

Safety Beyond the Training Data: Robust Out-of-Distribution MPC via Conformalized System Level Synthesis

Anutam Srinivasan¹

Antoine Leeman²

Glen Chou¹

ASRINIVASAN350@GATECH.EDU

ALEEMAN@ETHZ.CH

CHOU@GATECH.EDU

¹Georgia Institute of Technology. ²ETH Zurich.

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

Abstract

We present a novel framework for robust out-of-distribution planning and control using conformal prediction (CP) and system level synthesis (SLS), addressing the challenge of ensuring safety and robustness when using learned dynamics models beyond the training data distribution. We first derive high-confidence model error bounds using weighted CP with a learned, state-control-dependent covariance model. These bounds are integrated into an SLS-based robust nonlinear model predictive control (MPC) formulation, which performs constraint tightening over the prediction horizon via volume-optimized forward reachable sets. We provide theoretical guarantees on coverage and robustness under distributional drift, and analyze the impact of data density and trajectory tube size on prediction coverage. Empirically, we demonstrate our method on nonlinear systems of increasing complexity, including a 4D car and a 12D quadcopter, improving safety and robustness compared to fixed-bound and non-robust baselines, especially outside of the data distribution.

Keywords: safe learning-based control, conformal prediction, MPC, system level synthesis

1. Introduction

Provably-safe planning under unknown dynamics is crucial for real-world robotics. This poses difficulties for classical planners, which assume precise models; in practice, model error and uncertainty are unavoidable. Data-driven planners can learn dynamics from data but may exploit model inaccuracies, leading to unsafe behavior. This motivates uncertainty-constrained planners that quantify prediction error between true and learned dynamics to ensure robust, safe execution (Knuth et al., 2021, 2023; Wasielea et al., 2024). These methods estimate spatially-varying model error bounds to compute reachable tubes, which are used to constrain closed-loop behavior within a “trusted domain” where the model is accurate. This domain is typically selected as a bounded region around the training data, preventing generated plans from straying far from the data. However, in practice, data scarcity often forces robots to operate temporarily out of distribution (OOD), tolerating degraded performance while maintaining safety. Moreover, these planners are slow and do not exploit gradient information to efficiently optimize closed-loop behavior that actively reduces model uncertainty.

To address these gaps, we propose CP-SLS-MPC, a fast optimization-based robust feedback motion planner that uses the learned dynamics to jointly plan open-loop nominal trajectories and closed-loop tracking controllers via robust model predictive control (MPC). Our method uses weighted conformal prediction (CP) to bound the model error and system level synthesis (SLS) (Anderson et al., 2019) to compute reachable sets, which are used to constrain the true closed-loop system to be safe with high probability, enabling robust temporary OOD operation. CP-SLS-MPC updates the model error bounds with online data to encourage cautious behavior when OOD, such as staying further from obstacles. CP-SLS-MPC also uses gradient data from a model error predictor to efficiently guide plans toward low-error regions, enabling near-real-time replanning. Our contributions are:

- A fast optimization-based MPC method with learned dynamics ensuring finite-sample, probabilistic safety of the true system using state-control-dependent CP error bounds in the MPC.

- Theoretical analysis for high-probability safety and bounding the OOD coverage gap.
- Numerical validation on a simulated nonlinear 4D car and 12D quadcopter, improving safety and prediction accuracy even in OOD scenarios over uncertainty-unaware MPC baselines.

2. Related Work

Predictive Control Learning-based MPC unifies disturbance estimation with planning (Hewing et al., 2020), but typically lacks state-dependent error bounds and cannot steer plans towards low-error regions. Robust (Rawlings et al., 2017) and tube MPC (Mayne et al., 2005) ensure constraint satisfaction under disturbance, but in typical formulations, directly enforcing constraints on the closed-loop trajectory is nonconvex, leading to conservative overapproximations (Singh et al., 2018). In contrast, SLS (Anderson et al., 2019; Chen et al., 2024), or disturbance-feedback MPC (Goulart et al., 2006), provides a convex parameterization of closed-loop responses for linear time-varying (LTV) systems, enabling constraint satisfaction under disturbance (Bartos et al., 2025). SLS also extends to data-driven settings (Xue and Matni, 2021; Furieri et al., 2022) and nonlinear dynamics (Leeman et al., 2025b), where the problem is decomposed into an optimization over nominal trajectories and closed-loop LTV error dynamics. In this work, we inform SLS with state-dependent, CP-calibrated error bounds to enable safe planning toward low-error regions.

Planning with Learned Dynamics Many model-based reinforcement learning methods plan with learned models but lack safety guarantees (Moerland et al., 2023). Some methods estimate reachable tubes for safe exploration (Berkenkamp et al., 2016), but assume a controller is given. The most closely related method (Chou et al., 2021, 2022) safely plans with learned dynamics by constraining invariant tubes to lie in a “trusted domain” near training data. This demands dense data coverage; in contrast, we enable the robot to move beyond the training data while assuring probabilistic safety. Subsequent model error-aware planners (Suh et al., 2023) improve planning speed but lack formal guarantees. Also closely related is Marques and Berenson (2024), which bounds model error via CP but cannot extrapolate OOD as it uses *exchangeable* CP. Other methods aim to improve dynamics predictions in OOD scenarios (Lin et al., 2024) but do not focus on planning.

Conformal Prediction (CP) CP (Vovk et al., 2005) is a distribution-free method for constructing prediction regions from data from black-box models, making it well-suited for safe trajectory prediction and planning (Lindemann et al., 2023; Dixit et al., 2023; Sun et al., 2023; Muthali et al., 2023; Muenprasitvej et al., 2025). The most closely related methods are Contreras et al. (2025), which triggers a safe fallback policy in OOD scenarios for MPC, and Chee et al. (2023, 2024), which use WCP to address time-series correlations in MPC for tracking fixed trajectories. However, these methods do not use state-dependent uncertainty, and form prediction sets using a fixed spatially-invariant norm ball. Thus, the MPC cannot distinguish between state-space regions where the model is accurate and those where it is inaccurate, thereby degrading performance. In contrast, our method designs trajectories that are valid OOD and actively encourages uncertainty reduction, steering the MPC toward regions of lower model error, using a learned state-dependent error covariance matrix.

3. Preliminaries and Problem Statement

Let $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ be the true unknown deterministic nonlinear discrete-time dynamics,

$$x_{k+1} = f(x_k, u_k) = \hat{f}(x_k, u_k) + \varepsilon(x_k, u_k), \quad (1)$$

where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state space and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control space. Let $\hat{f} : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ denote an approximate learned dynamics model, where $\varepsilon : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ is the dynamics error. We model \hat{f}

with a neural network (NN), though our method is agnostic to model class. The goal is to reach a target set $\mathcal{X}_f \subseteq \mathcal{X}$ while satisfying pointwise-in-time state and control constraints $(x_k, u_k) \in \mathcal{F}$, where

$$\mathcal{F} := \{(x, u) \in \mathbb{R}^{n_x+n_u} \mid g_i(x, u) + b_i \leq 0, \quad \forall i \in \{1, \dots, n_c\}, \quad \text{for } g_i : \mathbb{R}^{n_x+n_u} \rightarrow \mathbb{R}\}. \quad (2)$$

Let $\mathcal{B}^m \subseteq \mathbb{R}^m$ be the unit 2-norm ball. Let the Minkowski sum be $A \oplus B := \{a+b \mid a \in A, b \in B\}$.

Conformal Prediction (CP) We use weighted CP to convert raw non-conformity scores, quantifying a mismatch between between $\hat{f}(x, u)$ and $f(x, u)$, into a calibrated threshold that yields $(1 - \alpha)$ coverage (Vovk et al., 2005). Let $\mathcal{D}_{\text{calib}} := (x_i, u_i, f(x_i, u_i))_{i=1}^n$ and choose a nonconformity score function $s : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$. Set $s_i := s(x_i, u_i, f(x_i, u_i))$. Calibration proceeds in three steps. We (i) form the weighted empirical distribution \hat{S} with weights, $\{\tilde{w}_1, \dots, \tilde{w}_n, \tilde{w}_{\text{test}}\} \subseteq [0, 1]^{n+1}$,

$$\hat{S} := \sum_{i=1}^n \tilde{w}_i \delta_{s_i} + \tilde{w}_{\text{test}} \delta_{+\infty}, \quad \text{s.t. } \tilde{w}_{\text{test}} + \sum_{i=1}^n \tilde{w}_i = 1, \quad (3)$$

where δ_{s_i} is the Dirac delta centered at s_i , (ii) calibrate the threshold $q_{1-\alpha} := Q_{1-\alpha}(\hat{S})$ with the $1 - \alpha$ quantile $Q_{1-\alpha}(\cdot)$, and (iii) define the conformal prediction set (CP set) at $(x_{\text{test}}, u_{\text{test}})$ as

$$\mathcal{C}(x_{\text{test}}, u_{\text{test}}; q_{1-\alpha}) := \{x \in \mathcal{X} \mid s(x_{\text{test}}, u_{\text{test}}, x) \leq q_{1-\alpha}\}. \quad (4)$$

Traditional (non-exchangeable) CP uses uniform weights, $\tilde{w}_i = \tilde{w}_{\text{test}} = \frac{1}{n+1}$, and assumes exchangeability, which may fail under spatio-temporal distribution shifts; WCP does not assume this. We denote $(s_1, s_2, \dots, s_{\text{test}}) \sim S^{\text{test}}$; moreover, let S^i be the joint distribution obtained by replacing s_i with the unknown test score, i.e., $(s_1, s_2, \dots, s_{i-1}, s_{\text{test}}, s_{i+1}, \dots, s_n) \sim S^i$. WCP guarantees

$$\Pr [f(x_{\text{test}}, u_{\text{test}}) \in \mathcal{C}(x_{\text{test}}, u_{\text{test}}; q_{1-\alpha})] \geq 1 - \alpha - \sum_{i=1}^n \tilde{w}_i d_{\text{TV}}(S^i, S^{\text{test}}), \quad (5)$$

where $d_{\text{TV}}(\cdot)$ is the total-variation distance and $\sum_{i=1}^n \tilde{w}_i d_{\text{TV}}(S^i, S^{\text{test}})$ quantifies the coverage gap due to distribution shift (it vanishes under exchangeability). In practice, the weights can be selected to trade off coverage gap size $\sum_i \tilde{w}_i d_{\text{TV}}(S^i, S^{\text{test}})$ and CP set size $|\mathcal{C}(x_{\text{test}}, u_{\text{test}}; q_{1-\alpha})|$, yielding uncertainty sets suitable for robust MPC with learned dynamics. WCP is essential for planning with learned dynamics to account for (1) distribution shifts between training and execution data, (2) state/control correlations during execution, and (3) visits to OOD regions of the state/control space.

SLS SLS (Anderson et al., 2019) is a robust control framework for uncertain LTV systems

$$x_{k+1} = A_k x_k + B_k u_k + E_k \xi_k \quad (6)$$

where $A_k \in \mathbb{R}^{n_x \times n_x}$, $B_k \in \mathbb{R}^{n_x \times n_u}$, and $E_k \in \mathbb{R}^{n_x \times n_x}$, and $\xi_k \in \mathcal{B}^{n_x}$ is a disturbance. SLS can be used to jointly optimize a nominal trajectory, a tracking controller, and a closed-loop reachable tube overapproximation computed by propagating worst-case disturbances over time. In particular, we denote a nominal length- T state trajectory $\mathbf{z} := \{z_1, \dots, z_T\} \in \mathbb{R}^{n_x T}$ and control trajectory $\mathbf{v} := \{v_1, \dots, v_T\} \in \mathbb{R}^{n_u T}$ satisfying the nominal LTV dynamics in (7). To track the nominal (\mathbf{z}, \mathbf{v}) , SLS defines a disturbance-feedback controller (8) and closed-loop state response (9), where $\Phi_{k,j}^u \in \mathbb{R}^{n_u \times n_x}$ and $\Phi_{k,j}^x \in \mathbb{R}^{n_x \times n_x}$ define the closed-loop response of (6) as an affine function of $\{\xi_k\}$,

$$z_{k+1} = A_k z_k + B_k v_k, \quad (7) \quad u_k = v_k + \sum_{j=1}^{k-1} \Phi_{k,j}^u \xi_j, \quad (8) \quad x_k = z_k + \sum_{j=1}^{k-1} \Phi_{k,j}^x \xi_j. \quad (9)$$

To ensure that (8)-(9) represent the closed-loop state/control response, $\Phi_{k,j}^x$ and $\Phi_{k,j}^u$ must satisfy

$$\Phi_{k+1,j}^x = A_k \Phi_{k,j}^x + B_k \Phi_{k,j}^u, \quad \Phi_{k+1,k}^x = E_k. \quad (10)$$

With SLS, we can guarantee that the closed-loop trajectories of (9) and (8) remain in the reachable tubes $\mathcal{R} := \{\mathcal{R}_k := (\mathcal{R}_k^x, \mathcal{R}_k^u)\}_{k=1}^T$, i.e., for all $k \in \{1, \dots, T\}$, $x_k \in \mathcal{R}_k^x$ and $u_k \in \mathcal{R}_k^u$, where

$$\mathcal{R}_k^x := (\bigoplus_{j=0}^{k-1} \Phi_{k,j}^x \mathcal{B}^{n_x}) \oplus \{z_k\}, \quad \mathcal{R}_k^u := (\bigoplus_{j=0}^{k-1} \Phi_{k,j}^u \mathcal{B}^{n_x}) \oplus \{v_k\}. \quad (11)$$

SLS has also been extended to control nonlinear dynamics (Leeman et al., 2025b; Zhan et al., 2025) with robust satisfaction of nonlinear constraints, both via optimized linearization error overbounds. For nonlinear dynamics \hat{f} of the form (1), we can obtain an LTV approximation using the Jacobian linearizations of \hat{f} around (\mathbf{z}, \mathbf{v}) : $A_k := \nabla_x \hat{f}(z_k, v_k)$, and $B_k := \nabla_u \hat{f}(z_k, v_k)$. To robustly satisfy nonlinear constraints of the form (2), we can enforce the tightened constraints (12), obtained by linearizing g_i , for all $i \in \{1, \dots, n_c\}$, around the nominal trajectory (7):

$$\sum_{j=0}^{k-1} \left\| \left[\nabla_x g_i(z_k, v_k) \Phi_{k,j}^x, \nabla_u g_i(z_k, v_k) \Phi_{k,j}^u \right] \right\|_2 + g_i(z_k, v_k) + b_i + g_i^{\text{lin}}(z_k, v_k) \leq 0, \quad (12)$$

where $g_i^{\text{lin}}(z_k, v_k)$ is a linearization error bound term (details are omitted for brevity; see (Zhan et al., 2025, Eq. (22))). This ensures that the closed-loop dynamics satisfy (2) for all $k \in \{1, \dots, T\}$.

