Data-Efficient Verification of Black-box Controllers via Flow Matching and Conformal Inference

Dhruv Metha Ramesh*[1], Sumanth Tangirala*[1], Mufang Ying[2], Riddhima Medatwal[1], Ewerton R. Vieira[3], Abdeslam Boularias[1], Ying Hung[2], Kostas E. Bekris[1]

***Abstract*— Verification of robot controllers is important for ensuring safe and reliable robot operation. It can involve the characterization of a controller's Region of Attraction (RoA), i.e., the initial states that lead to a successful outcome. This is challenging, however, for black-box, learned controllers where analytical models are unavailable. In such scenarios, verification is conducted using only trajectory data, making data efficiency critical as each data point represents a costly physical experiment. To address this, this paper presents a framework that integrates three components: (1) a Conditional Flow Matching (CFM) model that learns the distribution of final states from initial conditions, capturing multi-modal outcome distributions that naturally express uncertainty near the boundary between success and failure regions; (2) a probabilistic classification mechanism that interprets the learned distribution via optimized decision thresholds to categorize states into the RoA, the failure region, or an uncertain region where the model cannot commit to an outcome; and (3) an adaptive sampling strategy that concentrates trajectory collection on the uncertain region to reduce it with fewer rollouts than uniform sampling. The resulting predictions are certified via conformal inference to provide finite-sample coverage guarantees. Experiments on benchmark systems ranging from a 2D pendulum to a 13D quadrotor demonstrate that the proposed framework achieves high verification accuracy with low uncertainty compared to existing methods, and that adaptive sampling further improves data efficiency over the non-adaptive variant.**
Website: https://tinyurl.com/adaptive-roa-via-fm

I. Introduction

Problem Domain: Verification of robot controllers is fundamental for ensuring safety and reliability in robotics [1]. A central challenge in this context is characterizing the *Region of Attraction* (RoA): the set of initial states from which a controller achieves success, such as converging to a desired goal within a finite time or reaching an attractor of the underlying dynamical system. An effective verifier should also characterize the RoA for failure and operate effectively close to the boundary between the two corresponding regions, where the switch from success to failure occurs.

Challenges: Traditionally, RoA characterization relies on accurate analytical models, which are difficult to obtain due to numerical approximations, parameter uncertainty, and unmodeled dynamics [3], [4], and are unavailable for black-box learned policies [5]. Classical tools such as Lyapunov certificates or Sum-of-Squares optimization require analytical dynamics and are thus inapplicable, necessitating

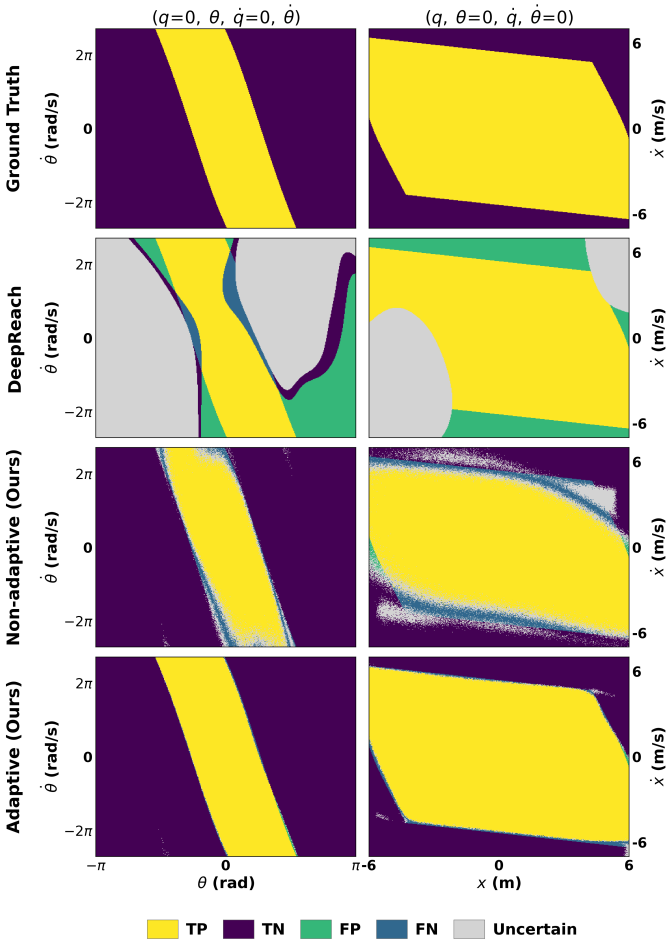


Fig. 1: RoA identification on two 2D slices of a 4D Cartpole ($(\theta, \dot{\theta})$ left, $(x, \dot{x})$ right; other DoFs zero). Rows, top to bottom: ground truth, DeepReach [2] (modified), and our non-adaptive and adaptive variants. Yellow/purple: correctly identified success/failure (TP/TN); green/teal: false positives/negatives; grey: uncertain (abstentions). The adaptive method closely recovers the ground truth, whereas DeepReach abstains on 16.7% of the space and misclassifies much of the true RoA.

data-driven verification from trajectory rollouts alone. This raises four challenges: **(A)** near the success/failure boundary, outcomes are *multi-modal*: nearby initial conditions diverge to qualitatively different final states, which a deterministic model cannot express; **(B)** scalar-sublevel-set representations such as DeepReach [2] and Neural Lyapunov functions [6] do not formally quantify prediction *confidence*; **(C)** trajectory data is the primary bottleneck, yet the boundary occupies a thin manifold underrepresented under uniform sampling,

*Equal contribution. [1]Department of Computer Science, [2]Department of Statistics, and [3]Department of Mathematics, Rutgers University, NJ, USA. {dm1487, st1122, kb572}@cs.rutgers.edu

motivating *informed* acquisition; and **(D)** empirical accuracy alone is insufficient and without calibration, confidence and abstention carry no guarantees.

