LEARNING RELIABLE ADAPTIVE REGION-OF-ATTRACTION REPRESENTATIONS FOR NONLINEAR DYNAMICAL SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning regions of attraction (RoAs) of nonlinear dynamical systems remains challenging because existing approaches often rely on a single global representation, creating a tradeoff between expressiveness, constructability, and scalability. Moreover, many practically relevant controllers are designed or learned primarily for control performance rather than jointly with an RoA certificate, motivating post-training RoA analysis without modifying the given controller.

We propose a hybrid framework that learns and represents the RoA through complementary local components. A quadratic Lyapunov representation derived from closed-loop linearization is refined under the nonlinear dynamics to obtain a trusted local core. Starting from this core, the framework learns convergent states, lifts them into continuous regions through locally validated geometric representations, and augments residual regions using flow-informed patches. The resulting hybrid representation is independently evaluated through finite-sample boundedness and reach-to-core validation, providing explicit statistical reliability without requiring the RoA to conform to a single global certificate.

The framework applies to fixed closed-loop nonlinear systems, requires no controller retraining or certificate-aware controller synthesis, and is independent of the controller representation. Experiments on CartPole and six-dimensional quadrotor systems demonstrate substantial expansion beyond the trusted local core while establishing 99.9% reliability at 99.9% confidence.

1 INTRODUCTION

Estimating the region of attraction (RoA) of a nonlinear dynamical system is a fundamental problem in control, verification, and scientific machine learning. The RoA characterizes the initial conditions whose closed-loop trajectories converge to a desired equilibrium and therefore provides an important measure of stability and robustness. Despite decades of research, obtaining an accurate, scalable, and reliable representation of the RoA remains challenging for general nonlinear systems.

The representation challenge. Existing approaches commonly represent the RoA through a global mathematical object, such as a Lyapunov function, Zubov solution, or Hamilton–Jacobi value function Khalil (2002); Zubov (1964); Mitchell et al. (2005). Such representations provide powerful stability or reachability characterizations, but constructing or computing them accurately can become difficult for nonlinear and higher-dimensional systems. Learning-based approaches improve representational flexibility and scalability Dawson et al. (2023); Bansal & Tomlin (2021), but still commonly seek a global certificate or value-function representation. This motivates a different question: rather than forcing the entire RoA into one global representation, can it be learned from complementary local representations adapted to different parts of its geometry?

This challenge is particularly relevant when the controller has already been designed or learned. Performance-driven controller design does not in general provide an RoA characterization, and post-training analysis should ideally assess the resulting closed-loop system without modifying the given controller Dawson et al. (2023).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This exposes three gaps: a post-training RoA method should analyze the given closed-loop system without controller retraining, should not require certificate-aware controller synthesis, and should avoid requiring the entire attraction region to conform to a single global representation. We therefore ask how the RoA of a fixed nonlinear feedback controller can be represented reliably and adaptively under these requirements.

Our approach. We develop an adaptive hybrid RoA representation organized around reach-to-core dynamics. The framework begins with a quadratic Lyapunov function obtained from the stable closed-loop linearization, which provides a structured local representation near the equilibrium. Because linearized Lyapunov decrease alone does not establish the extent of the corresponding region for the true nonlinear dynamics, we determine its practical local extent directly from the nonlinear closed-loop system. Candidate sublevel sets are conservatively shrunk using sampled nonlinear decrease tests and subsequently recovered through sampled bounded reach-to-core trajectories. The resulting trusted local core provides the stability anchor for subsequent RoA learning and representation.

Starting from the trusted local core, the framework first learns convergent trajectory seeds and then lifts these sparse states into continuous local regions. We study two complementary seed learning strategies: random exploration and dynamics-informed pullback exploration. The learned seeds are represented using locally validated geometric patches, for which we consider Euclidean balls and covariance-shaped ellipsoids. Residual convergent states not captured by the geometry layer are further represented using flow-informed local patches. The resulting hybrid representation is finally evaluated by independent finite-sample boundedness and reach-to-core validation.

The proposed method operates on a fixed closed-loop dynamical system and is independent of how its feedback controller is represented or obtained. In particular, it requires neither controller retraining nor joint controller–certificate synthesis. Neural feedback controllers nevertheless provide an important and challenging application: their expressive nonlinear policies can induce complex closed-loop attraction geometry that is difficult to capture with conventional RoA representations. We therefore evaluate the framework on fixed neural controllers for CartPole and quadrotor attitude stabilization; controller synthesis and training are outside the scope of this work.

Contributions. Our main contributions are:

- **Adaptive hybrid RoA representation.** We formulate post-training RoA estimation for a fixed closed-loop system as learning complementary local representations anchored to a trusted Lyapunov core, rather than constructing a single global Lyapunov, Zubov, or reachability certificate.
- **Seed learning and local representation.** We separate RoA expansion into learning convergent seeds and representing their neighborhoods, and study random versus pullback-guided seed learning together with ball versus ellipsoidal local representations. A complementary flow-informed stage represents residual convergent regions not captured by the geometry layer.
- **Multi-level statistical reliability.** We combine trajectory-based validation and recovery of the local Lyapunov representation with independent finite-sample validation of learned outer regions, providing explicit reliability bounds for the resulting adaptive representation.

We evaluate the framework on CartPole and six-dimensional quadrotor attitude stabilization with fixed neural feedback controllers. The results demonstrate substantial expansion beyond the local quadratic representation while maintaining explicit statistical reliability. Across both benchmarks, ablations further show that random seed learning with Euclidean-ball representations produces the largest represented regions among the evaluated configurations.

2 Background

2.1 Nonlinear Dynamics and Closed-Loop Systems

Consider a continuous-time nonlinear control system

$$\dot{x} = f(x, u), \qquad x \in \mathbb{R}^n, \tag{1}$$

where u denotes the control input. We consider a fixed feedback controller $u_\phi(x)$, which induces the autonomous closed-loop dynamics

$$\dot{x} = f(x, u_\phi(x)) \triangleq f_{\mathrm{cl}}(x). \tag{2}$$

The proposed RoA framework operates on the resulting closed-loop dynamics and does not depend on the particular representation or training procedure of the controller. In this work, we consider fixed neural-network feedback controllers as an important and challenging application; controller synthesis and training are outside the scope of the proposed method.

Let

$$\Phi_t(x)$$

denote the flow map generated by f_{cl}, satisfying the semigroup property

$$\Phi_{t+s}(x) = \Phi_t(\Phi_s(x)).$$

Throughout the theoretical development, we assume without loss of generality that the equilibrium of interest is located at the origin and is asymptotically stable. For a nonzero equilibrium x_{eq}, the same formulation applies after the coordinate translation $\tilde{x} = x - x_{\mathrm{eq}}$; the experiments retain the corresponding equilibrium explicitly.

For the neural controllers considered in our experiments, let $\mathcal{X}_{\mathrm{DOI}} \subset \mathbb{R}^n$ denote the domain of interest (DoI) used during controller training. The DoI specifies the state domain over which the fixed neural policy was trained; it is not assumed to define the RoA and does not serve as an intrinsic boundary of the proposed RoA learning procedure. When needed for finite-horizon rollout validation, we separately specify a bounded computational domain $\mathcal{X}_{\mathrm{bd}}$.

2.2 Region of Attraction and Local Lyapunov Functions

The *region of attraction* (RoA) of the equilibrium is

$$\mathcal{R} = \left\{ x_0 : \lim_{t \to \infty} \Phi_t(x_0) = 0 \right\}. \tag{3}$$

A continuously differentiable function

$$V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$$

is a Lyapunov function on a domain Ω if

$$V(x) > 0, \qquad x \neq 0, \qquad V(0) = 0, \tag{4}$$

$$\nabla V(x)^\top f_{\mathrm{cl}}(x) < 0, \qquad x \in \Omega \setminus \{0\}. \tag{5}$$

A sublevel set

$$\Omega_c = \{x : V(x) \leq c\}$$

contained in a domain on which these conditions hold is forward invariant and provides a local inner approximation of the RoA.

In the proposed framework, this classical Lyapunov result motivates the local quadratic representation used to initialize the learning procedure. The practical extent of the trusted local core is subsequently determined under the nonlinear closed-loop dynamics using sampled Lyapunov-decrease testing and bounded reach-to-core recovery, as described in Sec. 3.

