T}^{\text{ref},(0)}$ sampled from P , two efficient computations are performed: 1) we compute the nominal KRS $\hat{\mathcal{X}}_{0:T}$ specific to $x_{0:T}^{\text{ref},(0)}$ via (7), and 2) we inflate this set using the pre-computed offline bounds \bar{e}_t via the Minkowski sum $\bar{\mathcal{X}}_t^{1-\delta} = \hat{\mathcal{X}}_t \oplus \text{diag}(\bar{e}_t)\mathcal{B}_1(0)$, as defined in (10). This enables fast reachable set computation at runtime but may increase conservativeness of the resulting CKRS, since the bounds \bar{e}_t must now hold uniformly for all references drawn from P , rather than for a single fixed reference. The validity of this offline approach relies on the exchangeability of the non-conformity scores, which is met since each score $R^{(i)}$ is a deterministic function of the data pair $(x_{0:T}^{\text{ref},(i)}, x_0^{(i)})$. Each of these pairs is sampled i.i.d. from the joint distribution defined by $x_{0:T}^{\text{ref}} \sim P$ and $x_0 \sim \mathcal{B}_\epsilon(x_0^{\text{ref}})$. Hence, the resulting scores are i.i.d. and thus exchangeable, ensuring that the standard CP guarantee $\mathbb{P}(\bigwedge_{t=0}^T x_t^{(0)} \in \bar{\mathcal{X}}_t^{1-\delta}) \geq 1 - \delta$ holds.

6. Experiments

We compare our method against reachability baselines and evaluate it on reachability analysis for a unicycle (3D), planar quadcopter (6D), 3D quadcopter (12D), MuJoCo (Todorov et al., 2012) hopper (11D), and MuJoCo swimmer (28D). All results are run on a desktop with an Intel i9-14900K CPU, 64 GB RAM, and an NVIDIA RTX 4090 GPU. For the MuJoCo examples, we set the planner P to be an expert PPO (Schulman et al., 2017) policy; for the other systems, we use a trajectory optimizer on the analytical dynamics to serve as P . We report four key metrics: (i) *reachability runtime* (s), where lower is better; (ii) *average log-volume of reachable sets*, where lower indicates less conservativeness; (iii) *safety rate over rollouts*, where higher is better (safety is defined as zero intersection between the unsafe and reachable set); and (iv) *coverage variability for CP-based verification*, where lower Beta-posterior variance of empirical coverage across calibration sizes indicates stabler coverage. Some metrics apply only to specific baselines (e.g., safety filters report safety rate only). Numerical results are in Table 1; see App. 8.3 for full reachable set plots and experiment details.

Baselines We select baselines to highlight four gaps in data-driven reachability. (i) Learned reachability estimators (Bansal and Tomlin, 2021) lack overapproximation guarantees, motivating formal NN verification. (ii) Existing hybrid CP-NNV (CP-NNV) (Hashemi et al., 2024) methods struggle to scale to high-dimensional, nonlinear dynamics and long horizons, motivating the Koopman lift. (iii) Pure CP-based reachability approaches (Lindemann et al., 2024) suffer coverage degradation as calibration data or sample budgets shrink, motivating sample-efficient reachability analysis that remains reliable under limited data. (iv) Pure NNV-based RSOA computation (Nath et al., 2025) provides deterministic guarantees but can be slow, motivating fast Koopman-based linear reachability. *Unsafe behavior from learned reachability estimators.* Under unicycle dynamics (12), a value function based safety filter (Bansal and Tomlin, 2021) – an approximate reachability method – fails to consistently enforce reach-avoid safety, with only 70% of 250 rollouts satisfying the constraint (Fig. 5). Because the learned value function is an approximate neural PDE solution, small errors in the value and its gradient can cause safety violations, and the method lacks formal guarantees. In contrast, our method provides calibrated probabilistic guarantees that the closed-loop system remains in the CKRS computed via (10) (100% in our experiments) while preserving goal-reaching.

Pure CP based verification. Because our learned Koopman lift closely matches the training dynamics, the resulting KRS are already tight; conformalization then serves only to absorb the small residual model error. We compare to a pure SNSA baseline (Lindemann et al., 2024), which aggregates all heterogeneous trajectory and state errors into a single nonconformity score. As the horizon or state dimension grows, this forces the baseline to either inflate its sets or require much

Table 1: Numerical RSOA results for SCARe-Kro on unicycle, quadcopters and MuJoCo models.

Experiment	System dimension	Time steps	Confidence Level $(1 - \delta)$	Online CP time	KRS time	Total time (online)	Avg. Log volume (online)	Avg. Log volume (offline)	Calibration dataset size	Empirical coverage
Unicycle	3	100	99.0%	0.038s	0.277s	0.315s	-11.543	-8.690	100	100%
2D Quad	6	100	99.0%	0.055s	0.349s	0.404s	-21.491	-5.333	100	100%
3D Quad	12	200	99.0%	0.175s	0.500s	0.675s	-42.954	-35.275	100	100%
Hopper	11	225	99.0%	10.597s	0.387s	10.983s	-4.969	-1.806	100	100%
Swimmer	28	400	95.0%	20.818s	0.593s	21.411s	32.595	42.617	100	99.5%

larger calibration sets to achieve the target δ , causing it to suffer from slow convergence to the target coverage level. At practical data budgets (e.g., 1000 samples), a large coverage variance remains, e.g., as much as 5% under-coverage of the true system trajectories (Fig. 3), which can compromise system safety. In contrast, our method attains 100% empirical coverage across all tested calibration set sizes, avoiding this variance-data trade-off and achieving target coverage with far fewer samples.

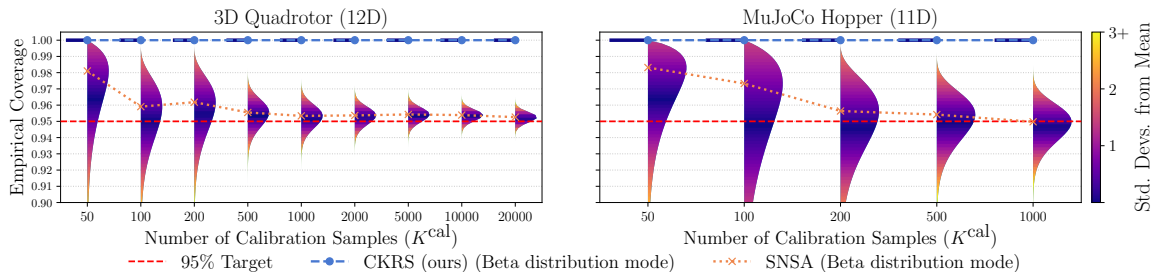


Figure 3: To visualize empirical reachable set coverage, we compute Beta posteriors for the SNSA baseline and ScaRe-Kro on the 3D quadcopter (left) and Hopper (right). The dotted lines indicate the mode of the Beta posterior. For SNSA, coverage rate slowly converges to the target $1 - \delta$ as calibration dataset size increases, whereas ScaRe-Kro maintains 100% coverage even for small K^{cal} .

Hybrid Reachability methods (CP+NNV). For a 3D quadcopter (14), we compare with a hybrid data-driven/NNV reachability method (Hashemi et al., 2024) (denoted CP-NNV), that computes an RSOA by applying NNV on a learned NN trajectory predictor and using CP to bound RSOA underapproximation error induced by learning error. As shown in Fig. 6, CP-NNV yields much larger reachable sets (avg. log volume: -23.35), especially in attitude and velocity. This reflects difficulties in scaling NNV and CP to high-dimensional state spaces and long horizons, which we address with Koopman operators. Our method produces tighter RSOAs (avg. log volume: -40.66) while maintaining coverage, improving scalability with horizon and reducing conservativeness.

Pure NNV Reachability. On a unicycle model, we compare with our prior work (Nath et al., 2025), which used NNV tools for one-shot RSOA computation, requiring 51.95s and achieving an average log-volume of -3.98. In contrast, our method computes the RSOA in 0.32s with a log-volume of -11.54 (Table 1). This highlights that our method is much faster and also reduces conservativeness.