Proposed Contributions: Motivated by these challenges, this paper proposes a framework that is *generative* (capturing multi-modal final-state distributions while decoupling dynamics from fixed success criteria), *certifiable* (providing finite-sample guarantees and principled abstention), and *adaptive* (prioritizing data where the model is uncertain). Its components are: **(1)** a Conditional Flow Matching (CFM) model [7] that learns a multi-modal distribution of final states from raw rollouts; **(2)** a probabilistic classifier that converts Monte Carlo samples into success/failure/uncertain predictions via an optimized threshold rule; **(3)** an uncertainty-guided adaptive sampling strategy that concentrates rollouts on the uncertain region to refine the boundary; and **(4)** a split conformal prediction layer that guarantees the true outcome lies in a calibrated prediction set, with non-singleton sets certifying abstention. The framework is evaluated on a 2D pendulum, 4D cartpole, and 13D spatial quadrotor under LQR control, and a 6D planar quadrotor under a learned policy, against six baselines spanning value-function, topological, and direct-prediction approaches, achieving higher F1 and a smaller uncertain region while maintaining target conformal coverage.

II. Problem Formulation

Let $\mathcal{X}$ be an n-dimensional state space (a smooth manifold embedded in $\mathbb{R}^d$) of a robotic system under a fixed policy. The deterministic closed-loop dynamics define a flow map $\Phi : \mathcal{X} \to \mathcal{X}$ taking each initial state x_0 to its terminal state $x_T = \Phi(x_0)$, where the trajectory runs until a horizon T or until success/failure, whichever comes first. Neither the dynamics nor the controller structure are known; the only information comes from rollouts $\tau = (x_0, x_1, \ldots, x_T)$. Two task-specific criteria classify the terminal state: a success criterion $\gamma_{\mathrm{s}} : \mathcal{X} \to \{0, 1\}$ (e.g., reaching a goal) and a failure criterion $\gamma_{\mathrm{f}} : \mathcal{X} \to \{0, 1\}$ (e.g., leaving admissible bounds). With T large enough that almost every trajectory terminates in a single outcome before timeout, this induces a.e. a ground-truth label:

$$y(x_0) = \begin{cases} \texttt{<Succ>} & \text{if } \gamma_{\mathrm{s}}(\Phi(x_0)) = 1, \\ \texttt{<Fail>} & \text{if } \gamma_{\mathrm{f}}(\Phi(x_0)) = 1, \end{cases} \tag{1}$$

partitioning $\mathcal{X}$ into the *region of attraction* $\texttt{RoA}_{\mathrm{s}} = \{x_0 : y(x_0) = \texttt{<Succ>}\}$ and the failure region $\texttt{RoA}_{\mathrm{f}} = \mathrm{int}(\mathcal{X} \setminus \texttt{RoA}_{\mathrm{s}})$, separated by the *true boundary*.

Problem (RoA Estimation): Given rollouts $\mathcal{D} = \{\tau_i\}_{i=1}^N$, determine whether the controller succeeds or fails from an initial state x_0, i.e., estimate $\texttt{RoA}_{\mathrm{s}}$ and $\texttt{RoA}_{\mathrm{f}}$. Since predictions inform deployment, each must carry a known confidence: the probability of a correct prediction must be at least $(1 - \alpha)$ for a user-specified $\alpha \in (0, 1)$.

III. Proposed Method

The proposed framework couples a probabilistic classifier for RoA estimation (Fig. 2a) with an uncertainty-guided adaptive sampling and training loop (Fig. 2b).

A. Generative Final-State Prediction

Rather than a direct classifier $x \mapsto \hat{y}$, the method learns a conditional generative model $p_\theta(\cdot \mid x)$ mapping each state x to a distribution over terminal states. This naturally captures multi-modal outcomes near the true boundary, where a single deterministic prediction may fail, and decouples the learned dynamics from any fixed success criterion: no success or failure labels are needed at training time. The model is a Conditional Flow Matching (CFM) [7] network with the rectified-flow parameterization [8]: a neural vector field $v_\theta(z_s, s, x)$ conditioned on x transports standard-Gaussian noise to the terminal-state distribution $p_\theta(\cdot \mid x)$. Training pairs every non-terminal state x_t along a trajectory with its terminal state x_T; since the closed-loop dynamics are deterministic, each length-T trajectory yields T supervision pairs $\mathcal{D}_{\mathrm{pairs}}^{\mathrm{train}} = \{(x_t, x_T)\}$ (with a labeled validation split $\mathcal{D}_{\mathrm{pairs}}^{\mathrm{val}}$ for threshold optimization). The dataset construction and CFM loss are detailed in Appendix VII-B.

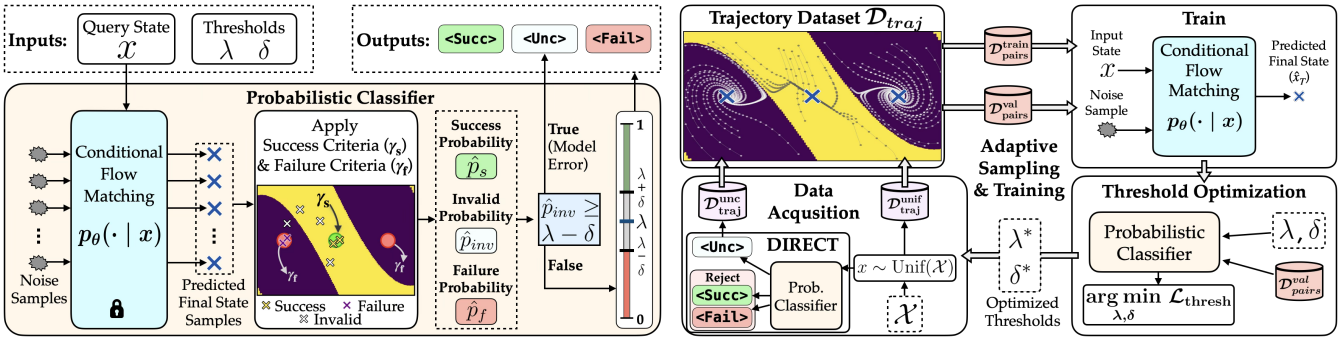
B. Probabilistic Classification

The generative model is converted into a probabilistic classifier (Fig. 2a) by estimating outcome probabilities via Monte Carlo sampling and a threshold-based decision rule.