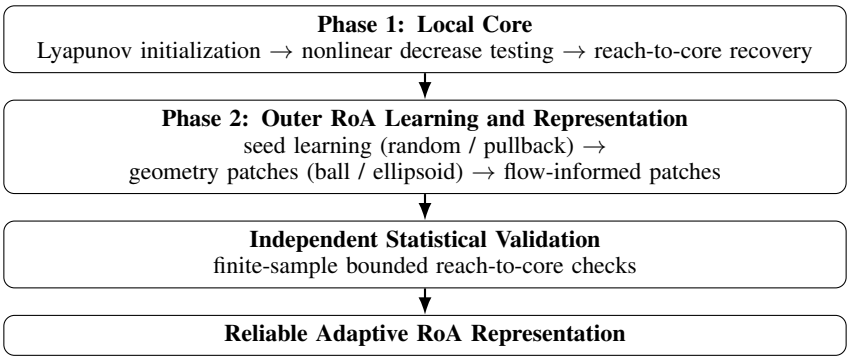


Figure 1: Overview of the proposed hybrid RoA learning framework.

2.3 Reach-to-Core Characterization

Let $\Omega_0 \subset \mathcal{R}$ denote a local region known to converge to the equilibrium. If, for some finite $T > 0$,

$$\Phi_T(x) \in \Omega_0, \tag{6}$$

then the semigroup property gives

$$\Phi_{T+t}(x) = \Phi_t(\Phi_T(x)).$$

Since every state in Ω_0 converges to the equilibrium, it follows that $x \in \mathcal{R}$.

Thus, finite-time reachability to a trusted local core provides a sufficient condition for RoA membership. This reach-to-core property forms the basis for the learning and validation procedures developed in the following sections.

3 The Framework

The framework combines a trusted local Lyapunov core with learned outer regions represented by complementary geometry- and flow-informed local patches, followed by independent statistical validation, as summarized in Fig. 1.

3.1 Phase 1: Local Lyapunov Core

Phase 1 constructs a trusted local core that anchors subsequent outer-region learning and representation. Let

$$A_{\mathrm{cl}} = \left.\frac{\partial f_{\mathrm{cl}}(x)}{\partial x}\right|_{x=0}$$

denote the closed-loop Jacobian at the equilibrium. Given a stable closed-loop linearization and a feasible target decay rate $\alpha > 0$, we compute $P \succ 0$ satisfying

$$A_{\mathrm{cl}}^{\top} P + P A_{\mathrm{cl}} + 2\alpha P \prec 0, \tag{7}$$

which defines the local quadratic representation

$$V_0(x) = x^{\top} P x.$$

The linearized condition does not determine how far this representation satisfies a decrease condition under the true nonlinear closed-loop dynamics. We therefore evaluate candidate sublevel sets $\Omega_c = \{x : V_0(x) \le c\}$ directly by sampling their boundaries and testing

$$\dot{V}_0(x) = \nabla V_0(x)^{\top} f_{\mathrm{cl}}(x) \le -\varepsilon V_0(x). \tag{8}$$

The level is conservatively reduced until the sampled decrease criterion is satisfied, yielding the locally validated seed Ω_{seed}.

Trajectory-based core recovery. The local seed may be conservative because it requires direct Lyapunov decrease at the sampled boundary. We therefore enlarge it using bounded reach-to-core behavior, which also provides the convergence criterion used in the subsequent outer-region learning and validation. For a candidate level $c \geq c_{\text{seed}}$, boundary samples are propagated under the nonlinear closed-loop dynamics and counted as successful when they remain inside a bounded computational domain $\mathcal{X}_{\text{bd}}$ and reach Ω_{seed} within the prescribed horizon:

$$\exists\, t \leq H : \Phi_t(x_0) \in \Omega_{\text{seed}}, \qquad \Phi_\tau(x_0) \in \mathcal{X}_{\text{bd}} \;\; \forall \tau \in [0, t]. \tag{9}$$

Candidate levels are expanded geometrically and refined by bisection according to a prescribed empirical success threshold. The largest passing search level c_{search} is conservatively backed off to obtain c_0, which defines the trusted local core

$$\widehat{\Omega}_0 = \{x : V_0(x) \leq c_0\}.$$

Here Ω_{seed} serves as the target during trajectory-based core recovery. Once recovery is complete, the resulting trusted core $\widehat{\Omega}_0$ becomes the target for all subsequent reach-to-core tests in Phase 2. This construction aligns the two phases of the framework: both local-core recovery and outer-region learning exploit bounded reach-to-core behavior of the nonlinear closed-loop dynamics.

3.2 Phase 2: Adaptive RoA Representation Learning

Starting from the trusted local core $\widehat{\Omega}_0$, Phase 2 learns and represents surrounding regions of attraction directly from closed-loop trajectories. It consists of three steps: (i) learning sparse convergent seeds, (ii) lifting these seeds into continuous regions through locally validated geometric patches, and (iii) representing residual convergent regions using flow-informed patches. The resulting hybrid representation is subsequently evaluated through independent finite-sample validation.

We consider two seed-learning strategies: uniform random exploration and dynamics-guided pullback exploration. Both use the same bounded reach-to-core criterion and produce the same form of convergent seed set, allowing the effect of seed learning to be separated from the subsequent representation stage. The random strategy is described below; complete algorithms for both variants are provided in the appendix.

3.2.1 Convergent Seed Learning

For a finite horizon T, define the bounded reach-to-core set

$$\mathcal{R}_T = \left\{ x_0 : \begin{array}{l} \exists\, t \in [0, T] \text{ such that } \Phi_t(x_0) \in \widehat{\Omega}_0, \\ \Phi_\tau(x_0) \in \mathcal{X}_{\text{bd}}, \quad \forall \tau \in [0, t] \end{array} \right\}. \tag{10}$$

Since $\widehat{\Omega}_0$ is a trusted local attraction region, states satisfying this condition provide candidate convergent seeds for the outer representation.

In the random variant, candidate states are sampled independently and uniformly from the exploration domain $\mathcal{X}_{\text{samp}}$,

$$x_i \sim \text{Unif}(\mathcal{X}_{\text{samp}}), \tag{11}$$

and propagated under the fixed closed-loop dynamics. States satisfying Eq. (10) are labeled successful, while the remaining samples are labeled unsuccessful. To focus the seed bank near the region being expanded, successful and unsuccessful samples are retained within prescribed shells in the local quadratic metric

$$r_P(x) = \sqrt{x^\top P x}. \tag{12}$$

The retained successful states form the convergent seed set $\mathcal{S}_{\text{conv}}$ used by the subsequent local-region construction.

As a dynamics-guided alternative, the pullback variant samples around the frontier of previously successful trajectories rather than uniformly over the exploration domain. Its exploration scale and frontier are updated from observed reach-to-core behavior, with additional local sampling near difficult frontier regions. Importantly, the two strategies differ only in how candidate seeds are discovered; the same reach-to-core criterion determines successful seeds, and the same downstream representation and validation procedures are applied to both.

3.2.2 CONTINUOUS REGION CONSTRUCTION

The convergent seeds provide trajectory evidence but do not themselves define a continuous RoA representation. We therefore lift them into local regions using geometry-based construction, followed by flow-informed representation of residual structures.

Geometry-based construction. The first layer represents convergent neighborhoods using local geometric patches. We consider two alternatives: isotropic Euclidean balls and covariance-adapted ellipsoids. Candidate patches are constructed around convergent seeds, with uncovered seeds iteratively selected as additional centers until the desired coverage is reached. Each candidate is evaluated using fresh interior and boundary-shell rollouts and retained only when its one-sided statistical failure bound satisfies the prescribed 1% construction threshold.

Flow-informed residual construction. Convergent seeds not covered by the geometry layer form a residual set that may contain thin or flow-informed structures. These residual seeds are organized using trajectory and time-to-core information, and flow-informed local patches are constructed to represent the remaining regions. Candidate flow patches are evaluated by fresh rollout samples using the same 1% construction threshold. The stricter final reliability requirement is imposed separately during independent validation of the completed representation.

The resulting adaptive representation is

$$\widehat{\mathcal{R}} = \widehat{\Omega}_0 \cup \bigcup_{i \in \mathcal{I}_{\mathrm{geo}}} E_i^{\mathrm{geo}} \cup \bigcup_{j \in \mathcal{I}_{\mathrm{flow}}} E_j^{\mathrm{flow}}. \tag{13}$$

Detailed construction algorithms for both geometric representations and the residual flow patches are provided in the appendix.