Problem Statement We assume that we are given an offline dataset $\mathcal{D} := \{(x_i, u_i, f(x_i, u_i))\}_{i=1}^N$, where each (x_i, u_i) is drawn from a joint distribution Y . We partition \mathcal{D} into a *training dataset* $\mathcal{D}_{\text{train}} := \{(x_i, u_i, f(x_i, u_i))\}_{i=1}^{N_{\text{train}}}$, to train the learned dynamics \hat{f} (1) and a model error predictor $\Sigma : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x \times n_x}$, described in Sec. 4, and *calibration dataset* $\mathcal{D}_{\text{calib}} := \{(x_i, u_i, f(x_i, u_i))\}_{i=1}^{N_{\text{calib}}}$. Our goal is to 1) estimate calibrated error bounds on the learned dynamics \hat{f} that hold with high probability and to 2) use these bounds within a robust MPC framework, with a prediction horizon of T steps, to guarantee robust constraint satisfaction with high probability. Specifically, we solve:

Problem 1: Learning and Calibration. Using $\mathcal{D}_{\text{train}}$, learn an approximate dynamics model $\hat{f} : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$. Using \hat{f} and $\mathcal{D}_{\text{calib}}$, provide a calibrated high-confidence bound $\mathcal{E} : \mathcal{X} \times \mathcal{U} \rightarrow 2^{\mathcal{X}}$ on the dynamics error $\varepsilon : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x}$, such that $\Pr\left[\bigcap_{k=1}^T (\varepsilon(x_k, u_k) \in \mathcal{E}(z_k, v_k))\right] \geq 1 - \tilde{\alpha}$, for nominal state (z_k, v_k) and for all reachable state/controls $(x_k, u_k) \in \mathcal{R}_k := (\mathcal{R}_k^x, \mathcal{R}_k^u)$, as defined in (11). Here, $\alpha \in (0, 1)$ is user-specified and $\tilde{\alpha} \in [\alpha, 1)$ absorbs the coverage gap in (5).

Problem 2: Planning. Using the learned model \hat{f} and error bound \mathcal{E} of Prob. 1, solve a robust MPC problem for \hat{f} ensuring that the constraints (2) are satisfied for the true closed-loop dynamics (i.e., under f in (1)) with probability at least $1 - \tilde{\alpha}$, jointly over T prediction steps, for a single MPC solve, and with probability at least $1 - \hat{\alpha}$ for R MPC solves (i.e., $R - 1$ replanning steps), where $\hat{\alpha} \in (\tilde{\alpha}, 1)$.

4. Method

We outline our method, CP-SLS-MPC, which uses WCP for high-probability state-control-dependent model error bounding (Sec. 4.1) and integrates it into an uncertainty-constrained SLS-based robust MPC optimization (Sec. 4.2). Finally, we discuss practical implementation (Sec. 4.3).

4.1. Model Training and Model Error Bounding via Conformal Prediction

To tackle Prob. 1, we first train the dynamics \hat{f} (1) on the training dataset $\mathcal{D}_{\text{train}}$ by minimizing a mean-squared-error loss $\sum_{k=1}^{N_{\text{train}}} \|f(x_k, u_k) - \hat{f}(x_k, u_k)\|_2$. Then, for any nominal state $z \in \mathcal{X}$ and control input $v \in \mathcal{U}$, we seek to bound the error of the learned dynamics model $\|\hat{f}(z, v) - f(z, v)\|_2$. To encode spatially-varying error, we train a heuristic uncertainty model $\Sigma : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x \times n_x}$, an NN, with a multivariate Gaussian negative-log-likelihood (MGNLL) loss (Dasgupta and Hsu, 2007):

$$\mathcal{L}(\Sigma; f, \hat{f}, x, u) = \frac{1}{2} \left((f(x, u) - \hat{f}(x, u))^\top \Sigma(x, u)^{-1} (f(x, u) - \hat{f}(x, u)) + \ln(\det(\Sigma(x, u))) \right). \quad (13)$$

MGNLL is selected to train Σ to output an uncalibrated covariance matrix which locally approximates the *shape* and *magnitude* of \hat{f} 's error dispersion at a given state-control pair. WCP's calibration procedure reconciles the inexact gaussian assumption (of MGNLL) made of the error dispersion

by scaling the covariance matrix. To capture the model error at timestep k for a nominal (z_k, v_k) , we adapt the non-conformity score of [Messoudi et al. \(2022\)](#):

$$s_{i,k} := s_k(x_i, u_i, f(x_i, u_i)) = \sqrt{(f(x_i, u_i) - \hat{f}(x_i, u_i))^\top \Sigma(z_k, v_k)^{-1} (f(x_i, u_i) - \hat{f}(x_i, u_i))}. \quad (14)$$

Using (14), we form the $(1 - \alpha_k)$ -confidence CP set for some nominal (z_k, v_k) , $\mathcal{C}(z_k, v_k; q_{1-\alpha_k})$, as

$$\mathcal{C}(z_k, v_k; q_{1-\alpha_k}) := \{x \in \mathcal{X} \mid \sqrt{(x - \hat{f}(z_k, v_k))^\top \Sigma(z_k, v_k)^{-1} (x - \hat{f}(z_k, v_k))} \leq q_{1-\alpha_k}(z_k, v_k)\}, \quad (15)$$

defining $q_{1-\alpha_k}(z_k, v_k) := Q_{1-\alpha_k}(\hat{S}_k)$, where $\hat{S}_k := \sum_{i=1}^n \tilde{w}_i^k \delta_{s_{i,k}} + \tilde{w}_{\text{test}}^k \delta_{+\infty}$ is formed using a set of normalized weights $\tilde{W}^k := \{\tilde{w}_1^k, \dots, \tilde{w}_{N_{\text{calib}}}^k, \tilde{w}_{\text{test}}^k\} \subseteq [0, 1]^{N_{\text{calib}}+1}$ tailored to the nominal state/control pair (z_k, v_k) . Intuitively, $\mathcal{C}(z_k, v_k; q_{1-\alpha_k})$ is an ellipsoid centered at the predicted next state $\hat{f}(z_k, v_k)$, with a radius given by the quantile of the weighted non-conformity scores from (14).

To construct the normalized weights \tilde{W}^k tailored for a nominal state/control pair (z_k, v_k) , we first define a set of unnormalized weights $W^k = \{w_1^k, \dots, w_{N_{\text{calib}}}^k, w_{\text{test}}^k = 1\} \subseteq [0, 1]^{N_{\text{calib}}+1}$, where

$$w_i^k = \rho^{\| (z_k, v_k) - (x_i, u_i) \|_2}, \quad (16)$$

where $\rho \in (0, 1]$, for all $(x_i, u_i) \in \mathcal{D}_{\text{calib}}$. The form of (16) encodes the assumption that model error is state/control-dependent and changes smoothly, such that nearby pairs have similar errors. We normalize the weights w^k as $\tilde{w}_i^k = \frac{w_i^k}{1 + \sum_{j=1}^n w_j^k}$ to be compatible with the WCP framework (Sec. 3).

Denote $(s_{1,k}, s_{2,k}, \dots, s_{i-1,k}, s_{k,k}, s_{i+1,k}, \dots, s_{N_{\text{calib}},k}, s_{i,k}) \sim S^{i,k}$ and $(s_{1,k}, \dots, s_{N_{\text{calib}},k}, s_{k,k}) \sim S^{k,k}$ (i.e., $S^{i,k}$ exchanges $s_{k,k}$ with $s_{i,k}$), where $s_{k,k} = s_k(z_k, v_k, f(z_k, v_k))$ is generally unknown as $f(z_k, v_k)$ may not be in $\mathcal{D}_{\text{calib}}$ for some candidate (z_k, v_k) . Via the WCP framework, (5) ensures

$$\Pr[f(z_k, v_k) \in \mathcal{C}(z_k, v_k; q_{1-\alpha_k})] \geq 1 - \alpha_k - \sum_{i=1}^n \tilde{w}_i^k d_{\text{TV}}(S^{i,k}, S^{k,k}). \quad (17)$$

Starting from (17), we see that the dynamics error is in an origin centered ellipsoid, $\varepsilon(z_k, v_k) = (f(z_k, v_k) - \hat{f}(z_k, v_k)) \in \mathcal{C}(z_k, v_k; q_{1-\alpha_k}) \oplus \{-\hat{f}(z_k, v_k)\}$, implying that it satisfies $\Pr[\varepsilon(z_k, v_k) \in \mathcal{E}(z_k, v_k)] \geq 1 - \alpha_k - \sum_{i=1}^n \tilde{w}_i^k d_{\text{TV}}(S^{i,k}, S^{k,k})$, where we set $\mathcal{E}(z_k, v_k) := \mathcal{C}(z_k, v_k; q_{1-\alpha_k}) \oplus \{-\hat{f}(z_k, v_k)\}$. In essence, by conformalizing the heuristic uncertainty model Σ , we obtain a calibrated spatially-dependent model error bound $\mathcal{E}(z_k, v_k)$ that holds with high probability. Equivalently, using the Cholesky factor $L(z_k, v_k)$ of $\Sigma(z_k, v_k)$, i.e., $\Sigma(z_k, v_k) = L(z_k, v_k)L(z_k, v_k)^\top$, (15)-(17), and selecting a miscoverage rate $\alpha_k \in (0, 1)$ tailored to timestep k , we can state that

$$\Pr[\varepsilon(z_k, v_k) \in \underbrace{V(z_k, v_k) \mathcal{B}^{n_x}}_{:= \mathcal{E}(z_k, v_k)}] \geq 1 - \alpha_k - \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i^k d_{\text{TV}}(S^{i,k}, S^{k,k}), \quad (18)$$

$$\text{where } V(z_k, v_k) := q_{1-\alpha_k}(z_k, v_k) L(z_k, v_k). \quad (19)$$

While (18)-(19) bounds model error at (z_k, v_k) , the system may be perturbed during execution within the reachable tube \mathcal{R}_k . Thus, at every timestep k , we must quantify the coverage of $\mathcal{E}(z_k, v_k)$ from any $(x_k, u_k) \in \mathcal{R}_k$. This can be achieved by inflating the coverage gap with an additional term, yielding $\Pr[\varepsilon(x_k, u_k) \in \mathcal{E}(z_k, v_k)] \geq 1 - \alpha_k - 2 \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i^k d_{\text{TV}}(S_{i,k}, S_{k,k}) - \gamma(\mathcal{R}_k)$, where $\gamma(\mathcal{R}_k)$ is defined in Thm. 4. Here, $S_{i,k}$ and $S_{k,k}$ are distributions such that $s_{i,k} \sim S_{i,k}$, $s_{k,k} \sim S_{k,k}$. Specifically, Thm. 1 derives the probability that $\varepsilon(x_k, u_k) \in \mathcal{E}(z_k, v_k)$ for all $(x_k, u_k) \in \mathcal{R}_k$, for all $k \in \{1, \dots, T\}$, providing our solution for Prob. 1 (proofs in App. A):

Theorem 1 Given a length- T trajectory of the true dynamics f (1) $\tau_f := \{(x_k, u_k)\}_{k=1}^T$ such that $(x_k, u_k) \in \mathcal{X} \times \mathcal{U}$ for all $k \in \{1, \dots, T\}$, desired miscoverage $\{\alpha_k\}_{k=1}^T$, total miscoverage $\sigma_k = \alpha_k + 2 \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i^k d_{TV}(S_{i,k}, S_{k,k}) + \gamma(\mathcal{R}_k)$, and independence between $\mathcal{D}_{\text{calib}}$ and (x_k, u_k) , we have:

$$\Pr[\bigcap_{k=1}^T (\varepsilon(x_k, u_k) \in \mathcal{E}(z_k, v_k))] \geq 1 - \sum_{k=1}^T \sigma_k := 1 - \tilde{\alpha}. \quad (20)$$

Overall, at every timestep, we predict an ellipsoid $\mathcal{E}(z_k, v_k)$ containing the one-step error $\varepsilon(x_k, u_k)$ with high probability. Next, we will use this data-driven, spatially-dependent bound $\mathcal{E}(z_k, v_k)$ to inform SLS-based robust MPC, shrinking tubes where data is dense and expanding them if OOD.