Unicycle We evaluate on CKRS computations on a unicycle model (12), where ϕ and ψ have 3 hidden layers with 128 neurons each, and $z \in \mathbb{R}^{10}$. As shown in Fig. 7, the certified reachable sets verify both safety and goal attainment. Full metrics are in Table 1. We view this benchmark as a sanity check on a simple system, paving the way for more substantive scalability results.

Planar Quadcopter We evaluate a 6-state planar quadcopter (13) navigating around a circular obstacle for $T = 100$. Here, ϕ and ψ have 3 hidden layers with 128 neurons each, and $z \in \mathbb{R}^{24}$. We visualize offline-calibrated CKRSs for multiple reference trajectories in Fig. 8; metrics are in Table 1. An NNV-only baseline (Nath et al., 2025) required 59.85s and achieved an average log-volume of -18.709. Overall, these results support our claim of reduced RSOA conservativeness.

3D Quadcopter We evaluate an analytical 12-state quadcopter (14) (Sabatino, 2015) navigating around a 1 m radius obstacle for $T = 200$. Here, ϕ and ψ have 3 hidden layers with 128 neurons each, and $z \in \mathbb{R}^{24}$. The computed CKRS (Fig. 4) verifies safety and goal reaching; metrics are in Table 1. To show that the offline CP inflation generalizes across controllers, we compute the CKRS for two references (Fig. 4, one orange, one blue) using the same offline-calibrated bounds; we visualize additional CKRSs in Fig. 9. An NNV-only baseline (Nath et al., 2025) is much slower and more conservative, requiring 2153.45 s with an average log-volume of -30.32 . Overall, these results show our method’s efficiency, tightness, and offline CP generalization for high-dimensional systems.

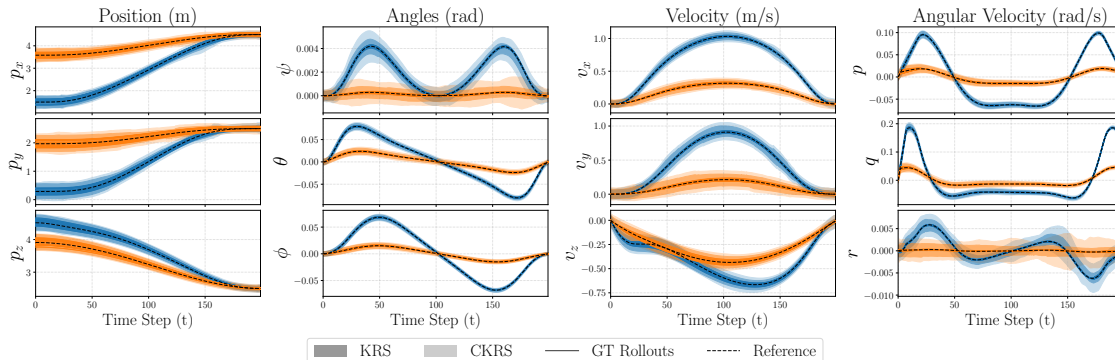


Figure 4: CKRS and KRS (pre-CP) computed for 3D quadcopter, plotted for each dimension.

MuJoCo: Hopper We evaluate on MuJoCo’s Hopper-v5 (Todorov et al., 2012), a nonlinear, underactuated system with black-box dynamics. A PPO agent (Stable-Baselines3, default hyperparameters) trained for 2×10^6 steps is used as the planner P that generates trajectories for the training dataset. Here, ϕ and ψ have 3 hidden layers with 128 neurons each, and $z \in \mathbb{R}^{24}$. Numerical results are in Table 1 and RSOA slices for a few state dimensions are shown in Fig. 1 (see Fig. 11 for all 11 dimensions). This experiment highlights that our approach scales to black-box, high-dimensional contact-rich dynamics while maintaining low runtimes and probabilistic coverage guarantees.

MuJoCo: Customized Swimmer We further assess scalability on a customized MuJoCo Swimmer-v5 (Todorov et al., 2012) with extended links, yielding a 28-dimensional state. The planner P is a PPO agent (Stable-Baselines3 MLP) trained for 1×10^7 steps with modified hyperparameters (LR 1×10^{-3} , $\gamma = 0.9999$, batch size 512; others default). Here, ϕ and ψ have 3 hidden layers with 128 neurons each, and $z \in \mathbb{R}^{38}$. Numerical results are in Table 1 and RSOA slices for a few state dimensions are shown in Fig. 1 (see Fig. 12 for all 28 dimensions). This experiment indicates that our approach maintains certified probabilistic safety while tractably computed RSOAs for high-dimensional systems over long horizons for black-box dynamical systems in MuJoCo.

7. Conclusion

We present ScaRe-Kro, a method for scalable reachability analysis of complex, nonlinear black-box dynamics with formal coverage guarantees. ScaRe-Kro learns a lifted linear representation of the nonlinear dynamics, facilitating tracking control design and efficient closed-loop reachability analysis. We then “conformalize” this reachable set by inflating it such that rollouts of the target nonlinear system are guaranteed to stay within the inflated set with a specified probability (i.e., at least $1 - \delta$). Across several benchmarks, including a unicycle, planar/3D quadrotors, MuJoCo hopper, and a 13-link MuJoCo swimmer, our method computes reachable sets efficiently and over long time horizons. Future work aims to: 1) remove the dependence on dynamically-feasible reference trajectories, 2) replace LQR with a robust controller enforcing constraints despite model error, and 3) use weighted CP (Tibshirani et al., 2019) to handle test-time shifts from the planning distribution.

References

- Matthias Althoff. An introduction to CORA 2015. In ARCH@CPSWeek, volume 34 of EPiC Series in Computing, pages 120–151, 2015.
- Matthias Althoff, Goran Frehse, and Antoine Girard. Set propagation techniques for reachability analysis. Annu. Rev. Control. Robotics Auton. Syst., 2021.
- Arash Amini, Qiyu Sun, and Nader Motee. Error Bounds for Carleman Linearization of General Nonlinear Systems, pages 1–8. 2021.
- Stanley Bak, Hoang-Dung Tran, and Taylor T. Johnson. Numerical verification of affine systems with up to a billion dimensions. In Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC '19, page 23–32, 2019.
- Stanley Bak, Sergiy Bogomolov, Parasara Sridhar Duggirala, Adam R. Gerlach, and Kostiantyn Potomkin. Reachability of black-box nonlinear systems after koopman operator linearization. In IFAC, volume 54, pages 253–258, 2021.
- Stanley Bak, Sergiy Bogomolov, Brandon Hency, Niklas Kochdumper, Ethan Lew, and Kostiantyn Potomkin. Reachability of koopman linearized systems using explicit kernel approximation and polynomial zonotope refinement. Formal Methods Syst. Des., 66(2):307–333, 2025.
- Somil Bansal and Claire J. Tomlin. Deepreach: A deep learning approach to high-dimensional reachability. In ICRA, pages 1817–1824, 2021.
- Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances, 2017.
- Fabia Bayer and Remco I. Leine. Sorting-free hill-based stability analysis of periodic solutions through koopman analysis. Nonlinear Dynamics, 111(9):8439–8466, May 2023.
- Shaoru Chen, Victor M. Preciado, and Mahyar Fazlyab. One-shot reachability analysis of neural network dynamical systems. In ICRA, 2023.
- Vamsi Krishna Chilakamarri, Zeyuan Feng, and Somil Bansal. Reachability analysis for black-box dynamical systems, 2024.
- Glen Chou, Necmiye Ozay, and Dmitry Berenson. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. In 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, December 14-17, 2021, pages 3576–3583, 2021.
- Glen Chou, Necmiye Ozay, and Dmitry Berenson. Safe output feedback motion planning from images via learned perception modules and contraction theory. In WAFR, volume 25 of Springer Proceedings in Advanced Robotics, pages 349–367, 2022.
- Elizabeth Dietrich, Alex Devonport, and Murat Arcaç. Nonconvex scenario optimization for data-driven reachability. In Proceedings of the 6th Annual Learning for Dynamics and Control Conference, volume 242, pages 514–527, 2024.