3.2.3 STATISTICAL PATCH SCREENING

For each candidate patch E, fresh rollout samples are drawn according to a prescribed distribution μ_E. A rollout is successful when it remains in the bounded validation domain until reaching the trusted core within the prescribed horizon:

$$S(x) = \left[\exists\, t \le H : \Phi_t(x) \in \widehat{\Omega}_0, \quad \Phi_\tau(x) \in \mathcal{X}_{\mathrm{bd}}\ \forall \tau \in [0,t]\right]. \tag{14}$$

Define the patch failure probability

$$\epsilon(E) = \Pr_{x \sim \mu_E}\left[\neg S(x)\right]. \tag{15}$$

From N independent rollout samples, we compute a one-sided Clopper–Pearson upper confidence bound $\epsilon^+(E,\delta)$. Given a prescribed construction threshold ϵ_{con}, a candidate patch is retained only if

$$\epsilon^+(E,\delta) \le \epsilon_{\mathrm{con}}. \tag{16}$$

In our experiments, both geometry and flow-informed candidate patches use the common construction requirement $\epsilon_{\mathrm{con}} = 0.01$ at 99.9% one-sided confidence, corresponding to a 99% required success level. This ensures consistent screening across all learning–representation variants. After construction, the completed representation is independently validated using the stricter $\epsilon_{\mathrm{final}} = 0.001$ (99.9% reliability) requirement.

Lemma 1 (Finite-Sample Patch Reliability)**.** *If a candidate patch E satisfies $\epsilon^+(E,\delta) \le \epsilon_{\mathrm{con}}$, then, with confidence at least $1-\delta$,*

$$\Pr_{x \sim \mu_E}\left[S(x)\right] \ge 1 - \epsilon_{\mathrm{con}}. \tag{17}$$

This provides a finite-sample reliability guarantee for unseen states drawn from the patch distribution μ_E.

4 EXPERIMENTS

We evaluate the framework on two nonlinear stabilization benchmarks: four-dimensional CartPole and six-dimensional quadrotor attitude stabilization, both with fixed neural feedback controllers.

Controller training is outside the proposed method. For each benchmark, we compare random and pullback seed learning and report a concise geometry comparison under random seeds; complete results for all four learning–geometry combinations are provided in the appendix.

4.1 CartPole

The CartPole state is $x = (p, \dot{p}, \theta, \dot{\theta})$ with bounded horizontal force $u \in [-10, 10]$. The controller is fixed throughout all RoA experiments. Phase 1 constructs the trusted local quadratic core through nonlinear decrease testing followed by trajectory-based reach-to-core recovery; this core is used as the common target for all subsequent learning and validation.

Local core and convergent seed learning. Phase 1 initializes the quadratic Lyapunov representation at $c_{\text{init}} = 2.8234 \times 10^{-3}$. Sampled nonlinear decrease testing gives $c_{\text{seed}} = 1.7339 \times 10^{-3}$, followed by trajectory-based recovery to the trusted core $c_0 = 1.9569 \times 10^{-2}$, which is fixed for all subsequent experiments.

Pullback exploration discovers $32{,}225$ convergent seeds in 160.4 s, compared with $17{,}363$ seeds in 577.5 s for random exploration. Thus, dynamics-guided pullback is substantially more efficient for seed discovery. As shown below, however, this advantage does not directly translate into a larger continuous RoA representation.

Geometry representation. To isolate the effect of local representation, we next hold seed learning fixed to random exploration and compare Euclidean balls with covariance-adapted ellipsoids. Table 1 shows that ellipsoids cover a larger fraction of the discrete seed bank, whereas balls produce a substantially larger continuous represented volume. Thus, discrete seed coverage alone is not a sufficient measure of the resulting RoA representation.

Table 1: CartPole geometry representation using random seeds.

Geometry	Patches	Seed cov.	DoI occ.
Ellipsoid	1,636	98.04%	13.48%
Ball	1,355	75.31%	**16.45%**

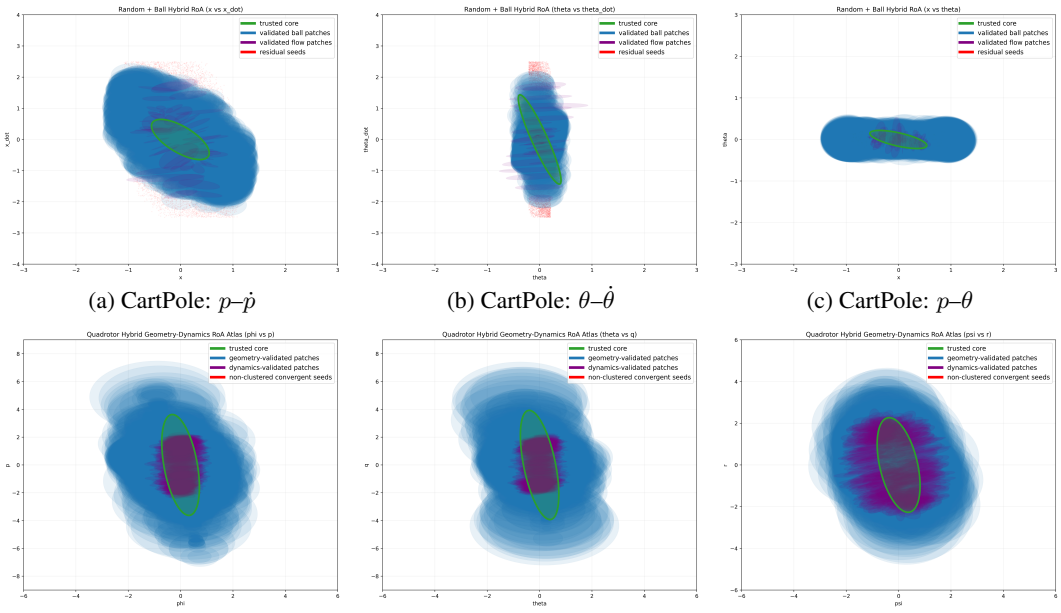
Hybrid representation and reliability. The complete comparison reported in the appendix identifies random seed learning with ball geometry as the strongest CartPole configuration. Its geometry layer occupies 16.448% of the DoI, compared with 2.103% for the trusted local core, corresponding to a $7.82\times$ volume ratio. The residual stage accepts 67 flow-informed patches and increases the final hybrid occupancy slightly to 16.449%, indicating that the ball geometry already captures nearly all of the represented volume.

Final reliability is evaluated only after the representation has been fixed, using samples independent of seed learning and patch construction. All $8,192$ samples drawn from the newly added hybrid region remain bounded and reach the trusted core. The resulting one-sided Clopper–Pearson failure upper bound is 8.43×10^{-4} at 99.9% confidence, satisfying the required 99.9% reliability criterion. Representative two-dimensional slices of the learned random–ball representation are shown in Fig. 2. The full-dimensional patches, rather than their displayed two-dimensional projections, are used for the reported volume and reliability calculations.

4.2 Quadrotor Attitude Stabilization

The second benchmark considers six-dimensional quadrotor attitude stabilization with three bounded torque inputs. Its coupled nonlinear rotational dynamics provide a higher-dimensional test of adaptive representation learning. The fixed neural controller and physical model are described in the appendix. Phase 1 constructs the trusted local quadratic core through nonlinear decrease testing and trajectory-based reach-to-core recovery; this core is used as the common target for all subsequent learning and validation.

Local core and convergent seed learning. Phase 1 initializes the quadratic Lyapunov representation at $c_{\text{init}} = 3.4111 \times 10^{-4}$. Sampled nonlinear decrease testing gives $c_{\text{seed}} = 3.4111 \times 10^{-4}$,



(a) CartPole: p–$\dot{p}$ (b) CartPole: θ–$\dot{\theta}$ (c) CartPole: p–θ

(d) Quadrotor: ϕ–p (e) Quadrotor: θ–q (f) Quadrotor: ψ–r

Figure 2: Learned adaptive RoA representations using the selected random-seed and Euclidean-ball configuration. The plots show representative two-dimensional slices of the full-dimensional hybrid representations for CartPole and quadrotor.

followed by trajectory-based recovery to the trusted core $c_0 = 1.3723 \times 10^{-1}$, which is fixed for all subsequent experiments.

Pullback exploration discovers $161{,}768$ convergent seeds in 461.1 s, compared with $125{,}194$ seeds in $2{,}298.0$ s for random exploration. Thus, dynamics-guided pullback is substantially more efficient for seed discovery. As shown below, however, this advantage does not directly translate into a larger continuous RoA representation.