4.2. Informing SLS-based Planning with CP Model Error Bounds

Using our conformed model error bounds, we construct an uncertain nonlinear system $z_{k+1} = \hat{f}(z_k, v_k) + V(z_k, v_k)\xi_k$, where $\xi_k \in \mathcal{B}^{n_x}$ and $V(z_k, v_k)\xi_k$ represent the unknown, bounded model error at timestep k . Let Φ^x and Φ^u collect all $\Phi_{k,j}^x$ and $\Phi_{k,j}^u$. Then we adapt Leeman et al. (2025b) to write a nonlinear program (NLP) in (21) to optimize for $\mathbf{z} = \{z_1, \dots, z_T\}$, $\mathbf{v} = \{v_1, \dots, v_{T-1}\}$, Φ^x , and Φ^u , given an initial state $\bar{x}_0 \in \mathcal{X}$:

$$\begin{aligned} \min_{\Phi^x, \Phi^u, \mathbf{z}, \mathbf{v}} \quad & J(\mathbf{z}, \mathbf{v}) + \tilde{H}_0(\Phi^x, \Phi^u) + J_{\mathcal{X}_f}(\mathbf{z}, \mathbf{v}), & (21a) \\ \text{s.t.} \quad & z_{k+1} = \hat{f}(z_k, v_k), \quad z_1 = \bar{x}_0, \quad \forall k = 1, \dots, T-1, & (21b) \\ & \Phi_{k+1,j}^x = A_k \Phi_{k,j}^x + B_k \Phi_{k,j}^u, \quad \forall j = 1, \dots, T-1, \quad \forall k = j+1, \dots, T-1, & (21c) \\ & \Phi_{j+1,j}^x = V(z_j, v_j), \quad \forall j = 1, \dots, T-1, & (21d) \\ & \sum_{j=1}^k \|\nabla_x g_i(z_k, v_k) \Phi_{k,j}^x, \nabla_u g_i(z_k, v_k) \Phi_{k,j}^u\|_2 + g_i(z_k, v_k) + b_i + g_i^{\text{lin}}(z_k, v_k) \leq 0, & (21e) \\ & \forall i = 1, \dots, n_c, \quad \forall k = 1, \dots, T. \end{aligned}$$

Here, (21a) penalizes a cost $J(\mathbf{z}, \mathbf{v})$ on the nominal trajectory, $J_{\mathcal{X}_f}(\mathbf{z}, \mathbf{v})$ penalizes reaching the terminal set \mathcal{X}_f , and $\tilde{H}_0(\Phi^x, \Phi^u)$ penalizes the size of the reachable tubes (see (49), (52), (53)). Since \hat{f} is more accurate in the training domain, we modify $J(\mathbf{z}, \mathbf{v})$ to include an active error reduction cost $J_{\text{active}}(\mathbf{z}, \mathbf{v})$ (see App. D.4) minimizing the distance of (\mathbf{z}, \mathbf{v}) to in-distribution calibration points. Constraint (21b) ensures nominal dynamic feasibility, while (21c) and (21d) enforce SLS conditions (10) for the LTV approximation obtained by linearizing \hat{f} around (\mathbf{z}, \mathbf{v}) , with model error bound $V(z_j, v_j)$ from (19). Constraint (21e) enforces robust state-control satisfaction using bounds on the closed-loop response derived by informing SLS with CP. Since SLS guarantees an overapproximation of the true reachable tube, (1) the closed-loop trajectory executing $(\mathbf{z}, \mathbf{v}, \Phi^x, \Phi^u)$ over the T -step horizon satisfies (2) with high probability, and (2) safety is maintained with high, though diminishing, probability over iterative replanning steps. The following result addresses Prob. 2:

Theorem 2 Define the forecasted nominal trajectory $\tau_{\hat{f}}^{(q)} := \{(z_k^{(q)}, v_k^{(q)})\}_{k=1}^T$ as the T -timestep future nominal trajectory predicted by the learned dynamics \hat{f} at the q th MPC solve (i.e., the $(q-1)$ st replanning step) in (21b). (a) Then the true closed-loop dynamics (1) under the controller defined by (21) and (8) satisfies $\Pr[\bigcap_{k=1}^T ((x_k^{(q)}, u_k^{(q)}) \in \mathcal{F})] \geq 1 - \sum_{k=1}^T \sigma_k^{(q)}$, where $\{(x_k^{(q)}, u_k^{(q)})\}_{k=1}^T$ is the realized state-control trajectory if executed for the full T timesteps, and (b) defining $\sigma_1^{(q)}$ as the miscoverage rate for the first timestep at MPC solve iteration q , we have that by executing (21) over R MPC solves (i.e., $R-1$ re-planning steps), $\Pr[\bigcap_{q=1}^R ((x_1^{(q)}, u_1^{(q)}) \in \mathcal{F})] \geq 1 - \sum_{q=1}^R \sigma_1^{(q)} := 1 - \hat{\alpha}$.

Overall, by solving (21), we reduce model error by 1) minimizing reachable tube volume and 2) penalizing distance to data. Coupling planning with CP-informed reachability ensures high-probability safety for the true dynamics. Finally, refining the CP error set \mathcal{C} with online data (Alg. 1) reduces the coverage gap, improving empirical model error coverage.

4.3. Practical Implementation via SCP

To efficiently solve the NLP (21), we propose CP-SLS-MPC (Alg. 1). CP-SLS-MPC uses sequential convex programming (SCP) (Malyuta et al., 2022; Messerer et al., 2021). Each SCP iteration solves a second-order cone program (SOCP) (Alg. 1, line 6), obtained by linearizing \hat{f} around the current nominal trajectory (\mathbf{z}, \mathbf{v}) and convexifying the constraint tightenings in (21e), yielding an update $(\mathbf{z}_{\text{lin}}^*, \mathbf{v}_{\text{lin}}^*)$ to the nominal trajectory. To solve this SOCP efficiently, we use the solver in Leeman et al. (2024), which iterates between solving a QP for (\mathbf{z}, \mathbf{v}) and Riccati recursions for (Φ^x, Φ^u) (line 7). For efficiency, we apply a real-time iteration scheme (Gros et al., 2020), limiting SCP to one iteration per step (Leeman et al., 2025a), and omit linearization error (Zhan et al., 2025), which is small in practice (Sec. 6). At each step, we execute the first control input v_1 (line 9), shift the previous solution for initialization (line 5), and re-solve via SCP. Moreover, we improve OOD robustness by updating the error set \mathcal{V} with online observations (line 10). Augmenting the error set updates the quantile $q_{1-\alpha_k}(z_k, v_k)$, which in turn alters constraint (21d) via (19), reducing the coverage gap and increasing the likelihood that the closed-loop dynamics remain within the computed reachable tubes.

Algorithm 1 CP-SLS-MPC

```

1: procedure CP-SLS-MPC( $\mathcal{D}_{\text{calib}}$ , horizon  $T$ ,  $\mathcal{F}$ ,  $\bar{x}_0$ , terminal set  $\mathcal{X}_f$ )
2:    $t \leftarrow 1$  and  $(\mathbf{z}, \mathbf{v}) \leftarrow$  Nominal Solution ▷ Initialize the problem with start state  $\bar{x}_0$ 
3:    $\mathcal{V} \leftarrow \{(x_i, u_i, f(x_i, u_i) - \hat{f}(x_i, u_i))\}_{i=1}^{N_{\text{calib}}}$ ,  $\forall (x_i, u_i, f(x_i, u_i)) \in \mathcal{D}_{\text{calib}}$  ▷ Initialize error set  $\mathcal{V}$ 
4:   while  $x_t \notin \mathcal{X}_f$  do ▷  $\mathcal{X}_f$  is included as a terminal cost (52), App D.3
5:      $\bar{x}_0 \leftarrow x_t$ ; initialize  $(\mathbf{z}, \mathbf{v})$  with shifted previous solution ▷ Initialize with the current state
6:     Linearize (21) around  $(\mathbf{z}, \mathbf{v})$ ; compute  $V(z_k, v_k)$  via (19),  $\mathcal{V}$ ; form SOCP
7:      $(\mathbf{z}_{\text{lin}}^*, \mathbf{v}_{\text{lin}}^*), \Phi^x, \Phi^u \leftarrow$  SOCP solution ▷ Single iteration of Leeman et al. (2024).
8:     Update  $(\mathbf{z}, \mathbf{v}) \leftarrow (\mathbf{z}, \mathbf{v}) + (\mathbf{z}_{\text{lin}}^*, \mathbf{v}_{\text{lin}}^*)$ .
9:      $u_t \leftarrow v_1$ ;  $x_{t+1} \leftarrow f(x_t, u_t)$  ▷ Closed loop control; state update
10:     $\mathcal{V} \leftarrow \mathcal{V} \cup \{(x_t, u_t, x_{t+1} - \hat{f}(x_t, u_t))\}$ ;  $t \leftarrow t + 1$  ▷ Online data update
11: end procedure

```

5. Theoretical Analysis: Bounding the Coverage Gap

To bound the coverage gap, we assume non-conformity scores from the calibration set $\mathcal{D}_{\text{calib}}$ are independent of the score at the nominal point (z_k, v_k) , though they may come from different distributions. We also assume a Lipschitz-type bound on the distribution drift between $S_{i,k}$ and $S_{k,k}$, i.e.,

$$d_{\text{TV}}(S_{i,k}, S_{k,k}) \leq \epsilon \| [z_k^\top, v_k^\top] - [x_i^\top, u_i^\top] \|_2, \quad (22)$$

for $\epsilon \in \mathbb{R}, \epsilon > 0$, where $s_i \sim S_i$ (14) and \tilde{S} is the non-conformity score distribution of (z, v) . (22) is a practical assumption, stating that the dynamics error distribution shifts gradually over the state-control space. The next theorem shows an overbound on the calibration error $d_{\text{TV}}(S_{i,k}, S_{k,k})$.

Theorem 3 *The Coverage Gap = $\sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i d_{\text{TV}}(S_{i,k}, S_{k,k})$ of our conformal predictor satisfies*

$$\text{Coverage Gap} \leq \underbrace{\sum_{i=1}^{N_{\text{calib}}} \left(\frac{\rho^{d_i}}{1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_j}} \right)}_{\text{Tight Bound}} \cdot 2\epsilon \cdot d_i \leq \underbrace{2\epsilon \left[\frac{d_1}{1 - \rho^{d_{\min}}} + \frac{d_{\max} \rho^{d_{\min}}}{(1 - \rho^{d_{\min}})^2} \right]}_{\text{Interpretable Bound}}, \quad (23)$$

where we sort the distance to points in the calibration set such that $d_i = \| [z_k^\top, v_k^\top] - [x_i^\top, u_i^\top] \|_2$, $d_1 \leq d_2 \leq \dots \leq d_n$, $d_{\min} = \min_{1 \leq i \leq n-1} \{d_{i+1} - d_i\}$, and $d_{\max} = \max_{1 \leq i \leq n-1} \{d_{i+1} - d_i\}$.

Thm. 3 (interpretable bound) shows that the gap depends on state-control calibration point density. Greater point density decreases d_{\min} , increasing the gap; reducing ρ mitigates this, but doing so can

cause the $(1 - \alpha)$ -quantile to diverge. The term $\frac{d_1}{1 - \rho^{d_{\min}}}$ shows that being closer to the training data reduces the gap, as d_1 is the minimal distance between $\mathcal{D}_{\text{calib}}$ and (z_k, v_k) . Further discussion of the intuition and validation of the empirical bounds is in App. E. Online augmentation of \mathcal{V} may violate independence (due to time-correlation), but for large N_{calib} , Thm. 3 suffices. As SLS requires model error to hold for all $(x, u) \in \mathcal{R}_k$, we also provide a coverage bound valid over the reachable tube:

Theorem 4 *Under the same assumption of Theorem 3, we have that for all $(x, u) \in \mathcal{R}_k$, the tube around the nominal point (z_k, v_k) , $\Pr[\varepsilon(x, u) \in \mathcal{E}(z_k, v_k)] \geq 1 - \alpha_k - 2 \sum_{i=1}^n \tilde{w}_i d_{TV}(S_{i,k}, S_{k,k}) - \gamma(\mathcal{R}_k)$, for $\gamma(\mathcal{R}_k) := 2\hat{\epsilon}M(\mathcal{R}_k)$, where $M(\mathcal{R}_k)$ is the maximum axis length of tube \mathcal{R}_k , and $\hat{\epsilon} \leq \epsilon$ is chosen such that (22) holds for all $(x, u) \in \mathcal{R}_k$.*

Using Thm. 4, we find that the coverage guarantees remain close to the desired coverage level α , assuming a gradual spatial-drift of the error distribution. Furthermore, the optimization problem (21) seeks to minimize the tube volume in the cost function, thereby minimizing $\gamma(\mathcal{R}_k)$.

6. Results

We evaluate our method on robust planning with learned 4D Dubins’ (see App. B.1) and 12D quadcopter dynamics (see App. B.2). Comparisons (see App. C for details) include naïve, uncalibrated MPC on the learned dynamics \hat{f} without robust constraint tightenings, denoted as vanilla MPC (V-MPC), and a fixed hyper-ball variant of our method (CP-Ball), where $\Sigma(z_k, v_k)$ in (14) is replaced with I_{n_x} , highlighting the benefit of state-dependent CP-Ellipsoid (our method) uncertainty sets. CP-Ball is an adaptation of Chee et al. (2024), using the L2-norm of errors as the non-conformity score. Direct use of Chee et al. (2024) was not feasible, as it only collects online data points, requiring horizon lengths longer than 150 points for successful deployment. Training details, model architectures, and cost functions are in App. D. Our implementation uses Acados (Verschueren et al., 2020) and L4CasADi (Salzmann et al., 2024) and is run on an M4 Pro MacBook (12 cores, 24 GB RAM). We report average MPC step time, prediction error $|f(x, u) - \hat{f}(x, u)|$, and minimum obstacle distance. All trials use horizon $T = 15$ with $\alpha_k = 0.1/15$.