- Michael Everett, Golnaz Habibi, Chuangchuan Sun, and Jonathan P. How. Reachability analysis of neural feedback loops. *IEEE Access*, 9, 2021.
- Marcelo Forets and Amaury Pouly. Explicit error bounds for carleman linearization. 2017.
- Marcelo Forets and Christian Schilling. Reachability of weakly nonlinear systems using carleman linearization. In *RP*, volume 13035, pages 85–99, 2021.
- Navid Hashemi, Lars Lindemann, and Jyotirmoy V. Deshmukh. Statistical reachability analysis of stochastic cyber-physical systems under distribution shift. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 43(11):4250–4261, 2024.
- Sylvia L. Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F. Fisac, and Claire J. Tomlin. Fastrack: A modular framework for fast and guaranteed safe motion planning. In *CDC*, pages 1517–1522, 2017.
- Saber Jafarpour, Akash Harapanahalli, and Samuel Coogan. Efficient interaction-aware interval analysis of neural network feedback loops. *TAC*, 2024.
- Craig Knuth, Glen Chou, Necmiye Ozay, and Dmitry Berenson. Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants. *IEEE Robotics Autom. Lett.*, 2021.
- Craig Knuth, Glen Chou, Jamie Reese, and Joseph Moore. Statistical safety and robustness guarantees for feedback motion planning of unknown underactuated stochastic systems. In *ICRA*, 2023.
- Niklas Kochdumper and Stanley Bak. Conformant synthesis for koopman operator linearized control systems. In *CDC*, pages 7327–7332, 2022.
- B. O. Koopman and J. V. Neumann. Dynamical systems of continuous spectra. In *Proceedings of the National Academy of Sciences*, 1932.
- Thomas Lew, Lucas Janson, Riccardo Bonalli, and Marco Pavone. A simple and efficient sampling-based algorithm for general reachability analysis. In Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer, editors, *L4DC*, volume 168, pages 1086–1099, 23–24 Jun 2022.
- Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J. Pappas, and Jyotirmoy V. Deshmukh. Formal verification and control with conformal prediction. 2024.
- Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. USA*, 118(35):e2026805118, 2021.
- Igor Mezic. Koopman operator, geometry, and learning. *arXiv preprint arXiv:2010.05377*, 2020.
- Devesh Nath, Haoran Yin, and Glen Chou. Formal safety verification and refinement for generative motion planners via certified local stabilization. *arXiv preprint arXiv:2509.19688*, 2025.

- Shaowu Pan, Eurika Kaiser, Brian M. de Silva, J. Nathan Kutz, and Steven L. Brunton. Pykoopman: A python package for data-driven approximation of the koopman operator. *Journal of Open Source Software*, 9(94):5881, 2024. doi: 10.21105/joss.05881.
- Hyunsang Park, Vishnu Vijay, and Inseok Hwang. Data-driven reachability analysis for nonlinear systems. *IEEE Control Systems Letters*, 8:2661–2666, 2024.
- Nicholas Rober and Jonathan P. How. Constraint-aware refinement for safety verification of neural feedback loops. *IEEE Control. Syst. Lett.*, 2024.
- F. Sabatino. Quadrotor control: Modeling, nonlinear control design, and simulation. M.s. thesis, KTH, Stockholm, Sweden, 2015.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- Lu Shi, Masih Haseli, Giorgos Mamakoukas, Daniel Bruder, Ian Abraham, Todd D. Murphey, Jorge Cortés, and Konstantinos Karydis. Koopman operators in robot learning. 2024.
- Sumeet Singh, Mo Chen, Sylvia L. Herbert, Claire J. Tomlin, and Marco Pavone. Robust tracking with model mismatch for fast and safe planning: An SOS optimization approach. In *WAFR*, volume 14 of *Springer Proceedings in Advanced Robotics*, pages 545–564, 2018.
- Robin Strässer, Karl Worthmann, Igor Mezić, Julian Berberich, Manuel Schaller, and Frank Allgöwer. An overview of koopman-based control: From error bounds to closed-loop guarantees. *Annual Reviews in Control*, 61:101035, 2026.
- Dawei Sun and Sayan Mitra. Neureach: Learning reachability functions from simulations. In Dana Fisman and Grigore Rosu, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 322–337, Cham, 2022. Springer International Publishing.
- Omanshu Thapliyal and Inseok Hwang. Approximate reachability for koopman systems using mixed monotonicity. *IEEE Access*, 10:84754–84760, 2022.
- Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033, 2012.
- Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In *NeurIPS*, 2020.
- Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, pages 4944–4953, 2018.

8. Appendix

We provide proofs for our theoretical results (App. 8.1), an algorithm block detailing our method (App. 8.2), and additional experiments and experimental details (App. 8.3).

8.1. Proofs

Lemma 1 (KRS overapproximation) *Let Γ denote the closed-loop CG of TKFL, and let \mathcal{X}_0 be the initial set. Suppose the KRS $\widehat{\mathcal{X}}_{0:T}$ is defined as in (7). Then, for any initial state $x_0 \in \mathcal{X}_0$, the decoded closed-loop trajectory (5), i.e., $\hat{x}_{0:T} = \Gamma(x_0, z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ satisfies $\hat{x}_t \in \widehat{\mathcal{X}}_t$ for all $t \in \mathcal{T}$.*

Proof The closed-loop graph Γ is a computational graph G as defined in Proposition 1. Using the `auto_LiRPA` library Xu et al. (2020), we compute sound, vector-valued affine bounding functions $\underline{\Gamma}_t$ and $\overline{\Gamma}_t$ (corresponding to \underline{G} and \overline{G} in Proposition 1) for the t -th state output.

By Proposition 1, these bounds are guaranteed to be sound. That is, for any specific input $x_0 \in \mathcal{X}_0$, the predicted state $\hat{x}_t = \Gamma_t(x_0, z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ satisfies

$$\underline{\Gamma}_t(x_0, z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}}) \leq \hat{x}_t \leq \overline{\Gamma}_t(x_0, z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}}).$$

The KRS $\widehat{\mathcal{X}}_t$ is defined in (7) as the interval $\text{Int}(\hat{x}_t^L, \hat{x}_t^U)$, where $\hat{x}_t^L = \min_{x'_0 \in \mathcal{X}_0} \underline{\Gamma}_t(x'_0, \dots)$ and $\hat{x}_t^U = \max_{x'_0 \in \mathcal{X}_0} \overline{\Gamma}_t(x'_0, \dots)$. By the definitions of the minimum and maximum, we have

$$\begin{aligned} \hat{x}_t &\geq \underline{\Gamma}_t(x_0, \dots) \geq \min_{x'_0 \in \mathcal{X}_0} \underline{\Gamma}_t(x'_0, \dots) = \hat{x}_t^L, \\ \hat{x}_t &\leq \overline{\Gamma}_t(x_0, \dots) \leq \max_{x'_0 \in \mathcal{X}_0} \overline{\Gamma}_t(x'_0, \dots) = \hat{x}_t^U. \end{aligned}$$

Hence, $\hat{x}_t \in \text{Int}(\hat{x}_t^L, \hat{x}_t^U) = \widehat{\mathcal{X}}_t$. Since this holds for all $t \in \mathcal{T}$, the trajectory $\hat{x}_{0:T}$ is contained within $\widehat{\mathcal{X}}_{0:T}$. ■

Theorem 1 (CKRS Coverage Guarantee) *Let $\widehat{\mathcal{X}}_{0:T}$ be the KRS defined in (7) and let $\bar{e}_t = [\bar{e}_{t,1}, \dots, \bar{e}_{t,n}]^\top \in \mathbb{R}^n$. Define the conformalized Koopman reachable set (CKRS) as*

$$\overline{\mathcal{X}}_{0:T}^{1-\delta} \equiv \left\{ \widehat{\mathcal{X}}_t \oplus \text{diag}(\bar{e}_t) \mathcal{B}_1(0) \right\}_{t=0}^T, \quad (11)$$

where $\text{diag}(\bar{e}_t) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with \bar{e}_t on its diagonal. Then, for a new reference trajectory $x_{0:T}^{\text{ref},(0)}$ drawn from the same distribution, the CKRS contains true closed-loop system trajectory generated by (1b) with probability at least $1 - \delta$:

$$\mathbb{P} \left(\bigwedge_{t=0}^T x_t^{(0)} \in \overline{\mathcal{X}}_t^{1-\delta} \right) \geq 1 - \delta.$$

Proof The proof combines a deterministic containment guarantee with a probabilistic coverage guarantee.