Monte Carlo probability estimation: For a state x, the model draws K final states $\hat{x}_T^{(1)}, \ldots, \hat{x}_T^{(K)} \sim p_\theta(\cdot \mid x)$. Each sample is evaluated against both criteria: $\hat{p}_{\mathrm{s}}(x) = \frac{1}{K}\sum_{k=1}^K \gamma_{\mathrm{s}}\big(\hat{x}_T^{(k)}\big)$, $\hat{p}_{\mathrm{f}}(x) = \frac{1}{K}\sum_{k=1}^K \gamma_{\mathrm{f}}\big(\hat{x}_T^{(k)}\big)$. For systems with complementary criteria, γ_{s} and γ_{f} partition the terminal state space, so $\hat{p}_{\mathrm{s}} + \hat{p}_{\mathrm{f}}$ would equal one. In general, however, the generative model may produce invalid samples that satisfy neither criterion. The remainder $\hat{p}_{\mathrm{inv}}(x) = 1 - \hat{p}_{\mathrm{s}}(x) - \hat{p}_{\mathrm{f}}(x)$ quantifies the model error, reflecting regions where the model has not learned well.

Threshold-based decision rule: A dual-threshold rule converts probability estimates to predictions $\hat{y}(x) \in \{\texttt{<Succ>}, \texttt{<Fail>}, \texttt{<Unc>}\}$, with decision boundaries $\theta_{\mathrm{s}} = \lambda + \delta$ and $\theta_{\mathrm{f}} = 1 - (\lambda - \delta)$, where $\lambda \in (0, 1)$ is the decision center and $\delta \geq 0$ controls the width of the uncertain region. The rule applies in three stages: first, if $\hat{p}_{\mathrm{inv}}(x) \geq \lambda - \delta$, the state is flagged as <Unc> due to *model error*, as too many invalid endpoints indicate the model has not learned this region. Otherwise, the rule predicts <Succ> if $\hat{p}_{\mathrm{s}}(x) \geq \theta_{\mathrm{s}}$, <Fail> if $\hat{p}_{\mathrm{f}}(x) \geq \theta_{\mathrm{f}}$, and <Unc> due to *outcome uncertainty* if neither threshold is satisfied, as the model generates valid but mixed outcomes and cannot confidently resolve the outcome at that state. Both sources of $\hat{y} = \texttt{<Unc>}$ form the uncertain region, but $\hat{p}_{\mathrm{inv}}$ distinguishes them. The uncertain region need not coincide with the true boundary; it may also include states where data coverage is insufficient. As the model improves, the uncertain region is expected to contract toward the true boundary.

Threshold optimization: Since the optimal trade-off depends on the data distribution, (λ, δ) are optimized on $\mathcal{D}_{\mathrm{pairs}}^{\mathrm{val}}$ rather than fixed, trading off the error rate among confident predictions against the uncertain-region rate via a weight $w \in$



(a) Probabilistic classifier via conditional flow matching.

(b) Adaptive sampling and training of the classifier.

Fig. 2: Overview of the proposed framework: (a) the generative probabilistic classifier, and (b) its uncertainty-guided adaptive sampling and training loop.

$[0, 1]$ (objective in Appendix VII-B). Larger w widens the uncertain region, yielding fewer misclassifications; smaller w tightens it at the cost of more errors. The optimized boundaries $\theta_{\mathrm{s}}^* = \lambda^* + \delta^*$ and $\theta_{\mathrm{f}}^* = 1 - (\lambda^* - \delta^*)$ are then used for probabilistic classification.

C. Adaptive Sampling and Training

Since evaluating $p_\theta(\cdot \mid x)$ is orders of magnitude cheaper than a rollout, candidate initial states can be screened before committing to data acquisition. The classifier (Sec. III-B) provides an acquisition signal: states receiving $\hat{y}(x) =$ <Unc> are where the model lacks confidence and where additional rollouts yield the greatest uncertainty reduction, enabling targeted acquisition that concentrates rollouts on the uncertain region rather than spreading them uniformly.

Acquisition strategy: DIRECT. Each acquisition round collects N new trajectories, governed by a *sampling ratio* $r_{\mathrm{unc}} \in [0, 1]$ that controls the exploration-exploitation split. $\lfloor N \cdot (1 - r_{\mathrm{unc}}) \rfloor$ initial states are drawn uniformly from $\mathcal{X}$, providing broad exploration of the state space. The remaining $\lceil N \cdot r_{\mathrm{unc}} \rceil$ trajectories target the uncertain region via rejection sampling: candidates are drawn uniformly from $\mathcal{X}$, screened by probabilistic classification, and retained only if $\hat{y}(x) =$ <Unc>; drawing and screening continues until $\lceil N \cdot r_{\mathrm{unc}} \rceil$ candidates are accepted.

Iterative procedure: Starting from $\mathcal{D}_{\mathrm{traj}}^{\mathrm{init}}$, trajectories collected uniformly from $\mathcal{X}$, the pipeline applies DIRECT over E iterations (Algorithm 1, Appendix VII-B). Each iteration retrains the generative model on the accumulated $\mathcal{D}_{\mathrm{pairs}}^{\mathrm{train}}$, re-optimizes (λ^*, δ^*) on $\mathcal{D}_{\mathrm{pairs}}^{\mathrm{val}}$, and acquires N new trajectories via DIRECT. The total trajectory budget is $N_{\mathrm{total}} = |\mathcal{D}_{\mathrm{init}}| + E \cdot N$.

Inference: To classify a new state x, the model draws K terminal-state samples from $p_\theta(\cdot \mid x)$, evaluates each against $\gamma_{\mathrm{s}}, \gamma_{\mathrm{f}}$ to obtain $\hat{p}_{\mathrm{s}}(x), \hat{p}_{\mathrm{f}}(x)$, and applies the threshold rule with the optimized boundaries, producing a point prediction $\hat{y}(x) \in \{$<Succ>, <Fail>, <Unc>$\}$.

IV. Probabilistic Guarantees

The point predictions of Sec. III are augmented with finite-sample coverage guarantees via split conformal prediction [9], producing prediction sets that contain the true label with probability at least $1 - \alpha$.