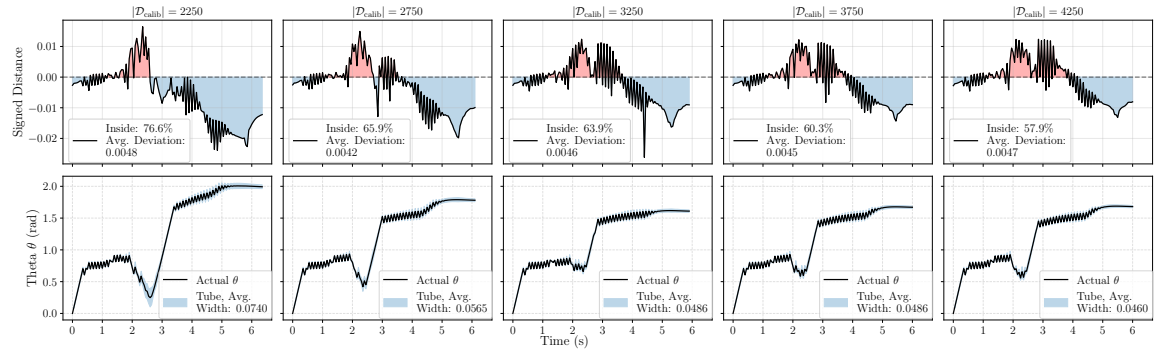


Figure 1: **OOD Car.** (Top) Minimum distance between prediction error and the ellipsoid edge is plotted for varying D_{calib} with a fixed start and goal; negative values (blue) indicate points inside the ellipsoid, i.e., valid coverage. (Bottom) Corresponding one-step θ tubes are shown. Smaller D_{calib} keeps prediction errors inside the ellipsoid more often and results in larger tubes.

Dubins’: In Domain (ID). First, we validate that our approach maintains performance compared to baselines in ID scenarios without a major loss of computational efficiency. We used the car dynamics in (39) and selected start and goal states training distribution (equivalently the feasible set for the ID results). We define $\mathcal{F} = \{(x, u) \mid p_x \in [0, 5], p_y \in [-5, 5], \|[p_x, p_y]^T - [2.5, 0]\|_2 \geq 1\}$,

with bounds on p_x and p_y and a 1-m radius obstacle centered at $(2.5, 0)$. Across 10 runs, all three approaches (V-MPC, CP-Ball, and CP-Ellipsoid) safely reached the goal while avoiding the obstacle; however, V-MPC does not perform constraint tightening and lacks safety guarantees. In Table 1, CP-Ball and CP-Ellipsoid have larger obstacle proximity—a byproduct of the constraint tightening (Table 1). Finally, the one-step prediction error for \hat{f} remains inside the ball and ellipsoid over $99.33\% = 100(1 - \frac{0.1}{15})\%$ during execution—validating the coverage guarantees (Fig. 7 in App. F). In the training domain, we expect V-MPC to have similar success rates as our method since \hat{f} is accurate; this changes in OOD settings. We find that our approach avoids the obstacle ID, without a major slowdown from constraint tightening computations.

Dubins’: OOD. To evaluate OOD performance, we train a dynamics and uncertainty model where $p_y \in [-12, -6] \cup [6, 12]$ and modify the dynamics to steer the car’s angle toward the obstacle for $p_y \in [-6, 6]$ (see App. B.1), while placing the feasible set of the rollout (same as the ID experiment) OOD. In this setting, we vary the calibration set size from 2,250 to 4,250 points (Fig. 1) for CP-Ellipsoid, ensuring the start and goal are outside the training domain. As shown in Fig. 1, the one-step prediction errors initially lie outside the predicted ellipsoid but gradually return within it as online calibration points are collected. Moreover, with fewer initial calibration points, the online adaptation occurs more rapidly, resulting in a higher coverage of errors in the ellipsoid. The plot also shows that the one-step tubes in θ are wider for smaller $\mathcal{D}_{\text{calib}}$, indicating that data scarcity increases conservativeness. The goal is reliably reached, suggesting that the SLS tubes, when informed by CP, enable robust OOD performance.

Dubins’: Disjoint Training Domains (Friction Car).

To evaluate OOD obstacle avoidance under limited online data, we construct disjoint training regions (similar to the OOD setup) where $p_y \in [-5, -2] \cup [2, 5]$. The start and goal positions are randomly sampled from different regions, ensuring that all feasible trajectories must traverse OOD areas to reach the goal. In the OOD region, we apply the same steering modification as in the previous experiment, while adding p_y dependent acceleration dynamics (see App. B.1 for details). Across 10 runs, V-MPC failed twice, the CP-Ball failed once, and CP-Ellipsoid was consistently successful. These differences arise from reduced obstacle clearance (Table 1) observed in the CP-Ball and V-MPC cases. Fig. 3 shows that both CP-Ball and CP-Ellipsoid reach the goal, but CP-Ellipsoid maintains a greater distance from the obstacle, as evident from its forecasted trajectories and uncertainty tubes. Fig. 3 also illustrates that the state-dependent ellipsoids expand along the direction of motion, as increased forecasted y -motion leads to higher y -direction uncertainty. A result from an extra run is in App. F.

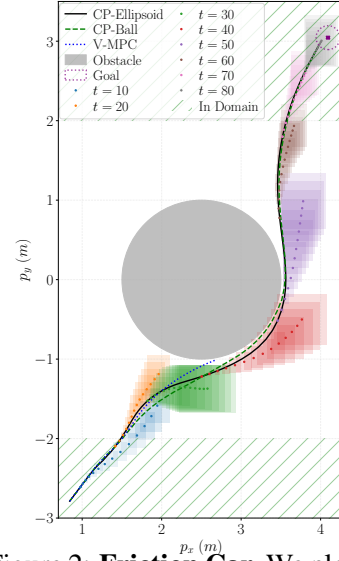


Figure 2: **Friction Car.** We plot all approaches and forecasted MPC steps & tubes for CP-Ellipsoid.

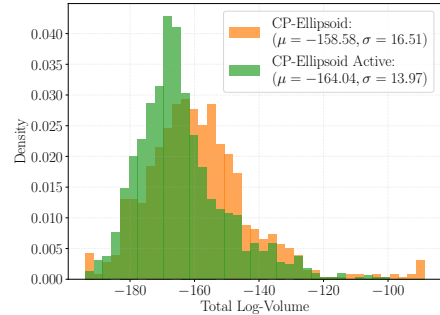
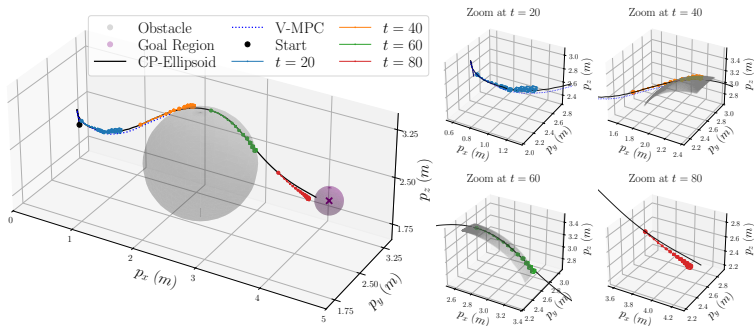


Figure 3: **Active Uncertainty.** We plot the histogram of summed log-volumes for each forecasted MPC plan. With active uncertainty reduction, the distribution shifts left, indicating less uncertainty at runtime.

Figure 4: **Quadcopter.** A rollout of the quadcopter with V-MPC and CP-Ellipsoid. CP-Ellipsoid maintains sufficient proximity to avoid crashing into the obstacle, while V-MPC crashes.



Dubins’: Active Uncertainty Reduction. Finally, to evaluate active uncertainty reduction, we trained \hat{f} and Σ without sampling data in a circular region centered at $(p_x, p_y) = (2.5, 0.0)$ with a radius of 1 meter, and introduced friction and attractive steering dynamics within this unseen region. Thus the feasible set contains an OOD region, where rollouts are permitted to traverse. Despite this, the active uncertainty reduction method achieves lower prediction error by planning a longer path that remains in the training domain (Table 2 App. F). Moreover, Fig. 3 shows the log-volume of the uncertainty tubes, showing that the active uncertainty cost ((54) in App. D.4) yields smaller tubes (Fig. 3, green) compared to the baseline without this cost (Fig. 3, orange).

Table 1: Mean and standard deviation of computation times and prediction errors and the mean minimum obstacle distance for vanilla-MPC (V-MPC), the CP-Ball (Ball), and CP-Ellipsoid (Ellipsoid).

| Model | Computation Time (ms) | | | Prediction Error (L_2) ($\times 10^{-2}$) | | | Min. Obstacle Dist. (m) | | |
|--------------|-----------------------|-----------------|-----------------|---|-----------------|-----------------|-------------------------|-------|-----------|
| | V-MPC | Ball | Ellipsoid | V-MPC | Ball | Ellipsoid | V-MPC | Ball | Ellipsoid |
| Car | 53.3 \pm 2.1 | 126.6 \pm 8.4 | 126.7 \pm 8.3 | 0.28 \pm 0.54 | 0.26 \pm 0.14 | 0.26 \pm 0.14 | 1.152 | 1.202 | 1.207 |
| Friction Car | 53.7 \pm 2.6 | 131.1 \pm 1.4 | 129.7 \pm 9.0 | 1.94 \pm 2.87 | 1.92 \pm 2.80 | 1.95 \pm 2.69 | 1.315 | 1.376 | 1.464 |
| Quadcopter | 63.1 \pm 4.6 | nan \pm nan | 222 \pm 20.9 | 3.89 \pm 5.49 | nan \pm nan | 3.30 \pm 4.17 | 0.804 | nan | 0.809 |

12D Quadcopter To evaluate the scalability of our method to higher-dimensional systems, we train a 12D quadcopter dynamics model. Using this model, we plan trajectories such that the spatial coordinates (p_x, p_y, p_z) remain outside a sphere centered at $(2.5, 2.5, 2.5)$ with a radius of 0.8 meters – leaving a 0.2-meter-thick shell around the sphere that is out of distribution and characterized by increased downward acceleration. To improve the performance of the uncertainty model, we modify its weighting by blending the covariance matrix with the identity matrix to reduce conservativeness, i.e., using $((1 - \tau)I + \tau\Sigma(z, v))$ instead of $\Sigma(z, v)$ in (15). With this approach, the CP-Ellipsoid method successfully planned 8 out of 10 trajectories (one failure was due to numerical instability, which did not result in collision), whereas V-MPC approach succeeded in only 4 out of 10 runs. While this demonstrates improved robustness over V-MPC, the failed case suggests faster tube expansion may be necessary to ensure consistent collision avoidance. Fig. 4 shows an example quadcopter trajectory, demonstrating that our CP-informed tubes maintain safe obstacle proximity.

7. Conclusion

In this work, we developed an efficient framework for robust out-of-distribution MPC with learned dynamics, using WCP and SLS. We provided theoretical guarantees ensuring that the rollout satisfy constraints with high probability, and introduced an uncertainty reduction cost function to encourage the system to remain within the training domain when possible. Through experiments on variants of the 4D Dubins car and 12D quadcopter, we demonstrated that the CP-SLS-MPC framework achieves robust performance even when using learned dynamics that generalize poorly OOD.

Acknowledgments

We thank Devesh Nath for insightful discussions and feedback on this work.

References

- James Anderson, John C Doyle, Steven H Low, and Nikolai Matni. System level synthesis. *Annual Reviews in Control*, 47:364–393, 2019.
- Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.
- Marcell Bartos, Alexandre Didier, Jerome Sieber, Johannes Köhler, and Melanie N Zeilinger. Stochastic mpc with online-optimized policies and closed-loop guarantees. *arXiv preprint arXiv:2502.06469*, 2025.
- Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *CDC*, pages 4661–4666, 2016.
- Kong Yao Chee, M. Ani Hsieh, and George J. Pappas. Uncertainty quantification for learning-based MPC using weighted conformal prediction. In *62nd IEEE Conference on Decision and Control, CDC 2023, Singapore, December 13-15, 2023*, pages 342–349. IEEE, 2023. doi: 10.1109/CDC49753.2023.10383587. URL <https://doi.org/10.1109/CDC49753.2023.10383587>.
- Kong Yao Chee, Thales C. Silva, M. Ani Hsieh, and George J. Pappas. Uncertainty quantification and robustification of model-based controllers using conformal prediction. In Alessandro Abate, Mark Cannon, Kostas Margellos, and Antonis Papachristodoulou, editors, *6th Annual Learning for Dynamics & Control Conference, 15-17 July 2024, University of Oxford, Oxford, UK*, volume 242 of *Proceedings of Machine Learning Research*, pages 528–540. PMLR, 2024. URL <https://proceedings.mlr.press/v242/chee24a.html>.
- Shaoru Chen, Victor M Preciado, Manfred Morari, and Nikolai Matni. Robust model predictive control with polytopic model uncertainty through system level synthesis. *Automatica*, 162:111431, 2024.
- Glen Chou, Necmiye Ozay, and Dmitry Berenson. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. In *2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, December 14-17, 2021*, pages 3576–3583. IEEE, 2021. doi: 10.1109/CDC45484.2021.9683354. URL <https://doi.org/10.1109/CDC45484.2021.9683354>.
- Glen Chou, Necmiye Ozay, and Dmitry Berenson. Safe output feedback motion planning from images via learned perception modules and contraction theory. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 349–367. Springer, 2022.
- Jose Leopoldo Contreras, Ola Shorinwa, and Mac Schwager. Safe, out-of-distribution-adaptive MPC with conformalized neural network ensembles. In Necmiye Ozay, Laura Balzano, Dimitra

