# AdaptNC: Adaptive Nonconformity Scores for Conformal Prediction under Distribution Shift

**Renukanandan Tumu**[1] * **Aditya Singh**[1] **Rahul Mangharam**[1]

## Abstract

Rigorous uncertainty quantification is essential for the safe deployment of autonomous systems in dynamic environments. Conformal Prediction (CP) provides a distribution-free framework for this task, yet its standard formulations rely on exchangeability assumptions that are violated by the distribution shifts inherent in real-world robotics. Existing online CP methods maintain target coverage by adaptively scaling the conformal threshold, but typically employ a static nonconformity score function. We show that this fixed geometry leads to highly conservative, volume-inefficient prediction regions when environments undergo structural shifts. To address this, we propose **AdaptNC**, a framework for the joint online adaptation of both the nonconformity score parameters and the conformal threshold. AdaptNC leverages an adaptive reweighting scheme to optimize score functions, and introduces a replay buffer mechanism to mitigate the coverage instability that occurs during score transitions. We evaluate AdaptNC on diverse robotic benchmarks involving multi-agent policy changes, environmental changes and sensor degradation. Our results demonstrate that AdaptNC significantly reduces prediction region volume compared to state-of-the-art threshold-only baselines while maintaining target coverage levels. Code is available at `https://github.com/nandantumu/conformal_region_designer/`.

## 1 Introduction

Autonomous systems operating in real-world environments are subject to constant change. These changes induce distribution shift, driven by changing environmental conditions, such wind or weather, system degradation due to damage or wear and tear, or through multi-agent interactions, which can all degrade the performance of predictive models. For safety-critical applications such as autonomous driving and robotic navigation, identifying these shifts and providing robust, uncertainty-aware predictions is not just a performance requirement; it is a safety necessity [1]. Distribution shift in robotics does not only change the magnitude or frequency of errors, it can change the structure of the errors. For example, tire damage in a ground vehicle or strong winds affecting a drone can induce directionally biased prediction errors. Because prediction regions are often used to define where robots should slow down, avoid, or plan conservatively [2], these structural changes directly affect the usefulness of uncertainty estimates. Thus, prediction regions should not only adapt in size, but also in shape, as environmental and system conditions evolve.

Uncertainty quantification has been a focus of much recent work, and conformal prediction (CP) [3, 4], has emerged as a popular method for its flexibility to be applied to predictors of all types, its statistical grounding, and its ease of use for the practitioner. CP constructs prediction sets which contain the true label with high $(1-\alpha)$ probability, under the assumption of exchangeability between the calibration and test distributions. In CP, the threshold controls the size of the prediction region, while the non-conformity score function controls its geometry. Online adaptation has largely focused on the former, leaving the latter fixed even when the structure of the data changes.

*[1] All authors are from Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania (email: {nandant, singhadi, rahulm}@engineering.upenn.edu)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

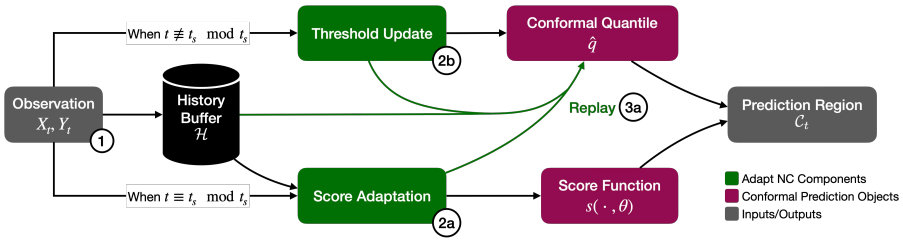



Figure 1: The AdaptNC framework is an online algorithm designed to handle distribution shifts. The above figure shows the procedure at a timestep $t$. (1) The observation for the current timestep arrives, and is added to the history buffer. Every $t_s$ timesteps, the score function is adapted by weighted score adaptation (2a). This adaptation yields a score function $s(\cdot, \theta_t)$, the replay algorithm (3a) compensates for the distribution shift generated by the change in score function by replaying recent samples from the history buffer $\mathcal{H}$ and generates the conformal quantile $\hat{q}$. When the score is not being adapted, only the conformal quantile is adapted through the threshold update step.

Recent online CP methods adapt the conformal threshold to maintain coverage under distribution shift[5–7]. However, because they keep the non-conformity score fixed, they can only resize prediction regions rather than reshape them. This limitation is important in robotic settings, where shifts such as sensor degradation or policy changes can alter the geometry of prediction errors. Separately, previous work has addressed the design of non-conformity score functions[8, 9], including control and robotics settings with multi-dimensional outputs [10–13], however, this work does not address the adaptation of the non-conformity score in response to distribution shift.

Unlike existing methods that only tune the threshold, AdaptNC optimizes the underlying score function to ensure that prediction regions remain tight and informative even as the data distribution evolves. AdaptNC also shifts the conformal threshold. In order to provide principled score function adaptation, we introduce an *adaptive reweighting scheme* that leverages DtACI expert weights to prioritize the most relevant historical observations during score optimization. The joint adaptation of the score function and threshold introduces the phenomenon of "coverage shock": the errors in coverage caused by evaluating old threshold values under a new score function. To mitigate "coverage shock" we employ a counterfactual *replay mechanism* to recalibrate the conformal threshold following a score update. Across three robotic settings with structural distribution shifts, AdaptNC maintains target coverage while producing smaller prediction regions than threshold-adaptation baselines, demonstrating that score adaptation improves efficiency without sacrificing empirical coverage.

**Contributions:** This paper addresses the challenge of constructing efficient, uncertainty-aware prediction regions under distribution shift by contributing:

1. **Online Score Function and Threshold Adaptation:** We present a method for the online optimization of non-conformity scores and their corresponding conformal thresholds that minimizes prediction region volume while enabling adaptation to shifting distributions.
2. **Dynamic Weighting and Replay Buffer:** We develop a weighting mechanism for data based on the rate of distribution shift, and a replay mechanism to maintain coverage stability during score parameter transitions.
3. **Empirical Validation on Robotic Benchmarks:** We evaluate AdaptNC on robotic tasks involving structural distribution shifts, including multi-agent policy changes and sensor degradation. We demonstrate that AdaptNC reduces prediction volume compared to state-of-the-art online CP baselines while maintaining target coverage levels.

Furthermore, we emphasize that AdaptNC is not a direct composition of prior adaptive conformal prediction and score-adaptation methods. Instead, it formulates a new setting involving coupled adaptation, characterizes the instabilities arising from a naive integration, and introduces the mechanisms necessary to ensure stable and effective joint adaptation. Additional discussion is provided in Appendix B.1.

## 2 Problem Setup and Preliminaries

### 2.1 Online Prediction Setting

We consider an online $\tau$-step prediction setting. At each time step $t = 1, \ldots, T$, the predictor observes features $X_t \in \mathcal{X}$ and produces a $\tau$-step prediction $\hat{Y}_t = h(X_t) = \{\hat{y}_{t+1}, \ldots, \hat{y}_{t+\tau}\} \in \mathcal{Y}^\tau$.

The future outcome $Y_t \in \mathcal{Y}^\tau$ is then observed. The joint distribution of $(X_t, Y_t)$ may change over time.

We seek to build a prediction region $\mathcal{C}_t$ that contains the future outcome $Y_t$ with probability $1-\alpha$, where $\alpha \in (0,1)$ is the target miscoverage rate. The prediction regions are constructed using a non-conformity score function $s(X_t, y; \theta_t)$ where $\theta_t$ are parameters of the score function at time $t$, as well as a conformal threshold, $q_t$. We define $q_{\mathcal{D}_t}(\delta)$ to be the empirical $\delta \in [0,1]$ quantile of the score dataset $\mathcal{D}_t$, and $q_t = q_{\mathcal{D}_t}(\delta_t)$. The prediction region is the set of points with a non-conformity score less than the conformal quantile: $\mathcal{C}_t(X_t; \theta_t, q_t) \coloneqq \{y \in \mathcal{Y}^\tau : s(X_t, y; \theta_t) \leq q_t\}$. The threshold $q_t$ controls the size of the prediction region, while the score parameters $\theta_t$ control its geometry.

**Desiderata:** We pursue the objectives of coverage that approaches the target, and efficiency of the prediction region under settings with distribution shift.

**Coverage:** Under distribution shift, exact finite-sample coverage at every time step is generally not attainable. Instead, we seek the asymptotic miscoverage rate $\alpha$, in settings where the magnitude of distribution shift decays over time. We define the error indicator $\mathbf{err}_t = \mathbb{1}\{Y_t \notin \mathcal{C}_t(X_t; \theta_t, q_t)\}$, and we target:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{err}_t = \alpha. \tag{1}$$

**Efficiency:** We would also like to ensure that the prediction regions $\mathcal{C}_t$ are small and informative to downstream algorithms. Subject to our coverage desideratum, our objective is to minimize the volume of the prediction set $\mathcal{C}_t$. We formulate the optimization problem:

$$\min_{\theta_t} \quad \text{Vol}(\mathcal{C}_t(X_{t+1}; \theta_t, q_t)), \quad \text{s.t.} \;\; \Pr[Y_{t+1} \in \mathcal{C}_t(X_{t+1}; \theta_t, q_t)] \geq 1-\alpha. \tag{2a}$$

In the online setting, the future distribution is unknown, so this problem serves as an idealized objective: AdaptNC seeks to reduce prediction-region volume while maintaining long-run coverage.

### 2.2 Preliminaries

**Conformal Prediction:** Conformal prediction (CP) and its split variants provide distribution-free, finite-sample prediction regions which guarantee ground truth coverage with $1-\alpha$ probability under the assumption of exchangeable calibration and test data [3, 4, 14]. Under distribution shift, this exchangeability assumption may fail, motivating online methods that adapt the threshold over time.

**Online Threshold Adaptation:** Online conformal prediction methods adapt the threshold to maintain long-run coverage in settings where distribution shift occurs. DtACI [6] targets long-run coverage under arbitrary distribution shifts by adaptively reweighting a set of candidate adaptation rates based on their historical performance. It obtains a long-term coverage guarantee of the form in Eq. (1) in settings where the size of the distribution shift decays over time.

The DtACI algorithm seeks to compensate for distribution shift by changing the adaptive miscoverage level $\alpha_t$ used to select the score threshold $q_t = q_{\mathcal{D}_t}(1-\alpha_t)$. The algorithm seeks to find $\alpha_t$ that minimizes the regret to the hindsight optimal quantile $\alpha_t^*$. This difference is bounded in Theorem D.1. Observations in the algorithm are quantiles $\beta_t$, which are the lowest empirical quantiles of the dataset $\mathcal{D}_t$ that would have covered the true outcome at time $t$. Formally,

$$\beta_t \coloneqq \inf\{\beta : Y_t \in \{y : s(X_t, y; \theta_t) \leq \hat{q}_{\mathcal{D}_t}(\beta)\}\} \tag{3}$$

To achieve this, DtACI maintains a set of $k$ candidate adaptation rates $\{\gamma_i\}_{i=1}^k$ and corresponding weights $\{w_t^i\}_{i=1}^k$. At each time step, the algorithm performs an exponential weights update on the weights based on the error incurred by each candidate rate. An averaged quantile $\bar{\alpha}_t$ is then computed using the weights, and the quantile threshold $q_{\mathcal{D}_t}(1-\bar{\alpha}_t)$ is updated accordingly. The algorithm is presented in Algorithm 3 (in Appendix C).

## 3 Related Work

**Online Threshold Adaptation.** Online conformal prediction methods address distribution shift primarily by adapting the conformal threshold or calibration quantile over time. Adaptive Conformal Inference (ACI) updates the quantile based on recent coverage errors [5], while related methods such as AgACI and multivalid conformal prediction provide alternative mechanisms for maintaining coverage under non-stationarity [7, 15]. DtACI further advances this line of work by adaptively

weighting multiple candidate adaptation rates and providing long-term coverage regret guarantees [6]. Bhatnagar et al. [16] develop Strongly Adaptive Online Conformal Prediction (SAOCP), which aggregates scale-free online gradient descent experts operating over different active intervals. This construction provides near-optimal strongly adaptive regret guarantees over all interval lengths. SAOCP directly adapts finite score thresholds under a bounded-score assumption. Consequently, this method produces finite prediction regions and cannot express uncertainty through an infinite threshold, as DtACI and AdaptNC can. However, these online CP methods adapt the threshold while leaving the non-conformity score fixed. As a result, they can resize prediction regions under distribution shift, but they cannot reshape them when the geometry of prediction errors changes. Other approaches address distribution shift by reweighting calibration points to better match the test distribution under covariate shift [17, 18]. These methods modify the effective calibration distribution, and have been used in reinforcement learning settings [19, 20] but they do not adapt the non-conformity score online.

**Efficient and Shape-Adaptive Non-Conformity Scores.** A separate line of work studies how the choice of non-conformity score affects the shape and volume of conformal prediction regions. Recent methods design volume-efficient scores for general prediction settings [8, 9], as well as for control and robotics problems with multi-dimensional outputs [10–13]. These methods show that intentional design of the non-conformity score function can substantially improve prediction-region efficiency and downstream usefulness. However, they typically design or optimize the score for a fixed data-generating process. They do not consider online score adaptation under distribution shift, where the geometry of prediction errors may change over time.