Geometry representation. Holding seed learning fixed to random exploration, Table 2 compares Euclidean balls with covariance-adapted ellipsoids. Balls achieve both higher seed coverage and substantially larger continuous DoI occupancy. Thus, for this benchmark, the simpler isotropic geometry provides the stronger continuous representation.

Table 2: Quadrotor geometry representation using random seeds.

Geometry	Patches	Seed cov.	DoI occ.
Ellipsoid	1,216	75.39%	62.75%
Ball	1,213	**88.65%**	**78.48%**

Hybrid representation and reliability. The complete comparison reported in the appendix identifies random seed learning with ball geometry as the strongest quadrotor configuration. The geometry layer occupies 78.48% of the DoI, compared with 13.11% for the trusted local core, corresponding to a $5.99\times$ volume ratio. Flow-informed residual construction accepts 212 patches and increases the final hybrid occupancy to 78.68%, yielding a $6.00\times$ volume ratio relative to the core. Thus, geometry captures most of the attraction structure, while the flow layer provides a small complementary refinement.

Final reliability is evaluated only after the representation has been fixed, using samples independent of seed learning and patch construction. All $8{,}192$ samples drawn from the newly added hybrid region remain bounded and reach the trusted core. The resulting one-sided Clopper–Pearson failure upper bound is 8.43×10^{-4} at 99.9% confidence, satisfying the required 99.9% reliability criterion. Representative two-dimensional slices of the learned random–ball representation are shown in Fig. 2.

5 Related Work

Lyapunov-based RoA estimation. Classical RoA methods construct Lyapunov functions whose sublevel sets provide inner approximations of the attraction region Colbert & Peet (2018); Henrion & Korda (2014); Tan & Packard (2008). Polynomial, SOS, and LMI-based techniques provide rigorous analysis for suitable system classes, but their estimates may be conservative when constrained to a single Lyapunov representation.

Learning-based methods improve representational flexibility using neural Lyapunov, Zubov, and physics-informed representations Richards et al. (2018); Chang et al. (2019); Dai et al. (2021); Meng et al. (2024); Liu et al. (2025); Dawson et al. (2023). Related methods incorporate Lyapunov structure into controller learning or jointly synthesize controllers and stability certificates Berkenkamp et al. (2017); Kushwaha & Biron (2026); Li et al. (2025). For example, Li et al. (2025) jointly learn a neural controller and Lyapunov function through Zubov sampling and iterative domain expansion. Such certificate-aware synthesis is complementary to our setting, where the controller is fixed and its closed-loop RoA is estimated post-training.

Reachability-based RoA estimation. Reachability provides a complementary view by propagating sets under the system dynamics. Forward reachability and Hamilton–Jacobi (HJ) methods can capture complex attraction geometry El-Guindy et al. (2017); Mitchell et al. (2005); Bansal et al. (2017), but state-space discretization limits scalability as dimension increases. Simulation-driven methods instead lift numerical trajectories to continuous reach tubes using local discrepancy or sensitivity bounds Fan et al. (2016; 2017). This is related to our use of trajectory information to construct local regions, although our geometry- and flow-informed patches are learned from convergent trajectory samples and evaluated statistically rather than constructed as formally bloated reach tubes. Neural HJ methods alleviate the discretization cost by learning value-function approximations Bansal & Tomlin (2021); Hofgard (2024); Yang et al. (2025), while still primarily seeking a global reachability representation.

Positioning of this work. We study post-training RoA representation for a fixed closed-loop system, without controller retraining or certificate-aware synthesis. Rather than requiring a single global certificate or value function, we anchor the representation in a trajectory-validated local Lyapunov core, learn convergent trajectory seeds through alternative exploration strategies, and lift these seeds into complementary geometry- and flow-informed local regions with finite-sample reliability estimates.

6 Concluding Remarks

We developed an adaptive RoA representation for fixed nonlinear closed-loop systems combining a trajectory-validated local Lyapunov core, trajectory-based seed learning, geometry- and flow-informed local regions, and finite-sample validation. We compare dynamics-guided pullback and random seed learning with covariance-adapted ellipsoid and Euclidean-ball representations. Experiments on CartPole and six-dimensional quadrotor show substantial expansion beyond the local quadratic representation. Pullback provides more efficient seed discovery, whereas random exploration with ball geometry produces the largest continuous representation on both benchmarks, demonstrating that discrete seed coverage need not predict represented volume. Flow refinement complements the geometry layer by capturing residual trajectory-supported structures. These results support adaptive local representations without requiring a single global Lyapunov or reachability representation.

The framework provides finite-sample reliability rather than formal set-wise certification. Formal verification of finite-horizon reach-to-core remains challenging for controllers not designed for verifiability; recent verification-aware methods also obtain only partial formal results on harder systems. For example, Li et al. (2025) resort to PGD for a planar quadrotor with simpler dynamics than the 3D attitude dynamics considered here. Formal reach-to-core verification and higher-dimensional scalability remain important future directions, as does adaptive selection of seed-learning strategies and local geometries.

AI USE STATEMENT

In this work, we used ChatGPT to assist with the design and implementation of the proposed algorithms, development and debugging of experimental code, analysis and organization of experimental results, and organization and polishing of the manuscript. Generative AI was also used to support technical discussion during method development, including refinement of algorithmic ideas and experimental procedures.

All AI-assisted algorithmic designs and implementations were reviewed by the authors. Generated or modified code was tested through execution and its outputs were examined against the intended mathematical and algorithmic behavior. Experimental results reported in the paper were obtained by executing the implemented methods and were independently inspected by the authors; they were not generated or fabricated by the AI system. AI-assisted technical and writing suggestions were similarly reviewed, revised, and incorporated only when judged appropriate by the authors.

The authors take full responsibility for the methodology, implementation, experiments, analysis, claims, and final content of this work, including all text and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies computational methods for region-of-attraction analysis of nonlinear dynamical systems and does not involve human subjects, personal data, or sensitive datasets. The proposed framework provides statistically validated reliability estimates. Accordingly, its learned RoA representations should not be interpreted as formal safety certificates for deployment in safety-critical systems without additional application-specific verification and validation. All experimental results are obtained from simulated dynamical-system benchmarks.

REPRODUCIBILITY STATEMENT

The appendix provides detailed algorithms and implementation information for local-core construction, random and adaptive pullback seed learning, geometry- and flow-informed region construction, and independent statistical validation. Additional implementation details, hyperparameters, and experimental results are also reported there. The complete source code, configuration files, and experimental data used to produce the reported results will be made publicly available upon acceptance of the paper.

REFERENCES

Somil Bansal and Claire J. Tomlin. DeepReach: A deep learning approach to high-dimensional reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1817–1824. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561949.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire Tomlin. Hamilton–Jacobi reachability: A brief overview and recent advances. *IEEE Conference on Decision and Control*, pp. 2242–2253, 2017.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Science Robotics*, 2(7), 2017. doi: 10.1126/scirobotics.aah9867.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural Lyapunov control. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Brendon K. Colbert and Matthew M. Peet. Using trajectory measurements to estimate the region of attraction of nonlinear systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2341–2347. IEEE Press, 2018. doi: 10.1109/CDC.2018.8618959.

Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. In *Proceedings of Robotics: Science and Systems*, July 2021. doi: 10.15607/RSS.2021.XVII.063.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3):1749–1767, 2023.

Ahmed El-Guindy, Dongkun Han, and Matthias Althoff. Estimating the region of attraction via forward reachable sets. In *2017 American Control Conference (ACC)*, pp. 1263–1270, 2017. doi: 10.23919/ACC.2017.7963126.

Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Locally optimal reach set over-approximation for nonlinear systems. In *Proceedings of the 13th International Conference on Embedded Software*, pp. 6:1–6:10. ACM, 2016. doi: 10.1145/2968478.2968482.

Chuchu Fan, James Kapinski, Xiaoqing Jin, and Sayan Mitra. Simulation-driven reachability using matrix measures. *ACM Transactions on Embedded Computing Systems*, 17(1):21:1–21:28, 2017. doi: 10.1145/3126685.

Didier Henrion and Milan Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312, 2014. doi: 10.1109/TAC.2013.2283095.

William Hofgard. Convergence guarantees for neural network-based Hamilton–Jacobi reachability. *arXiv preprint arXiv:2410.02904*, 2024.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

Dhruv S. Kushwaha and Zoleikha A. Biron. A review on safe reinforcement learning using Lyapunov and barrier functions. *Artificial Intelligence Review*, 2026. doi: 10.1007/s10462-026-11611-9.