- Panagou, and Alessandro Abate, editors, *7th Annual Learning for Dynamics & Control Conference, Ann Arbor, MI, USA, 4-6 June 2025*, volume 283 of *Proceedings of Machine Learning Research*, pages 194–207. PMLR, 2025. URL <https://proceedings.mlr.press/v283/contreras25a.html>.
- Sanjoy Dasgupta and Daniel Hsu. On-line estimation with the multivariate gaussian distribution. In *International Conference on Computational Learning Theory*, pages 278–292. Springer, 2007.
- Anushri Dixit, Lars Lindemann, Skylar X. Wei, Matthew Cleaveland, George J. Pappas, and Joel W. Burdick. Adaptive conformal prediction for motion planning among dynamic agents. In Nikolai Matni, Manfred Morari, and George J. Pappas, editors, *Learning for Dynamics and Control Conference, LADC 2023, 15-16 June 2023, Philadelphia, PA, USA*, volume 211 of *Proceedings of Machine Learning Research*, pages 300–314. PMLR, 2023. URL <https://proceedings.mlr.press/v211/dixit23a.html>.
- Luca Furieri, Clara Lucía Galimberti, and Giancarlo Ferrari-Trecate. Neural system level synthesis: Learning over all stabilizing policies for nonlinear systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2765–2770. IEEE, 2022.
- Paul J. Goulart, Eric C. Kerrigan, and Jan M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006. ISSN 0005-1098. doi: <https://doi.org/10.1016/j.automatica.2005.08.023>.
- Sébastien Gros, Mario Zanon, Rien Quirynen, Alberto Bemporad, and Moritz Diehl. From linear to nonlinear mpc: bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020. doi: [10.1080/00207179.2016.1222553](https://doi.org/10.1080/00207179.2016.1222553). URL <https://doi.org/10.1080/00207179.2016.1222553>.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. *The elements of statistical learning*, 2009.
- Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.
- Craig Knuth, Glen Chou, Necmiye Ozay, and Dmitry Berenson. Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants. *IEEE Robotics Autom. Lett.*, 6(3):5129–5136, 2021. doi: [10.1109/LRA.2021.3068889](https://doi.org/10.1109/LRA.2021.3068889). URL <https://doi.org/10.1109/LRA.2021.3068889>.
- Craig Knuth, Glen Chou, Jamie Reese, and Joseph L. Moore. Statistical safety and robustness guarantees for feedback motion planning of unknown underactuated stochastic systems. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 12700–12706. IEEE, 2023. doi: [10.1109/ICRA48891.2023.10161001](https://doi.org/10.1109/ICRA48891.2023.10161001). URL <https://doi.org/10.1109/ICRA48891.2023.10161001>.
- Antoine P Leeman, Johannes Kohler, Florian Messerer, Amon Lahr, Moritz Diehl, and Melanie N Zeilinger. Fast system level synthesis: Robust model predictive control using riccati recursions. *IFAC-PapersOnLine*, 58(18):173–180, 2024.

- Antoine P Leeman, Johannes Köhler, and Melanie N Zeilinger. Guaranteed robust nonlinear mpc via disturbance feedback. *arXiv preprint arXiv:2509.18760*, 2025a.
- Antoine P. Leeman, Johannes Köhler, Andrea Zanelli, Samir Bennani, and Melanie N. Zeilinger. Robust nonlinear optimal control via system level synthesis. *IEEE Trans. on Automatic Control*, 70(7):4780–4787, 2025b. doi: 10.1109/TAC.2025.3552482.
- Yating Lin, Glen Chou, and Dmitry Berenson. Improving out-of-distribution generalization of learned dynamics by learning pseudometrics and constraint manifolds. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16970–16976. IEEE, 2024.
- Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics Autom. Lett.*, 8(8):5116–5123, 2023. doi: 10.1109/LRA.2023.3292071. URL <https://doi.org/10.1109/LRA.2023.3292071>.
- Danylo Malyuta, Taylor P Reynolds, Michael Szmuk, Thomas Lew, Riccardo Bonalli, Marco Pavone, and Behçet Açıkmeşe. Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022.
- Luís Marques and Dmitry Berenson. Quantifying aleatoric and epistemic dynamics uncertainty via local conformal calibration, 2024. URL <https://arxiv.org/abs/2409.08249>.
- David Q. Mayne, Maria M. Seron, and Saša V. Raković. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 41(2):219–224, 2005.
- Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized gauss-newton methods. *ESAIM: Proceedings and Surveys*, 71:64–88, 2021.
- Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Conformal and Probabilistic Prediction with Applications*, pages 294–306. PMLR, 2022.
- Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118, 2023. doi: 10.1561/22000000086. URL <https://doi.org/10.1561/22000000086>.
- Kasidit Muenprasitvej, Ye Zhao, and Glen Chou. Probabilistically-safe bipedal navigation over uncertain terrain via conformal prediction and contraction analysis. *arXiv preprint arXiv:2510.07725*, 2025.
- Anish Muthali, Haotian Shen, Sampada Deglurkar, Michael H. Lim, Rebecca Roelofs, Aleksandra Faust, and Claire J. Tomlin. Multi-agent reachability calibration with conformal prediction. In *62nd IEEE Conference on Decision and Control, CDC 2023, Singapore, December 13-15, 2023*, pages 6596–6603. IEEE, 2023. doi: 10.1109/CDC49753.2023.10383723. URL <https://doi.org/10.1109/CDC49753.2023.10383723>.
- James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

- F. Sabatino. Quadrotor control: Modeling, nonlinear control design, and simulation. M.s. thesis, KTH, Stockholm, Sweden, 2015.
- Tim Salzmann, Jon Arrizabalaga, Joel Andersson, Marco Pavone, and Markus Ryll. Learning for casadi: Data-driven models in numerical optimization. In *Learning for Dynamics and Control Conference (LADC)*, 2024.
- Sumeet Singh, Mo Chen, Sylvia L Herbert, Claire J Tomlin, and Marco Pavone. Robust tracking with model mismatch for fast and safe planning: an sos optimization approach. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 545–564. Springer, 2018.
- H. J. Terry Suh, Glen Chou, Hongkai Dai, Lujie Yang, Abhishek Gupta, and Russ Tedrake. Fighting uncertainty with gradients: Offline reinforcement learning via diffusion score matching. In Jie Tan, Marc Toussaint, and Kouros Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 2878–2904. PMLR, 2023. URL <https://proceedings.mlr.press/v229/suh23a.html>.
- Jiankai Sun, Yiqi Jiang, Jianing Qiu, Parth Nobel, Mykel J. Kochenderfer, and Mac Schwager. Conformal prediction for uncertainty-aware planning with diffusion dynamics model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/fe318a2b6c699808019a456b706cd845-Abstract-Conference.html.
- Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados: a modular open-source framework for fast embedded optimal control, 2020. URL <https://arxiv.org/abs/1910.13753>.
- Vladimir Vovk, Alexander Gammernan, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.
- Simon Wasiela, Smail Ait Bouhsain, Marco Cognetti, Juan Cortés, and Thierry Simeon. Learning uncertainty tubes via recurrent neural networks for planning robust robot motions. In *ECAI 2024*, pages 4385–4392. IOS Press, 2024.
- Anton Xue and Nikolai Matni. Data-driven system level synthesis. In *Learning for dynamics and control*, pages 189–200. PMLR, 2021.
- Shuyu Zhan, Chih-Yuan Chiu, Antoine P Leeman, and Glen Chou. Robustly constrained dynamic games for uncertain nonlinear dynamics. *arXiv preprint arXiv:2509.16826*, 2025.

Appendix A. Proofs:

A.1. Proof of Theorem 3 and Corollary 5

Proof (Theorem 3) From Theorem 2 and Lemma 1 in Barber et al. (2023), the coverage gap for independent data is bounded by:

$$\text{Coverage Gap} \leq \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot d_{TV}(S^{i,k}, S^{k,k}) \leq \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot 2d_{TV}(S_{i,k}, S_{k,k}) \quad (24)$$

Given the normalized weights are $\tilde{w}_i = \frac{w_i}{1 + \sum_{j=1}^{N_{\text{calib}}} w_j}$, and $w_i = \rho^{d_i}$ for $d_i = \|[z_k^\top, v_k^\top] - [x_i^\top, u_i^\top]\|_2$

We can substitute into Equation 24, and prove the *Tight Bound* in Theorem 3:

$$\text{Coverage Gap} \leq \sum_{i=1}^{N_{\text{calib}}} \left(\frac{\rho^{d_i}}{1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_j}} \right) \cdot 2\epsilon \cdot d_i \quad (25)$$

Without loss of generality, we can rearrange the sum to create a decaying series $\{\rho^{d_i}\}_{i=1}^{N_{\text{calib}}}$, where d_i is increasing. While this series is not purely geometric, we can bound it by two geometric series. Let $d_{\min} = \min\{1 \leq i \leq N_{\text{calib}} - 1 \mid d_{i+1} - d_i\}$ and $d_{\max} = \max\{1 \leq i \leq N_{\text{calib}} - 1 \mid d_{i+1} - d_i\}$. Then, $\rho^{d_1} \rho^{d_{\max}(i-1)} \leq \rho^{d_i} \leq \rho^{d_1} \rho^{d_{\min}(i-1)}$. Thus, we can bound Equation 25,

$$\text{Coverage Gap} \leq \sum_{i=1}^{N_{\text{calib}}} \left(\frac{\rho^{d_i}}{1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_j}} \right) \cdot 2\epsilon \cdot d_i \leq 2\epsilon \left(\frac{\sum_{i=1}^{N_{\text{calib}}} \rho^{d_1} \rho^{d_{\min}(i-1)} \cdot (d_1 + d_{\max}(i-1))}{1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_1} \rho^{d_{\max}(j-1)}} \right)$$

Then observe the sum of the numerator,

$$N_{\text{sum}} = \sum_{i=1}^{N_{\text{calib}}} \rho^{d_1} \rho^{d_{\min}(i-1)} \cdot (d_1 + d_{\max}(i-1)) \quad (26)$$

$$= \rho^{d_1} \left[d_1 \left(\frac{1 - \rho^{N_{\text{calib}} \cdot d_{\min}}}{1 - \rho^{d_{\min}}} \right) + d_{\max} \left(\frac{\rho^{d_{\min}} - N_{\text{calib}} \rho^{N_{\text{calib}} \cdot d_{\min}} + (N_{\text{calib}} - 1) \rho^{(N_{\text{calib}} + 1) d_{\min}}}{(1 - \rho^{d_{\min}})^2} \right) \right] \quad (27)$$

Similarly, the sum the of the denominator,

$$D_{\text{sum}} = 1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_1} \rho^{d_{\max}(j-1)} \quad (28)$$

$$= 1 + \rho^{d_1} \left(\frac{1 - \rho^{N_{\text{calib}} \cdot d_{\max}}}{1 - \rho^{d_{\max}}} \right) \quad (29)$$

Then, the coverage gap is bounded by,

$$\text{Coverage Gap} \leq 2\epsilon \frac{N_{\text{sum}}}{D_{\text{sum}}} \quad (30)$$

$$\leq 2\epsilon \left[d_1 \left(\frac{1 - \rho^{N_{\text{calib}} \cdot d_{\min}}}{1 - \rho^{d_{\min}}} \right) + d_{\max} \left(\frac{\overbrace{\rho^{d_{\min}} - N_{\text{calib}} \rho^{N_{\text{calib}} \cdot d_{\min}} + (N_{\text{calib}} - 1) \rho^{(N_{\text{calib}} + 1) d_{\min}}}^{\leq 0}}{(1 - \rho^{d_{\min}})^2} \right) \right] \underbrace{\frac{1 - \rho^{d_{\max}}}{1 - \rho^{N_{\text{calib}} \cdot d_{\max}}}}_{\leq 1} \quad (31)$$

$$\leq 2\epsilon \left[\frac{d_1}{1 - \rho^{d_{\min}}} + \frac{d_{\max} \rho^{d_{\min}}}{(1 - \rho^{d_{\min}})^2} \right]. \quad (32)$$

Hence, we have proved the *Interpretable Bound*. \blacksquare

For large N_{calib} , (i.e., as $N_{\text{calib}} \rightarrow \infty$), we can achieve a tighter upper bound

Corollary 5 *Under the same assumptions of Theorem 3 we have that as $N_{\text{calib}} \rightarrow \infty$ the coverage gap is more tightly bounded by,*

$$\text{Coverage Gap} \leq 2\epsilon \left[\frac{d_1}{1 - \rho^{d_{\min}}} + \frac{d_{\max} \rho^{d_{\min}}}{(1 - \rho^{d_{\min}})^2} \right] (1 - \rho^{d_{\max}}) \quad (33)$$

Proof (Corollary 5) Taking Equation 31 as $n \rightarrow \infty$ we complete the proof. \blacksquare

A.2. Proof of Theorem 4

Proof (Theorem 4) Using Equation 24, we have that

$$\text{Coverage Gap} \leq \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot d_{TV}(\tilde{S}^{i,k}, \tilde{S}^{k,k}) \leq \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot 2d_{TV}(S_{i,k}, \tilde{S}_{k,k}), \quad (34)$$

where $(s_{1,k}, s_{2,k}, s_{i-1,k}, \tilde{s}_{k,k}, s_{i+1,k}, \dots, s_{N_{\text{calib}},k}) \sim \tilde{S}^{i,k}$, $(s_{1,k}, \dots, s_{N_{\text{calib}},k}, \tilde{s}_{k,k}) \sim \tilde{S}^{k,k}$, $\tilde{s}_{k,k} = s_k(x_k, u_k, f(x_k, u_k))$ is unknown due to uncertainty in $f(x_k, u_k)$, and $\tilde{s}_{k,k} \sim \tilde{S}_{k,k}$ is the distribution of the non-conformity score corresponding to the observed state and control (x_k, u_k) . Then, using the triangle inequality we have that,

$$\text{Coverage Gap} \leq \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot d_{TV}(\tilde{S}^{i,k}, \tilde{S}^{k,k}) \leq \underbrace{\sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot 2d_{TV}(S_{i,k}, S_{k,k})}_{(*) \text{, Bounded in Eq. 32}} + \underbrace{\sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot 2d_{TV}(S_{k,k}, \tilde{S}_{k,k})}_{(**)}. \quad (35)$$

Observing that the left sum (*) is already bounded by Theorem 3, we proceed to bound the right sum (**). Then, realizing the sum of normalized weights is ≤ 1 , we obtain,

$$\sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \cdot 2d_{TV}(S_{k,k}, \tilde{S}_{k,k}) = 2d_{TV}(S_{k,k}, \tilde{S}_{k,k}) \sum_{i=1}^{N_{\text{calib}}} \tilde{w}_i \leq 2d_{TV}(S_{k,k}, \tilde{S}_{k,k}) \quad (36)$$

Then, using the Lipschitz-type bound,

$$2d_{TV}(S_{k,k}, \tilde{S}_{k,k}) \leq 2\hat{\epsilon} \|[x_k^\top, u_k^\top] - [z_k^\top, v_k^\top]\|_2 \leq 2\hat{\epsilon}M(\mathcal{R}_k) =: \gamma(\mathcal{R}_k), \quad (37)$$

where, $(x_k, u_k) \in \mathcal{R}_k$, $M(\mathcal{R}_k)$ is the maximum axis length of the tube, and $\hat{\epsilon} \leq \epsilon$ is the local Lipschitz constant in the tube. The remaining proof follows substituting the bound, $\gamma(\mathcal{R}_k)$ for $(**)$ in (35). ■

A.3. Proof Sketch of Theorem 1 and Theorem 2

Proof Sketch (Theorem 1) Using Theorem 4 we get the probability of error coverage for each step in the rollout. Using the approach of Proposition 2 and Theorem 1 from Chee et al. (2024) and Lindemann et al. (2023) respectively we complete the proof. ■

Proof Sketch (Theorem 2) (a) Since SLS provides guarantees conditioned on valid disturbance bounds, Theorem 1 directly implies the result in Corollary 2(a).