Step 1: KRS Containment. By Lemma 1, the decoded closed-loop lifted trajectory $\hat{x}_{t,j}^{(0)}$ is deterministically contained within the hyper-rectangular KRS:

$$\hat{x}_{t,j}^{(0)} \in \widehat{\mathcal{X}}_{t,j} = \text{Int}(\hat{x}_{t,j}^L, \hat{x}_{t,j}^U), \quad \forall t \in \mathcal{T}, \quad \forall j \in \{1, \dots, n\}.$$

Step 2: Probabilistic Coverage. The CP guarantee requires exchangeability of the nonconformity scores $\{R^{(i)}\}_{i=0}^{K^{\text{cal}}}$. In our framework, all calibration and test trajectories share the same reference $(z_{0:T}^{\text{ref}}, u_{0:T-1}^{\text{ref}})$ and controller G_t , and their initial states $x_0^{(i)}$ are sampled i.i.d. from $\mathcal{B}_\epsilon(x_0^{\text{ref}})$. Since the decoded closed-loop lifted trajectory (8), true closed-loop trajectory (9), and the nonconformity score are deterministic functions of $x_0^{(i)}$, the resulting scores $R^{(i)}$ are also i.i.d., and hence exchangeable. By the CP guarantee, the event

$$\mathcal{E} \equiv \{|e_{t,j}^{(0)}| \leq \bar{e}_{t,j}, \forall t, j\}$$

occurs with probability at least $1 - \delta$.

Step 3: Combined Guarantee. The true state can be decomposed as $x_{t,j}^{(0)} = \hat{x}_{t,j}^{(0)} + e_{t,j}^{(0)}$. When \mathcal{E} occurs, we have

$$\hat{x}_{t,j}^L - \bar{e}_{t,j} \leq \hat{x}_{t,j}^{(0)} + e_{t,j}^{(0)} \leq \hat{x}_{t,j}^U + \bar{e}_{t,j},$$

which by definition implies $x_{t,j}^{(0)} \in \bar{\mathcal{X}}_{t,j}^{1-\delta}$. Since this holds for all t and j , the trajectory $x_{0:T}^{(0)}$ is contained in the CKRS $\bar{\mathcal{X}}_{0:T}^{1-\delta}$ with probability at least $1 - \delta$. ■

8.2. Algorithm

Algorithm 1 details the procedure for computing the CKRS. The process begins by lifting the planner-provided reference trajectory $x_{0:T}^{\text{ref}}$ into the latent space to compute the feedforward control u^{ref} via least-squares, and synthesizing a time-varying LQR controller to track this reference. To account for the discrepancy between the learned Koopman dynamics and the true system, the algorithm generates calibration data from system rollouts and utilizes SNSA-based conformal prediction to derive statistically-valid error bounds \bar{e}_t . In parallel, the nominal KRS, $\bar{\mathcal{X}}_{0:T}$, is computed by constructing a computational graph of the closed-loop tracking dynamics in the lifted Koopman state space and computing reachable sets for the linear Koopman dynamics via `auto_LIRPA`. Finally, the algorithm returns a conformalized Koopman reachable set (CKRS), $\mathcal{X}_{0:T}^{1-\delta}$, obtained by inflating the nominal KRS with the derived conformal error bounds via a Minkowski sum. Crucially, the CKRS guarantees coverage of the trajectories generated by the *true system* with probability $1 - \delta$.

8.3. Experiment Details and Additional Experiments

The following section contains the the analytical dynamics we used in this paper (App. 8.3.1) and additional experiment results. In particular, we benchmark our approach with a baseline Koopman dynamics learning approach that leverages polynomial liftings (App. 8.3.2), a baseline learned reachable set prediction method (App. 8.3.3), and a baseline conformal prediction-based reachable set computation method that does not leverage Koopman-based propagation (App. 8.3.4). We also present additional experiment figures, including full reachable set visualizations on a per-state breakdown for the unicycle system (App. 8.3.5), planar quadcopter system (App. 8.3.6), MuJoCo hopper system (App. 8.3.8), and MuJoCo swimmer system (App. 8.3.9).

Algorithm 1 Online Conformalized Koopman Reachable Set (CKRS)

Input: Koopman model (ϕ, ψ, K_A, K_B) ; true dynamics f (simulator); reference trajectory $x_{0:T}^{\text{ref}}$; initial set $\mathcal{X}_0 = \mathcal{B}_\epsilon(x_0^{\text{ref}})$; RSOA confidence $1 - \delta$; calibration sizes $M^\lambda, K^{\text{cal}}$; cost matrices Q, R ; verification tool `auto_LiRPA`

Output: Conformalized Koopman reachable set $\bar{\mathcal{X}}_{0:T}^{1-\delta}$ (CKRS)

```

// 1. Initialization
1:  $z_{0:T}^{\text{ref}} \leftarrow \phi(x_{0:T}^{\text{ref}})$  // encode reference state trajectory
2:  $u_{0:T-1}^{\text{ref}} \leftarrow \text{Equation (2)}$  // compute feedforward control sequence
3:  $G_{0:T-1} \leftarrow \text{LQR}(K_A, K_B)$  // compute LQR gains
4:  $\pi(\cdot) \leftarrow \text{Equation (3)}$  // compute lifted controller using LQR and feedforward inputs
// 2. Error bounds  $\bar{e}_t$  via CP
5:  $\mathcal{D}_E, \mathcal{D}_N \leftarrow \text{GenerateCalibrationData}(M^\lambda, K^{\text{cal}})$ 
6:  $\bar{e}_t \leftarrow \text{SNSA}(\mathcal{D}_N, \mathcal{D}_E)$ 
// 3. Nominal KRS  $\hat{\mathcal{X}}_{0:T}$ 
7:  $\Gamma(\cdot) \leftarrow \psi(\text{PropagateLatent}(\phi(\cdot), z^{\text{ref}}, u^{\text{ref}}, G))$  // define CG
8:  $\hat{\mathcal{X}}_{0:T} \leftarrow \text{AutoLiRPA}(\Gamma, \mathcal{X}_0)$  // compute KRS via CG
9:  $\bar{\mathcal{X}}_{0:T}^{1-\delta} \leftarrow \hat{\mathcal{X}}_{0:T} \oplus \text{diag}(\bar{e}_t)\mathcal{B}_1(0)$  // 4. combine to obtain CKRS, and return
10: return  $\bar{\mathcal{X}}_{0:T}^{1-\delta}$ 

```

8.3.1. DYNAMICS.

For unicycle experiments ($x \in \mathbb{R}^3, u \in \mathbb{R}^2$), we used the dynamics model, time-discretized via forward Euler ($dt = 0.1\text{s}$)

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} u_1 \cos(x_2) \\ u_1 \sin(x_2) \\ u_2 \end{bmatrix}. \quad (12)$$

For planar quadcopter experiments ($x \in \mathbb{R}^6, u \in \mathbb{R}^2$), we used the parameters $m = 0.5\text{kg}$, $g = -9.81\text{m/s}^2$, $I_y = 0.01\text{kg} \cdot \text{m}^2$ for the dynamics model, and time-discretized with $dt = 0.05\text{s}$:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ -\frac{u_1}{m} \sin(x_3) \\ g + \frac{u_1}{m} \cos(x_3) \\ \frac{u_2}{I_y} \end{bmatrix} \quad (13)$$

For 3D quadcopter experiments, we used the parameters $m = 1\text{kg}$, $g = -9.81\text{ m/s}^2$, $I_{x,y,z} = [0.5, 0.1, 0.3]\text{ kg}\cdot\text{m}^2$ for the dynamics, with a time-discretization of $dt = 0.025\text{ s}$:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q \cdot \sin(\phi) / \cos \theta + r \cdot \cos \phi / \cos \theta \\ q \cdot \cos \phi - r \cdot \sin \phi \\ p + q \cdot \sin \phi \cdot \tan \theta + r \cdot \cos \phi \cdot \tan \theta \\ \frac{u_1}{m} \cdot (\sin \phi \cdot \sin \psi + \cos \phi \cdot \cos \psi \cdot \sin \theta) \\ \frac{u_1}{m} \cdot (\cos \phi \cdot \sin \phi - \cos \phi \cdot \sin \psi \cdot \sin \theta) \\ g + u_1 \cdot (\cos \phi \cdot \cos \theta) / m \\ ((I_y - I_z) / I_x) \cdot q \cdot r + \frac{u_2}{I_x} \\ ((I_z - I_x) / I_y) \cdot p \cdot r + \frac{u_3}{I_y} \\ ((I_x - I_y) / I_z) \cdot p \cdot q + \frac{u_4}{I_z} \end{bmatrix} \quad (14)$$