A nonconformity score $s(x, y)$ measures how much the model's probability estimates would need to change for a candidate label $y \in \{$<Succ>, <Fail>, <Unc>$\}$ to become the natural prediction; it is zero when y is already consistent with the optimized decision boundaries $\theta_{\mathrm{s}}^*, \theta_{\mathrm{f}}^*$ (i.e., the corresponding probability exceeds its boundary for <Succ>/<Fail>, or neither does for <Unc>). Calibrating s on a held-out set $\mathcal{D}'_{\mathrm{cal}}$ (the model-error filter of Sec. III-B applied) yields a quantile $\hat{q}$, and the prediction set $\mathcal{C}(x) = \{y : s(x, y) \leq \hat{q}\}$ collects all labels scoring below it. The score, calibration, and prediction-set construction are detailed in Appendix VII-B.

Theorem 1 [9]: If the calibration data $\mathcal{D}'_{\mathrm{cal}}$ and the test point $(x_{\mathrm{test}}, y_{\mathrm{test}})$ are exchangeable, then the prediction set satisfies $\Pr\big(y_{\mathrm{test}} \in \mathcal{C}(x_{\mathrm{test}})\big) \geq 1 - \alpha$.

A singleton $\{$<Succ>$\}$ or $\{$<Fail>$\}$ is a confident prediction carrying the $(1-\alpha)$ guarantee; any non-singleton set, or $\{$<Unc>$\}$, classifies the state as uncertain. The guarantee applies to states not removed by the model-error filter, which are assigned <Unc> directly without a coverage claim.

V. Experiments

This section evaluates 7 methods on 4 dynamical systems of varying dimension (Figure 3). All methods operate over the same trajectory data without access to the dynamics or controller structure.

A. Benchmark Systems

Four systems of increasing dimension are used: a **Pendulum (2D)** and **Cartpole (4D)** under LQR control, a **Planar Quadrotor (6D)** under a learned PPO policy (testing a black-box controller with an irregular boundary), and a **Spatial Quadrotor (13D)** under LQR control (testing scalability). The pendulum uses complementary criteria ($\gamma_{\mathrm{f}} = 1 - \gamma_{\mathrm{s}}$); for the other three, failure is exceeding the state-space bounds. The latter three, including controllers, use `safe-control-gym` [10]. Eval-

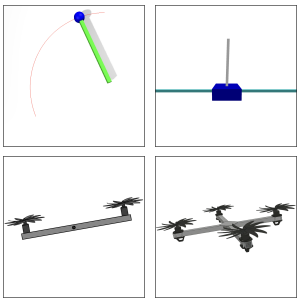


Fig. 3: Benchmark systems. Top: Pendulum (2D), Cartpole (4D). Bottom: Planar Quad (6D), Spatial Quad (13D).

Method	Pend. (2D) F1↑	Unc.↓	Cartp. (4D) F1↑	Unc.↓	P.Quad (6D) F1↑	Unc.↓	S.Quad (13D) F1↑	Unc.↓
DeepReach	.982	3.6%	.901	16.7%	.353	0%	.701	42.8%
Lyapunov NN	.851	0%	.978	71.9%	.571	97.5%	.921	32.1%
MORALS	.905	24.4%	.699	17.9%	.292	32.4%	.194	81.8%
Determ. FS	.768	0.6%	.346	1.0%	.067	1.0%	.029	1.0%
Classification	.974	0.2%	.946	0.2%	.777	0.5%	.763	0.6%
Non-adaptive	**.986**	4.2%	.984	7.6%	.855	13.2%	.947	25.3%
Adaptive	.974	6.0%	**.990**	1.0%	**.933**	6.0%	**.953**	25.2%

TABLE I. Comparison across the four dynamical systems (Non-adaptive and Adaptive are ours; Adaptive uses $r_{\text{unc}} = 1.0$).

uation sets range from 50K (2D) to 1M (13D) states with 8–39% success; full specifications are in Appendix VII-C.

B. Methods Compared

All methods receive the same trajectory data $\mathcal{D}_{\text{traj}}$ without an analytical model and produce point predictions $\hat{y} \in \{\texttt{<Succ>}, \texttt{<Fail>}, \texttt{<Unc>}\}$ without conformal calibration (coverage is validated separately in Sec. V-E). *Structured baselines* learn an intermediate control-theoretic or topological representation: *DeepReach* [2] approximates the Hamilton-Jacobi value function via a neural PDE solver, adapted to the model-free setting by estimating the Hamiltonian from data; *Neural Lyapunov* [6] learns $V(x)$ with a sampled decrease condition (via `NeuroMANCER` [11]); for both, a threshold band on the value function is calibrated post-training. *MORALS* [12] builds a Morse graph over a learned latent representation. *Direct-prediction baselines* map states to outcomes directly: *Deterministic Final-State* regresses a single $\hat{x}_T$ evaluated against the criteria, and *Classification* predicts $\Pr(\texttt{<Succ>})$ with a threshold band for `<Unc>`. The *proposed variants* are *Non-adaptive* (the classifier of Sec. III with uniform sampling) and *Adaptive* (trajectories concentrated on the separatrix per Sec. III-C); comparing them isolates adaptive sampling. Both use $K = 20$ samples and $w = 0.9$, evaluated on point predictions only.

C. Metrics

Metrics are computed on a held-out set with ground-truth labels: the *uncertain region rate* (fraction classified $\hat{y} = \texttt{<Unc>}$) and the *F1 score* for the success class over confident predictions ($\hat{y} \neq \texttt{<Unc>}$). The two are interpreted jointly, as high F1 is meaningful only at moderate uncertain-region rates.

D. Results

Table I compares 7 methods across 4 systems using $N_{\text{total}} \in \{500, 1{,}000, 12{,}000, 25{,}000\}$ trajectories respectively. The proposed method achieves the highest F1 on all four benchmarks (non-adaptive leading on the pendulum, adaptive on the rest).