Haoyu Li, Xiangru Zhong, Bin Hu, and Huan Zhang. Two-stage learning of stabilizing neural controllers via Zubov sampling and iterative domain expansion. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Physics-informed neural network Lyapunov functions: PDE characterization, learning, and verification. *Automatica*, 175:112193, 2025. doi: 10.1016/j.automatica.2025.112193.

Yiming Meng, Ruikun Zhou, Amartya Mukherjee, Maxwell Fitzsimmons, Christopher Song, and Jun Liu. Physics-informed neural network policy iteration: algorithms, convergence, and verification. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Ian M. Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005. doi: 10.1109/TAC.2005.851439.

Spencer Richards, Felix Berkenkamp, and Andreas Krause. Lyapunov-based safe policy optimization for continuous control. In *Advances in Neural Information Processing Systems*, pp. 7865–7874, 2018.

Weehong Tan and Andrew Packard. Stability region analysis using polynomial and composite polynomial lyapunov functions and sum-of-squares programming. *IEEE Transactions on Automatic Control*, 53:565–571, 2008.

Yujie Yang, Hanjiang Hu, Tianhao Wei, Shengbo Eben Li, and Changliu Liu. Scalable synthesis of formally verified neural value function for Hamilton–Jacobi reachability analysis. *Journal of Artificial Intelligence Research*, 83, 2025. doi: 10.1613/jair.1.16946.

VI Zubov. *Methods of A.M. Lyapunov and Their Application*. Groningen: Noordhoff, 1964.

A Detailed Algorithms

This appendix gives the detailed procedures underlying the framework. The local-core construction is common to all experiments. For outer-region learning, we consider two convergent-seed strategies—adaptive pullback and uniform random exploration—and two geometry representations— covariance-adapted ellipsoids and Euclidean balls. This yields four learning–geometry combinations, followed by the same flow-informed residual construction.

A.1 Trajectory-Validated Local-Core Construction

Starting from the quadratic Lyapunov representation

$$V_0(x) = x^\top P x,$$

the local-core procedure first determines a conservative level through sampled nonlinear decrease testing and then recovers a larger trusted level through bounded reach-to-seed rollouts.

Algorithm 1 Trajectory-Validated Local-Core Construction

Require: Closed-loop dynamics f_{cl}, quadratic $V_0(x) = x^\top P x$, initial level c_{init}
Ensure: Trusted local core $\widehat{\Omega}_0 = \{x : V_0(x) \le c_0\}$

$c \leftarrow c_{\mathrm{init}}$
repeat
  Sample boundary states from $V_0(x) = c$
  Evaluate $\dot{V}_0(x) = \nabla V_0(x)^\top f_{\mathrm{cl}}(x)$
  if the sampled decrease test fails **then**
    $c \leftarrow \gamma_{\mathrm{shrink}} c$
  end if
until the sampled decrease test is satisfied
$c_{\mathrm{seed}} \leftarrow c$
$\Omega_{\mathrm{seed}} \leftarrow \{x : V_0(x) \le c_{\mathrm{seed}}\}$
Initialize $c_{\mathrm{lo}} \leftarrow c_{\mathrm{seed}}$ and choose $c_{\mathrm{hi}} > c_{\mathrm{lo}}$
while the candidate level is accepted **do**
  Sample states from $V_0(x) = c_{\mathrm{hi}}$
  Roll out the nonlinear closed-loop dynamics
  Measure bounded reach-to-Ω_{seed} success
  if the empirical acceptance criterion is satisfied **then**
    $c_{\mathrm{lo}} \leftarrow c_{\mathrm{hi}}$
    Increase c_{hi}
  else
    break
  end if
end while
Refine $[c_{\mathrm{lo}}, c_{\mathrm{hi}}]$ by bisection using the same rollout criterion
$c_{\mathrm{search}} \leftarrow$ largest passing search level
$c_0 \leftarrow \max\{c_{\mathrm{seed}}, \gamma_{\mathrm{backoff}} c_{\mathrm{search}}\}$
Validate the recovered boundary $V_0(x) = c_0$ using fresh rollout samples

return $\widehat{\Omega}_0 = \{x : V_0(x) \le c_0\}$

The decrease test is used to obtain a conservative local seed, whereas the recovery stage enlarges this seed using the actual nonlinear closed-loop trajectories. The resulting $\widehat{\Omega}_0$ is subsequently used as the common reach-to-core target.

A.2 Convergent Seed Learning

The outer representation is learned from states whose closed-loop trajectories exhibit bounded reach-to-core behavior. We investigate two strategies for discovering these states. The first uses dynamics-guided pullback exploration; the second provides a simple uniform-random baseline under the same rollout criterion.

A.2.1 Adaptive Pullback Seed Learning

Adaptive pullback learning expands outward from previously discovered convergent states. Successful trajectories are organized into a segment graph whose frontier guides subsequent sampling, while the outward scale is adjusted according to observed rollout success.

Algorithm 2 Adaptive Pullback Seed Learning

Require: Trusted core $\widehat{\Omega}_0$, closed-loop dynamics f_{cl}
Ensure: Convergent seed set $\mathcal{S}_{\mathrm{conv}}$
Initialize frontier $\mathcal{F}_0 \leftarrow \partial\widehat{\Omega}_0$, $\mathcal{S}_{\mathrm{conv}} \leftarrow \emptyset$, and outward scale α
for $k = 0, \ldots, K-1$ **do**
  Generate directional shell samples around $\mathcal{F}_k$
  Perform long-horizon closed-loop rollouts
  Retain bounded states that reach $\widehat{\Omega}_0$ and update $\mathcal{S}_{\mathrm{conv}}$
  Adapt α according to the observed success rate
  Build/update the trajectory-segment graph from successful trajectories
  Extract graph frontier nodes along successful flow segments
  Perform short-horizon correction near difficult frontier regions
  Update $\mathcal{F}_{k+1}$
end for

return $\mathcal{S}_{\mathrm{conv}}$

This strategy concentrates rollout effort near the evolving convergence frontier and uses observed trajectory structure to guide subsequent exploration.

A.2.2 Random Seed Learning

The random alternative removes frontier adaptation and trajectory-guided sampling. At every iteration, states are drawn independently and uniformly from the prescribed sampling domain $\mathcal{X}_{\mathrm{samp}}$ and evaluated using the same bounded reach-to-core rollout criterion. The same radial seed-bank filters used by pullback learning are applied so that the downstream representation receives comparable seed evidence.

Algorithm 3 Uniform-Random Convergent Seed Learning

Require: Trusted core $\widehat{\Omega}_0$, sampling domain $\mathcal{X}_{\mathrm{samp}}$, closed-loop dynamics f_{cl}
Ensure: Convergent seed set $\mathcal{S}_{\mathrm{conv}}$
$\mathcal{S}_{\mathrm{conv}} \leftarrow \emptyset$
for $k = 0, \ldots, K-1$ **do**
  Draw N iid samples $x_i \sim \mathrm{Unif}(\mathcal{X}_{\mathrm{samp}})$
  Perform the common bounded reach-to-core rollouts
  Separate successful and unsuccessful initial states
  Apply the common P-metric seed-bank shell filters
  Add retained successful states to $\mathcal{S}_{\mathrm{conv}}$
end for

return $\mathcal{S}_{\mathrm{conv}}$

No frontier, adaptive outward scale, or correction stage is used in random exploration. A trajectory-segment graph may be constructed from successful trajectories for compatibility with downstream region construction, but it does not influence subsequent random samples.

A.3 Geometry-Based Region Construction

The discrete convergent seeds are next lifted into continuous local regions. We use the same support-selection, coverage repair, and rollout-validation procedure with two alternative local primitives: covariance-adapted ellipsoids and isotropic Euclidean balls.

For a candidate center, a local neighborhood of convergent seeds is first selected. For the ellipsoidal representation, the local covariance determines the patch orientation and relative axis lengths, while a Mahalanobis-distance quantile sets its scale. For the ball representation, the local support is selected using Euclidean distance, the patch is recentered at the support mean, and its radius is obtained from a quantile of the support distances to that mean.