# 4 AdaptNC Method

Our method, Adaptation of Non-Conformity Scores (AdaptNC), extends online conformal prediction by jointly adapting the conformal threshold and non-conformity score function (NCSF). (1) **Threshold adaptation** maintains long-run coverage, (2) **Score Parameter Adaptation** improves efficiency under distribution shift, and (3) **Replay** recalibrates the threshold after score updates to mitigate coverage shock. At each timestep, the conformal threshold is updated using the quantile $q_{\mathcal{D}_t}(1-\bar{\alpha}_t)$ based on the current NCSF and newly observed data. Score parameters $\theta_t$ are periodically updated using estimates of distribution shift to obtain more efficient regions, and replay rescores recent observations under the updated NCSF ccount for the induced change in the score distribution . Due to computational intensity, steps (2) and (3) are performed every $t_s$ timesteps, while (1) is updated at every timestep. An overview is shown in Fig. 1, with full details in Algorithm 2 (Appendix C).

The data observed is saved as tuples $(X_t, Y_t, \hat{Y}_t)$ in a history buffer $\mathcal{H}$. For each observation, $\beta_t$ is calculated according to Eq. (3). The dataset $\mathcal{D}_t$ used in the quantile calculation is a rolling window of the last $W$ timesteps.

**Conformal Threshold Update** At every timestep, AdaptNC uses DtACI (Algorithm 3) to adapt the conformal threshold online. The goal of this step is to select an adaptive miscoverage level $\bar{\alpha}_t$ and the corresponding quantile $q_t = q_{\mathcal{D}_t}(1-\bar{\alpha}_t)$, where $\mathcal{D}_t$ is a rolling window of the last $W$ timesteps. To enable coverage under unknown distribution shift, $k$ ACI [5] experts are maintained and dynamically reweighted. The procedure involves calculating an inverse quantile to transform the raw non-conformity score, expert reweighting, and finally expert update. This procedure aims to estimate $\alpha_t^*$, the optimal value for the miscoverage rate that would yield the correct coverage.

Based on the error between the predicted quantile for each expert $\alpha_t^i$ and the observed $\beta_t$, the weight $w_t^i$ is exponentially reweighted. When these weights are applied to the experts, we get the algorithm's estimate of the current conformal quantile $\bar{\alpha}_t$. Finally, each expert is updated according to the expert learning rate $\gamma_i$ and the error indicator $\mathbf{err}_t^i$, which is 1 if the observation is not covered by the $i$-th expert, and 0 otherwise. The complete DtACI algorithm is presented in Algorithm 3 (in Appendix C).

**Score Parameter Adaptation** Threshold adaptation can maintain coverage under distribution shift, but it can only resize prediction regions for a fixed score function. To adapt the geometry of the prediction region, AdaptNC periodically updates the score parameters $\theta_t$. The objective of this step is to find score parameters that reduce prediction-region volume while preserving the desired coverage behavior. Since the future distribution is unknown, AdaptNC approximates this objective using a relevance-weighted history buffer, where the relevance weights are derived from the current DtACI expert weights $w_t^i$ and adaptation rates $\gamma_i$.

Let $\mathcal{H}_t$ denote the history buffer at time $t$, and let $p_t^i = w_t^i / \sum_{h=1}^k w_t^h$ be the normalized DtACI expert weights. We associate each expert with a geometric memory rate $\lambda_i = \rho\gamma_i/\gamma_{\max}$, where

$\gamma_{\max} = \max_i \gamma_i$, $0 < \rho \leq 1$, and $\rho = 1$ in our experiments. For each historical observation $j \leq t$, we normalize the geometric weights within each expert and combine them using the DtACI mixture:

$$\kappa_{j,t}^i = \frac{(1-\lambda_i)^{t-j}}{\sum_{\ell \in \mathcal{H}_t}(1-\lambda_i)^{t-\ell}}, \qquad \omega_{j,t} = \sum_{i=1}^{k} p_t^i \kappa_{j,t}^i. \tag{4}$$

When $\lambda_i = 1$, the expert assigns all its history weight to the most recent observation.

This weighting scheme induces a weighted empirical distribution $\mathcal{H}_t^\omega$. Larger adaptation rates correspond to faster threshold adjustment and are assigned shorter geometric memory, while smaller rates retain information over longer histories. Normalizing each expert's weights before mixing ensures that its total contribution equals its DtACI probability $p_t^i$. As DtACI shifts weight among adaptation rates, the score-fitting distribution shifts between recent observations and longer-term history. This connects the timescales favored by threshold adaptation to the data used for adapting region geometry.

Using the weighted history distribution $\mathcal{H}_t^\omega$, AdaptNC seeks score parameters that minimize prediction-region volume subject to weighted empirical coverage. This gives the objective

$$\min_{\theta} \quad \mathbb{E}_{X \sim \mathcal{H}_t^\omega}\left[\mathrm{Vol}\left(\mathcal{C}(X;\theta,q_\theta)\right)\right] \quad \text{s.t. } \mathbb{P}_{(X,Y)\sim\mathcal{H}_t^\omega}\left[Y \in \mathcal{C}(X;\theta,q_\theta)\right] \geq 1-\alpha, \tag{5a}$$

where $q_\theta$ denotes the threshold induced by the score function with parameters $\theta$ on the weighted history. In practice, AdaptNC approximates this objective through two stages: estimating a high-density region of the weighted residual distribution and constructing its convex-hull template. The density-estimation stage identifies residual configurations carrying approximately $1-\alpha$ probability mass, while convexification produces geometry suitable for downstream planning and control. Convexification can include low-density space between modes and therefore introduces additional volume. The consistency analysis concerns estimation of the limiting region; it does not bound the volume gap between the implemented procedure and the optimum of Eq. (5).

To estimate the high-density region, AdaptNC forms residual samples $Y_j - h(X_j)$ from the weighted history buffer and constructs a weighted kernel density estimate (KDE) [21]. We then sample candidate residuals from the KDE, evaluate their estimated density, and retain the highest-density samples whose empirical mass is approximately $1-\alpha$. We denote this retained set by $\hat{R}_t$. This Monte Carlo procedure provides an empirical approximation of the high-density residual region under the current weighted data distribution; the full algorithmic details are given in Appendix C.4.

Given the estimated high-density region $\hat{R}_t$, AdaptNC fits a convex shape template to cover these points, in a manner similar to [10]. We parameterize the template by $\theta_t = \{A \in \mathbb{R}^{r\times p}, b \in \mathbb{R}^r\}$, where $r$ is the number of facets. The resulting non-conformity score is: $s(X,y;\theta_t) = \max_{j\in\{1,\ldots,r\}}\left(A_j(y-h(X)) - b_j\right)$.

Under this parameterization, the zero sublevel set of the score corresponds to the fitted convex template centered at the predictor output $h(X)$. We fit the convex hull of $\hat{R}_t$ using the QuickHull algorithm [22]. This produces convex prediction-region geometry, which is useful for downstream planning and control applications [23].

**Replay** Changing the score parameters alters the distribution of non-conformity scores. Consequently, the DtACI state prior to an update may remain calibrated to the previous score function. This mismatch induces *coverage shock*, a transient calibration error immediately following a score update. In the DtACI regret bound, this appears as a potentially large change in the hindsight-optimal miscoverage level $\alpha_t^*$ across consecutive timesteps (see Sections E.1 and 5.3 for theoretical and empirical evidence ).

To mitigate coverage shock, AdaptNC replays the most recent $W$ observations after each score update. The replay procedure rescans these observations under the updated score parameters $\theta_t$, recomputes the corresponding quantiles $\beta_t$, and reruns the threshold updates. This can be interpreted as a counterfactual reconstruction of the DtACI state under the new score function. The resulting miscoverage level and threshold are then used for subsequent online predictions, effectively resetting the the threshold adaptation process under the new score distribution and reducing the abrupt calibration error induced by score updates.

## 5 Theoretical Analysis

AdaptNC modifies adaptive conformal prediction in two ways: it updates the nonconformity score online, and it replays recent observations after score updates to realign the ACI expert states with the updated score. These two mechanisms serve different goals. Score adaptation controls the geometry and efficiency of the prediction region, while the ACI expert updates control long-run coverage. Our theoretical analysis therefore separates these effects: we first study long-run stability of prediction-region size, then show that single-expert ACI coverage is preserved whenever replay induces only a vanishing average perturbation of the expert recursion. $\theta_t$.

### 5.1 Long-Run Score Function and Region Stability

We make a series of assumptions in the long-run stability and coverage analysis, which are analogous to the long-run regimes considered in prior work [6].

**Assumption 5.1** (Weighted Residual Stability)**.** We assume that the weighted distribution of residuals $\mathcal{Z}_t$ converges in $f$-divergence to the steady state $\mathcal{Z}_*$ in the asymptote: $D_\phi(\mathcal{Z}_t, \mathcal{Z}_*) \to 0$ as $t \to \infty$. We additionally assume that the MCKDE parameters $M, N \to \infty$.

In AdaptNC, because the score function is fit from weighted residuals, the corresponding stability object is the weighted residual distribution used by the score-adaptation step. We require the regularity conditions for the HDR stability result in Assumption D.4. We also assume that the MCKDE procedure converges in Assumption C.1. The proof of Theorem 5.2 can be found in Section D.2.

**Theorem 5.2** (Score Function Stability)**.** *Suppose Assumptions C.1, D.4 and 5.1 hold. Let $\widehat{R}_m$ be the MCKDE high-density residual region estimated at score-update time $t_m$, and define$K_m := \mathrm{conv}(\widehat{R}_m)$. Let$R_\star := \{z : f_\star(z) \geq \tau_\star\}$ and $K_\star := \mathrm{conv}(R_\star)$. Let $s_{K_m}$ and $s_{K_\star}$ denote the canonical support-function scores induced by $K_m$ and $K_\star$. Then, for any evaluation domain $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{Y}$,*

$$\sup_{(x,y)\in\mathcal{A}} |s_{K_m}(x,y) - s_{K_\star}(x,y)| \xrightarrow{p} 0 \quad \therefore \quad \sup_{(x,y)\in\mathcal{A}} |s_{K_{m+1}}(x,y) - s_{K_m}(x,y)| \xrightarrow{p} 0.$$

**Corollary 5.3** (Stability of prediction-region size)**.** *Suppose* $\sup_{(x,y)\in\mathcal{A}} |s_t(x,y) - s_\star(x,y)| \leq \delta_t$ *with $\delta_t \to 0$ in probability due to Theorem 5.2. Under Assumption D.4, for every fixed threshold $q$,*

$$|\mathrm{Vol}(C_t(x;q)) - \mathrm{Vol}(C_\star(x;q))| \leq L_{\mathrm{vol}}\delta_t \quad \textit{in probability}.$$

*Hence the size of the prediction region stabilizes as the score function stabilizes in probability.*

Thus, the score-stability result implies that the geometry and volume of the prediction regions stabilize around the limiting convex-template region. In this sense, the score-adaptation analysis is an efficiency result: once the residual geometry stabilizes, AdaptNC no longer incurs persistent changes in prediction region size due to score updates.

### 5.2 Coverage Guarantees

The long-term coverage guarantee of our method follows from a variation on the per-expert guarantee of ACI, which we present in Theorem 5.4. This modification is necessary because the recursion used to prove the original ACI guarantee is perturbed by the score updates and replay mechanism. We show that the single-expert ACI guarantee is preserved whenever replay induces only a vanishing average perturbation of the expert recursion. We then give a sufficient condition for this perturbation to vanish: successive score updates must become small on replayed observations, and replayed observations must not concentrate near the conformal boundary.

**Theorem 5.4** (Per-Expert Coverage Guarantee with AdaptNC)**.** *Fix an expert $i \in \{1, \ldots, k\}$ with step size $\gamma_i > 0$. At time $t$, let the expert's prediction set be $\mathcal{C}_t^i = \mathcal{C}_t(X_t; \theta_t, q_{\mathcal{D}_t}(1-\alpha_t^i))$, and define its realized miscoverage indicator $\boldsymbol{err}_t^i = \mathbb{1}\{Y_t \notin \mathcal{C}_t^i\}$. Suppose the expert state evolves according to the perturbed ACI recursion $\alpha_{t+1}^i = \alpha_t^i + \gamma_i(\alpha - \boldsymbol{err}_t^i) + \Delta_t^i$, where $\Delta_t^i$ is the perturbation caused by score updates and replay. Assume that the expert states remain uniformly bounded in the range $(0,1)$, then:*

$$\frac{1}{T}\sum_{t=1}^{T} |\Delta_t^i| \longrightarrow 0 \implies \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{err}_t^i \longrightarrow \alpha \tag{6}$$

**Theorem 5.5** (Averaged Replay Perturbation Bound)**.** *Fix an ACI expert $i$ with step size $\gamma_i > 0$. Let $\{t_m\}_{m\geq 1}$ be the score-update times, and let $\mathcal{R}_m$ be the replay window used at update $t_m$. Let $s_m^-$ and $s_m^+$ denote the score functions immediately before and after the update. Define $\varepsilon_m := \sup_{r\in\mathcal{R}_m} |s_m^+(X_r, Y_r) - s_m^-(X_r, Y_r)|$ . Let $q_{m,r}^{i,-}$ and $q_{m,r}^{i,+}$ be the conformal thresholds used by expert $i$ at replay step $r$ under the pre-update and post-update scores. Suppose there exists $L_q < \infty$ such that $|q_{m,r}^{i,+} - q_{m,r}^{i,-}| \leq L_q\varepsilon_m$ for all $m$ and $r \in \mathcal{R}_m$. Suppose further that*

$$\frac{1}{T}\sum_{m:t_m\leq T}\sum_{r\in\mathcal{R}_m} \mathbf{1}\left\{\left|s_m^-(X_r, Y_r) - q_{m,r}^{i,-}\right| \leq (1 + L_q)\varepsilon_m\right\} \longrightarrow 0.$$

*Then the replay-induced perturbations satisfy $\frac{1}{T}\sum_{t=1}^{T} |\Delta_t^i| \longrightarrow 0$.*

The second supposition requires that replayed observations do not accumulate in the shrinking band around the conformal boundary where a small score update can change the miscoverage indicator. Score stability makes the width of this band shrink, but it does not by itself rule out many replayed points lying near the boundary. The assumption can be interpreted as a regularity condition on the data distribution. Taken together, the preceding results establish the single-expert long-run coverage property underlying AdaptNC. Proofs are provided in Section D. Since the DtACI long-run coverage guarantee holds for arbitrary sequences, AdaptNC inherits this guarantee for any sequence of score functions.