Baselines: The structured baselines each fail on one axis: DeepReach is either overconfident (.35 F1 on the planar quadrotor) or over-conservative (42.8% abstention, .70 F1 on the spatial quadrotor); Neural Lyapunov misclassifies directly (.85 F1 at 0% abstention on the pendulum) or abstains aggressively (up to 97.5% on the planar quadrotor); and MORALS degrades on both axes (F1 .91→.19, abstention 24.4%→81.8%) as its bounded-state assumption is violated. The classification baseline is the strongest direct method (.76–.97 F1), but its gap to ours widens with dimensionality (.78 vs. .93 and .76 vs. .95 on the planar and spatial quadrotors), indicating a scaling advantage of generative modeling.

Adaptive sampling and data efficiency: Both variants

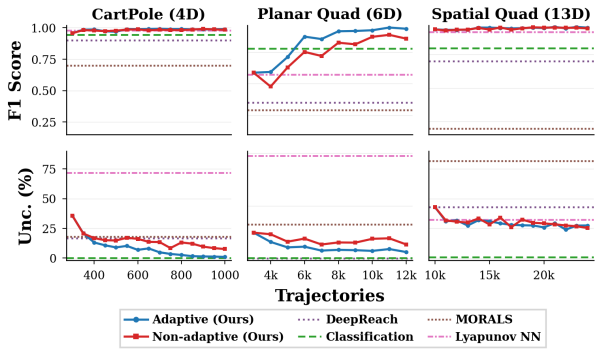


Fig. 4: F1 score (top) and uncertain region rate (bottom) vs. trajectory budget across 3 systems. Horizontal lines show baseline performance at full data budget.

improve F1 and reduce the uncertain region rate with increasing budget, surpassing baselines trained at full data. Adaptive sampling yields ∼1.4× savings on the cartpole (uncertain rate dropping to ∼1% vs. ∼10%) and ∼1.5× on the planar quadrotor, where it accelerates the crossover past the classification baseline and reaches higher final F1. On the spatial quadrotor F1 exceeds the baseline throughout, though the uncertain rate plateaus at ∼25%. False positive rates stay below 1% everywhere; the adaptive F1 gains come from reduced false negatives (10.4% vs. 22.6% on the planar quadrotor). The uncertain region's model-error and outcome-uncertainty components, and the 13D plateau, are analyzed in Appendix VII-C.

E. Coverage Guarantees

At $\alpha = 0.1$, across all four systems at the final adaptive iteration, $\hat{q} = 0$ and empirical coverage exceeds 0.98: the point predictions already meet the 0.90 target without conformal widening. Average prediction-set size is 1.0 (1.01 on the planar quadrotor), i.e., nearly all predictions are singletons.

VI. Conclusion

This paper presents a data-driven framework for `RoA` estimation that combines generative final-state prediction, uncertainty-guided adaptive sampling, and conformal coverage guarantees, without access to system dynamics or controller structure. Across four systems (2D–13D), it achieves the highest F1 on all benchmarks with a formal $(1 - \alpha)$ guarantee and up to 1.5× data savings over uniform sampling; the advantage over discriminative baselines widens with dimensionality, and the conformal layer adds guarantees at no cost to accuracy ($\hat{q} = 0$). Extending it to stochastic systems and image-based states is future work.

REFERENCES

[1] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," *CSUR*, 2019.

[2] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *ICRA*, 2021.

[3] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *SSCI*, 2020.

[4] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning," *IEEE Access*, 2021.

[5] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE TRO*, 2023.

[6] S. M. Richards, F. Berkenkamp, and A. Krause, "The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *CoRL*, 2018.

[7] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *ICLR*, 2023.

[8] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *ICLR*, 2023.

[9] V. Vovk, A. Gammerman, and G. Shafer, *ALRW*. Springer, 2005.

[10] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, "Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics," *IEEE RAL*, 2022.

[11] J. Drgona, A. Tuor, J. Koch, M. Shapiro, B. Jacob, and D. Vrabie, "NeuroMANCER: Neural Modules with Adaptive Nonlinear Constraints and Efficient Regularizations," 2023. https://github.com/pnnl/neuromancer.

[12] E. R. Vieira, A. Sivaramakrishnan, S. Tangirala, E. Granados, K. Mischaikow, and K. Bekris, "Morals: Analysis of high-dimensional robot controllers via topological tools in a latent space," in *ICRA*, 2024.

[13] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of backward reachable sets for continuous dynamic games," *IEEE TAC*, 2005.

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

[15] S. Sastry, "Nonlinear systems," *IAM*, 1999.

[16] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," *NeurIPS*, 2019.

[17] E. R. Vieira, E. Granados, A. Sivaramakrishnan, M. Gameiro, K. Mischaikow, and K. E. Bekris, "Morse graphs: Topological tools for analyzing the global dynamics of robot controllers," in *WAFR*, 2023.

[18] M. Janner, I. Mordatch, and S. Levine, "Gamma-models: Generative temporal difference learning for infinite-horizon prediction," *NeurIPS*, 2020.

[19] L. Schramm and A. Boularias, "Bellman diffusion models for offline reinforcement learning," in *18th EWRL*.

[20] C. Zhu, X. Wang, T. Han, S. S. Du, and A. Gupta, "Transferable reinforcement learning via generalized occupancy models," in *ICML 2024 Workshop*.

[21] S. M. Katz, A. L. Corso, C. A. Strong, and M. J. Kochenderfer, "Verification of image-based neural network controllers using generative models," *JAIS*, 2022.

[22] N. Botteghi, F. Califano, M. Poel, and C. Brune, "Trajectory generation, control, and safety with denoising diffusion probabilistic models," *arXiv preprint arXiv:2306.15512*, 2023.

[23] A. N. Angelopoulos, S. Bates, *et al.*, "Conformal prediction: A gentle introduction," *Foundations and trends® in machine learning*, 2023.

[24] A. Lin and S. Bansal, "Verification of neural reachable tubes via scenario optimization and conformal prediction," in *L4DC*, 2024.

[25] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *L4DC*, 2023.

[26] L. Lindemann, Y. Zhao, X. Yu, G. J. Pappas, and J. V. Deshmukh, "Formal verification and control with conformal prediction," *arXiv preprint arXiv:2409.00536*, 2024.

[27] F. Cairoli, L. Bortolussi, J. V. Deshmukh, L. Lindemann, and N. Paoletti, "Conformal predictive monitoring for multi-modal scenarios," in *ICRV*, 2025.