8.3.2. COMPARISON BETWEEN NEURAL KOOPMAN LIFTINGS AND POLYNOMIAL LIFTINGS

We compare root mean square error (RMSE) between a polynomial-lifted extended dynamic mode decomposition baseline and our neural lifting approach on the Hopper system. For the polynomial lift, we use PyKoopman (Pan et al., 2024) with degree-3 monomials and identical training data/preprocessing. During open-loop evaluation, the polynomial model becomes numerically unstable, that is, its predictions grow without bound and diverge by the eighth timestep, so we report RMSE only over the first 5 prediction steps in Table 2. In contrast, our neural lifting yields a stable latent dynamics model that tracks accurately across three complete hops, enabling longer-horizon evaluation. These results highlight that for certain systems, e.g., those with strongly nonlinear, contact-rich dynamics (like the MuJoCo hopper), fixed polynomial bases are insufficient, whereas a neural network lifting function captures the necessary structure and remains stable over extended rollouts. Together, these results motivate the development of a Koopman reachable-set computation framework that incorporates neural networks in the loop, which we enable using `auto_LIRPA`.

State	Polynomial Lifting	Ours
z-coord (height)	2728.481591	0.009848
torso angle	117.374908	0.004201
thigh joint	2832.025934	0.013581
leg joint	10791.162550	0.015690
foot joint	8063.167273	0.017901
vel x-coord	321013.541866	0.047526
vel z-coord	1691803.060196	0.059763
vel torso	4512148.680118	0.082879
vel thigh	596074.452102	0.073777
vel leg	6164730.041985	0.164258
vel foot	135554800.971296	0.259034

Table 2: RMSE comparison for the hopper (11D) system on rollouts performed by the learned model and the polynomial fitted model.

8.3.3. BASELINE COMPARISON WITH BANSAL AND TOMLIN (2021).

Figure 5 offers a qualitative comparison of safety violations between the baseline DeepReach method (Bansal and Tomlin, 2021) and ScaRe-Kro. While the main text statistics indicate a lower safety rate for the baseline, this visualization explicitly captures the “reach-avoid” failure mode: the orange trajectories generated by the DeepReach value function clearly drift into the red obstacle region. This highlights the risks of relying on approximate neural PDE solutions for safety-critical constraints. In contrast, the reachable set (shown in blue) computed by our method creates a tight, verified corridor that contains the ground truth rollouts with high, calibrated probability, visually confirming that the controller successfully steers the dynamics clear of the hazard while adhering to the probabilistic bounds.

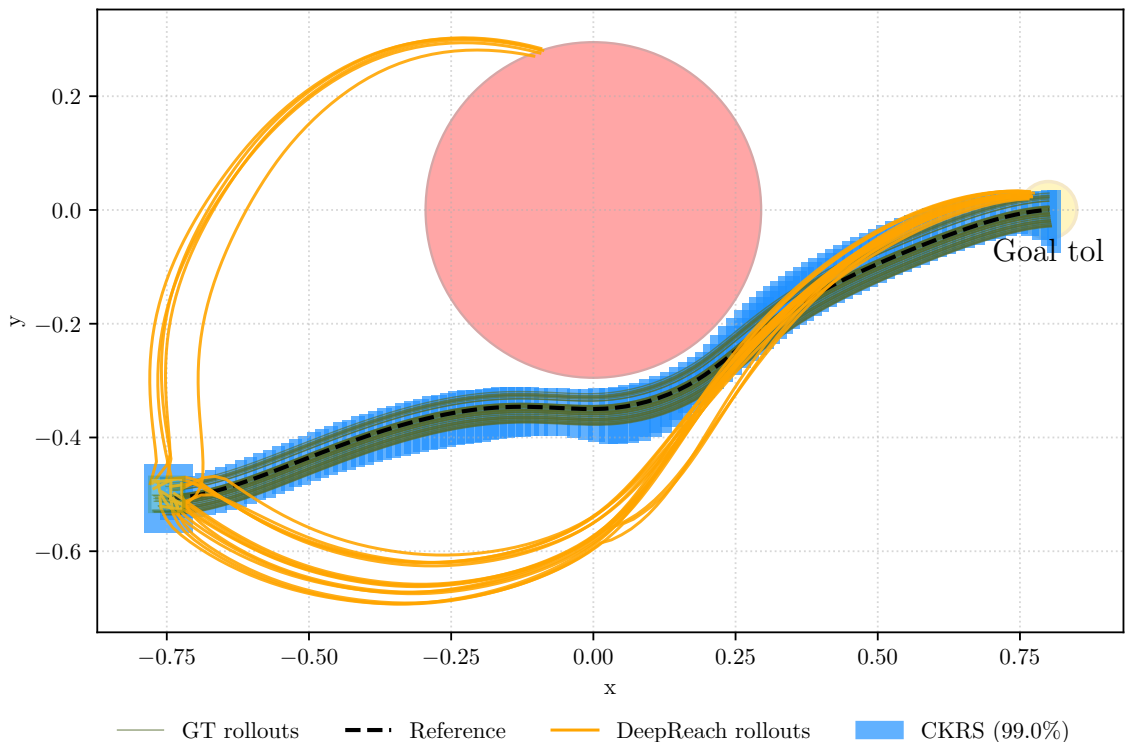


Figure 5: Closed-loop trajectories obtained from DeepReach-based controller (orange) can fail to enforce the reach-avoid constraint (reach yellow goal region while avoiding red unsafe set), while our controller (green rollouts) guarantees safety and containment within the blue CKRS with probability 0.99.

8.3.4. BASELINE COMPARISON WITH HASHEMI ET AL. (2024).

Figure 6 provides a state-by-state breakdown of the conservativeness gap between the reachable sets computed via the baseline CP-NNV (Hashemi et al., 2024) approach (yellow) and our proposed method (blue). While the tabular results in Section 6 quantify the volume difference, Figure 6 illustrate where that difference originates visually. Overall, one of the key differences between our approach and Hashemi et al. (2024) is in using a learned multi-step predictor instead of learned

Koopman dynamics to compute reachable sets. We note that the CP-NNV method suffers from more conservative reachable sets, particularly in the angular velocity (p, q, r) and angle (ϕ, θ, ψ) dimensions. We attribute this to the difficulty of computing reachable sets accurately with a learned multi-step predictor over long horizons. Conversely, ScaRe-Kro maintains consistently tight bounds across all 12 dimensions, demonstrating that the Koopman linearization effectively mitigates the “wrapping effect” that causes over-approximation in standard neural network verification.