### 5.3 Replay

We formally define *coverage shock* as the $\alpha^*$ shift term from Remark D.2: coverage shock $= \left|\alpha_t^* - \alpha_{t-1}^*\right|$ . Together with the term that varies inversely with the window size $|W|$, this quantity dominates the regret. Minimizing coverage shock, i.e., the change in the optimal miscoverage level across consecutive timesteps, will improve regret performance. However, changes to the NCSF can have an outsized impact on this difference, motivating the counterfactual replay mechanism. Here, we present an intuitive explanation paired with evidence from a toy experiment in support.

The hindsight-optimal $\alpha_t^*$ depends on the distribution of non-conformity scores at time $t$. We decompose changes in this distribution into those arising from new observations and those induced by updates to the NCSF. First, we note that replacing a single data point yields a bounded change in the score distribution in Proposition D.9. Next, we show that the distance caused by the change in NCSF parameter is not bounded by the data-distribution shift.

*Remark* 5.6. The score distribution depends on both the underlying data and the NCSF used to process it. We can decompose the distribution shift as follows due to Proposition D.9.

$$\mathbf{TV}(S_{\theta_t,t}, S_{\theta_{t+1},t+1}) \leq \delta + \mathbf{TV}(S_{\theta_t,t+1}, S_{\theta_{t+1},t+1}) \tag{7}$$

### 5.4 Computational Complexity Analysis

The computational complexity is dominated by replay associated costs, proportional to the size of the replay window $W$.

*Remark* 5.7 (Computational Complexity of AdaptNC). Let $W$ denote the window size, $k$ the number of experts, $M$ the number of Monte Carlo samples, $d$ the state dimension, $r$ the number of facets of the convex hull, and $n = |\hat{R}_t|$ the number of high-density points retained after thresholding. Under the assumption $d \leq 3$, the amortized per-timestep cost of AdaptNC is $O(W + k + MWd/t_s)$ .

We defer a detailed analysis of computational complexity to Section C.5, and a discussion in Section B.3

## 6 Experiments

We evaluate AdaptNC across three case studies capturing common forms of distribution shift in robotics and dynamical systems: (i) environmental shift from unmodeled sensor noise, (ii) policy-induced shift from changing interaction dynamics, and (iii) dynamics shift from gradual system degradation. Together, these settings provide a comprehensive testbed for adaptive uncertainty quantification under evolving conditions.

We present three experimental settings. In the **Indoor Localization** experiments, a robot performs localization using wireless signal strength measurements. Distribution shifts arise in this setting

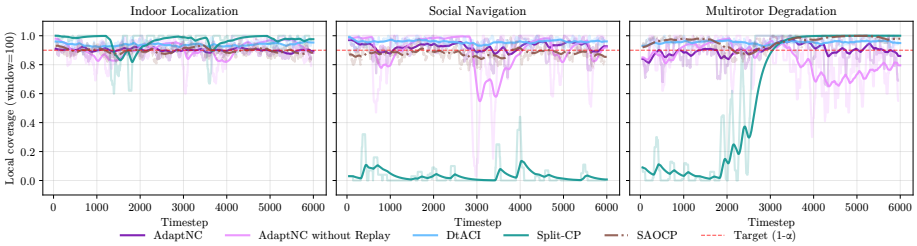



Figure 2: This figure shows the evolution of local coverage over a 100-step sliding window for a representative seed. For readability, an exponential moving average is shown, with the raw data in a lighter color. AdaptNC consistently maintains local coverage near the target level.

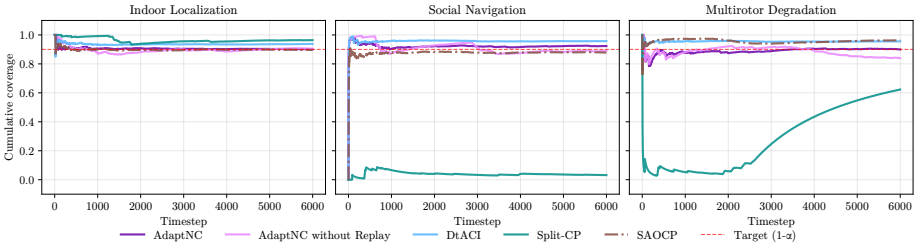



Figure 3: This figure illustrates the evolution of empirical coverage over the evaluation horizon relative to the target coverage level (shown in red) for a representative seed. AdaptNC's performance highlights that score-function adaptation enables sustained coverage with tighter regions.

because the underlying localization model fails to account for realistic wireless channel dynamics which vary over time. We additionally evaluate abrupt RSSI attenuation in this environment, comparing AdaptNC with and without replay across three attenuation levels; details are provided in Section E.2.6. In the **Social Navigation** setting, 8 agents navigate a shared environment based on information from agents within a collaboration radius. A trajectory predictor seeks to estimate the future poses of the agents. The distribution shift in this scenario stems from a gradual increase in the collaboration radius. In the **Multirotor Tracking** experiment, a predictor seeks to estimate the future position of a multirotor tracking a reference trajectory while maintaining a fixed altitude, but encounters distribution shift due to gradual degradation in the multirotor actuators. These three experiments showcase AdaptNC's benefits in settings due to environmental changes, policy shift in multi-agent settings, and actuator degradation. Further details are in Appendices E.2, E.3, and E.4.

For each case study, we evaluate all methods across 50 different random seeds to assess performance variability and robustness. For each seed, evaluation is conducted over a single 6000-step rollout without resets, and results are aggregated across all seeds.

**Evaluation Metrics**: **(1) Global Coverage**, defined as $1 - \frac{1}{T}\sum_{t=1}^{T} \mathbf{err}_t$, which measures recovery of the target coverage level, **(2)** Average Finite Volume per Covered Timestep, which quantifies conservatism. **(3) Local Coverage**, defined by $1 - \frac{1}{w}\sum_{i=t-\frac{w}{2}+1}^{t+\frac{w}{2}} \mathbf{err}_i$, a sliding-window average of size $w = 100$ that captures temporal behavior under distribution shift; we report its mean over all complete windows. We separately report the percentage of timesteps with an infinite threshold, or vacuous coverage, in Table 8.

**Baselines**: Our main comparison includes six baselines and ablations. We additionally evaluate fixed-step ACI across ten step sizes and Scale-Free Online Gradient Descent (SF-OGD), with results reported in Tables 9 and 10. **(1) DtACI:** We use DtACI [6] as our primary baseline for online conformal prediction under distribution shift. **(2) Split Conformal Prediction:** assumes exchangeability and does not account for distribution shift. **(3) AdaptNC without Replay:** an ablation to assess the role of replay in stabilizing adaptation and mitigating coverage shock. **(4) AdaptNC without Adaptive Reweighting (AdaptNC Uniform):** an ablation where score adaptation occurs at fixed intervals, and equal weight is assigned to historical data within the window $W$. No replay is performed in this setting. **(5) Strongly Adaptive Online Conformal Prediction (SAOCP):** [16] is a method that instantiates experts of varying lifespans, each of which aims to adapt the conformal threshold using online gradient descent in response to changing data distributions. **(6) Shape-CP:** [10] uses optimized but static nonconformity scores, highlighting the limitations of static calibration under distribution shift.

Table 1: This table reports mean $\pm$ population standard deviation across 50 evaluation seeds. Among the methods shown, AdaptNC achieves global coverage closest to the $90\%$ target in all three environments. It also produces smaller average finite volumes per covered timestep than DtACI, SAOCP, and both AdaptNC ablations.

| | Indoor Localization | | | Social Navigation | | | Multirotor Tracking | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Global Cov. (%) | Vol. ($m^2$) | Mean Local Cov. (%) | Global Cov. (%) | Vol. ($m^2$) | Mean Local Cov. (%) | Global Cov. (%) | Vol. ($m^2$) | Mean Local Cov. (%) |
| **ANC** | **90.04 ± 0.30** | **35.78 ± 2.87** | **89.99 ± 0.31** | **91.25 ± 1.03** | **41.96 ± 7.15** | **91.23 ± 1.04** | **90.13 ± 0.85** | **0.0394 ± 0.0102** | **90.15 ± 0.85** |
| ANC NR | 90.13 ± 1.65 | 43.90 ± 4.78 | 90.05 ± 1.68 | 85.28 ± 2.77 | 80.35 ± 12.85 | 85.19 ± 2.78 | 85.19 ± 0.90 | 0.1351 ± 0.0329 | 85.39 ± 0.91 |
| DtACI | 93.47 ± 0.22 | 39.26 ± 3.29 | 93.43 ± 0.21 | 95.40 ± 0.26 | 50.83 ± 5.01 | 95.38 ± 0.27 | 95.26 ± 0.24 | 0.2973 ± 0.0106 | 95.26 ± 0.24 |
| Sp. CP | 97.12 ± 0.85 | 98.91 ± 3.12 | 97.12 ± 0.86 | 2.99 ± 0.78 | 0.71 ± 0.02 | 2.97 ± 0.81 | 60.11 ± 1.57 | 0.2090 ± 0.0001 | 60.17 ± 1.59 |
| Sh. CP | 96.91 ± 1.07 | 102.09 ± 7.03 | 96.92 ± 1.06 | 2.98 ± 0.82 | 0.70 ± 0.02 | 2.96 ± 0.84 | 0.68 ± 0.24 | 0.0027 ± 0.0003 | 0.54 ± 0.24 |
| ANC uniform | 91.65 ± 0.94 | 49.57 ± 4.94 | 91.60 ± 0.93 | 84.80 ± 2.16 | 83.70 ± 12.58 | 84.75 ± 2.16 | 91.19 ± 1.01 | 0.1204 ± 0.0317 | 91.23 ± 1.00 |
| SAOCP | 89.73 ± 0.44 | 44.45 ± 3.45 | 89.69 ± 0.44 | 88.22 ± 0.87 | 51.65 ± 5.31 | 88.22 ± 0.87 | 96.56 ± 0.41 | 0.2954 ± 0.0072 | 96.58 ± 0.41 |

**Abbreviations:** ANC= AdaptNC; NR = No replay; Sh. CP = Shape CP; Sp. CP = Split CP; CP = Conformal Prediction.

SAOCP and SF-OGD directly adapt finite score thresholds, whereas DtACI and AdaptNC can return infinite quantile thresholds. SAOCP and SF-OGD assume bounded scores; therefore, their zero vacuity rates do not establish stronger coverage performance under distribution shift [16].

### 6.1 Performance Evaluation and Comparative Analysis

Table 1 shows that AdaptNC achieves global coverage closest to the $90\%$ target among the methods shown, together with smaller average finite volumes per covered timestep than DtACI and SAOCP.

**Global and Local Coverage:** Across 50 evaluation seeds, AdaptNC achieves mean global coverage of $90.04\%$, $91.25\%$, and $90.13\%$ in indoor localization, social navigation, and multirotor tracking, respectively. These values are closest to the $90\%$ target among the methods in Table 1. DtACI exceeds the target in all three environments while producing larger average finite volumes per covered timestep. Conversely, Split CP and Shape-CP exhibit severe undercoverage in social navigation, with coverage of approximately $3\%$. Figures 2 and 3 show the temporal evolution of local and global coverage for a representative seed in each environment, complementing the aggregate results.

**Uncertainty Volume:** AdaptNC produces smaller finite uncertainty regions at covered timesteps than DtACI in all three environments. The respective mean volumes are 35.78 versus 39.26 for indoor localization, 41.96 versus 50.83 for social navigation, and 0.0394 versus 0.2973 for multirotor tracking. Figure 4 illustrates selected finite prediction regions qualitatively.

**Importance of Replay:** Table 1 demonstrates the benefits of replay across 50 evaluation seeds. Replay brings mean global coverage closer to the target in all three environments, with the largest improvements in social navigation and multirotor tracking: coverage increases from $85.28\%$ to $91.25\%$ and from $85.19\%$ to $90.13\%$, respectively. Replay also reduces the across-seed standard deviation of global coverage and the average finite volume per covered timestep in every environment. Figures 2 and 3 illustrate the corresponding coverage trajectories for representative seeds. These results highlight the importance of replay for recovering near-target coverage and tighter finite uncertainty regions when adapting the score function.