[28] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *NeurIPS*, 2017.

[29] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes," in *CDC*, 2016.

[30] X.-S. Wang, S. A. Moore, J. D. Turner, and B. P. Mann, "A model-free sampling method for basins of attraction using hybrid active learning (hal)," *Commun. Nonlinear Sci. Numer. Simul.*, 2022.

[31] Z. Kharazian, T. Lindgren, S. Magnússon, and H. Boström, "Copal: Conformal prediction for active learning with application to remaining useful life estimation in predictive maintenance," *PMLR*, 2024.

[32] M. D. Zhao, H. Admoni, R. Simmons, A. Ramdas, and A. Bajcsy, "Conformalized interactive imitation learning: Handling expert shift and intermittent feedback," in *ICLR*, 2024.

[33] S. Dasgupta, "Two faces of active learning," *TCS*, 2011.

[34] H. Ha, S. Gupta, S. Rana, and S. Venkatesh, "High dimensional level set estimation with bayesian neural network," in *AAAI*, 2021.

VII. APPENDIX

A. Related Work

Verification of Control Systems: Hamilton-Jacobi (HJ) reachability computes backward reachable sets [13], [14], but these grid-based solvers scale poorly with dimension. Neural approximations, such as DeepReach [2], extend to higher-dim. systems. Lyapunov stability theory certifies convergence via energy-like functions satisfying $\dot{V}(x) < 0$ [15]. Classical Lyapunov and Sum-of-Squares methods require the dynamics or the certificate to admit a functional form, but neural Lyapunov methods [6], [16], [5] remove this restriction. Topological methods decompose state spaces into Morse sets that capture qualitative dynamics without explicit certificates [17]. The MORALS framework [12] extends this to higher-dim. systems via a learned latent embedding. These topological methods assume the system stabilizes in the bounded domain, which is violated when failure trajectories diverge rather than converging to a failure attractor.

Generative Models for State Distribution Learning: An emerging paradigm uses generative models as predictive surrogates that learn the distribution of future states a system will visit. Training a generative model on discounted state occupancy measures [18] produces a reward-independent representation that can be queried under different success criteria without retraining. Subsequent work uses diffusion models to learn successor state measures with Bellman flow constraints [19] and policy-agnostic occupancy distributions from offline data [20]. Generative surrogates have also been applied directly to verification: conditional GANs that map system states to sensor observations enable formal reachability analysis of image-based neural network controllers [21]. Diffusion-based pipelines have been combined with control barrier functions for safe trajectory planning [22].

Conformal Prediction in Robotics: Conformal prediction (CP) [9], [23] provides a model-agnostic framework for uncertainty quantification with finite-sample statistical guarantees. Applications include calibrating forward reachable tubes [24], safe motion planning [25], and runtime monitoring of temporal logic specifications [26]. GenQPM [27] pairs diffusion models with conformalized quantile regression for runtime safety monitoring over a finite future horizon.

Adaptive Sampling and Active Learning: Adaptive sampling for `RoA` estimation has relied on Gaussian processes (GPs) [28], [29] or boundary-focused heuristics, such as

Hybrid Active Learning [30]. These methods improve data efficiency but rely on kernel assumptions or heuristic confidence measures that lack distribution-free guarantees. Some recent works also explore combining conformal prediction with active learning: CoPAL [31] and ConformalDAgger [32] do so in regression settings unrelated to control verification.

B. Method and Inference Details

1) Generative model and training: **Training data.** From the trajectory dataset $\mathcal{D}_{\text{traj}}$, trajectories are split into a training set $\mathcal{D}_{\text{train}}$ and a validation set $\mathcal{D}_{\text{val}}$. Training pairs are constructed by pairing every non-terminal state x_t along a trajectory ($t = 0, \ldots, T-1$) with its terminal state x_T. Since the closed-loop dynamics are deterministic, any state along a trajectory uniquely determines the same terminal outcome, so each trajectory of length T contributes T valid supervision pairs. The resulting collections $\mathcal{D}_{\text{pairs}}^{\text{train}} = \{(x_t, x_T)\}$ and $\mathcal{D}_{\text{pairs}}^{\text{val}} = \{(x_t, x_T, y)\}$ are constructed from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ respectively, where $\mathcal{D}_{\text{pairs}}^{\text{val}}$ additionally carries ground-truth labels y for threshold optimization. Crucially, $\mathcal{D}_{\text{pairs}}^{\text{train}}$ requires no success or failure labels at training time, decoupling the learned dynamics from any fixed success criterion.

Conditional flow matching loss. For a training pair (x_t, x_T), noise samples $z \sim p_0$ (standard Gaussian) and flow times $s \sim \text{Unif}[0, 1]$, the network is trained to match the straight-line velocity between z and x_T: $\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}\big\|v_\theta(z_s,\, s,\, x_t) - (x_T - z)\big\|^2$, where $z_s = (1-s)\, z + s\, x_T$.

Threshold optimization objective. Thresholds (λ, δ) are optimized over $\mathcal{D}_{\text{pairs}}^{\text{val}}$ by minimizing $\mathcal{L}_{\text{thresh}} = w \cdot R_{\text{err}}(\lambda, \delta) + (1-w) \cdot R_{\text{unc}}(\lambda, \delta)$, where R_{err} is the fraction of confident predictions ($\hat{y} \neq$ <Unc>) that disagree with the ground-truth label, R_{unc} is the fraction of all states classified $\hat{y} =$ <Unc>, and $w \in [0, 1]$ controls the trade-off. Both terms are computed only over states with $\hat{p}_{\text{inv}}(x) < \lambda - \delta$, excluding model-error states that are flagged <Unc> upstream regardless of the threshold values.

Training procedure. Algorithm 1 details the adaptive training loop.