Algorithm 4 Geometry-Based Region Construction

Require: Convergent seeds $\mathcal{S}_{\mathrm{conv}}$, geometry type $G \in \{\text{ellipsoid}, \text{ball}\}$, trusted core $\widehat{\Omega}_0$
Ensure: Accepted geometry patches $\mathcal{E}_{\mathrm{geo}}$

```
Select initial candidate centers from available seed and trajectory evidence
for each candidate center z do
    Select a local convergent neighborhood N_z
    if G = ellipsoid then
        Estimate regularized covariance Σ_z from N_z
        Set the patch scale from a prescribed Mahalanobis-distance quantile
        Construct covariance-adapted ellipsoid E_z
    else
        Recenter at μ_z = mean(N_z)
        Set r_z from a prescribed quantile of {‖x − μ_z‖_2 : x ∈ N_z}
        Construct Euclidean ball E_z = {x : ‖x − μ_z‖_2 ≤ r_z}
    end if
end for
while seed coverage is below the target and the patch budget is not exhausted do
    Select farthest uncovered convergent seeds as repair centers
    Fit additional patches using the same geometry rule
end while
Draw interior and boundary-shell rollout samples for each candidate patch
Compute the one-sided failure upper bound ε^+(E, δ)
Accept patches satisfying ε^+(E, δ) ≤ ε_con

return E_geo
```

Thus, the ellipsoid and ball variants differ only in the local geometric representation fitted to the convergent evidence; coverage repair and trajectory-based statistical screening are otherwise shared. This permits the effect of geometry choice to be separated from the seed-learning strategy.

A.4 Flow-informed Residual Region Construction

The geometry layer may leave convergent structures that are poorly represented by its local primitives. Let

$$\mathcal{S}_{\mathrm{res}} = \mathcal{S}_{\mathrm{conv}} \setminus \bigcup_{E \in \mathcal{E}_{\mathrm{geo}}} E$$

denote the convergent seeds not covered by the accepted geometry patches. These residual seeds are first replayed using the same bounded reach-to-core semantics used during region construction. Only seeds whose successful behavior is reproduced are retained for flow-informed refinement.

The reproduced residual seeds are partitioned into hitting-time shells. Within each shell, clustering uses features combining the direction toward the core in the P metric, the local closed-loop flow direction, the state direction, and normalized log hitting time. A local covariance patch is then fitted to each sufficiently populated cluster in P-whitened coordinates.

Algorithm 5 Flow-informed Residual Region Construction

Require: Trusted core $\widehat{\Omega}_0$, convergent seeds $\mathcal{S}_{\mathrm{conv}}$, accepted geometry patches $\mathcal{E}_{\mathrm{geo}}$, closed-loop dynamics f_{cl}
Ensure: Accepted flow patches $\mathcal{E}_{\mathrm{flow}}$
Identify residual seeds

$$\mathcal{S}_{\mathrm{res}} \leftarrow \mathcal{S}_{\mathrm{conv}} \setminus \bigcup_{E \in \mathcal{E}_{\mathrm{geo}}} E$$

Replay $\mathcal{S}_{\mathrm{res}}$ using bounded reach-to-core rollouts
Retain reproduced successes and record their first core-hitting times
Partition retained seeds into hitting-time shells
for each hitting-time shell **do**
  Compute local closed-loop flow directions
  Form clustering features from P-metric core direction, flow direction, state direction, and log hitting time
  Cluster the residual seeds in the resulting feature space
  for each sufficiently populated cluster **do**
    Transform the cluster to P-whitened coordinates
    Fit a local covariance patch and determine its radius from the prescribed distance quantile
    Draw fresh samples from the candidate patch and perform bounded reach-to-core rollouts
    Compute the one-sided failure upper bound $\epsilon^{+}(E, \delta)$
    if $\epsilon^{+}(E, \delta) \leq \varepsilon_{\mathrm{con}}$ **then**
      $\mathcal{E}_{\mathrm{flow}} \leftarrow \mathcal{E}_{\mathrm{flow}} \cup \{E\}$
    end if
  end for
end for

return $\mathcal{E}_{\mathrm{flow}}$

The flow layer is applied separately to the residual set produced by each learning–geometry combination and therefore depends on both preceding design choices. In the reported experiments, flow-patch construction uses $\varepsilon_{\mathrm{con}} = 0.01$ with one-sided confidence $1 - \delta = 0.999$, matching the geometry-stage construction criterion. The completed hybrid representation is subsequently subjected to the stricter independent reliability validation reported in the benchmark appendices.

B CartPole Benchmark Details

This appendix provides additional implementation details and ablation results for the CartPole benchmark. Controller synthesis is performed separately from the proposed RoA framework, and the selected controller is fixed before local-core construction, seed learning, region construction, and statistical validation.

B.1 Dynamics and Domain

The CartPole state is

$$x = (p, \dot{p}, \theta, \dot{\theta}) \in \mathbb{R}^4,$$

where p is the cart position and θ is the pole angle relative to the upright equilibrium. The control input is the horizontal cart force $u \in [-10, 10]$.

We use

$$g = 9.8, \qquad m_c = 1.0, \qquad m_p = 0.1, \qquad \ell = 0.5,$$

with $M = m_c + m_p = 1.1$ and $m_p \ell = 0.05$. Define

$$a(x, u) = \frac{u + m_p \ell \dot{\theta}^2 \sin\theta}{M}, \qquad d(\theta) = \ell \left(\frac{4}{3} - \frac{m_p \cos^2\theta}{M} \right).$$

The nonlinear dynamics are

$$\ddot{\theta} = \frac{g \sin\theta - a(x,u)\cos\theta}{d(\theta)}, \qquad \ddot{p} = a(x,u) - \frac{m_p \ell \cos\theta}{M}\ddot{\theta}. \tag{18}$$

Together with the corresponding first-order kinematics, these equations define the continuous-time vector field used in the experiments. Numerical integration is used only for trajectory propagation; no state clipping or modification of the continuous dynamics is introduced.

Domain of interest. The controller-development and benchmark domain of interest is

$$\mathcal{X}_{\mathrm{DoI}} = \left\{ x : |p| \le 2.4,\ |\dot{p}| \le 2.5,\ |\theta| \le 12^\circ,\ |\dot{\theta}| \le 2.5 \right\}. \tag{19}$$

The DoI specifies the operating region used during controller development and for common volume comparisons; it does not define the RoA. For rollout validation we separately use the bounded computational domain

$$\mathcal{X}_{\mathrm{bd}} = \{ x : |x_i| \le 10, \quad i = 1, \dots, 4 \}. \tag{20}$$

B.2 Fixed Neural Feedback Controller

The experiments use a compact neural feedback controller obtained in a separate controller-learning stage. The proposed RoA method is independent of the controller representation, and no controller parameters are updated during any RoA experiment.

An initial stabilizing policy is learned using curriculum PPO with a larger two-hidden-layer tanh network. This policy is subsequently used as a teacher for a compact controller with architecture

$$4 \to 16 \to 16 \to 1,$$

using tanh nonlinearities and an output scaled by $u_{\max} = 10$.

As in the main text, the theoretical presentation uses coordinates translated to the closed-loop equilibrium.

The controller parameters remain fixed during all subsequent experiments. Hence, the proposed method estimates the RoA of a fixed nonlinear closed-loop system; the method itself does not depend on the controller being represented by a neural network.

B.3 Trusted Local Core Construction

The closed-loop linearization is stable and yields the initial quadratic level $c_{\mathrm{init}} = 2.8234 \times 10^{-3}$. The nonlinear decrease condition $\dot{V}_0 \le -0.01 V_0$ is then evaluated directly on the true nonlinear closed-loop dynamics. Using $5{,}000$ boundary samples per round and a shrink factor of 0.85, the initial level is reduced to $c_{\mathrm{seed}} = 1.7339 \times 10^{-3}$, at which no sampled decrease violations remain.

The conservative seed is subsequently enlarged by bounded reach-to-seed recovery. Each candidate level is tested using $4{,}000$ boundary rollouts with a zero-failure construction gate. Geometric expansion followed by bisection gives the largest passing search level $c_{\mathrm{search}} = 1.9968 \times 10^{-2}$. Applying the prescribed 0.98 backoff gives the trusted local core $c_0 = 1.9569 \times 10^{-2}$.

A fresh independent test of $8{,}192$ boundary states at the recovered level produces no reach-to-seed or boundedness failures. The corresponding one-sided Clopper–Pearson failure upper bound is 8.43×10^{-4} at 99.9% confidence. The recovered core is thereafter fixed and used as the reach-to-core target throughout Phase 2.

B.4 Convergent Seed Learning

We evaluate two approaches for discovering convergent states: uniform random exploration and dynamics-guided pullback exploration. Both use the same bounded reach-to-core criterion and ultimately produce a set of successful seeds for subsequent region construction.