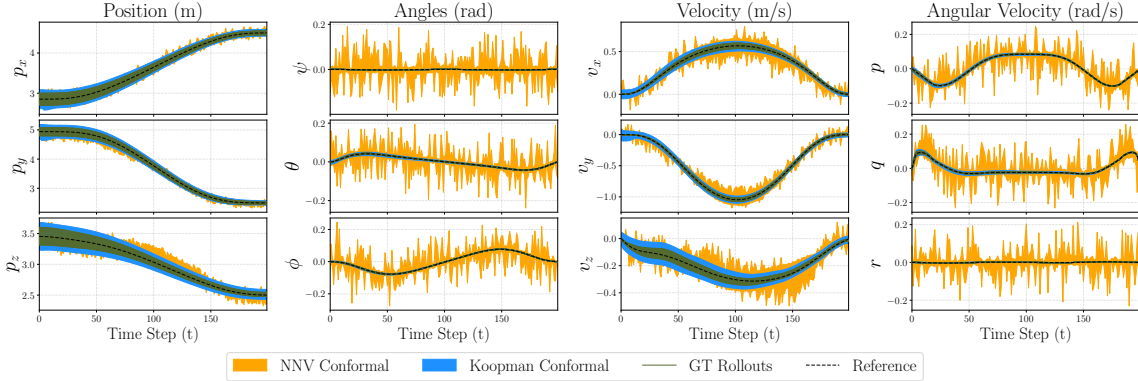


Figure 6: Baseline comparison between CP NNV approach and our method

8.3.5. ADDITIONAL RESULTS ON UNICYCLE SYSTEM.

Figure 7 depicts the spatial evolution of the CKRS for the unicycle. Beyond the numerical coverage rates of Sec. 6, this plot serves as a geometric sanity check, showing the reachable set’s ability to contain the reference trajectory (black dashed line) and closed-loop trajectories (green) while navigating tight constraints. The blue tube is the CKRS that represents the region where the system is guaranteed to remain with with probability at least $1 - \delta$, illustrating that even with the reachable set inflation induced by the conformalization step, the resulting set remains sufficiently compact to verify safety despite the closeness of the closed-loop rollouts to the red obstacle.

8.3.6. ADDITIONAL RESULTS ON PLANAR QUADCOPTER SYSTEM.

We present the full dimensional trajectory evolution for the 6D planar quadcopter in Figure 8. These plots isolate the performance of the Koopman tracking controller on an underactuated system. Notably, the reachable sets (light blue) and their conformalized counterparts (dark blue) accurately capture the behavior in all state dimensions, bounding the closed-loop trajectories. This confirms that the learned linear structure in the lifted Koopman space preserves the critical coupling effects of the original nonlinear dynamics, while the conformal bounds effectively absorb the residual error in the predicted KRS. We also note that the conformalization of the KRS in this example has a relatively small impact on the overall volume of the reachable set estimates, though we note that this is not always the case, e.g., the hopper system shown in Fig. 11. Moreover, this experiment showcases the offline CP inflation bounds (described in Sec. 5.4) generalize across multiple reference trajectories, we compute the CKRS for four reference trajectories total (Fig. 8a-d) and show that the resulting closed-loop trajectories remain within the computed set. This indicates that a single offline CP calibration can be reused to efficiently inflate reachable sets computed online.

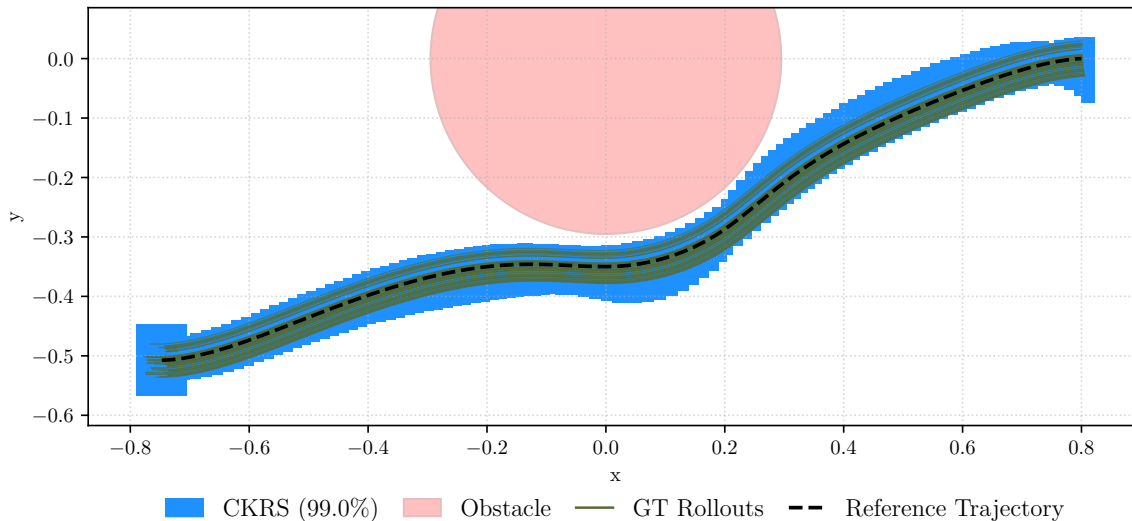


Figure 7: CKRS computed for 3D unicycle model, enabling safety verification despite the small distance of the closed-loop trajectories to the obstacle (red).

8.3.7. ADDITIONAL RESULTS ON 3D QUADROTOR SYSTEM.

To verify that the offline CP inflation bounds (described in Sec. 5.4) generalize across multiple reference trajectories, we compute the CKRS for three alternative reference trajectories and show that the resulting closed-loop trajectories remain within the computed set (Fig. 9). This indicates that a single offline CP calibration can be reused to efficiently inflate reachable sets computed online. We also visualize two of these CKRSs together with the obstacle that the quadrotor is navigating around (Fig. 10), showing that since the RSOAs do not intersect with the obstacle, the closed-loop system is guaranteed to avoid collision when tracking either reference.

8.3.8. ADDITIONAL RESULTS ON MUJoCo HOPPER SYSTEM.

Figure 11 illustrates the full state-space evolution of the 11-dimensional MuJoCo Hopper (Todorov et al., 2012) system during a forward hopping maneuver. This system poses a unique challenge due to its hybrid dynamics, characterized by non-smooth transitions during ground impacts. The periodic stability of the gait is clearly visible in the “height z ” and “torso angle” trajectories, where the ScaRe-Kro framework successfully captures the complex dynamics associated with the flight and stance phases. Notably, while the position states exhibit tight confinement around the reference trajectory (dashed black line), the velocity dimensions (e.g., “Velocity Foot”) display a necessary inflation of the conformal bounds. This expansion correctly accounts for the instantaneous jumps in state caused by contact forces and modeling uncertainties at impact, yet the ground truth rollouts remain strictly contained within the predicted tube, validating the robustness of the Koopman tracking controller under discontinuous contact dynamics.

8.3.9. ADDITIONAL RESULTS ON MUJoCo SWIMMER SYSTEM.

Figure 12 presents the component-wise reachability analysis for the 28-dimensional MuJoCo Swimmer (Todorov et al. (2012)), representing the most high-dimensional validation of our framework. The

plots display the highly coupled, undulatory gait required for locomotion, evidenced by the phase-shifted oscillations across the chain of link joints and velocities. Despite the significant increase in state dimensionality, the CKRS maintain consistent coverage of the closed-loop trajectories across all 28 dimensions. The uniform tightness of the bounds across the swimmer link segments confirms that the learned globally-linear dynamics structure effectively encodes the multi-body dynamics, enabling safe, long-horizon verification for complex, articulated robots while maintaining tractable computation.