**Runtime Analysis:** We report runtimes in Table 2, averaged over 50 seeds on a desktop workstation (AMD Ryzen 7900X, 64GB RAM). AdaptNC's threshold path averages less than $0.1$ ms in all environments. The score path, including score fitting and replay, averages $130.28$, $124.49$, and $125.92$ ms for indoor localization, social navigation, and multirotor tracking, respectively. Score updates occur periodically—every $40$, $80$, and $80$ steps in these environments—while threshold adaptation continues at every timestep. This separation yields whole-run mean costs of $4.146$, $2.282$, and $2.200$

Table 2: Recorded runtimes **in milliseconds**, reported as mean $\pm$ population standard deviation across 50 evaluation seeds. Whole-method rows report recorded runtime divided by rollout length. AdaptNC and its ablations include initialization and calibration; DtACI and SAOCP measure online evaluation and updates; Split-CP and Shape-CP amortize batch calibration/fitting and evaluation over the rollout. Threshold-path and score-path rows report mean time per call to the indicated path.

| Method / path | Indoor Localization | Multirotor Tracking | Social Navigation |
|---|---|---|---|
| AdaptNC | 4.146 ± 2.542 | 2.200 ± 1.161 | 2.282 ± 1.162 |
| AdaptNC, no replay | 1.292 ± 0.141 | 14.739 ± 1.614 | 1.761 ± 0.479 |
| DtACI | 0.206 ± 0.004 | 0.165 ± 0.007 | 0.206 ± 0.070 |
| Split-CP | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 |
| Shape-CP | 1.738 ± 0.531 | 1.535 ± 0.264 | 1.236 ± 0.253 |
| AdaptNC, uniform window | 1.079 ± 0.132 | 2.643 ± 0.469 | 1.199 ± 0.177 |
| SAOCP | 0.314 ± 0.006 | 0.311 ± 0.009 | 0.319 ± 0.031 |
| AdaptNC: threshold path | 0.089 ± 0.002 | 0.075 ± 0.002 | 0.081 ± 0.002 |
| AdaptNC: score path | 130.283 ± 100.845 | 125.921 ± 91.862 | 124.485 ± 92.639 |
| No replay: threshold path | 0.154 ± 0.003 | 0.158 ± 0.005 | 0.142 ± 0.002 |
| No replay: score path | 7.235 ± 0.218 | 7.324 ± 0.119 | 6.976 ± 0.700 |

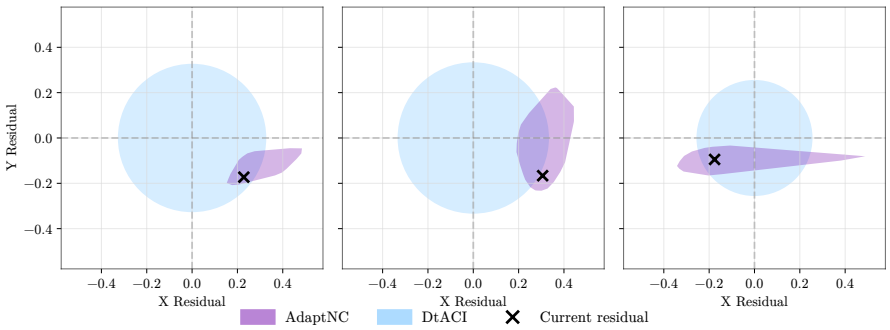



Figure 4: This figure shows uncertainty regions from AdaptNC and DtACI at representative timesteps ($t \in \{720, 2218, 3193\}$) in the multirotor tracking task for a representative seed, along with the realized residual. These timesteps were selected to illustrate finite prediction-region geometries. **AdaptNC recovers tighter regions containing the realized residuals**, illustrating the benefit of adapting region geometry, whereas DtACI scales a fixed circular region.

ms per timestep, respectively, including initialization and calibration. Efficient online computation is also central to safe robotics methods such as Control Barrier Function (CBF)-based safety filters [24], which enforce safety constraints through real-time optimization. AdaptNC addresses the computational demands of adapting uncertainty regions through inexpensive threshold updates and less frequent updates to region geometry. Further discussion is provided in Appendices B.6 and B.7.

**Hyperparameter Selection:** For each environment, we jointly select the history-window size $W$ and score-update interval $t_s$ over the 42-configuration Cartesian grid $W \in \{25, 50, 75, 100, 125, 150\}$ and $t_s \in \{20, 40, 60, 80, 100, 120, 150\}$ using three pilot rollouts with seeds 0, 1, and 2. We minimize the absolute difference between mean pilot coverage and the $90\%$ target. Exact ties are resolved by lower mean finite prediction-region volume, followed by smaller $W$ and smaller $t_s$. The selected settings are fixed across 50 evaluation rollouts whose seeds are disjoint from the pilot seeds. The same settings are used for the AdaptNC ablations, while DtACI and fixed-rate ACI share the selected reference-window size.

**Hyperparameter Sensitivity:** Table 7 in Appendix E.5 reports slices of the joint grid search through the selected configurations, evaluated on the three pilot seeds. These validation results are distinct from the 50-seed evaluation. The effects on coverage and volume differ across environments, motivating environment-specific validation rather than a claim of insensitivity to these parameters.

**Results Summary:** Across all three case studies, AdaptNC achieves mean global coverage near the target under structural distribution shifts while producing smaller average finite volumes per covered timestep than DtACI and SAOCP. The replay ablation demonstrates the contribution of replay to recovering near-target coverage and tighter finite regions during score adaptation. These finite-volume gains coexist with more frequent vacuous predictions, as reported in Table 8 and discussed in the limitations. The expert weight dynamics in Figures 10, 11, and 12 illustrate how the allocation among adaptation rates evolves over each rollout. Appendices E.2, E.3, and E.4 provide additional details.

## 7 Conclusion

We present AdaptNC, an online conformal prediction method that adapts nonconformity scores under distribution shift. It leverages a reweighted data distribution and a replay mechanism to stabilize adaptation and maintain coverage as scores evolve. We establish regret and long-term coverage guarantees and validate the method on three robotics case studies involving **environmental change, multi-agent interaction,** and **system degradation**. Across all three environments, AdaptNC achieves global coverage closer to the target and smaller average finite volumes per covered timestep than DtACI, demonstrating the benefit of jointly adapting the score function and conformal threshold. Future work will focus on improving computational efficiency and addressing the limitations discussed in Appendix A.

## Acknowledgments and Disclosure of Funding

This work was partially supported by US DoT Safety21 National University Transportation Center and NSF grants CISE-2431569. Renukanandan Tumu was supported by NSF GRFP award DGE-2236662.

## References


[1] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(Volume 5, 2022):411–444, May 2022. ISSN 2573-5144. doi: 10.1146/annurev-control-042920-020211. URL https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-020211.

[2] Lars Lindemann, Yiqi Zhao, Xinyi Yu, George J. Pappas, and Jyotirmoy V. Deshmukh. Formal verification and control with conformal prediction: Practical safety guarantees for autonomous systems. *IEEE Control Systems*, 45(6):72–122, 2025. doi: 10.1109/MCS.2025.3611545.

[3] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer International Publishing, Cham, 2022. ISBN 978-3-031-06648-1 978-3-031-06649-8. doi: 10.1007/978-3-031-06649-8. URL https://link.springer.com/10.1007/978-3-031-06649-8.

[4] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113(523):1094–1111, July 2018. ISSN 0162-1459. doi: 10.1080/01621459.2017.1307116. URL https://doi.org/10.1080/01621459.2017.1307116. _eprint: https://doi.org/10.1080/01621459.2017.1307116.

[5] Isaac Gibbs and Emmanuel Candes. Adaptive Conformal Inference Under Distribution Shift. In *Advances in Neural Information Processing Systems*, volume 34, pages 1660–1672. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html.

[6] Isaac Gibbs and Emmanuel J. Candès. Conformal Inference for Online Prediction with Arbitrary Distribution Shifts. *Journal of Machine Learning Research*, 25(162):1–36, 2024. ISSN 1533-7928. URL http://jmlr.org/papers/v25/22-1218.html.

[7] Margaux Zaffran, Olivier Feron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. Adaptive Conformal Predictions for Time Series. In *Proceedings of the 39th International Conference on Machine Learning*, pages 25834–25866. PMLR, June 2022. URL https://proceedings.mlr.press/v162/zaffran22a.html.

[8] Shayan Kiyani, George Pappas, and Hamed Hassani. Length Optimization in Conformal Prediction. *Advances in Neural Information Processing Systems*, 37:99519–99563, December 2024. doi: 10.52202/079017-3158. URL https://doi.org/10.52202/079017-3158.

[9] Chao Gao, Liren Shan, Vaidehi Srinivas, and Aravindan Vijayaraghavan. Volume Optimality in Conformal Prediction with Structured Prediction Sets. June 2025. URL https://openreview.net/forum?id=oNDhnGrD51&noteId=7kR09SC5BY.

[10] Renukanandan Tumu, Matthew Cleaveland, Rahul Mangharam, George Pappas, and Lars Lindemann. Multi-modal conformal prediction regions by optimizing convex shape templates. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, pages 1343–1356. PMLR, June 2024. URL https://proceedings.mlr.press/v242/tumu24a.html.

[11] Renukanandan Tumu, Lars Lindemann, Truong Nghiem, and Rahul Mangharam. Physics Constrained Motion Prediction with Uncertainty Quantification. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, Anchorage, AK, USA, June 2023. IEEE. ISBN 979-8-3503-4691-6. doi: 10.1109/IV55152.2023.10186812. URL https://ieeexplore.ieee.org/document/10186812/.

[12] Matthew Cleaveland, Insup Lee, George J. Pappas, and Lars Lindemann. Conformal Prediction Regions for Time Series Using Linear Complementarity Programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):20984–20992, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i19.30089. URL https://ojs.aaai.org/index.php/AAAI/article/view/30089.

[13] Sacha Braun, Liviu Aolaritei, Michael I. Jordan, and Francis Bach. Minimum Volume Conformal Sets for Multivariate Regression, March 2025. URL http://arxiv.org/abs/2503.19068. arXiv:2503.19068 [stat].

[14] Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+, May 2020. URL http://arxiv.org/abs/1905.02928. arXiv:1905.02928 [stat].

[15] Osbert Bastani, Varun Gupta, Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Practical Adversarial Multivalid Conformal Prediction. October 2022. URL https://openreview.net/forum?id=QNjyrDBx6tz.

[16] Aadyot Bhatnagar, Huan Wang, Caiming Xiong, and Yu Bai. Improved Online Conformal Prediction via Strongly Adaptive Online Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2337–2363. PMLR, July 2023.

[17] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal Prediction Under Covariate Shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html.

[18] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, April 2023. ISSN 0090-5364, 2168-8966. doi: 10.1214/23-AOS2276. URL https://doi.org/10.1214/23-AOS2276.

[19] Muhammad Faaiz Taufiq, Jean-Francois Ton, Rob Cornish, Yee Whye Teh, and Arnaud Doucet. Conformal Off-Policy Prediction in Contextual Bandits. *Advances in Neural Information Processing Systems*, 35:31512–31524, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cc84bfabe6389d8883fc2071c848f62a-Abstract-Conference.html.

[20] Tom Kuipers, Renukanandan Tumu, Shuo Yang, Milad Kazemi, Rahul Mangharam, and Nicola Paoletti. Conformal Off-Policy Prediction for Multi-Agent Systems. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 1067–1074, December 2024. doi: 10.1109/CDC56724.2024.10886791. URL https://ieeexplore.ieee.org/document/10886791. ISSN: 2576-2370.

[21] Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. ISSN 0003-4851. URL https://www.jstor.org/stable/2237880.

[22] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996. ISSN 0098-3500. doi: 10.1145/235815.235821. URL https://dl.acm.org/doi/10.1145/235815.235821.

[23] Björn Lindqvist, Sina Sharif Mansouri, Ali-akbar Agha-mohammadi, and George Nikolakopoulos. Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles. *IEEE Robotics and Automation Letters*, 5(4):6001–6008, October 2020. ISSN 2377-3766. doi: 10.1109/LRA.2020.3010730. URL https://ieeexplore.ieee.org/abstract/document/9145644.

[24] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017. doi: 10.1109/tac.2016.2638961.

[25] A. W. Van Der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1 edition, October 1998. ISBN 978-0-511-80225-6 978-0-521-49603-2 978-0-521-78450-4. doi: 10.1017/CBO9780511802256. URL https://www.cambridge.org/core/product/identifier/9780511802256/type/book.

[26] Yury Polyanskiy and Yihong Wu. *Information theory: from coding to learning*. Cambridge University Press, Cambridge, United Kingdom ; New York, NY, 2025. ISBN 978-1-108-83290-8 978-1-108-96635-1. doi: 10.1017/9781108966351.

[27] Theodore S Rappaport. *Wireless communications: Principles and practice, 2/E*. Pearson Education India, 2010.

[28] Mikael Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronics letters*, 27(23):2145–2146, 1991.

[29] William C Jakes and Donald C Cox. *Microwave mobile communications*. Wiley-IEEE press, 1994.

[30] Richard Hedley Clarke. A statistical theory of mobile-radio reception. *Bell system technical journal*, 47(6):957–1000, 1968.