Random exploration. Candidate states are sampled independently and uniformly over $\mathcal{X}_{\mathrm{DoI}}$ and propagated under the fixed closed-loop dynamics. Successful states are those that remain bounded and reach $\widehat{\Omega}_0$ within the prescribed rollout horizon. Retained successful and unsuccessful states are restricted to prescribed shells in the quadratic P-metric so that the resulting seed bank concentrates on the region relevant to outward representation.

The random experiment evaluates $200{,}000$ candidate states and retains $17{,}363$ convergent seeds. The complete seed-learning stage requires approximately 577.5 s.

Pullback exploration. The pullback variant begins from the boundary of $\widehat{\Omega}_0$ and repeatedly generates candidates around the current successful frontier in the whitened P-metric. The outward exploration scale adapts to the observed reach-to-core fraction. Successful trajectories are organized into short overlapping segments whose endpoints update the frontier, while a short-horizon correction stage probes difficult local regions.

The principal pullback settings are summarized in Table 3.

Table 3: CartPole pullback seed-learning configuration.

Parameter	Value
Pullback iterations	20
Initial frontier samples	2,000
Exploration samples / iteration	10,000
Exploration step size	0.02 s
Exploration horizon	100 steps (2.0 s)
Exploration perturbation	0.15
Correction samples / frontier	1,500
Correction step size	0.02 s
Correction horizon	10 steps (0.2 s)
Correction perturbation	0.08
Initial outward scale	1.05
Outward-scale range	$[1.01, 1.25]$
Scale increase / decrease	$1.03/0.95$
Segment length / stride	$5/2$
Graph clusters	8–64

Pullback exploration retains $32{,}225$ convergent seeds in approximately 160.4 s. It therefore discovers more retained convergent seeds substantially faster than uniform random exploration. As shown below, however, this advantage in seed discovery does not imply a larger final continuous representation.

B.5 Geometry-Representation Ablation

For each seed-learning strategy, we independently construct continuous regions using two local primitives: isotropic Euclidean balls and covariance-adapted ellipsoids. Candidate patches are generated from local convergent neighborhoods, expanded through repair centers when necessary, and screened using fresh interior and boundary-shell rollout samples.

All geometry variants use the same construction criterion: candidate patches are retained only when their one-sided statistical failure bound satisfies the prescribed 1% failure threshold. Thus, differences in the resulting regions reflect the learning and representation choices rather than different patch-reliability requirements.

Table 4 reports the complete 2×2 comparison. Occupancy is estimated under a common uniform measure over $\mathcal{X}_{\mathrm{DoI}}$ using 2^{20} samples.

Table 4: CartPole seed-learning and representation comparison.

Learning	Geometry	Patches	Seed cov.	Geom. occ.	Hybrid occ.
Pullback	Ellipsoid	2,048	94.18%	2.84%	2.84%
Pullback	Ball	1,606	99.36%	4.73%	4.73%
Random	Ellipsoid	1,636	98.04%	13.48%	13.48%
Random	Ball	1,355	75.31%	**16.45%**	**16.45%**

The comparison illustrates that discrete seed coverage and continuous represented volume measure different properties of the learned representation. Pullback–ball and random–ellipsoid cover 99.36% and 98.04% of their respective convergent seeds, whereas random–ball covers only 75.31%. Nevertheless, random–ball produces by far the largest continuous representation, occupying 16.45% of the DoI. Thus, maximizing coverage of a finite learned seed set does not necessarily maximize the continuous RoA representation.

Flow-informed residual refinement changes the represented volume only slightly for all four configurations. The final random–ball hybrid representation remains the largest, occupying 16.45% of the DoI compared with 2.10% for the trusted core, corresponding to a $7.82\times$ volume ratio. This configuration is therefore used as the final CartPole representation.

B.6 Flow-informed Residual Construction

After geometry construction, convergent states not represented by the accepted geometry patches form a residual set. Their closed-loop trajectories are replayed to obtain core-hitting information, from which local flow-informed patches are constructed. The flow stage therefore provides a complementary representation of residual structures left by the corresponding geometry layer.

For consistency across the ablation, candidate flow patches use the same 1% construction failure threshold as the geometry patches. Table 5 reports the resulting flow refinement for all four learning–geometry combinations.

Table 5: CartPole residual flow-patch results.

Learning	**Geometry**	**Flow patches**	**Geom. occ.**	**Hybrid occ.**
Pullback	Ellipsoid	42	2.8403%	2.8423%
Pullback	Ball	5	4.7317%	4.7330%
Random	Ellipsoid	7	13.4755%	13.4755%
Random	Ball	67	16.4483%	**16.4492%**

Across all four configurations, flow refinement adds little represented volume beyond the geometry layer. This indicates that the geometry layer captures most of the continuous attraction structure on CartPole, while flow patches primarily represent residual flow-dependent structures. For the selected random–ball configuration, 67 accepted flow patches increase DoI occupancy only slightly, from 16.4483% to 16.4492%.

B.7 Expansion and Independent Reliability

The random–ball configuration is used for the final CartPole representation because it produces the largest represented volume in the complete ablation. Using the common 2^{20} uniform DoI samples, the trusted core occupies 2.1033% of the DoI, the random–ball geometry layer occupies 16.4483%, and the completed hybrid representation occupies 16.4492%. The final representation therefore has a $7.8205\times$ volume ratio relative to the trusted core.

Final reliability is evaluated only after the complete representation has been fixed. Fresh samples are generated independently of seed learning, patch construction, and volume estimation. We separately test the geometry-added region, the flow-added residual region, and the complete newly added region. Each test uses $8{,}192$ initial states and the same bounded reach-to-core criterion.

Table 6: CartPole random–ball expansion and independent reliability validation.

Quantity	Value
Trusted-core DoI occupancy	2.1033%
Geometry-stage DoI occupancy	16.4483%
Final hybrid DoI occupancy	16.4492%
Hybrid/core volume ratio	$7.8205\times$
Geometry-added validation	$8192/8192$
Flow-added validation	$8192/8192$
Total-added validation	$8192/8192$
CP failure upper bound (99.9% conf.)	8.43×10^{-4}

No failures are observed in any of the three independent validation sets. For the total newly added region, the one-sided Clopper–Pearson upper failure bound is 8.43×10^{-4} at 99.9% confidence, below the prescribed 10^{-3} threshold. The selected CartPole representation therefore satisfies the required 99.9% statistical reliability at 99.9% confidence.

For the total-added samples, the median reach-to-core time is 1.04 s, the 90th percentile is 2.08 s, and the maximum observed reach time is 5.06 s, all well below the 40-s independent-validation horizon.

Overall, the CartPole ablation separates the roles of seed discovery and continuous representation. Pullback provides more efficient convergent-seed discovery, whereas random exploration produces seeds that lead to substantially larger continuous representations. Moreover, high coverage of the discrete convergent seed set does not necessarily imply a large continuous representation. Among the four combinations, random exploration with Euclidean-ball geometry produces the largest validated representation, despite not having the highest seed coverage.

C Quadrotor Attitude Stabilization Benchmark

This appendix provides additional implementation details for the quadrotor attitude-stabilization benchmark and the complete comparison of the seed-learning and geometry-representation variants. Controller training is performed separately from the proposed RoA-learning framework. The resulting controller is fixed throughout local-core construction, seed learning, region construction, and statistical validation.

C.1 Dynamics and Domain

The quadrotor attitude state is

$$x = (\phi, \theta, \psi, p, q, r) \in \mathbb{R}^6,$$

where (ϕ, θ, ψ) denote roll, pitch, and yaw angles, and (p, q, r) are the body-frame angular rates. The control input consists of the three body torques

$$u = (\tau_\phi, \tau_\theta, \tau_\psi).$$

The rigid-body inertias used in the experiments are

$$J_x = J_y = 4.856 \times 10^{-3}, \qquad J_z = 8.801 \times 10^{-3}.$$

The continuous-time attitude kinematics are

$$\begin{aligned} \dot{\phi} &= p + q \sin\phi \tan\theta + r \cos\phi \tan\theta, \\ \dot{\theta} &= q \cos\phi - r \sin\phi, \\ \dot{\psi} &= \frac{q \sin\phi + r \cos\phi}{\cos\theta}, \end{aligned} \tag{21}$$

and the body-rate dynamics are

$$\dot{p} = \frac{\tau_\phi - (J_y - J_z)qr}{J_x},$$
$$\dot{q} = \frac{\tau_\theta - (J_z - J_x)pr}{J_y},$$
$$\dot{r} = \frac{\tau_\psi - (J_x - J_y)pq}{J_z}. \tag{22}$$

Eqs. (21) and (22) are the nonlinear dynamics used in the reported experiments. In particular, (p, q, r) are body-frame angular rates rather than Euler-angle derivatives. No state clipping, denominator modification, or artificial saturation is introduced into the continuous-time vector field. Numerical integration is used only for trajectory propagation.