(b) Instead of applying the union bound approach across the forecasted MPC plan, we bound the intersection probability of the first-step errors remaining in the predicted ellipsoids for all R planning steps. Then, using the same approach as Theorem 4 we get:

$$\Pr\left[\bigcap_{k=1}^T \varepsilon(z_1^{(q)}, v_1^{(q)}) \in \mathcal{E}(z_1^{(q)}, v_1^{(q)})\right] \geq 1 - \sum_{q=1}^R \sigma_1^q, \quad (38)$$

Notably, we omit the tubes since the tubes at the first step of each replan, $\mathcal{R}_1^q = \{(z_1^{(q)}, v_1^{(q)})\}$ since the planner knows the initial state. Then, using the same approach as **(a)** we get the result in **(b)**. ■

Appendix B. Dynamics Models

Below we discuss the dynamics models used in our experiments. We provide the continuous time dynamics which were used to train our models. We obtained the discrete model by using Euler Discretization.

B.1. Car Dynamics Models

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega \\ a \end{bmatrix} \quad (39)$$

Equation 39 describes the control-affine dynamics models for Dubins' car, where p_x and p_y are the car's x and y positions. θ and v are the direction and velocity of the car. Lastly, the control inputs directly steer (ω) and accelerate (a) the car. The dynamics model in (39) was used strictly for the in-distribution experiments.

OOD Experiment Dynamics For the out of distribution (OOD) Dubins’ car experiment we modified the θ dynamics by adding attractive steering towards the obstacle centered at (2.5, 0) in the region without training data,

$$\dot{\theta} = \omega_{\text{attr}} \mathbf{1}[|p_y| < 6] + \omega \quad (40)$$

$$d^2 = (p_x - 2.5)^2 + p_y^2 \quad (41a)$$

$$\theta_a = \text{atan2}(p_y, p_x) \quad (41b)$$

$$\Delta\theta_{\text{norm}} = \text{atan2}(\sin(\theta_a - \theta), \cos(\theta_a - \theta)) \quad (41c)$$

$$\omega_{\text{attr}} = \left(\frac{k_{\text{attr}}}{d^2} \right) \Delta\theta_{\text{norm}}, \quad (41d)$$

where the attractive steering dynamics are defined in Equation 41, and atan2 is the arctangent function accounting for the x-y quadrant, and $\mathbf{1}[\cdot]$ is the indicator function. For the OOD experiment we set k_{attr} , the strength of the attractive force to -0.5 (positive values repel). Notably, the strength of the force increases as the car gets closer to the obstacle.

Active Uncertainty Experiment Dynamics Similar to the OOD dynamics we include attractive steering (with $k_{\text{attr}} = -0.5$, we modify the the v dynamics of the car such that,

$$\dot{\theta} = \tilde{\omega}_{\text{attr}} \mathbf{1}[d^2 < 1] + \omega \quad (42a)$$

$$\dot{v} = -(1 + \cos(\pi\sqrt{d^2})) \mathbf{1}[d^2 < 1] + a \quad (42b)$$

$$\tilde{\omega}_{\text{attr}} = \left(\frac{k_{\text{attr}}}{d^2 + 0.1} \right) \Delta\theta_{\text{norm}}, \quad (42c)$$

where d^2 and $\Delta\theta_{\text{norm}}$ are defined in Equation 41. We modify the denominator of ω_{attr} to prevent a singularity at (2.5, 0.0)—this was not a problem in the OOD case since there was an obstacle present at (2.5, 0.0). The augmented velocity dynamics add a friction term to reduce the speed of the car in this region.

Friction Car Dynamics Lastly, we modified Dubins’ car such that the velocity dynamics were modified over the whole state-space, while including attractive steering (with $k_{\text{attr}} = -1.5$):

$$\dot{\theta} = \omega_{\text{attr}} \mathbf{1}[|p_y| < 2] + \omega \quad (43a)$$

$$\dot{v} = a - \text{sign}(v) \left(0.1 \cos\left(\frac{2\pi}{5} p_y\right) + 0.1 - a_{\text{slip}} \mathbf{1}[|p_y| < 2] \right) \quad (43b)$$

$$a_{\text{slip}} = 0.7 e^{-2p_y^2} \quad (43c)$$

In this model, we include a slippage term to increase the velocity of the car for $-2 < p_y < 2$, while providing a changing friction term over the whole state-space.

B.2. 12D Quadcopter Dynamics Model

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \\ \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = f(x, u) := \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ q \sin(\phi) / \cos(\theta) + r \cos(\phi) / \cos(\theta) \\ q \cos(\phi) - r \sin(\phi) \\ p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \\ \frac{u_1}{m} (\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta)) \\ \frac{u_1}{m} (\cos(\psi) \sin(\phi) - \cos(\phi) \sin(\psi) \sin(\theta)) \\ g + \frac{u_1}{m} (\cos(\phi) \cos(\theta)) - a_{\text{fall}} \\ \frac{I_y - I_z}{I_x} qr + \frac{u_2}{I_x} \\ \frac{I_z - I_x}{I_y} pr + \frac{u_3}{I_y} \\ \frac{I_x - I_y}{I_z} pq + \frac{u_4}{I_z} \end{bmatrix} \quad (44)$$

$$a_{\text{fall}} = (0.1 + 0.1 \cos(2\pi d^2 - \pi)) \mathbf{1}[d^2 < 1] \quad (45)$$

$$d^2 = (p_x - 2.5)^2 + (p_y - 2.5)^2 + (p_z - 2.5)^2 \quad (46)$$

For our experiments, we used the 12D Quadcopter dynamics model described in Equation B.2 (Sabatino, 2015). p_x, p_y , and p_z describe the x, y, and z positions of the quadcopter while \dot{p}_x, \dot{p}_y , and \dot{p}_z describe the x, y, and z velocities. ψ, θ , and ϕ are the roll, pitch, and yaw. Similarly, p, q , and r are the roll rate, pitch rate, and yaw rates, respectively are directly controlled by inputs u_2, u_3 , and u_4 . The control input, u_1 , directly impacts the thrust–z-direction acceleration. Lastly, the constants g, m, I_x, I_y , and I_z are the acceleration due to gravity, the mass of the quadcopter, and the principle moments of inertia. For our experiments, we let $g = -9.81m/s^2$, $m = 1.0kg$, $I_x = 0.5kg \cdot m^2$, $I_y = 0.1kg \cdot m^2$, and $I_z = 0.3kg \cdot m^2$. Notably, the expression for \ddot{p}_z includes a_{fall} to introduce a falling term for OOD dynamics. The falling dynamics are not encountered while training the model as described in App. D.1.

Appendix C. Baseline Problem Definitions

We define the baselines used in Sec. 6. First, we define Vanilla MPC (V-MPC) (47), which is a nominal naïve MPC planning under the learned dynamics without any constraint tightenings or model error calibration:

$$\min_{\mathbf{z}, \mathbf{v}} J(\mathbf{z}, \mathbf{v}) + J_{\mathcal{X}_t}(\mathbf{z}, \mathbf{v}) \quad (47a)$$

$$\text{s.t. } z_{k+1} = \hat{f}(z_k, v_k), z_1 = \bar{x}_0, \quad \forall k = 1, \dots, T, \quad (47b)$$

$$g_i(z_k, v_k) + b_i \leq 0, \quad (47c)$$

$$\forall i = 1, \dots, n_c, \quad \forall k = 1, \dots, T.$$

We also define CP-Ball (48), which is a variant of our method which computes CP-calibrated model error bounds, except using a spherical, constant error bound representation, which is more

conservative than our ellipsoidal bound:

$$\min_{\Phi^x, \Phi^u, \mathbf{z}, \mathbf{v}} J(\mathbf{z}, \mathbf{v}) + J_{\mathcal{X}_i}(\mathbf{z}, \mathbf{v}) + \tilde{H}_0(\Phi^x, \Phi^u), \quad (48a)$$

$$\text{s.t.} \quad z_{k+1} = \hat{f}(z_k, v_k), \quad z_1 = \bar{x}_0, \quad \forall k = 1, \dots, T, \quad (48b)$$

$$\Phi_{k+1,j}^x = \nabla_x \hat{f}(z_k, v_k) \Phi_{k,j}^x + \nabla_u \hat{f}(z_k, v_k) \Phi_{k,j}^u, \quad \forall j = 1, \dots, T, \forall k = j+1, \dots, T, \quad (48c)$$

$$\Phi_{j+1,j}^x = V, \quad \forall j = 1, \dots, T, \quad (48d)$$

$$\sum_{j=1}^k \left\| \left[\nabla_x g_i(z_k, v_k) \Phi_{k,j}^x, \nabla_u g_i(z_k, v_k) \Phi_{k,j}^u \right] \right\|_2 + g_i(z_k, v_k) + b_i + g_i^{\text{lin}}(z_k, v_k) \leq 0, \quad (48e)$$

$$\forall i = 1, \dots, n_c, \quad \forall k = 1, \dots, T.$$

Surrogate Cost: All SLS formulations use (49) to minimize the volume of the tubes—a surrogate for uncertainty, where Φ^x and Φ^u collects all $\Phi_{k,j}^x$ and $\Phi_{k,j}^u$, respectively, and $P^{\frac{1}{2}}, Q^{\frac{1}{2}} \in \mathbb{R}^{n_x \times n_x}, R^{\frac{1}{2}} \in \mathbb{R}^{n_u \times n_u}$ are symmetric positive definite matrices.

$$\tilde{H}_0(\Phi^x, \Phi^u) := \sum_{j=0}^{T-1} \left(\|P^{\frac{1}{2}} \Phi_{T,j}^x\|_{\mathcal{F}}^2 + \sum_{k=j}^{T-1} \left(\|Q^{\frac{1}{2}} \Phi_{k,j}^x\|_{\mathcal{F}}^2 + \|R^{\frac{1}{2}} \Phi_{k,j}^u\|_{\mathcal{F}}^2 \right) \right), \quad (49)$$

Appendix D. Experiment Setup

D.1. Data Sampling

Dubins' Car & Friction Car: To train the in-distribution Dubins' car dynamics and uncertainty models, we applied uniform sampling to the following state-control space:

$$[p_x, p_y, \theta, v] \times [\omega, a] \in [0, 5] \times [-5, 5] \times [-\pi, \pi] \times [-10, 10] \times [-10, 10] \times [-10, 10]. \quad (50)$$

Similarly, for the out-of-distribution (OOD) car, we instead sampled $p_y \in [-12, -6] \cup [6, 12]$ with the remaining state and controls remaining the same as (50). For the friction car model, we sampled $p_y \in [-5, -2] \cup [2, 5]$, while using (50) for the remaining dimensions. Lastly, for our active uncertainty experiments, we also sampled from Equation 50, and enforced that $(p_x - 2.5)^2 + (p_y)^2 > 1$ to create an OOD region. For both the OOD car and active uncertainty experiments, we ensured our learned model was restricted to training on areas where modifications did not impact the car dynamics model.

Using the sampled points, we computed the ground truth using the expert dynamics model, as formulated in Equation 39, to create our dynamics training dataset (1,000,000 points), uncertainty training dataset (1,000,000 points), and calibration datasets (varying numbers of points).

12D Quadcopter: To get informative data points to train our neural-network model, we collected expert trajectories such that the states satisfied,

$$p_x, p_y, p_z \in [0, 5] \text{ s.t. } (p_x - 2.5)^2 + (p_y - 2.5)^2 + (p_z - 2.5)^2 > 1 \quad (51a)$$

$$\psi, \theta, \phi \in [-\pi/4, \pi/4] \quad (51b)$$

$$\dot{p}_x, \dot{p}_y, \dot{p}_z \in [-5, 5] \quad (51c)$$

$$p, q, r \in [-2, 2] \quad (51d)$$

$$u_1 \in [0, -2gm], \quad g = -9.81m/s^2, \quad m = 1.0kg \quad (51e)$$

$$u_2, u_3, u_4 \in [-0.1, 0.1] \quad (51f)$$

to create an OOD region centered at $(2.5, 2.5, 2.5)$ with a radius of 1. Additionally, the goal position was $[p_x = 4.5, p_y = 2.5, p_z = 2.5]$, while the start positions was arbitrarily sampled from Equation 51. Our discrete timestep was $0.05s$ and our trajectory length was 100 timesteps. We sampled 100,000 trajectories and included state-control pairs from trajectories that the optimizer (IpOpt) was able to solve. This resulted in a dynamic training and uncertainty-trained set with 4, 702, 400 points each, leaving up to 100, 000 points for the calibration dataset.