## Appendix Contents


**A Limitations** **15**

**B Additional Clarifications and Extended Discussion** **15**

B.1 AdaptNC is not a direct combination of DtACI and prior score-adaptation methods 15

B.2 AdaptNC's Theoretical Support . . . . . 15

B.3 Role of the Low-Dimensional Assumption in Complexity Analysis . . . . . 16

B.4 Effects of Vacuous Prediction Regions on the Next Cycle of Learning and Their Mitigation . . . . . 16

B.5 Detecting Distribution Shift and Adapting to its Rate . . . . . 16

B.6 Adaptation Between Score Updates . . . . . 16

B.7 Tackling More Complex Tasks and Computational Considerations . . . . . 16

**C Method** **18**

C.1 Score Optimization Algorithm . . . . . 18

C.2 AdaptNC Algorithm . . . . . 18

C.3 Dynamically Tuned Adaptive Conformal Inference (DtACI) . . . . . 18

C.4 High Density Region Identification . . . . . 18

C.5 Computational Complexity Analysis of AdaptNC . . . . . 19

**D Theoretical Details and Proofs** **21**

D.1 Background ACI and DtACI Guarantees . . . . . 21

D.2 HDR and Score-Stability Proofs . . . . . 21

D.3 Long-Run Coverage under Replay . . . . . 24

D.4 Coverage Shock from Score Updates . . . . . 25

**E Experiments** **26**

E.1 Gaussian Mixture Model: A Minimal Example of Coverage Shock . . . . . 26

E.2 Indoor Localization . . . . . 29

E.3 Social Navigation . . . . . 33

E.4 Multirotor Tracking . . . . . 36

E.5 Hyperparameter Sensitivity Analysis . . . . . 38

# A Limitations

While AdaptNC demonstrates the ability to recover tight, valid uncertainty regions under distribution shift, it remains, as with other adaptive conformal prediction methods, susceptible to producing trivial uncertainty sets. These sets are not entirely uninformative, as their occurrence can serve as an operational signal that the environment is undergoing change, however their frequency represents a limitation. AdaptNC returns vacuous regions on 15.55%, 33.12%, and 44.30% of timesteps in indoor localization, social navigation, and multirotor tracking, respectively, compared with 14.00%, 17.74%, and 33.88% for DtACI. Thus, smaller finite-region volumes coexist with more frequent unbounded predictions. We emphasize that vacuous coverage timesteps serve as indicators of a potential regime change in the environment, at which point the system can transition to a fallback safe controller during the transient phase. We believe that score function regularization may be able to address this issue, and consider it an avenue for future work.

# B Additional Clarifications and Extended Discussion

## B.1 AdaptNC is not a direct combination of DtACI and prior score-adaptation methods

While AdaptNC builds upon these important prior works, it is not merely a direct combination of these methods. The key novelty of AdaptNC lies in identifying and addressing a new setting that prior work does not consider: the joint, coupled adaptation of both the nonconformity score and the conformal threshold under distribution shift. This coupling introduces a non-trivial feedback loop. Changes in the score function alter the induced nonconformity distribution, which subsequently shifts the optimal quantile and destabilizes threshold updates. As a result, the update dynamics of prior independent methods no longer perform their function in the same manner. To support this new setting, we introduced an adaptive reweighting scheme that enables fully online score optimization, alongside new theoretical results including score function stability under joint adaptation (Theorem 5.3) in an asymptotic setting, using a guarantee on the stability of the score function adaptation in (Theorem B.1).

Furthermore, the direct amalgamation of DtACI with score adaptation, corresponds to our "AdaptNC without Replay" baseline, which includes our historical data reweighting scheme. As demonstrated in our ablation studies, this naive combination fails, exhibiting substantial misses in coverage and significantly inflated prediction regions because composition alone is insufficient to handle the feedback problem created through this joint optimization. The intuition behind this is presented in (Remark 5.5), which shows that there are two sources of shift in the score distribution; the bounded component due to the additional data point observed at the next timestep, and the potentially unbounded component which is due to the shift in the score function. To solve this, we introduced the counterfactual replay mechanism. By reinitializing and recalibrating the DtACI expert chain using recent data under the updated score, the replay mechanism prevents coverage shocks caused by mismatched quantiles. As shown empirically, replay is not a minor implementation detail but a fundamentally necessary component that restores both valid coverage and tight uncertainty sets.

Overall, AdaptNC goes significantly beyond combining existing techniques by formulating a new problem setting of coupled adaptation, identifying the instabilities of a naive synthesis, and introducing the novel mechanisms required to make joint adaptation both stable and effective.

## B.2 AdaptNC's Theoretical Support

AdaptNC's various components have asymptotic guarantees. It is important to note that finite-sample guarantees are generally unavailable in online conformal prediction (CP) due to the absence of assumptions on distribution shift. Guarantees are often given in the asymptotic setting. [6] show that the regret of their expert weighting algorithm will have a bounded regret to the best expert. [5] establish asymptotic convergence of each expert to the target coverage. In this asymptotic stability setting, we guarantee that our method yields tight and consistent prediction regions. This is shown in Theorem 5.2 and Corollary 5.3. Taken together, our method provides convergence guarantees in the asymptotic setting, including for the score adaptation procedure. Our theoretical analysis gives sufficient conditions under which replay induces a vanishing average perturbation of the ACI expert recursion. These conditions are asymptotic and rely on boundary anti-concentration; finite-sample replay guarantees remain an important direction for future work.

### B.3 Role of the Low-Dimensional Assumption in Complexity Analysis

The assumption $d \leq 3$ in Section 5.4 is introduced to derive tractable bounds for the geometric operations underlying AdaptNC, particularly convex hull construction via the QuickHull algorithm. In low dimensions, QuickHull exhibits near-linear complexity, which supports the stated amortized per-timestep bound. This assumption is therefore analytical in nature and is used to characterize the regime where AdaptNC achieves favorable computational efficiency.

Importantly, AdaptNC itself does not depend on $d \leq 3$ and remains applicable in higher dimensions, although the associated geometric computations scale less favorably. In particular, QuickHull has complexity $O(n^{\lfloor d/2 \rfloor})$ for $d > 3$, where $\lfloor \cdot \rfloor$ denotes the greatest integer function. However, this growth remains manageable up to moderate dimensions (e.g., $d \leq 5$), where the scaling is at most quadratic, consistent with the empirical runtimes in Table 2. Moreover, in autonomous systems applications, which are the primary focus of this work, conformal prediction is typically performed over low-dimensional predictors such as **scalar** value functions or spatial trajectory predictors (up to 3 dimensions), and thus AdaptNC remains applicable without incurring prohibitive computational overhead.

### B.4 Effects of Vacuous Prediction Regions on the Next Cycle of Learning and Their Mitigation

The "infinite" safety margins or vacuous bounds do not corrupt the next cycle of learning, as the underlying data is not perturbed or destroyed. These vacuous bounds are often encountered when the system undergoes change, or sees significant shift in error patterns. When the non-conformity scores the system sees are very large, larger than those we have previously seen, AdaptNC correctly identifies that the system is undergoing change, and surfaces to the user that the change is major, and the system cannot yet accurately predict what will happen. In this way, vacuous bounds *are* informative; they show that AdaptNC cannot provide guarantees in high-change regimes. After this period of change is complete, AdaptNC will return to predicting safety regions, as the residuals stabilize. Finally, prediction of these vacuous bounds happens when the current quantile prediction $\bar{\alpha}_t$ becomes very small. The data is preserved in its unperturbed state in the history buffer $\mathcal{H}$. These independent mechanisms ensure that data is kept in a manner that permits learning from samples, even if the predicted safety margin is infinite.

### B.5 Detecting Distribution Shift and Adapting to its Rate

Identifying when the data distribution has shifted is one of the major problems in the online conformal prediction setting. There are advantages to having more data, namely that more data allows for more data to develop bounds on the non-conformity scores. However, using old or stale data can yield inaccurate coverage which can lead to catastrophe for robots. AdaptNC is fundamentally a weighted mixture of a series of “expert" predictors. Each of these “experts" functions under the hypothesis that the distribution is shifting at a different rate, characterized by $\gamma$. A lower value of $\gamma$ indicates a low expected rate of change, and a higher $\gamma$ indicates a higher expected rate of change. AdaptNC uses the weights allocated to each expect to dynamically identify how quickly the distribution is changing, and use that estimate to change how much historical data it uses at every timestep. This is also illustrated in the expert weight dynamics (Figures 10, 11, and 12).

### B.6 Adaptation Between Score Updates

Between score updates, AdaptNC retains the current score function and updates the conformal threshold after each observed outcome. The updated threshold is used for the next prediction, allowing region scale to adapt before the next geometry update. At a scheduled score update, the method refits the score and replays recent observations to recalibrate the threshold state. The abrupt-shift experiment in Section E.2.6 evaluates this adaptation following an instantaneous change in the observation model.

### B.7 Tackling More Complex Tasks and Computational Considerations

AdaptNC separates frequent threshold adaptation from periodic score fitting and replay. The threshold path averages less than $0.1$ ms in all three environments, while the score path averages

approximately $124$–$130$ ms per call. The selected score-update interval $t_s$ is $40$ steps for indoor localization and $80$ steps for social navigation and multirotor tracking. This separation allows the method to adapt region scale at every timestep while updating region geometry less frequently. Table 2 reports both the per-call path costs and the whole-run average costs.

# C Method

## C.1 Score Optimization Algorithm

**Algorithm 1** OptimizeScore: Score Parameter Optimization

**Require:** History buffer $\mathcal{H}_t^w$, current parameters $\theta$, target miscoverage rate $\alpha$

1: Generate high-density points $\hat{R}_t$ covering proportion $1-\alpha$ of density mass through Algorithm 4
2: Fit shape template $\mathcal{S}$ using QuickHull algorithm on $\hat{R}_t$
3: $\theta_{t+1} \leftarrow$ parameters encoding shape templates $\{\mathcal{S}\}$
4: **Return:** Updated parameters $\theta_{t+1}$

## C.2 AdaptNC Algorithm

**Algorithm 2** AdaptNC Algorithm

**Require:** Target miscoverage rate $\alpha$, adaptation interval $t_s$, initial parameters $\theta_0$, DtACI expert gammas $\Gamma = [\gamma_1, \ldots, \gamma_k]$, window size $W$, learning rate $\eta$, calibration data $\mathcal{D}_{cal}$

```
Initialize history buffer H ← ∅
Fit initial conformal region with θ_0 on D_cal
Initialize DtACI with s(·; θ_0), α, Γ, W, η, D_cal
for t = 1, 2, ..., T do
    Observe X_t, produce prediction Ŷ_t
    Output prediction region Ĉ_t(X_t; θ_t, q_{1−α_t, D_t})
    Observe Y_t
    Append (X_t, Y_t, s_t) to H
    Compute score s_t = s(X_t, Y_t; θ_t)
    Update DtACI with (X_t, Y_t) to obtain ᾱ_t
    Obtain new threshold: q_{1−α_t, D}
    if t mod t_s = 0 then
        Compute distribution H_t^ω using Eq. (4)
        θ_{t+1} ← OPTIMIZESCORE(H_t^ω, θ_t, α)
        Re-initialize DtACI with score function s(·; θ_{t+1})
        for each (X_i, Y_i) ∈ [t − w + 1, t] do
            Update experts and expert weights with (X_i, Y_i)
        end for
    end if
end for
```

## C.3 Dynamically Tuned Adaptive Conformal Inference (DtACI)

See Algorithm 3.

## C.4 High Density Region Identification

Under standard KDE regularity conditions and sufficiently dense Monte Carlo sampling, the retained high-density samples approximate the corresponding KDE level set. We therefore model the score-optimization step through Assumption C.1.

**Assumption C.1** (Consistent residual-template estimation)**.** At score-update times $t_m$, let $\widehat{R}_m$ be the residual set returned by the score-optimization procedure and let $K_m = \mathrm{conv}(\widehat{R}_m)$. Let $R_\star = \{z : f_\star(z) \geq \tau_\star\}$ and $K_\star = \mathrm{conv}(R_\star)$. We assume

$$d_H(K_m, K_\star) \xrightarrow{p} 0.$$

**Theorem C.2** (Glivenko-Cantelli Theorem as written in Vaart [25])**.** *If $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables with cumulative distribution function $f$, and empirical distribution function $f_n$, then:*

$$\|f_n - f\|_\infty \xrightarrow{as} 0 \tag{8}$$

**Algorithm 3** Dynamically tuned Adaptive Conformal Inference (DtACI) [6, Algorithm 2]

**Require:** Observed values $\{\beta_t\}_{1\le t\le T}$, set of candidate $\gamma$ values $\{\gamma_i\}_{1\le i\le k}$, starting points $\{\alpha_1^i\}_{1\le i\le k}$, and parameters $\sigma$ and $\eta$.