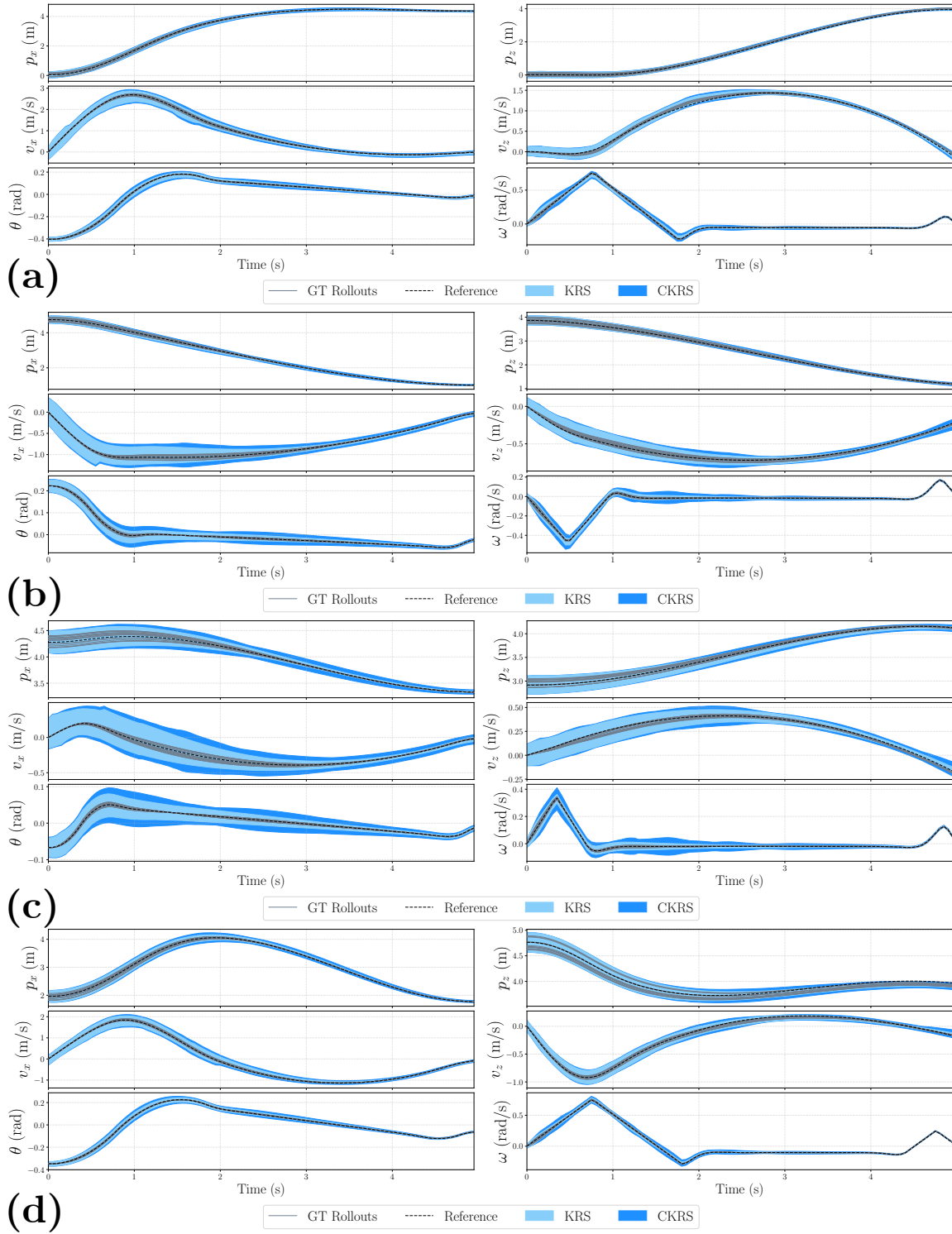


Figure 8: CKRS computed for four reference trajectories on a 2D quadcopter model, with offline-calibrated CP bounds.

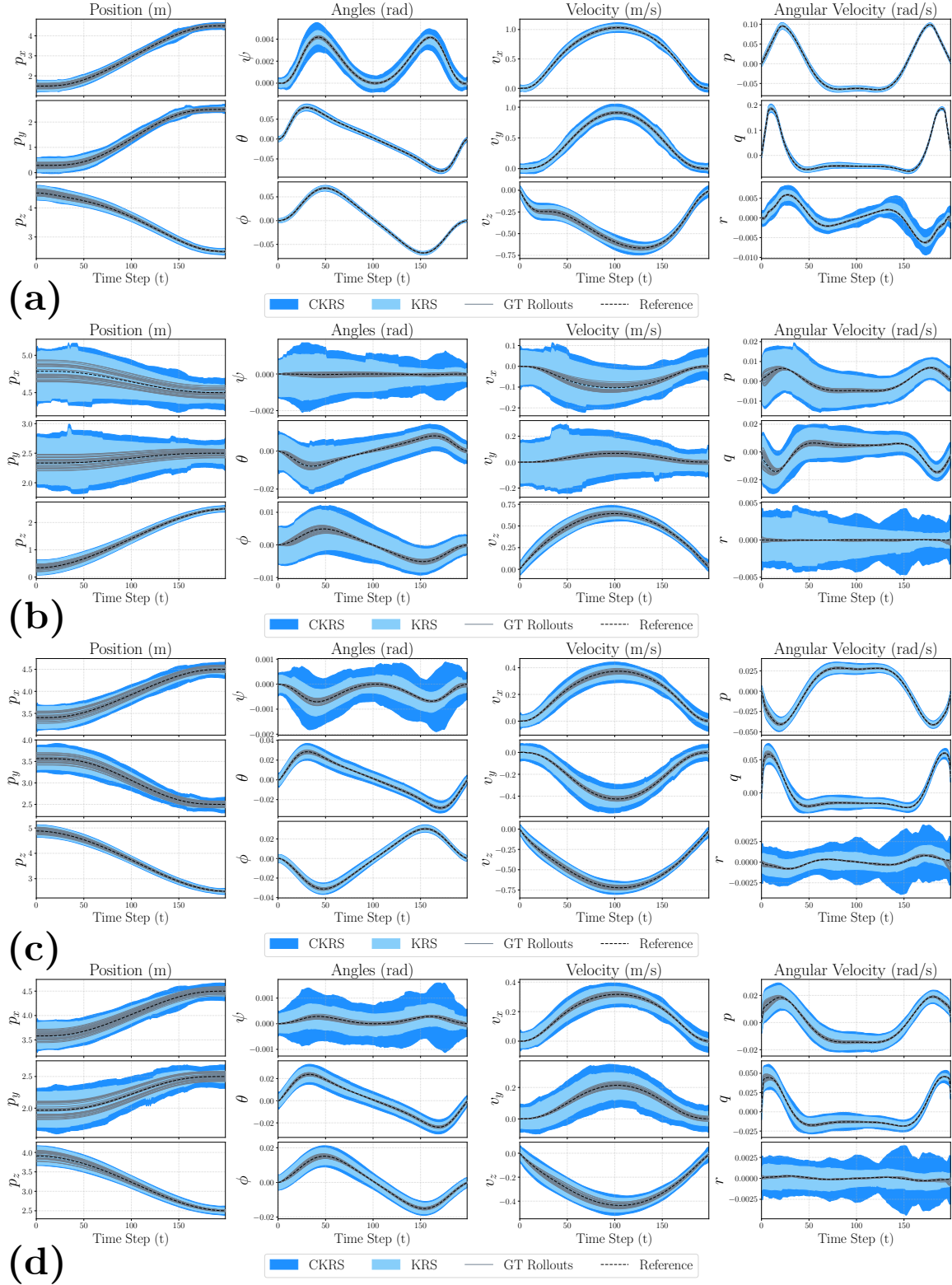


Figure 9: Alternative CKRSs computed for 3D quadcopter model for a different reference trajectory.