Domain of interest. The controller-development and benchmark domain of interest is

$$\mathcal{X}_{\mathrm{DoI}} = \left\{ x : \begin{array}{ll} |\phi|, |\theta| \le 0.35, & |\psi| \le 1.57, \\ |p|, |q|, |r| \le 2.5 & \end{array} \right\}. \tag{23}$$

The DoI specifies the operating region used during controller development and benchmark analysis. It does not truncate the subsequent RoA expansion.

C.2 Fixed Neural Feedback Controller

The benchmark uses a neural attitude controller obtained in a separate controller-learning stage. Controller synthesis is not part of the proposed RoA method; the following information is provided only for reproducibility.

An initial stabilizing policy is learned using PPO over an expanding attitude-state curriculum. A compact controller is subsequently obtained for the stability-analysis pipeline. The fixed controller used in the reported RoA experiments is a tanh MLP with architecture

$$6 \to 16 \to 16 \to 3,$$

using weight-normalized linear layers. Its normalized outputs are scaled by

$$u_{\max} = 0.1$$

to produce the three body torques.

The controller parameters remain fixed during all subsequent experiments. Hence, the proposed method estimates the RoA of a fixed nonlinear closed-loop system; the method itself does not depend on the controller being represented by a neural network.

The learned controller induces a shifted closed-loop equilibrium, approximately

$$x^\star = (-0.164075, -0.175200, -0.120367, 0, 0, 0)^\top.$$

Consistent with the zero-equilibrium convention in the main text, the RoA analysis is performed in translated coordinates $\tilde{x} = x - x^\star$.

C.3 Trusted Local Core Construction

The closed-loop linearization is stable and yields the initial quadratic level $c_{\mathrm{init}} = 3.4111 \times 10^{-4}$. The nonlinear decrease condition $\dot{V}_0 \le -0.01 V_0$ is then evaluated directly on the true nonlinear closed-loop dynamics. Using 8,192 boundary samples per round and a shrink factor of 0.85, the initial level passes directly with no sampled decrease violations, giving $c_{\mathrm{seed}} = 3.4111 \times 10^{-4}$.

The conservative seed is subsequently enlarged by bounded reach-to-seed recovery. Each candidate level is tested using $4{,}000$ boundary rollouts with a zero-failure construction gate. Geometric expansion followed by bisection gives the largest passing search level $c_{\mathrm{search}} = 1.4003 \times 10^{-1}$. Applying the prescribed 0.98 backoff gives the trusted local core $c_0 = 1.3723 \times 10^{-1}$.

A fresh independent test of $8{,}192$ boundary states at the recovered level produces no reach-to-seed or boundedness failures. The corresponding one-sided Clopper–Pearson failure upper bound is $8.43 \times$

10^{-4} at 99.9% confidence. The recovered core is thereafter fixed and used as the reach-to-core target throughout Phase 2.

C.4 Convergent Seed Learning

We compare the dynamics-guided pullback strategy with random exploration under the same bounded reach-to-core criterion. Pullback exploration produces $161{,}768$ validated convergent seeds, whereas random exploration produces $125{,}194$. Pullback is also substantially faster, requiring approximately 461.1 s compared with $2{,}298.0$ s for random exploration.

Thus, dynamics-guided exploration provides both more convergent evidence and substantially lower seed-discovery cost. As the continuous-region results below demonstrate, however, the number of discovered seeds alone does not determine the quality of the resulting RoA representation.

For pullback exploration, the principal implementation parameters are summarized in Table 7.

Table 7: Quadrotor pullback-learning configuration.

Parameter	Value
Pullback iterations	20
Exploration samples / iteration	10,000
Exploration step size	0.02 s
Exploration horizon	100 steps (2.0 s)
Correction samples / frontier	1,500
Correction step size	0.02 s
Correction horizon	10 steps (0.2 s)
Bounded rollout domain	$\lvert x_i\rvert \le 10$
Initial outward scale	1.05
Outward-scale range	$[1.01, 1.25]$
Scale increase / decrease	$1.03/0.95$
Exploration jitter	0.15
Correction jitter	0.08
Initial frontier samples	2,000

A rollout is retained only when it remains inside the bounded rollout domain and reaches the trusted core within the prescribed horizon. The boundedness domain

$$\mathcal{X}_{\mathrm{bd}} = \{x : |x_i| \le 10,\ i = 1, \dots, 6\}$$

is deliberately much larger than $\mathcal{X}_{\mathrm{DoI}}$ and is used only as a divergence guard. Thus, the controller-training DoI does not impose an artificial boundary on the learned RoA.

C.5 Geometry Representation Ablation

The convergent seeds are converted into continuous regions using either covariance-adapted ellipsoids or isotropic Euclidean balls. Combining the two seed-learning strategies with the two geometric primitives gives four configurations. All geometry variants use the same construction criterion: candidate patches are retained only when their one-sided statistical failure bound satisfies the prescribed 1% failure threshold.

Table 8 reports the complete comparison. The DoI occupancies are evaluated using a common set of 2^{20} uniformly distributed samples.

Table 8: Quadrotor seed-learning and representation comparison.

Learning	Geometry	Patches	Seed cov.	Geom. occ.	Hybrid occ.
Pullback	Ellipsoid	178	99.56%	68.35%	68.54%
Pullback	Ball	196	71.85%	27.33%	27.55%
Random	Ellipsoid	1,216	75.39%	62.75%	63.21%
Random	Ball	1,213	88.65%	**78.48%**	**78.68%**

The comparison illustrates that discrete seed coverage, geometry-stage occupancy, and final hybrid occupancy capture different aspects of representation quality. In particular, pullback–ellipsoid

achieves the highest seed coverage with only 178 accepted geometry patches, but this does not produce the largest continuous representation. Random–ball instead achieves the largest geometry-stage occupancy and remains the strongest configuration after flow-informed residual refinement, reaching 78.68% final DoI occupancy.

The relatively small differences between geometry and hybrid occupancy also show that the geometry layer captures most of the represented volume for all four configurations. The flow layer provides a complementary refinement whose contribution depends on the residual structure left by the corresponding geometry representation. We therefore select random–ball for the final quadrotor representation.

C.6 Flow-informed Residual Construction

Flow-informed refinement is applied to the residual structure left by each corresponding geometry representation. Because the residual set depends directly on the seed-learning and geometry combination, the resulting hybrid occupancies are reported together with the geometry-stage occupancies in Table 8.

For the selected random–ball configuration, seeds not covered by the geometry layer are replayed using the same bounded reach-to-core semantics. Reproduced residual seeds are grouped using local state geometry, closed-loop flow direction, and time-to-core information, and flow-informed micro-patches are constructed for the resulting residual structures.

The flow stage accepts 212 residual patches. Its contribution to represented volume is relatively small because the ball geometry already captures most of the learned attraction structure: occupancy increases from 78.48% after geometry construction to 78.68% for the final hybrid representation. Thus, the flow layer acts primarily as a complementary refinement rather than as the dominant source of expansion.

C.7 Expansion and Independent Reliability

To quantify the continuous representation, we estimate the occupied volume of the trusted core, geometry representation, and final hybrid representation using the same 2^{20} uniform DoI samples. Final reliability is evaluated only after the representation is fixed, using independent samples drawn from the newly represented volume.

Table 9: Quadrotor random–ball expansion and independent reliability validation.

Quantity	Value
Trusted-core DoI occupancy	13.11%
Geometry-stage DoI occupancy	78.48%
Final hybrid DoI occupancy	78.68%
Hybrid/core volume ratio	$6.003\times$
Geometry-added validation	$8192/8192$
Flow-added validation	$8192/8192$
Total-added validation	$8192/8192$
CP failure upper bound (99.9% conf.)	8.43×10^{-4}

No failures are observed in the geometry-added, flow-added, or total-added validation sets. For the total newly added region, the one-sided Clopper–Pearson failure upper bound is 8.43×10^{-4} at 99.9% confidence, below the prescribed 10^{-3} threshold.

The other three learning–geometry configurations likewise achieve $8{,}192/8{,}192$ successful total-added-region rollouts under the same independent validation. Thus, the random–ball configuration is selected for its larger continuous representation rather than a weaker reliability criterion.