$w_1^i \leftarrow 1,\ 1 \le i \le k$;
**for** $t = 1, 2, \ldots, T$ **do**
  Define the probabilities $p_t^i := w_t^i / \sum_{1\le j\le k} w_t^j, \quad \forall 1 \le i \le k$;
  Output $\bar{\alpha}_t = \sum_{1\le i\le k} p_t^i \alpha_t^i$;
  $\bar{w}_t^i \leftarrow w_t^i \exp(-\eta \ell(\beta_t, \alpha_t^i)), \quad \forall 1 \le i \le k$;
  $\bar{W}_t \leftarrow \sum_{1\le i\le k} \bar{w}_t^i$;
  $w_{t+1}^i \leftarrow (1-\sigma)\bar{w}_t^i + \bar{W}_t \sigma / k$;
  $\mathbf{err}_t^i := \mathbb{1}\{Y_t \notin \hat{C}_t(\alpha_t^i; \theta_t, \mathcal{D}_t)\}, \quad \forall 1 \le i \le k$;
  $\mathbf{err}_t := \mathbb{1}\{Y_t \notin \hat{C}_t(\bar{\alpha}_t; \theta_t, \mathcal{D}_t)\}$;
  $\alpha_{t+1}^i = \alpha_t^i + \gamma_i(\alpha - \mathbf{err}_t^i), \quad \forall 1 \le i \le k$;
**end for**

**Algorithm 4** Monte Carlo Kernel Density Estimation (MCKDE)

**Input:** Dataset $X \in \mathbb{R}^{N\times d}$, sample weights $w \in \mathbb{R}^N$, Monte Carlo samples $M$, miscoverage rate $\alpha$, bandwidth factor $\beta$.
**Output:** Set of high-density samples $\hat{R}_{N,M}$.
**// 1. Bandwidth Selection**
**if** bandwidth method is "Scott" **then**
  $h \leftarrow N^{\frac{-1}{d+4}}$
**else if** bandwidth method is "Silverman" **then**
  $h \leftarrow \left(\frac{N(d+2)}{4}\right)^{\frac{-1}{d+4}}$
**end if**
$h \leftarrow h \times \beta$
**// 2. Fit Kernel Density Estimator**
Fit weighted Gaussian KDE $\hat{f}_N$ using $(X, w)$
**// 3. Monte Carlo Sampling & Scoring**
Draw $M$ samples $\mathcal{Z} = \{z_1, \ldots, z_M\}$ where $z_i \sim \hat{f}_N$
Calculate density scores $\xi_i \leftarrow \hat{f}_N(z_i)$ for $i = 1, \ldots, M$
**// 4. Thresholding**
Let $\tau$ be the $\alpha$-quantile of the set $\{\xi_1, \ldots, \xi_M\}$
$\hat{R}_{N,M} \leftarrow \{z_i \in \mathcal{Z} \mid \xi_i \ge \tau\}$
**return** $\hat{R}_{N,M}$

### C.5 Computational Complexity Analysis of AdaptNC

**Theorem C.3** (Computational Complexity of AdaptNC)**.** *AdaptNC consists of two components: a threshold update executed at every timestep and a score function update executed every $t_s$ timesteps. Let $W$ denote the window size, $k$ the number of experts, $M$ the number of Monte Carlo samples, $d$ the residual dimension, $r$ the number of facets of the convex hull, and $n = |\hat{R}_t|$ the number of retained high-density points. Then, the per-timestep cost of the threshold update is $O(r + W + k)$, and the cost of the score function update is $O(MWd + M\log M + n\log n + W(r+k))$, which is dominated by the KDE evaluation term $O(MWd)$ in the low-dimensional regime considered here.*

*Under the assumption $d \le 3$, the amortized per-timestep cost of AdaptNC is*

$$O(W + k + MWd/t_s)\,.$$

*Proof.* We analyze the computational cost of AdaptNC by decomposing it into two components: the threshold update and the score function update. We assume that the residual dimensionality satisfies $d \le 3$, which holds for all experiments and typical real-world trajectory prediction tasks.

**Threshold update:** The threshold update executes at every timestep and consists of three operations. First, the nonconformity score $s(X_t, Y_t; \theta_t)$ is evaluated for the current observation, requiring $O(r)$ time where $r$ is the number of facets of the convex hull. Second, the inverse quantile $\beta_t$ is computed by evaluating the score against the rolling window $\mathcal{D}_t$, requiring $O(W)$ time. Third, the $k$ expert weights are updated via exponential reweighting and each expert's quantile $\alpha_t^i$ is updated, requiring $O(k)$ time. The total per-timestep cost of the fast path is therefore $O(r + W + k)$.

**Score function update:** The score function update executes every $t_s$ timesteps and consists of three stages: reweighting, score optimization, and replay.

*Reweighting:* Computing the adaptive weights $\omega_\tau$ for all history buffer entries requires evaluating the exponential decay across $k$ experts for each of the $W$ buffered points, yielding $O(kW)$.

*Score optimization:* This stage comprises density estimation and shape template fitting. The weighted KDE is constructed from $W$ samples in $O(Wd)$ time. Drawing $M$ Monte Carlo samples from the KDE costs $O(Md)$. Evaluating the KDE at each Monte Carlo sample requires summing over all $W$ kernel centers, yielding $O(MWd)$. Selecting the top $1 - \alpha$ fraction by density requires sorting in $O(M \log M)$. Finally, the QuickHull algorithm fits the convex hull to the $n$ retained high-density points in $O(n \log n)$ expected time. The dominant cost of this stage is the KDE evaluation at $O(MWd)$.

*Replay:* The replay mechanism re-initializes the DtACI expert chain and processes the $W$ most recent observations under the updated score function. Each replayed observation requires $O(r + k)$ for score evaluation and expert updates, yielding a total replay cost of $O(W(r + k))$.

The total cost of the score update is $O(MWd + M \log M + n \log n + W(r + k))$, dominated by the KDE evaluation term $O(MWd)$.

**Amortized cost:** Since the score function update is executed every $t_s$ timesteps, its per timestep cost amortizes to: $O(MWd/t_s)$. Adding the threshold update cost yields the amortized per-timestep cost of AdaptNC:

$$O(W + k + MWd/t_s)$$

□

# D Theoretical Details and Proofs

## D.1 Background ACI and DtACI Guarantees

We present the theorem that yields Remark D.2

**Theorem D.1** (Theorem 4, [6]). *Let $\gamma_{max} := \max_{1\le i\le k}\gamma_t$ and assume that $\gamma_1 < \gamma_2 < ... < \gamma_k$ where $\gamma_{i+1}/\gamma_i \le 2$ for all $1 < i \le k$, and that $\ell(\beta_t, \theta) := \alpha(\beta_t - \theta) - \min\{0, \beta_t - \theta\}$ Then, for any window $W = [r, s] \in [T]$ and any sequence $\alpha_r^*, \dots, \alpha_s^* \in [0, 1]$,*

$$\begin{aligned}\frac{1}{|W|}\sum_{t=r}^{s}\ell((\beta_t,\bar{\alpha}_t)) - \frac{1}{|W|}\sum_{t=r}^{s}\ell((\beta_t,\alpha_t^*)) \le \\ \frac{\log(k/\sigma)+2\sigma|W|}{\eta|W|} + \frac{\eta}{|W|}\sum_{t=r}^{s}\mathbb{E}\big[\ell(\beta_t,\alpha_t)^2\big] + \\ 4(1+\gamma_{\max})^2\max\left\{\sqrt{\frac{\sum_{t=r+1}^{s}\left|\alpha_t^*-\alpha_{t-1}^*\right|+1}{|W|}},\gamma_1\right\}.\end{aligned} \tag{9}$$

*Remark* D.2. The bound of the error presented here can be written in simpler form for specific choices of $\sigma$ and $\eta$. Specifically, $\eta = \sqrt{\frac{\log(2k|W|)+1}{\sum_{t=r}^{s}\mathbb{E}[\ell(\beta_t,\alpha_t)^2]}}$ and $\sigma = 1/(2|W|)$

$$\frac{1}{|W|}\sum_{t=r}^{s}\ell((\beta_t,\bar{\alpha}_t)) - \frac{1}{|W|}\sum_{t=r}^{s}\ell((\beta_t,\alpha_t^*)) \le O\left(\sqrt{\frac{\log(|W|)}{|W|}}\right) + O\left(\sqrt{\frac{\sum_{t=r+1}^{s}\left|\alpha_t^*-\alpha_{t-1}^*\right|+1}{|W|}}\right). \tag{10}$$

This derivation is due to Gibbs and Candès [6]

**Theorem D.3** (Theorem 6, [6]). *Consider a modified version of Algorithm 3 in which on iteration $t$ the parameters $\eta$ and $\sigma$ are replaced by values $\eta_t$ and $\sigma_t$. Let $\gamma_{\min} := \min_i \gamma_i$,$\gamma_{\max} := \max_i \gamma_i$, and that $\lim_{t\to\infty}\eta_t = \lim_{t\to\infty}\sigma_t = 0$. Then,*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathrm{err}_t \overset{a.s.}{=} \alpha$$

*where the expectation is over the randomness in the randomized variant of DtACI and the data $\beta_1, \dots, \beta_T$ can be viewed as fixed.*

## D.2 HDR and Score-Stability Proofs

**Assumption D.4** (Regularity for HDR stability). Let $\mathcal{Z}_t$ denote the effective weighted residual distribution used by AdaptNC at score-update time $t$, and let $f_t$ denote its density. Let $\mathcal{Z}_\star$ be the limiting weighted residual distribution with density $f_\star$.

1. The residual-distribution convergence is strong enough to imply uniform density convergence:
$$\|f_t - f_\star\|_\infty \to 0.$$
2. The KDE kernel, bandwidth sequence, and weighted effective sample size satisfy the standard conditions for uniform KDE consistency:
$$\|\widehat{f}_t - f_t\|_\infty \xrightarrow{p} 0.$$
3. The density-score quantile at level $\alpha$ is regular: if
$$G_\star(u) := \mathbb{P}_{Z\sim\mathcal{Z}_\star}(f_\star(Z) \le u),$$
then $G_\star$ is continuous and strictly increasing in a neighborhood of $\tau_\star := G_\star^{-1}(\alpha)$.
4. The limiting HDR boundary is regular: there exist constants $c > 0$ and $\varepsilon_0 > 0$ such that
$$\|\nabla f_\star(z)\|_2 \ge c \qquad \text{whenever } |f_\star(z) - \tau_\star| \le \varepsilon_0.$$

5. The relevant HDRs are nonempty compact subsets of $\mathbb{R}^p$.

6. There exists $L_{\mathrm{vol}} < \infty$ such that for all $x$, all thresholds $q$ in the relevant range, and all sufficiently small $\varepsilon > 0$

$$\mathrm{Vol}\{y : q - \varepsilon \le s_\star(x, y) \le q + \varepsilon\} \le L_{\mathrm{vol}}\varepsilon.$$

**Lemma D.5** (Hausdorff consistency of the MCKDE high-density region)**.** *Let*

$$R_\star := \{z \in \Omega : f_\star(z) \ge \tau_\star\}$$

*be the limiting* $(1-\alpha)$ *high-density residual region, where* $\tau_\star$ *is chosen so that*

$$\int_{R_\star} f_\star(z)\, dz = \int_\Omega \mathbf{1}\{f_\star(z) \ge \tau_\star\} f_\star(z)\, dz = 1 - \alpha.$$

*At score-update time* $t$*, let* $\widehat{f}_t$ *be the KDE formed from* $N_t$ *weighted residual samples, and let* $\widehat{\tau}_t$ *be the Monte Carlo estimate of the density threshold obtained from* $M_t$ *samples. Define the MCKDE estimated high-density region*

$$\widehat{R}_t := \{z \in \Omega : \widehat{f}_t(z) \ge \widehat{\tau}_t\}.$$

*Under Assumptions 5.1 and D.4,*

$$d_H(\widehat{R}_m, R_\star) \xrightarrow{p} 0.$$

*Proof.* By the assumed stability of the weighted residual distributions and the assumed relation between $D_\phi$ and the residual density class,

$$\|f_t - f_\star\|_\infty \to 0.$$

By uniform KDE consistency,

$$\|\widehat{f}_t - f_t\|_\infty \xrightarrow{p} 0.$$

Therefore,

$$\|\widehat{f}_t - f_\star\|_\infty \xrightarrow{p} 0.$$

The Monte Carlo threshold $\widehat{\tau}_t$ is the empirical quantile of the density scores $\widehat{f}_t(Z)$, where $Z$ is sampled from the KDE. Since $M_t \to \infty$, the empirical density-score CDF converges uniformly to its population counterpart, by the Glivenko–Cantelli theorem. Together with the regularity of the density-score quantile and the convergence $\widehat{f}_t \to f_\star$, this gives

$$|\widehat{\tau}_t - \tau_\star| \xrightarrow{p} 0.$$

Define

$$\varepsilon_t := \|\widehat{f}_t - f_\star\|_\infty + |\widehat{\tau}_t - \tau_\star|.$$

Then $\varepsilon_t \xrightarrow{p} 0$.