D.2. Model Architecture and Training

For our learned model (dynamics and uncertainty), we use an MLP with one hidden layer and the tanh activation function. Our uncertainty model was trained to output Cholesky factors, ensuring the matrices were positive-definite and facilitating faster performance in our optimization problem.

For Dubins' car and the friction car dynamics models, we used 4, 096 and 2, 048 hidden nodes for the dynamics and uncertainty models, respectively. Similarly, for the 12D quadcopter, we used 1, 024 hidden nodes for both the dynamics and uncertainty models, and were trained for 200 epochs. Lastly, we used a learning rate of 10^{-4} and 10^{-5} for the dynamics and uncertainty models, respectively.

D.3. Optimization Parameters and Cost Functions

For the SLS forward propagation and constraint tightenings, we used $\bar{P}^{\frac{1}{2}} = 10^6 I_{n_x}$, $\bar{Q}^{\frac{1}{2}} = 10^6 I_{n_x}$ and $\bar{R}^{\frac{1}{2}} = 10^6 I_{n_u}$, as found in Equation 49. For trajectory optimization, we considered the following LQR cost,

$$J(\mathbf{z}, \mathbf{v}) := \sum_{k=1}^{T-1} \left(z_k^\top Q z_k + (z_{k+1} - z_k)^\top Q_s (z_{k+1} - z_k) + u_k^\top R u_k \right), \quad (52)$$

$$J_{\mathcal{X}_t}(\mathbf{z}, \mathbf{v}) := (z_T - z_{\text{goal}})^\top Q_f (z_T - z_{\text{goal}}) \quad (53)$$

where $Q_f, Q_s, Q \in \mathbb{R}^{n_x \times n_x}$ are positive-semi-definite matrices, $R \in \mathbb{R}^{n_u \times n_u}$ is a positive-definite matrix, and z_{goal} is the goal state. For all dynamics models, we used diagonal matrices, represented as $\text{diag}(a_1, a_2, \dots, a_n)$ and $\text{block_diag}(\dots)$ representing block diagonal matrices. For Dubins' car (excluding the active uncertainty experiment) and the friction car, we let $Q_f = \text{diag}(1, 1, 0, 1)$, $Q = Q_s = 0_{4 \times 4}$ (the 4×4 zero matrix), and $R = \text{diag}(0.1, 0.1)$. For the Quadcopter, $Q_f = \text{block_diag}(I_3, 0_{3 \times 3}, I_3, 0.2I_3)$, $Q = \text{block_diag}(0_{3 \times 3}, 0_{3 \times 3}, 0.01I_3, 0_{3 \times 3})$, $Q_s = 0_{12 \times 12}$, and $R = \text{diag}(0.01, 0.1, 0.1, 0.1)$.

Lastly, regarding conformal prediction parameters, we used $\rho = 0.97$ for the car model and $\rho = 0.8$ for the quadcopter model, respectively. Outside of the OOD car experiment, we used 10, 000 calibration points for the car models and 30, 000 calibration points for the quadcopter model. For all the models we set $\alpha_k = \frac{0.1}{15}$ s.t. the sum total of α would be 0.1.

D.4. Active Uncertainty

To enable active uncertainty reduction, we sought to minimize the distance to the positional-state variables (i.e., p_x and p_y) from $\mathcal{D}_{\text{calib}}$. To achieve efficient performance, we used K-means clustering [Hastie et al. \(2009\)](#) to find a representative set of L points, $\tilde{\mathcal{D}} = \{(p_{x_i}, p_{y_i})\}_{i=1}^L$. Then, we

minimized the following cost for Dubins' car,

$$\begin{aligned}
 & J_{\text{active}}(\mathbf{z}, \mathbf{v}) \\
 &= \sum_{k=0}^{T-1} \exp \left[-\frac{1}{L+G} \left(G e^{-\beta \| (p_{x_{\text{goal}}, p_{y_{\text{goal}}, v_{\text{goal}}) - (p_{x_k, p_{y_k}, v_k)} \|^2} + \sum_{j=1}^L e^{-\beta \| (p_{x_j, p_{y_j}) - (p_{x_k, p_{y_k}}) \|^2} \right) \right], \tag{54}
 \end{aligned}$$

where $p_{x_{\text{goal}}}, p_{y_{\text{goal}}}$, and v_{goal} are the goal x-position, y-position, and velocities, p_{x_k}, p_{y_k} , and v_k are the x-position, y-position, and velocities of the nominal point z_k while p_{x_i} , and p_{y_i} are the x and y positions of points from $\tilde{\mathcal{D}}$, and G is a weight for the goal node (effectively creating G points at the goal). Lastly, β is a sharpness parameter to ensure that we minimize the distance to a particular point in $\tilde{\mathcal{D}}$ as opposed to minimizing the distance to all points. The inclusion of the goal point enabled us to provide a high weight to the cost without hinder the MPC optimizer from reaching the goal. Since our goal is to minimize the distance to the representative points by minimizing the cost function we scale the internal sum by -1 . For efficient implementation we included this as a part of the quadratic cost, thus requiring the outer-exponential to provide a lower-bound to the cost. In our experiments we choose $L = 800$, $G = 200$, and $\beta = 3$, and our total cost was $J_{\text{total}}(\mathbf{z}, \mathbf{v}) = J(\mathbf{z}, \mathbf{v}) + J_{\mathcal{X}_i}(\mathbf{z}, \mathbf{v}) + 1000J_{\text{active}}(\mathbf{z}, \mathbf{v})$. For Q_f, Q , and R we used the same parameters as the remaining Dubins' car experiments and let $Q_s = 0.1I_4$.

Appendix E. Theorem 3 Analysis

For completeness we restate the *Coverage Gap* bound in Theorem 3:

$$\text{Coverage Gap} \leq \underbrace{\sum_{i=1}^{N_{\text{calib}}} \left(\frac{\rho^{d_i}}{1 + \sum_{j=1}^{N_{\text{calib}}} \rho^{d_j}} \right) \cdot 2\epsilon \cdot d_i}_{\text{Tight Bound}} \leq 2\epsilon \underbrace{\left[\frac{d_1}{1 - \rho^{d_{\min}}} + \frac{d_{\max} \rho^{d_{\min}}}{(1 - \rho^{d_{\min}})^2} \right]}_{\text{Interpretable Bound}}$$

As discussed in Section 5, an increase in the density of state-control calibration points reduces d_{\min} , thereby increasing the (interpretable) coverage gap bound in (23). We can reduce the bound by reducing ρ . To build intuition, on the results, we consider the following toy example: we have calibration points uniformly sampled from a unit circle with a test point located at the origin, $(0, 0)$, such that the distribution of the non-conformity score is spatially varying (i.e. $d_{\text{TV}} \neq 0$). Without defining the distribution, Figure 5 demonstrates that at smaller ρ values, greater emphasis is provided to non-conformity scores located near the test point, with smaller TV distances to the test point. Hence, the coverage gap decreases for a fixed N_{calib} as ρ decreases. The interpretable bound also implies that when d_{\min} is reduced, the coverage gap increases, which can be compensated for by requiring ρ to decrease. Figure 5, illustrates that the coverage gap increases at larger d_{\min} stemming from increased data density and N_{calib} , due to a higher weightage being provided to the test point, represented as a point-mass at ∞ in the empirical weighted non-conformity score distribution, resulting in a larger $1 - \alpha$ quantile value. The larger quantile value then increases coverage, thereby reducing the coverage gap.

To understand the tightness of the derived bound, we use a gamma distribution to randomly sample non-conformity scores such that $s(x_i, y_i) \sim S(x_i, y_i) = \Gamma(\alpha = 5 - \frac{1}{3}(\sqrt{x_i^2 + y_i^2})^{0.8}, \theta = 2.0)$,

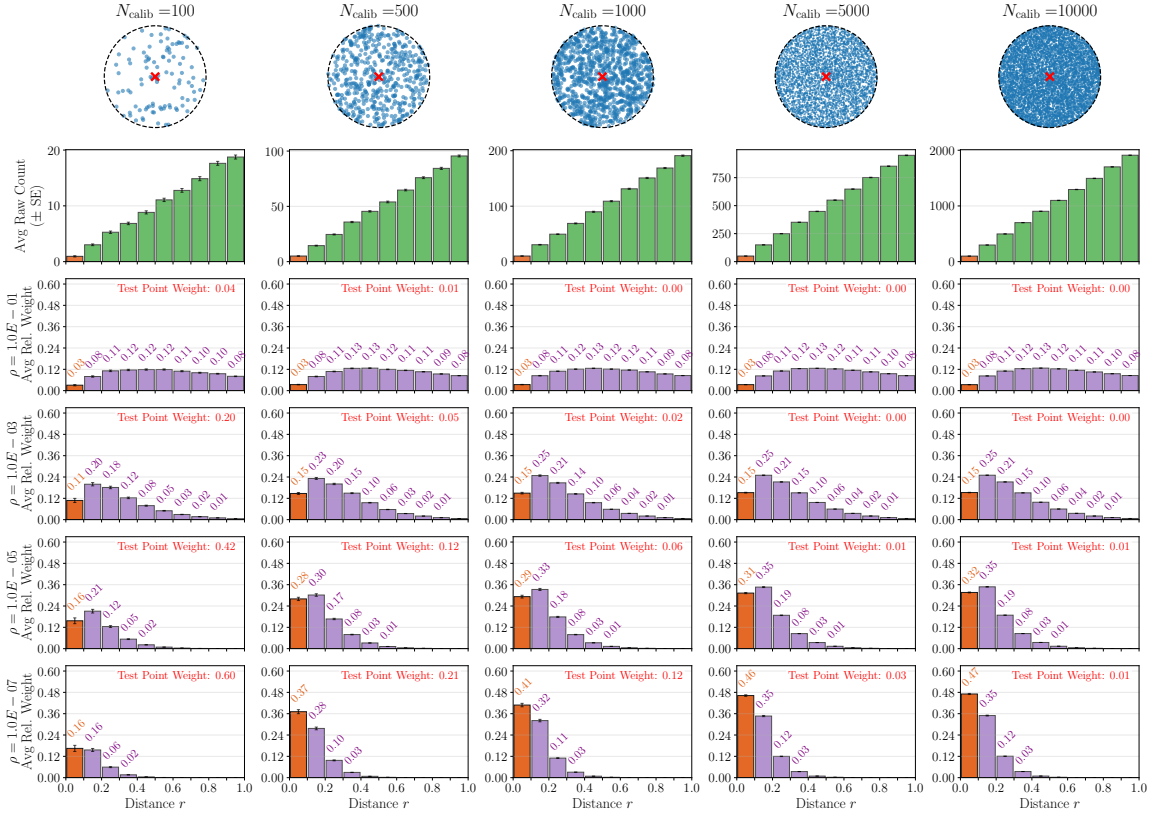


Figure 5: **Toy Example.** We plot the effect of the calibration set size, N_{calib} , and compare the weightage provided to points using the exponential weight decay function at different ρ values. The results are averaged across 100 trials of randomly sampling N_{calib} points in the unit circle, with an example of the calibration data location in the top row. The second row (a histogram) shows that, under uniform sampling on the unit circle, the number of points at each radius (Distance r) increases linearly. The remaining rows provide the weight contribution of each histogram bin by normalizing the weights (as discussed in Section 4.1) and summing the normalized weights for each bin. We observe that smaller ρ values place more emphasis on points closer to the test point, and the weight of the test point, represented by a point mass at ∞ in the empirical score distribution, increases. Lastly, in the plot, red denotes the test point, while orange highlights the bin of points closest to it.

where $\mathbb{E}[s(x_i, y_i)] = 10 - \frac{2}{3}(\sqrt{x_i^2 + y_i^2})^{0.8}$ and $\text{Var}[s(x_i, y_i)] = 20 - \frac{4}{3}(\sqrt{x_i^2 + y_i^2})^{0.8}$. To estimate ϵ , we randomly sampled several pairs of points in the unit circle, then computed $\frac{d_{\text{TV}}(S(x_1, y_1), S(x_2, y_2))}{\|(x_1, y_1) - (x_2, y_2)\|}$ and took the maximum value computed to be ϵ . Then, we sampled several N_{calib} values and ρ values on a log scale and computed the coverage across 100 trials. In each trial, we randomly sampled N_{calib} points on the unit circle, then sampled a non-conformity score from the distribution defined by the coordinates of each sampled point. Since we know the distribution, we can directly compute the TV distance (24) from Barber et al. (2023). We also computed the tight bound in (23) using the estimated ϵ value.