Algorithm 1: Adaptive RoA Estimation (Training)

Input: Initial trajectories $\mathcal{D}_{\text{traj}}^{\text{init}}$, state space $\mathcal{X}$, rollout budget N, sampling ratio r_{unc}, trade-off weight w, iterations E
Output: Trained model p_θ, optimized thresholds (λ^*, δ^*)

Initialize $\mathcal{D}_{\text{pairs}}^{\text{train}}$, $\mathcal{D}_{\text{pairs}}^{\text{val}}$ from $\mathcal{D}_{\text{traj}}^{\text{init}}$;
for $e = 1$ **to** E **do**
  Train p_θ via $\mathcal{L}_{\text{CFM}}$ on $\mathcal{D}_{\text{pairs}}^{\text{train}}$;
  Optimize (λ^*, δ^*) on $\mathcal{D}_{\text{pairs}}^{\text{val}}$;
  $\mathcal{D}_{\text{traj}}^{\text{unif}} \leftarrow$ roll out $\lfloor N(1 - r_{\text{unc}}) \rfloor$ states drawn uniformly from $\mathcal{X}$;
  $\mathcal{D}_{\text{traj}}^{\text{unc}} \leftarrow$ roll out $\lceil N \cdot r_{\text{unc}} \rceil$ states with $\hat{y}(x) =$ <Unc>;
  Split $(\mathcal{D}_{\text{traj}}^{\text{unif}} \cup \mathcal{D}_{\text{traj}}^{\text{unc}})$ into $\mathcal{D}_{\text{new}}^{\text{train}}$ and $\mathcal{D}_{\text{new}}^{\text{val}}$;
  $\mathcal{D}_{\text{pairs}}^{\text{train}} \leftarrow \mathcal{D}_{\text{pairs}}^{\text{train}} \cup \{(x_t, x_T)$ from $\mathcal{D}_{\text{new}}^{\text{train}}\}$;
  $\mathcal{D}_{\text{pairs}}^{\text{val}} \leftarrow \mathcal{D}_{\text{pairs}}^{\text{val}} \cup \{(x_t, x_T, y)$ from $\mathcal{D}_{\text{new}}^{\text{val}}\}$;
end
Train p_θ on $\mathcal{D}_{\text{pairs}}^{\text{train}}$;
Optimize (λ^*, δ^*) on $\mathcal{D}_{\text{pairs}}^{\text{val}}$;

2) Conformal prediction: **Nonconformity score.** Using the optimized decision boundaries θ_{s}^* and θ_{f}^*, the nonconformity score $s(x, y)$ measures how inconsistent the model's probability estimates are with each candidate label $y \in \{$<Succ>, <Fail>, <Unc>$\}$:

$$s(x,y) = \begin{cases} \max\big(0,\ \theta_{\text{s}}^* - \hat{p}_{\text{s}}(x)\big) & y = \texttt{<Succ>}, \\ \max\big(0,\ \theta_{\text{f}}^* - \hat{p}_{\text{f}}(x)\big) & y = \texttt{<Fail>}, \\ \max\big(0,\ \hat{p}_{\text{s}}(x) - \theta_{\text{s}}^*,\ \hat{p}_{\text{f}}(x) - \theta_{\text{f}}^*\big) & y = \texttt{<Unc>}. \end{cases}$$

A score of zero means y is already consistent with the model's output: for $y \in \{$<Succ>, <Fail>$\}$ when the corresponding probability exceeds its decision boundary, and for $y =$ <Unc> when neither probability exceeds its boundary.

Calibration. A held-out calibration set $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n_{\text{cal}}}$ with ground-truth labels $y_i \in \{$<Succ>, <Fail>$\}$, disjoint from training and test data, determines the conformal quantile. Because the true labels are always <Succ> or <Fail>, states where the model produces mostly invalid endpoints would contribute large scores and inflate $\hat{q}$ to accommodate modeling artifacts rather than genuine boundary ambiguity. Points with $\hat{p}_{\text{inv}}(x_i) \geq \lambda^* - \delta^*$ (the model-error filter of Sec. III-B) are therefore removed; let $\mathcal{D}'_{\text{cal}} \subseteq \mathcal{D}_{\text{cal}}$ denote the filtered set. The quantile is $\hat{q} = \text{Quantile}\Big(\frac{\lceil (1-\alpha)(|\mathcal{D}'_{\text{cal}}|+1) \rceil}{|\mathcal{D}'_{\text{cal}}|},\ \{s(x_i, y_i)\}_{(x_i,y_i) \in \mathcal{D}'_{\text{cal}}}\Big)$.

Prediction sets. For a test state x, $\mathcal{C}(x) = \big\{y \in \{$<Succ>, <Fail>, <Unc>$\} : s(x, y) \leq \hat{q}\big\}$. Since $\hat{p}_{\text{s}} + \hat{p}_{\text{f}} \leq 1$ but $\theta_{\text{s}}^* + \theta_{\text{f}}^* > 1$, both probabilities cannot simultaneously exceed their boundaries, so any set containing both <Succ> and <Fail> must also contain <Unc>.

Coverage and certified classification. Theorem 1 applies only to states not removed by the model-error filter; filtered states are classified <Unc> directly without a coverage claim. Since the ground truth is always $y_{\text{test}} \in \{$<Succ>, <Fail>$\}$, prediction sets containing <Unc> alongside the true label (e.g., $\{$<Succ>, <Unc>$\}$ with $y =$ <Succ>) still count as coverage hits, while $\mathcal{C}(x) = \{$<Unc>$\}$ is always a miss. The score is computed from finite Monte Carlo estimates of $\hat{p}_{\text{s}}, \hat{p}_{\text{f}}$, so the guarantee holds for the implemented randomized predictor. A singleton $\{$<Succ>$\}$ or $\{$<Fail>$\}$ carries the $(1-\alpha)$ guarantee; a non-singleton set or $\{$<Unc>$\}$ is classified <Unc>.