If $z \in \widehat{R}_t$, then $\widehat{f}_t(z) \ge \widehat{\tau}_t$, and hence

$$f_\star(z) \ge \widehat{f}_t(z) - \|\widehat{f}_t - f_\star\|_\infty \ge \widehat{\tau}_t - \|\widehat{f}_t - f_\star\|_\infty \ge \tau_\star - \varepsilon_t.$$

Therefore,

$$\widehat{R}_t \subseteq \{z \in \Omega : f_\star(z) \ge \tau_\star - \varepsilon_t\}.$$

Similarly, if $z \in R_\star$, then $f_\star(z) \ge \tau_\star$. Points in $R_\star$ with $f_\star(z) \ge \tau_\star + \varepsilon_t$ satisfy

$$\widehat{f}_t(z) \ge f_\star(z) - \|\widehat{f}_t - f_\star\|_\infty \ge \tau_\star + \varepsilon_t - \|\widehat{f}_t - f_\star\|_\infty \ge \widehat{\tau}_t,$$

and therefore belong to $\widehat{R}_t$. Thus, any point of $R_\star$ that may fail to belong to $\widehat{R}_t$ must lie in the shrinking level band

$$\{z \in \Omega : |f_\star(z) - \tau_\star| \le \varepsilon_t\}.$$

By the regular-boundary assumption, these level bands shrink to $\partial R_\star$ in Hausdorff distance. More precisely, for all sufficiently small $\varepsilon$, every point satisfying

$$|f_\star(z) - \tau_\star| \le \varepsilon$$

lies within distance $O(\varepsilon)$ of the boundary $\{z : f_\star(z) = \tau_\star\}$. Consequently,

$$d_H(\widehat{R}_t, R_\star) \le C\varepsilon_t$$

for some constant $C > 0$ and all sufficiently large $t$, with probability tending to one. Since $\varepsilon_t \xrightarrow{p} 0$, we conclude that

$$d_H(\widehat{R}_t, R_\star) \xrightarrow{p} 0.$$

□

**Lemma D.6** (Convex-hull stability)**.** *Let $A_m, A_\star \subset \mathbb{R}^p$ be nonempty compact sets.*

$$d_H(A_m, A_\star) \to 0 \implies d_H(\mathrm{conv}(A_m), \mathrm{conv}(A_\star)) \to 0.$$

*Proof.* Let $\rho_m = d_H(A_m, A_\star)$. For any $x \in \mathrm{conv}(A_m)$, write

$$x = \sum_{\ell=1}^{L} \lambda_\ell a_\ell, \qquad a_\ell \in A_m, \quad \lambda_\ell \ge 0, \quad \sum_{\ell=1}^{L} \lambda_\ell = 1.$$

For each $a_\ell$, choose $a_\ell^\star \in A_\star$ such that $\|a_\ell - a_\ell^\star\|_2 \le \rho_m$. Then

$$x^\star := \sum_{\ell=1}^{L} \lambda_\ell a_\ell^\star \in \mathrm{conv}(A_\star),$$

and

$$\|x - x^\star\|_2 \le \sum_{\ell=1}^{L} \lambda_\ell \|a_\ell - a_\ell^\star\|_2 \le \rho_m.$$

where the last inequality follows from the definition of the Hausdorff distance. Thus every point of $\mathrm{conv}(A_m)$ lies within $\rho_m$ of $\mathrm{conv}(A_\star)$. The reverse inclusion is identical. Hence

$$d_H(\mathrm{conv}(A_m), \mathrm{conv}(A_\star)) \le \rho_m.$$

□

**Corollary D.7.** *In the setting of Lemma D.6,*

$$d_H(\mathrm{conv}(A_m), \mathrm{conv}(A_\star)) \le d_H(A_m, A_\star).$$

Because halfspace representations of a convex set are not unique, we state score stability for the canonical support-function score induced by the fitted convex template. Equivalently, the implemented halfspace score should be normalized to this canonical representation before comparing scores across updates. Our implementation, which relies on QuickHull [22], performs this normalization.

**Lemma D.8** (Score stability from hull stability)**.** *For a compact convex set $K \subset \mathbb{R}^p$, define*

$$s_K(x, y) := \sup_{\|v\|_2 = 1} \{v^\top (y - h(x)) - H_K(v)\}, \qquad H_K(v) := \sup_{z \in K} v^\top z.$$

*If*

$$d_H(K_t, K_\star) \to 0,$$

*then*

$$\sup_{(x,y) \in \mathcal{A}} |s_{K_t}(x, y) - s_{K_\star}(x, y)| \to 0$$

*for a set $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{Y}$.*

*Proof.* For any residual $z = y - h(x)$,

$$\begin{aligned} |s_{K_m}(x, y) - s_{K_\star}(x, y)| &= \left| \sup_{\|v\|_2 = 1} \{v^\top z - h_{K_m}(v)\} - \sup_{\|v\|_2 = 1} \{v^\top z - h_{K_\star}(v)\} \right| \\ &\le \sup_{\|v\|_2 = 1} |H_{K_m}(v) - H_{K_\star}(v)|. \end{aligned}$$

For compact convex sets,

$$\sup_{\|v\|_2=1} |H_{K_m}(v) - H_{K_\star}(v)| = d_H(K_m, K_\star).$$

Therefore,

$$|s_{K_m}(x,y) - s_{K_\star}(x,y)| \le d_H(K_m, K_\star).$$

Taking the supremum over $(x,y) \in \mathcal{A}$ gives the result. □

**Theorem 5.2** (Score Function Stability)**.** *Suppose Assumptions C.1, D.4 and 5.1 hold. Let $\widehat{R}_m$ be the MCKDE high-density residual region estimated at score-update time $t_m$, and define$K_m := \mathrm{conv}(\widehat{R}_m)$. Let$R_\star := \{z : f_\star(z) \ge \tau_\star\}$ and $K_\star := \mathrm{conv}(R_\star)$. Let $s_{K_m}$ and $s_{K_\star}$ denote the canonical support-function scores induced by $K_m$ and $K_\star$. Then, for any evaluation domain $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{Y}$,*

$$\sup_{(x,y)\in\mathcal{A}} |s_{K_m}(x,y) - s_{K_\star}(x,y)| \xrightarrow{p} 0 \quad \therefore \quad \sup_{(x,y)\in\mathcal{A}} |s_{K_{m+1}}(x,y) - s_{K_m}(x,y)| \xrightarrow{p} 0.$$

*Proof.* By Lemma D.5,

$$d_H(\widehat{R}_m, R_\star) \xrightarrow{p} 0.$$

By Lemma D.6,

$$d_H(K_m, K_\star) = d_H(\mathrm{conv}(\widehat{R}_m), \mathrm{conv}(R_\star)) \xrightarrow{p} 0.$$

By Lemma D.8,

$$\sup_{(x,y)\in\mathcal{A}} |s_{K_m}(x,y) - s_{K_\star}(x,y)| \xrightarrow{p} 0.$$

Finally, by the triangle inequality,

$$\begin{aligned}&\sup_{(x,y)\in\mathcal{A}} |s_{K_{m+1}}(x,y) - s_{K_m}(x,y)| \\ &\quad\le \sup_{(x,y)\in\mathcal{A}} |s_{K_{m+1}}(x,y) - s_{K_\star}(x,y)| + \sup_{(x,y)\in\mathcal{A}} |s_{K_m}(x,y) - s_{K_\star}(x,y)|.\end{aligned}$$

Both terms converge to zero in probability, so the successive score functions are asymptotically stable. □

*Proof of Corollary 5.3.* Suppose

$$\sup_{(x,y)\in\mathcal{A}} |s_t(x,y) - s_\star(x,y)| \le \delta_t.$$

Under Assumption D.4, for every fixed threshold $q$,

$$|\mathrm{Vol}(C_t(x;q)) - \mathrm{Vol}(C_\star(x;q))| \le L_{\mathrm{vol}}\delta_t.$$

Since $\delta_t \to 0$ in probability by Theorem 5.2, the prediction-region volume stabilizes in probability. □

### D.3 Long-Run Coverage under Replay

*Proof of Theorem 5.4.* Expanding the perturbed recursion gives

$$\alpha_{T+1}^i = \alpha_1^i + \gamma_i \sum_{t=1}^{T} (\alpha - \mathbf{err}_t^i) + \sum_{t=1}^{T} \Delta_t^i.$$

Rearranging,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{err}_t^i - \alpha = \frac{\alpha_1^i - \alpha_{T+1}^i}{T\gamma_i} + \frac{1}{T\gamma_i}\sum_{t=1}^{T} \Delta_t^i.$$

The first term vanishes because the expert states are bounded. The second term vanishes by the assumed sublinear cumulative perturbation.

$$\left| \frac{1}{T\gamma_i}\sum_{t=1}^{T} \Delta_t^i \right| \le \frac{1}{T\gamma_i}\sum_{t=1}^{T} |\Delta_t^i| \to 0.$$

Hence:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{err}_t^i \to \alpha.$$

□

*Proof of Theorem 5.5.* For the analysis, define the pre-update replay trajectory as the counterfactual trajectory obtained by replaying $\mathcal{R}_m$ using $s_m^-$, and the post-update replay trajectory as the trajectory obtained by replaying the same window using $s_m^+$, both initialized from the same expert state. Let

$$e_{m,r}^{i,-} := \mathbf{1}\{s_m^-(X_r, Y_r) > q_{m,r}^{i,-}\}, \qquad e_{m,r}^{i,+} := \mathbf{1}\{s_m^+(X_r, Y_r) > q_{m,r}^{i,+}\}.$$

At non-update times, $\Delta_t^i = 0$. At update time $t_m$, the difference between the replayed post-update expert state and the corresponding pre-update expert state is controlled by the number of replayed observations whose coverage indicators change:

$$|\Delta_{t_m}^i| \le \gamma_i \sum_{r\in\mathcal{R}_m} |e_{m,r}^{i,+} - e_{m,r}^{i,-}|.$$

Now observe that the indicators can differ only if the replayed score lies near the pre-update threshold. Indeed, if

$$|s_m^-(X_r, Y_r) - q_{m,r}^{i,-}| > (1 + L_q)\varepsilon_m,$$

then

$$|s_m^+(X_r, Y_r) - s_m^-(X_r, Y_r)| \le \varepsilon_m$$

and

$$|q_{m,r}^{i,+} - q_{m,r}^{i,-}| \le L_q\varepsilon_m$$

together imply that $s_m^+(X_r, Y_r)$ and $s_m^-(X_r, Y_r)$ remain on the same side of their respective thresholds. Hence

$$|e_{m,r}^{i,+} - e_{m,r}^{i,-}| \le \mathbf{1}\left\{|s_m^-(X_r, Y_r) - q_{m,r}^{i,-}| \le (1 + L_q)\varepsilon_m\right\}.$$

Therefore

$$\frac{1}{T}\sum_{t=1}^{T}|\Delta_t^i| \le \frac{\gamma_i}{T}\sum_{m:t_m\le T}\sum_{r\in\mathcal{R}_m}\mathbf{1}\left\{|s_m^-(X_r, Y_r) - q_{m,r}^{i,-}| \le (1 + L_q)\varepsilon_m\right\}.$$

The assumed averaged boundary condition implies that the right-hand side converges to zero. The final coverage claim follows from the perturbed single-expert ACI guarantee. □

### D.4 Coverage Shock from Score Updates

**Proposition D.9.** *Let $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$ be the joint distributions of inputs and observations $(X_t, Y_t)$ at times $t$ and $t+1$. Assume that the total variation distance ($\mathbf{TV}$) between the two distributions is bounded $\mathbf{TV}(\mathcal{D}_t, \mathcal{D}_{t+1}) \le \delta$. For a fixed NCSF $s(\cdot)$, and a fixed score parameter $\theta$ let $S_{\theta,t}$ be the random variable $s(X, Y, \theta)$ where $(X, Y) \sim \mathcal{D}_t$, and $S_{\theta,t+1}$ be the random variable where $(X, Y) \sim \mathcal{D}_{t+1}$ Then, the $\mathbf{TV}$ distance between the score distributions will be less than $\delta$: $\mathbf{TV}(S_{\theta,t}, S_{\theta,t+1}) \le \delta$*

*Proof.* This follows from the Data Processing Inequality [26, Theorem 7.4] with NCSF $s(\cdot;\theta)$ as channel, and $f$-divergence as $\mathbf{TV}$ distance. □

# E Experiments

## E.1 Gaussian Mixture Model: A Minimal Example of Coverage Shock

The replay mechanism in AdaptNC is motivated by a simple failure mode: the state of an adaptive thresholding algorithm is tied to the nonconformity score used to map past residuals into ranks and coverage errors. When the score function changes, the same recent residuals can induce a different score distribution and therefore a different calibration state. Thus, even if the underlying residual distribution changes smoothly, a score update can produce a transient mismatch between the inherited threshold state and the updated score. We refer to this transient mismatch as *coverage shock*. The following Gaussian mixture example isolates this effect in a setting where the only source of nonstationarity is a smooth change in mixture weights.

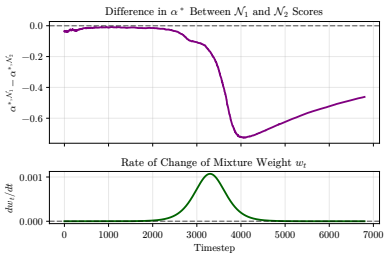



Figure 5: Score-induced coverage shock in GMM example. The top panel shows the gap between history-relative effective miscoverage levels induced by the two diagnostic scores, while the bottom panel shows the rate of change of the mixture weight. Despite smooth distributional changes, altering the score can induce large shifts in the calibration state that DtACI must track.

**Data-generating process.** At time $t$, a residual sample $z_t \in \mathbb{R}^2$ is drawn from a two-component Gaussian mixture

$$p_t(z) = (1 - w_t)p_1(z) + w_t p_2(z), \tag{11}$$

where $w_t$ varies smoothly over time. Thus the stream begins in a regime dominated by $\mathcal{N}_1$ and gradually transitions to a regime dominated by $\mathcal{N}_2$. The two components are

$$\mathcal{N}_1 = \mathcal{N}\left([1.0 \quad -1.2], \begin{bmatrix} 1.2 & 0.6 \\ 0.6 & 0.9 \end{bmatrix}\right), \qquad \mathcal{N}_2 = \mathcal{N}\left([-1.0 \quad 1.2], \begin{bmatrix} 0.8 & -0.3 \\ -0.3 & 1.1 \end{bmatrix}\right).$$

The resulting stream is shown in Fig. 6.

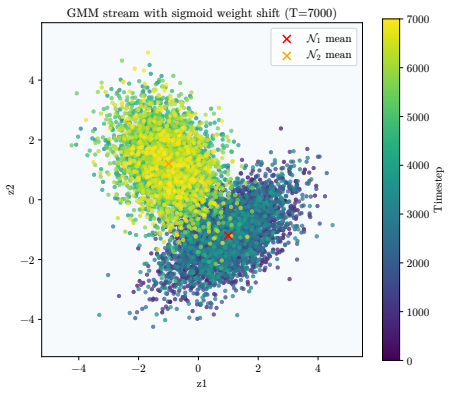



Figure 6: The 7000 samples drawn from the shifting Gaussian mixture stream. Color indicates time, showing the transition from the $\mathcal{N}_1$-dominated regime to the $\mathcal{N}_2$-dominated regime.