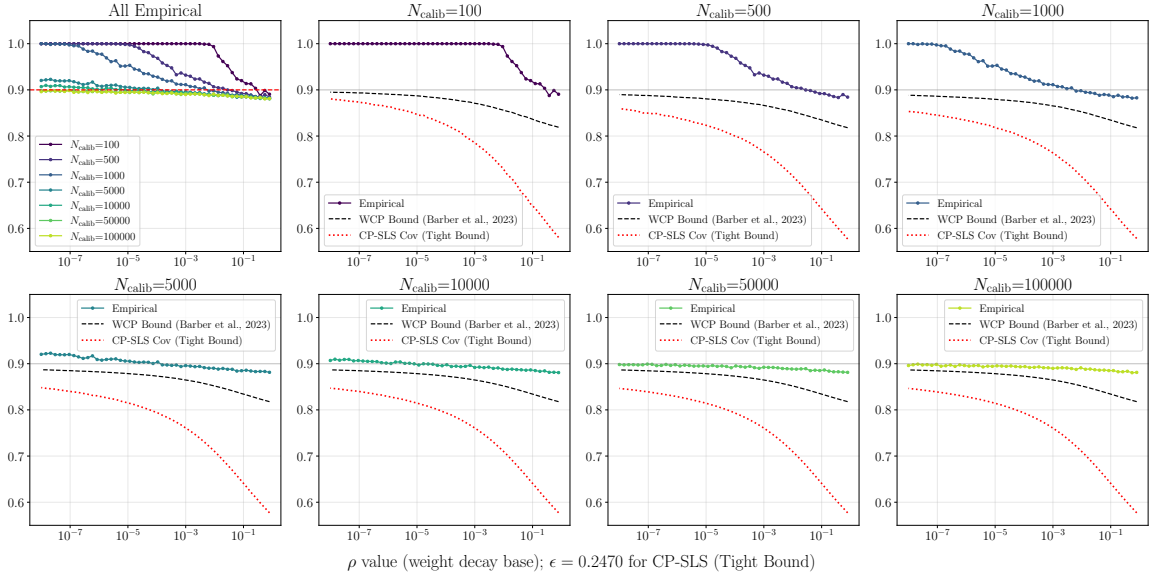


Figure 6: Toy Example. We plot the empirical coverage, along with the theoretical bounds from (24), Barber et al. (2023), and our derived bound (labeled) as CP-SLS cov in (23) for ρ values selected on a log scale and several N_{calib} values. We observe that the lower bounds and empirical coverage increase as ρ decreases, and that the bounds increase more rapidly for smaller ρ values.

The results in Figure 6 demonstrate that the lower bound on coverage ($= 1 - \alpha - \text{Coverage Gap}$) increases for large ρ as well as the empirically observed coverage. Additionally, the tightness of our derived bound, relative to Barber et al. (2023), improves for smaller ρ values. Similarly, we find the lower-bound increases for fixed ρ values as N_{calib} increases. While these results suggest a smaller ρ is optimal, for coverage, it is essential to consider the utility sacrifice. Smaller ρ values lead to larger, potentially infinite, values being computed as the $1 - \alpha$ quantile, thereby making the uncertainty set harder to use and potentially impractical.

Appendix F. Additional Results

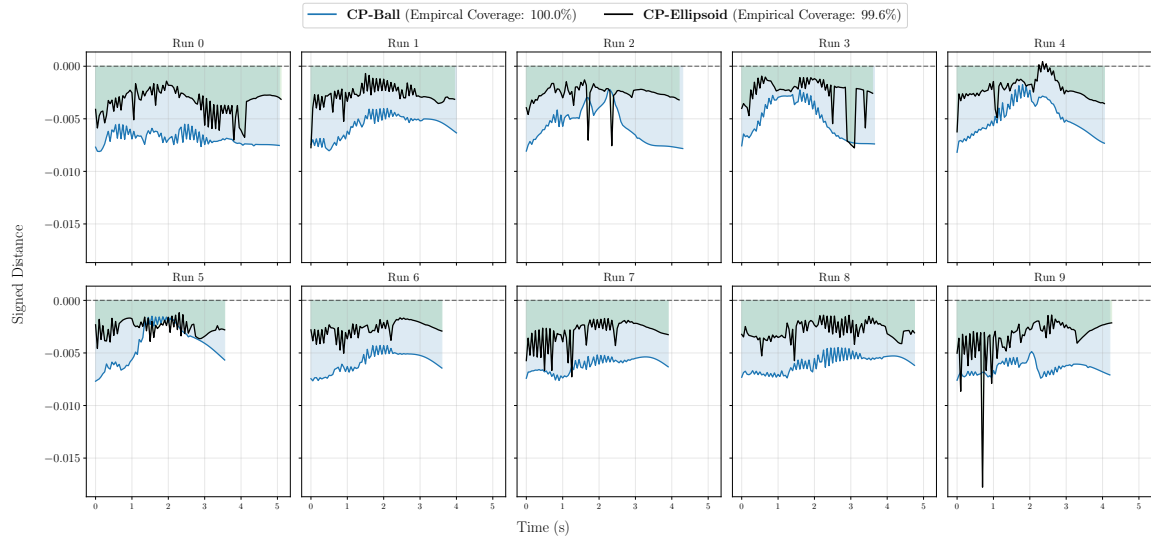


Figure 7: **In Domain Car.** We plot the minimum distance of the prediction error to the edge of the ellipsoid. Both the fixed ball and adaptive ellipsoid method empirically satisfy the coverage guarantee and remain inside over $99.33\% = (1 - \frac{0.1}{15}) \times 100\%$ of the time.

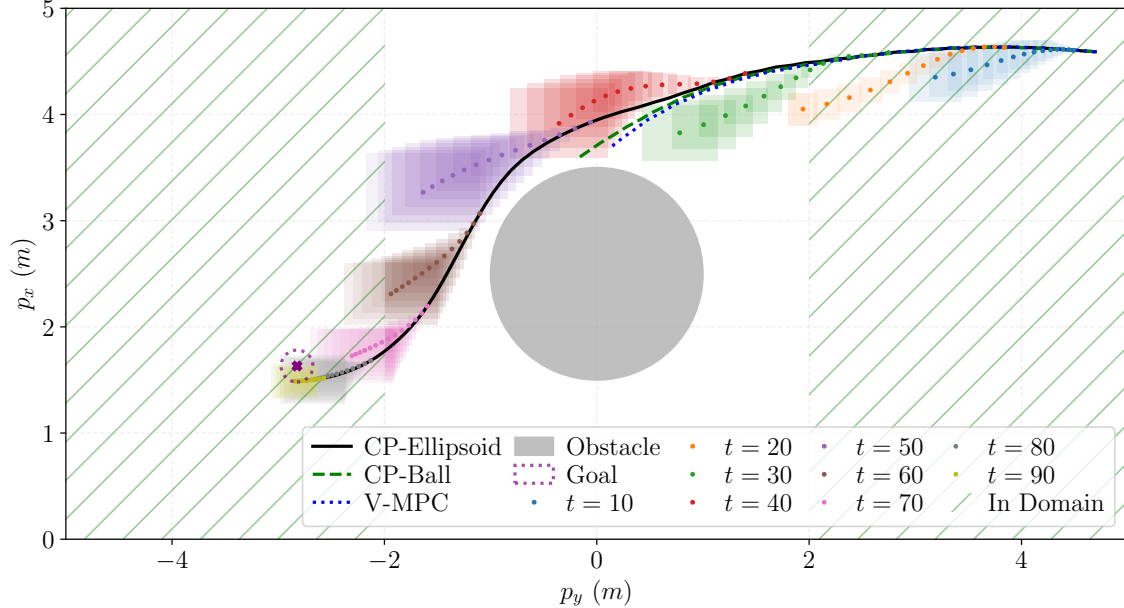


Figure 8: **Friction Car.** We provide an example run of the friction area where the adaptive ellipsoid approach is successful in avoiding the obstacle. The remaining approaches fail because they do not maintain sufficient obstacle proximity and thus crash into the obstacle.

Table 2: **Active Uncertainty.** The computation times (ms), and prediction errors when planning with vanilla-MPC, the adaptive ellipsoid with and without active uncertainty.

| Model | V-MPC | CP-Ellipsoid | CP-Ellipsoid + Active Uncertainty |
|--|-----------------|---------------------|--|
| Time (ms) | 56.1 ± 1.1 | 132.0 ± 9.9 | 131.4 ± 8.2 |
| Pred. Err. (L_2) ($\times 10^{-2}$) | 2.82 ± 7.85 | 2.83 ± 7.89 | $.34 \pm .59$ |

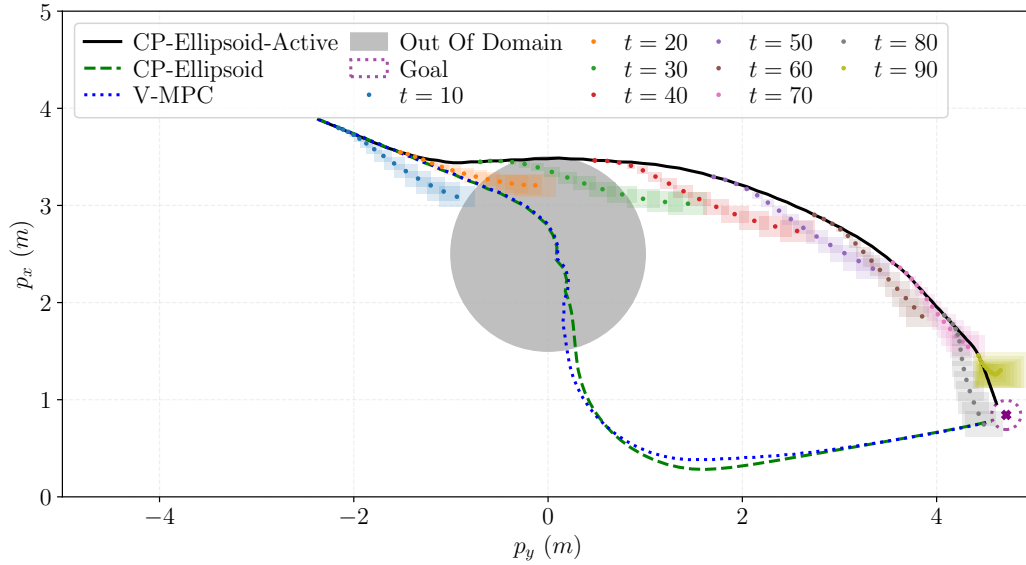


Figure 9: **Active Uncertainty.** We provide an example run of the rollouts when using V-MPC and CP-Ellipsoid with active uncertainty. We find that the active uncertainty approach mostly avoids the OOD region, and thus has a smoother, more certain trajectory.

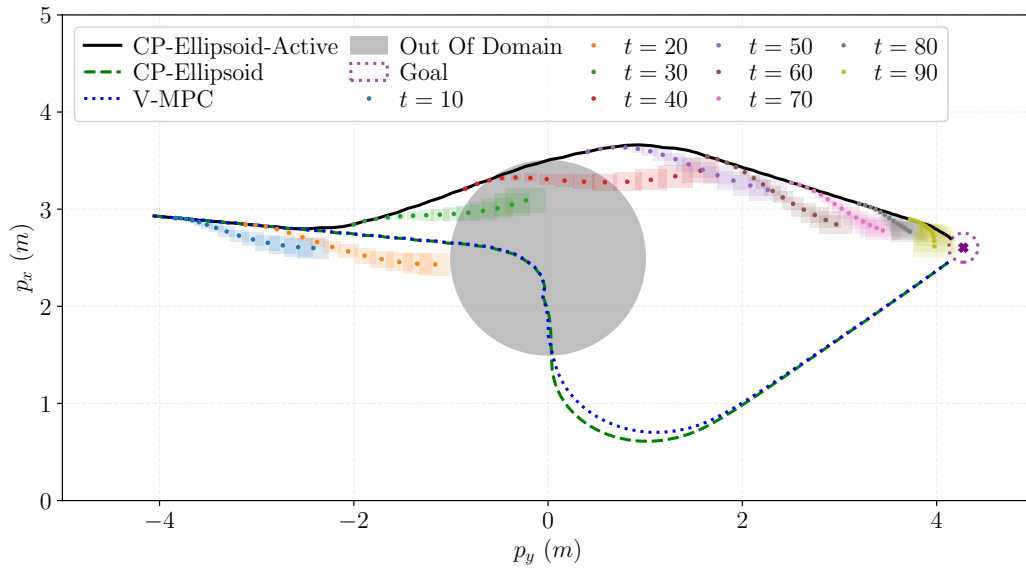


Figure 10: **Active Uncertainty.** We provide another example run of the rollouts when using V-MPC and CP-Ellipsoid with active uncertainty. Similar to Figure 9, the active uncertainty rollout is smoother by mitigating time in the OOD region. Although the active uncertainty approach has more steps in the OOD region than shown in Figure 9, it is near the in-distribution region and thus encounters limited OOD errors during the rollout.

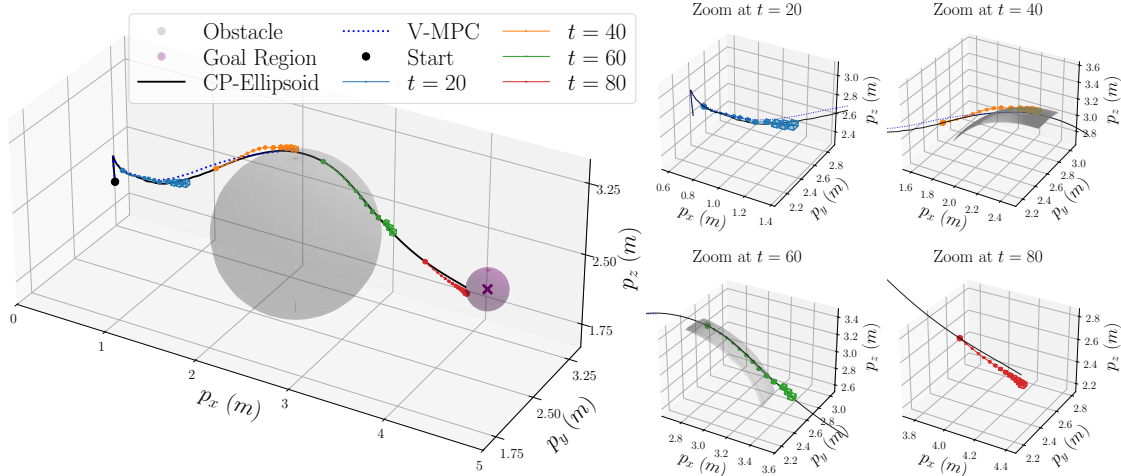


Figure 11: **Quadcopter.** We provide another example run of the quadcopter rollout, with a different initial condition than in Figure 4. Again, we observe the quadcopter is successful in avoiding the obstacle when using CP-Ellipsoid (unlike V-MPC) when entering the OOD region.