C. Experimental Details

1) Benchmark systems: **Pendulum (2D):** an inverted pendulum with state $(\theta, \dot{\theta})$ under LQR control [17], stabilizing at the upright equilibrium $(0, 0)$; success and failure criteria are complementary ($\gamma_{\text{f}} = 1 - \gamma_{\text{s}}$). Evaluation: 50K grid states, 39% success. **Cartpole (4D):** state $(q, \theta, \dot{q}, \dot{\theta})$ under LQR control, stabilizing at the origin (pole balance and cart regulation). Evaluation: 115K grid states, 18% success. **Planar Quadrotor (6D):** state $(q_x, q_z, \phi, \dot{q}_x, \dot{q}_z, \dot{\phi})$ under a learned PPO RL policy, reaching a hover at unit height; tests a black-box controller with an irregular decision boundary. Evaluation: 500K grid states, 8% success. **Spatial Quadrotor**

(13D): state $(q, \dot{q}, q_{\text{quat}}, \omega) \in \mathbb{R}^{13}$ under LQR control, reaching hover at unit height with identity orientation. Grid evaluation is infeasible at this dimension, so the 1M-state evaluation set was built by discretizing the space, identifying occupied cells from held-out trajectories, and sampling equally from each; 22% success. The cartpole and both quadrotors, including controllers, use `safe-control-gym` [10]; for these three, failure is exceeding the state-space bounds.

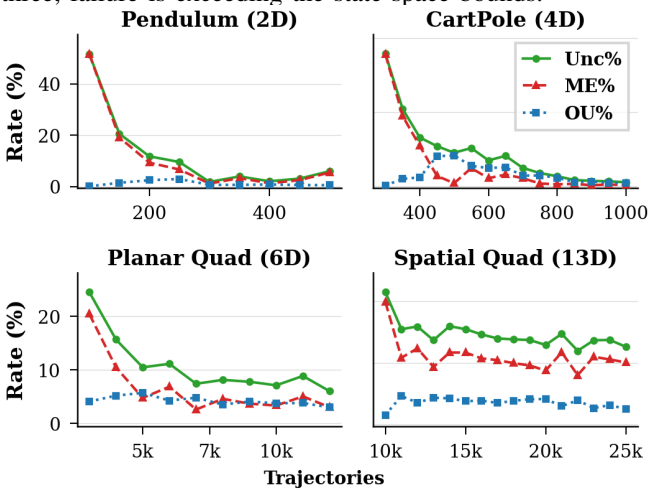


Fig. 5: Uncertain region decomposition across adaptive iterations. Green: total uncertain region rate. Red (dashed): invalid final state predictions classified (model-error) as `<Unc>`. Blue (dotted): true classification uncertainty (outcome uncertainty).

2) Uncertain region decomposition: Figure 5 reveals a three-phase pattern: a *model-error phase*, where invalid predictions dominate the uncertain region; a *transfer phase*, where the model begins producing valid final states in previously poorly-modeled regions, revealing multimodal outcome distributions that convert model error into outcome uncertainty; and a *resolution phase*, where both components decline together. The transition from the first to the second phase defines the adaptive divergence point: ∼450 trajectories on the cartpole (model error falling from 35.6% to 0.2%) and ∼5,000 on the planar quadrotor (∼3%). The 13D quadrotor remains in the transfer phase throughout the trajectory budget (model error ∼20%, outcome uncertainty ∼5–8%), explaining the r_{unc} plateau in Figure 6. This is consistent with the expectation that adaptive sampling requires a minimum dataset size before outperforming uniform sampling [30], with the divergence point scaling with dimensionality [33] as the true boundary occupies a vanishingly thin manifold in the 13D state space [34].

Variant	Planar Quad(6D) F1↑	Planar Quad(6D) Unc.↓	Spatial Quad(13D) F1↑	Spatial Quad(13D) Unc.↓
Non-Adaptive	.855	13.2%	.947	25.3%
Adaptive (fixed threshold)	.945	9.3%	.960	31.9%
Adaptive (optimized)	.933	6.0%	.953	25.2%

TABLE II. Adaptive (fixed) uses guided sampling with classification thresholds ($\lambda = 0.5, \delta = 0.1$); Adaptive (optimized) jointly optimizes thresholds (λ^*, δ^*) with the adaptive sampling loop.

3) Ablation studies: **Fixed vs. Optimized Thresholds:** Table II shows that with fixed thresholds (λ=0.5, δ=0.1), adaptive sampling improves F1 over the non-adaptive baseline on both systems but increases the uncertain region rate on the spatial quadrotor (25.3% to 31.9%). This occurs because adaptive sampling shifts the training distribution toward the boundary, and fixed thresholds do not account for this shift. Jointly optimizing thresholds at each iteration corrects this: uncertain region rates drop to 6.0% and 25.2% respectively, recovering a rate below the non-adaptive baseline on the spatial quadrotor.

Sensitivity to r_{unc}: Figure 6 varies r_{unc} from 0 to 1.0 across all four systems. On the cartpole and planar quadrotor, increasing r_{unc} monotonically reduces the uncertain region rate while F1 remains stable or improves. On the pendulum, moderate values ($r_{\text{unc}} = 0.75$) achieve the lowest uncertain region rate (1.3%), but $r_{\text{unc}} = 1.0$ reverses the gain (6.0%), suggesting that in this low-dimensional setting, fully concentrating sampling near the boundary deprives the model of coverage in the unambiguous success and failure regions.

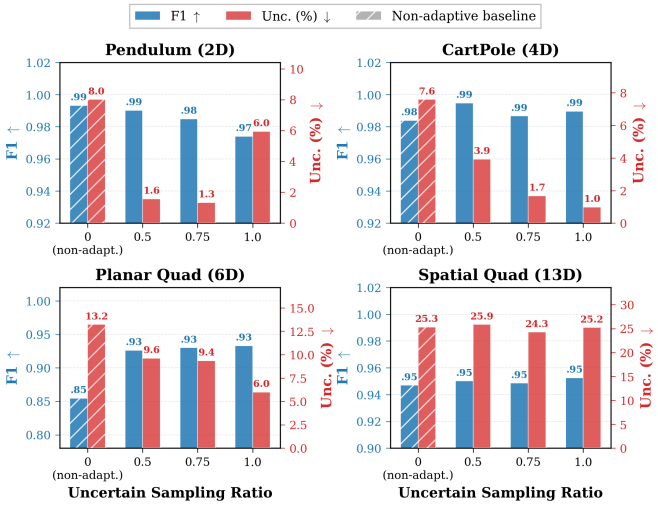


Fig. 6: Effect of r_{unc} on classification quality across systems. Blue bars: F1 score (left axis). Red bars: uncertain region rate (right axis). Hatched bars at $r_{\text{unc}} = 0$ denote the non-adaptive baseline.