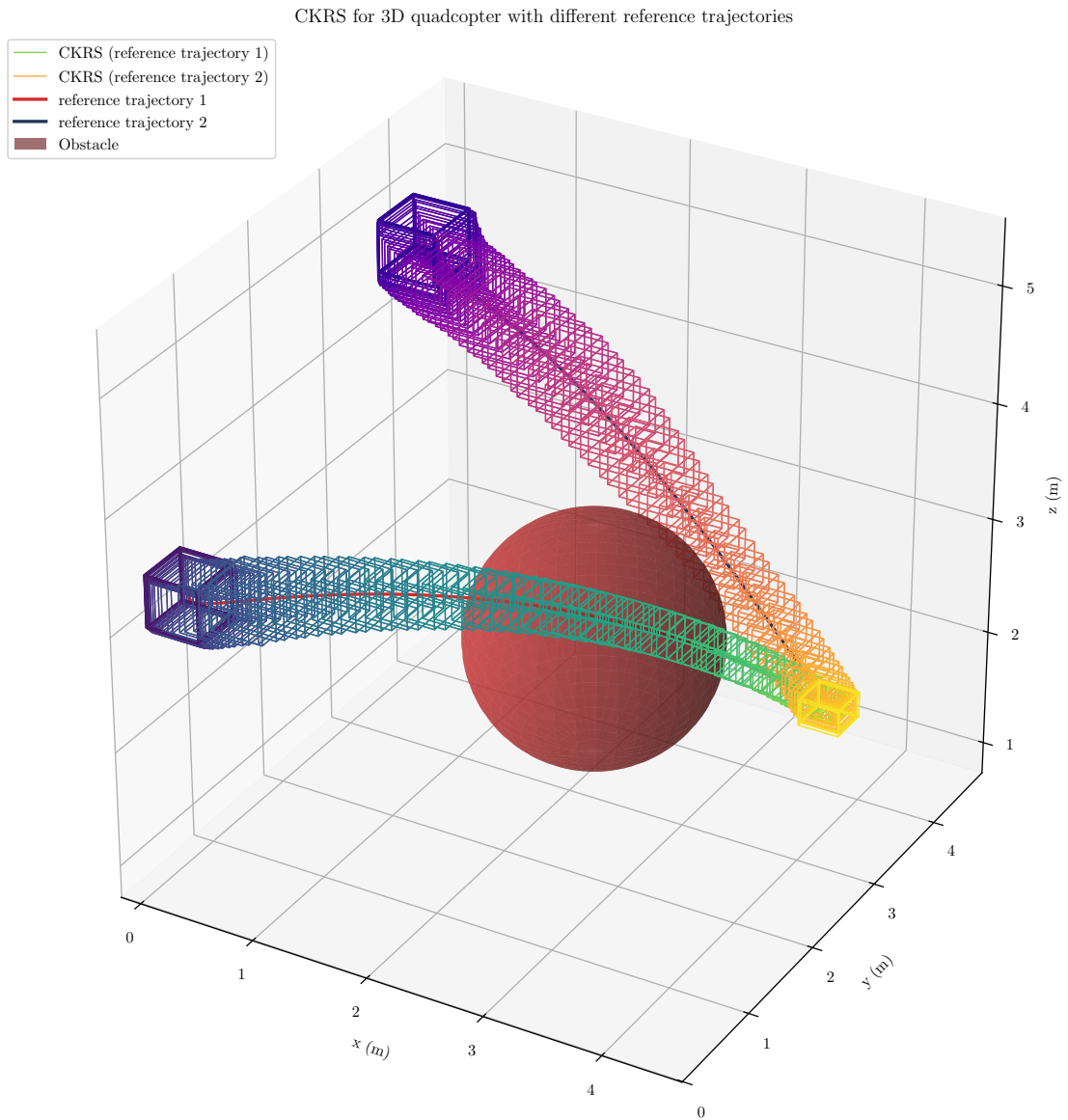


Figure 10: CKRSs computed for a 3D quadcopter model (position components), overlaid with obstacle.

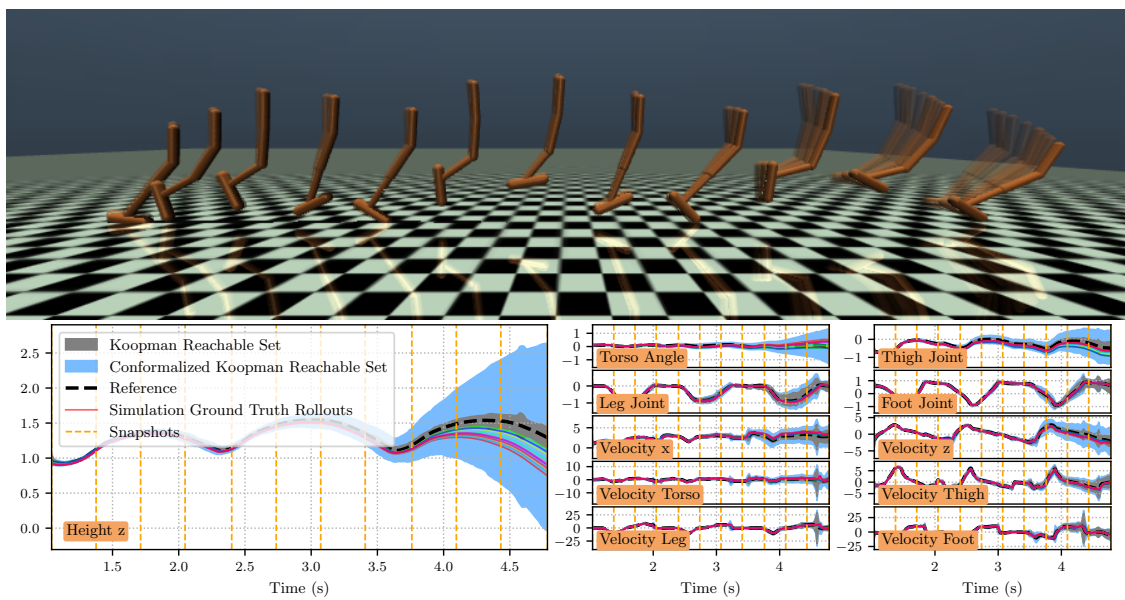


Figure 11: CKRS and KRS computed for 11-D MuJoCo Hopper. The thin multi-colored lines denote distinct closed-loop trajectory rollouts.

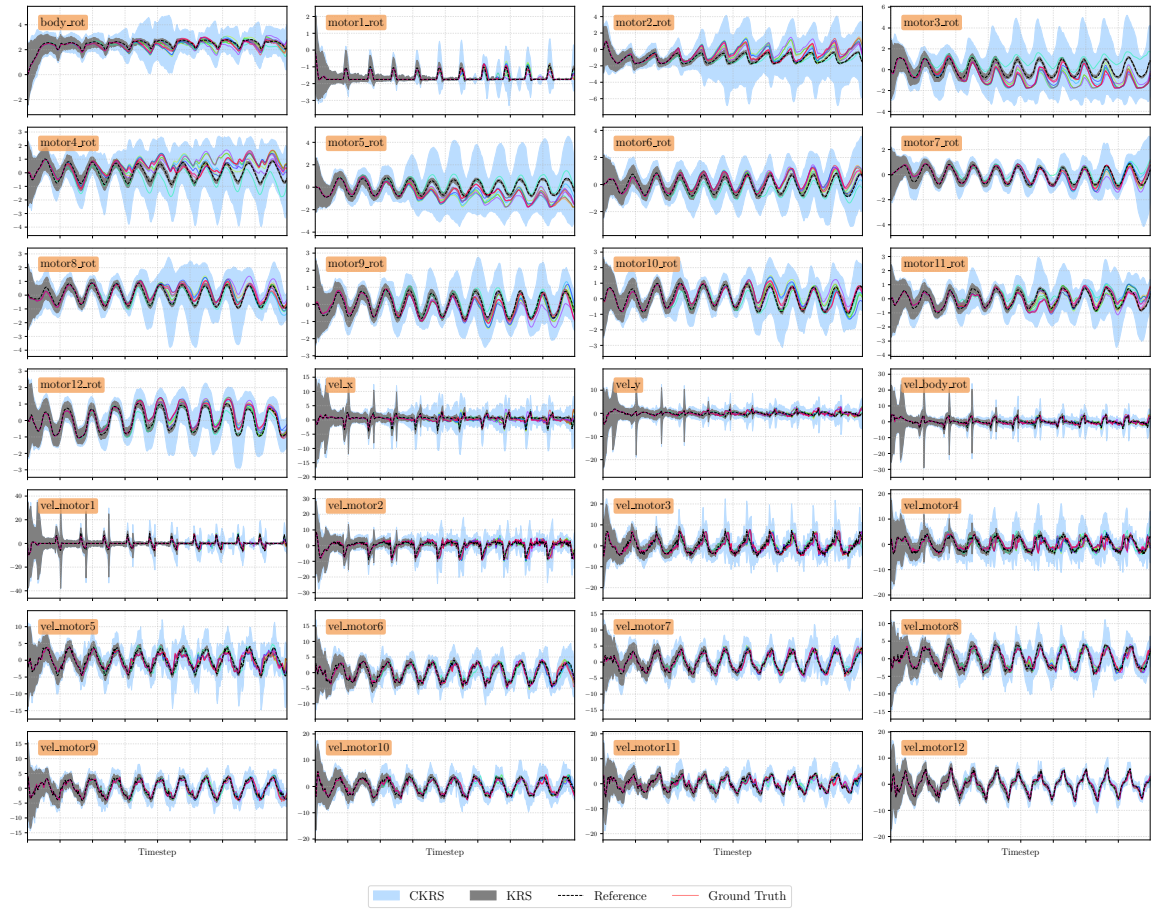


Figure 12: CKRS and KRS computed for 28-D MuJoCo Swimmer. The thin multi-colored lines denote distinct closed-loop trajectory rollouts.