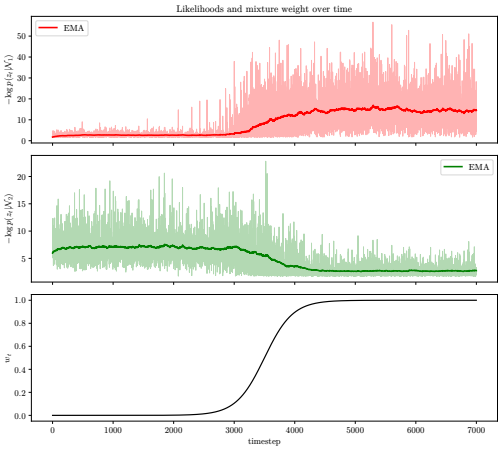



Figure 7: Negative log-likelihood scores under the two component scores as the mixture weight changes. The smooth mixture transition induces a systematic change in the score distribution.

**Score-dependent reference levels.** We compare two fixed nonconformity scores, each matched to one component:

$$s_a(z) = -\log p_a(z), \qquad a \in \{1, 2\}. \tag{12}$$

These scores are used as interpretable diagnostic endpoints: $s_1$ is matched to the early regime and $s_2$ is matched to the late regime. AdaptNC does not literally switch between these two scores; rather, the pair illustrates how a change in score geometry can change the calibration state even on the same residual stream.

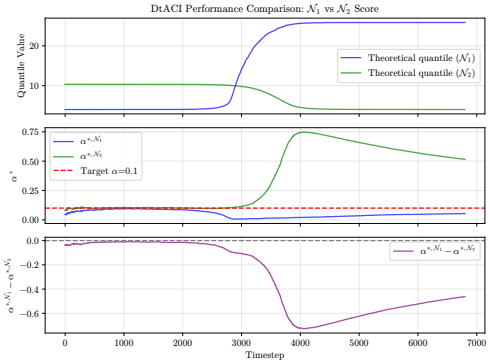



Figure 8: Coverage shock in the shifting GMM. **Top:** The nominal $(1-\alpha)$ score quantiles for the two diagnostic scores evolve in opposite directions as the dominant mixture component changes. **Middle:** The corresponding effective miscoverage levels $\widehat{\alpha}^{\star}_{a,t}$ deviate from the nominal target $\alpha = 0.1$ and from one another. **Bottom:** The gap between these levels quantifies the change in calibration state induced by changing scores.

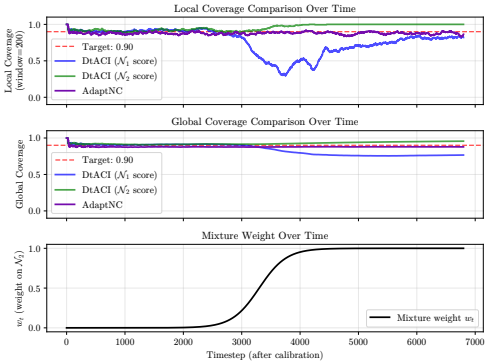



Figure 9: Coverage behavior under the shifting GMM. The fixed $s_1$ baseline undercovers after the mixture shifts, while the fixed $s_2$ baseline is conservative. AdaptNC remains closer to the target than the under-covering fixed score and avoids the persistent conservatism of the late-regime score by adapting its score and replaying recent observations.

For each score $s_a$, let

$$q^{\mathrm{mix}}_{a,t} = \inf\left\{ q : \Pr_{Z\sim p_t}\left[s_a(Z) \le q\right] \ge 1-\alpha \right\} \tag{13}$$

be the nominal $(1-\alpha)$ score quantile under the instantaneous mixture $p_t$. In the experiment, this quantity is estimated by Monte Carlo from the current mixture distribution. We then evaluate where this instantaneous quantile falls relative to the accumulated score history,

$$\widehat{F}^{\mathrm{hist}}_{a,t}(u) = \frac{1}{t+1}\sum_{\tau=0}^{t} \mathbb{1}\{s_a(z_\tau) \le u\}, \qquad \widehat{\alpha}^{\star}_{a,t} = 1 - \widehat{F}^{\mathrm{hist}}_{a,t}\left(q^{\mathrm{mix}}_{a,t}\right). \tag{14}$$

The quantity $\widehat{\alpha}^{\star}_{a,t}$ is a diagnostic effective miscoverage level: it is the miscoverage rate that would make the accumulated score history place its $(1-\widehat{\alpha}^{\star}_{a,t})$ quantile at the instantaneous mixture quantile. Because both $q^{\mathrm{mix}}_{a,t}$ and $\widehat{F}^{\mathrm{hist}}_{a,t}$ depend on the score map, changing from $s_1$ to $s_2$ can induce a large jump in $\widehat{\alpha}^{\star}_{a,t}$ even when $w_t$ changes smoothly.

**Why this motivates replay.** Fig. 8 shows that a smooth change in mixture weight can still create a large separation between the calibration reference levels associated with different scores. If a score-adaptive method updates its score without recalibrating the threshold state, DtACI continues from a state shaped by old score observations while receiving future observations through a new score. In the regret bound of Remark D.2, this appears as a large $|\alpha^{\star}_t - \alpha^{\star}_{t-1}|$ term, i.e., coverage shock.

Replay mitigates this effect by reconstructing the recent DtACI state under the updated score. Instead of applying the new score to future observations while retaining a threshold state learned from the old score, AdaptNC rescores the recent window, recomputes the corresponding ranks and coverage errors, and reruns the threshold updates. In the GMM example, this replaces a counterfactual abrupt jump between the diagnostic endpoint scores with a recalibration step under the updated score used by AdaptNC.

This behavior is reflected in Fig. 9. When a fixed score becomes mismatched with the dominant mixture component, DtACI either undercovers or becomes conservative. AdaptNC adapts the score and uses replay to realign the threshold state, yielding more stable near-target coverage through the transition. The example is intentionally simple: its purpose is not to model a robotics task, but to

isolate the mechanism by which score changes create coverage shock and to motivate replay as the corresponding state-realignment step.

### E.2 Indoor Localization

This section provides a complete and reproducible description of the environment dynamics, wireless observation model, trajectory predictor, and evaluation policy used in all experiments.

#### E.2.1 Environment Dynamics

We consider a planar indoor localization environment with discrete-time dynamics and sampling period $\Delta t = 0.1$ s. The true system state at time $t$ is

$$\mathbf{s}_t = [x_t, y_t, v_{x,t}, v_{y,t}]^\top \in \mathbb{R}^4,$$

where $(x_t, y_t)$ denotes the agent position and $(v_{x,t}, v_{y,t})$ denotes velocity.

Given an acceleration control input $\mathbf{a}_t = [a_{x,t}, a_{y,t}]^\top$, the dynamics evolve according to

$$\begin{aligned} x_{t+1} &= x_t + \Delta t\, v_{x,t}, \\ y_{t+1} &= y_t + \Delta t\, v_{y,t}, \\ v_{x,t+1} &= v_{x,t} + \Delta t\, (a_{x,t} + \varepsilon_{x,t}), \\ v_{y,t+1} &= v_{y,t} + \Delta t\, (a_{y,t} + \varepsilon_{y,t}), \end{aligned}$$

where $\varepsilon_{x,t}, \varepsilon_{y,t} \sim \mathcal{N}(0, \sigma_{\text{proc}}^2)$ model additive process noise on acceleration. We use $\sigma_{\text{proc}} = 0.02$.

The agent operates within a bounded workspace defined by $x, y \in [-6, 6]$. Reflective boundary conditions are enforced by reversing the corresponding velocity component whenever the agent reaches the workspace boundary.

#### E.2.2 Access Point Geometry

The environment contains four fixed access points located at

$$(-5, -5),\ (5, -5),\ (5, 5),\ (-5, 5).$$

These access points are static and known to the environment simulator and predictor.

#### E.2.3 RSSI Observation Model

At each time step, the agent observes a received signal strength indicator vector

$$\mathbf{o}_t = [\text{RSSI}_t^{(1)}, \ldots, \text{RSSI}_t^{(4)}]^\top \in \mathbb{R}^4.$$

Each RSSI measurement is generated as the sum of path loss, shadowing, and small-scale fading components, expressed in decibels:

$$\text{RSSI}_t^{(i)} = P_0 - 10n \log_{10}(d_t^{(i)}) + S_t^{(i)} + F_t^{(i)}.$$

**Path Loss.** The path loss component follows the standard log-distance path loss model [27]. Here, $P_0 = -30$ dB denotes the reference received power at unit distance, $n = 2.2$ is the path loss exponent, and

$$d_t^{(i)} = \|[x_t, y_t] - \mathbf{a}_i\|_2$$

is the Euclidean distance between the agent and access point $i$. Distances are lower bounded by $10^{-2}$ m to ensure numerical stability. The path loss exponent controls the rate of signal attenuation with distance and reflects indoor propagation characteristics.

**Shadowing.** Large-scale shadowing is modeled as a temporally correlated log-normal process [28]. For each access point $i$, the shadowing term evolves according to a first-order Gauss–Markov model:

$$S_t^{(i)} = \rho S_{t-1}^{(i)} + \sqrt{1 - \rho^2}\, \sigma_{\text{sh}}\, \xi_t^{(i)}, \quad \xi_t^{(i)} \sim \mathcal{N}(0, 1),$$

where $\sigma_{\text{sh}} = 4.0$ dB controls the shadowing variance and $\rho = 0.97$ determines temporal correlation. Shadowing states are initialized to zero at environment reset and evolve independently across access points.

**Small-Scale Fading.** Small-scale fading is modeled using a Rayleigh fading process generated via a sum-of-sinusoids [29] method. For each access point, a complex channel coefficient $h_t^{(i)}$ is generated, and the fading contribution is computed as

$$F_t^{(i)} = 10 \log_{10}(|h_t^{(i)}|^2 + \epsilon),$$

with $\epsilon = 10^{-12}$ for numerical stability.

The Doppler frequency governing temporal fading correlation is velocity-dependent:

$$f_{d,t} = \frac{\|\mathbf{v}_t\|_2}{\lambda}, \quad \lambda = \frac{c}{f_c},$$

where $f_c = 2.4$ GHz is the carrier frequency and $c$ is the speed of light. Fading processes are independent across access points. The Rayleigh distribution of $|h_t^{(i)}|$ follows from the assumption of rich multipath with no dominant line-of-sight component [30, 27].

### E.2.4 Trajectory Predictor

We employ a lightweight model-based trajectory predictor that estimates the agent position from RSSI measurements using a simplified propagation model. The predictor is mismatched with the true environment observation model and does not explicitly account for shadowing or small-scale fading. This mismatch induces systematic and time-varying localization error.

**Internal State.** The predictor maintains an internal estimate of planar position and velocity,

$$\hat{\mathbf{x}}_t \in \mathbb{R}^2, \quad \hat{\mathbf{v}}_t \in \mathbb{R}^2,$$

which are reset to zero at the beginning of each rollout.

**Prediction Step.** At each time step, the predictor applies a constant-velocity motion model:

$$\hat{\mathbf{x}}_{t|t-1} = \hat{\mathbf{x}}_{t-1} + \Delta t\, \hat{\mathbf{v}}_{t-1}.$$

This prediction does not incorporate control inputs or process noise and serves only as a kinematic extrapolation of past estimates.

**RSSI-to-Distance Conversion.** Given an RSSI observation vector $\mathbf{o}_t$, the predictor converts each RSSI measurement into a distance estimate by inverting the log-distance path loss model:

$$\hat{d}_t^{(i)} = 10^{\frac{P_0 - \text{RSSI}_t^{(i)}}{10n}}.$$

This inversion assumes that RSSI measurements are generated solely by path loss and ignores the presence of shadowing and fading. As a result, the inferred distances are biased and noisy, with error statistics that depend on the instantaneous channel realization.

**Multilateration via Gauss–Newton.** The predicted position is refined by solving a nonlinear least-squares multilateration problem of the form

$$\min_{\mathbf{x} \in \mathbb{R}^2} \sum_{i=1}^{N} w_i \left( \|\mathbf{x} - \mathbf{a}_i\|_2 - \hat{d}_t^{(i)} \right)^2, \quad w_i = \frac{1}{(\hat{d}_t^{(i)})^2}.$$

This problem is solved using Gauss–Newton iterations initialized at $\hat{\mathbf{x}}_{t|t-1}$. A small damping term is added for numerical stability, and update steps are clipped to avoid large jumps. The resulting solution is denoted $\mathbf{x}_t^{\text{meas}}$.

**Alpha–Beta Filtering.** The final position and velocity estimates are obtained using an $\alpha$–$\beta$ filter:

$$\begin{aligned}
\mathbf{e}_t &= \mathbf{x}_t^{\text{meas}} - \hat{\mathbf{x}}_{t|t-1}, \\
\hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t-1} + \alpha \mathbf{e}_t, \\
\hat{\mathbf{v}}_t &= \hat{\mathbf{v}}_{t-1} + \frac{\beta}{\Delta t} \mathbf{e}_t,
\end{aligned}$$

with $\alpha = 0.25$ and $\beta = 0.05$. This filter smooths high-frequency measurement noise while allowing the predictor to track gradual motion trends.

**Model Mismatch and Error Characteristics.** Since the predictor assumes a deterministic path loss model and neglects shadowing and small-scale fading, its position estimates exhibit residual errors that are neither independent nor identically distributed. In particular, temporally correlated fading induces time-correlated localization errors whose magnitude and direction vary over time. This mismatch produces nonstationary residual streams, making the predictor suitable for evaluating adaptive conformal uncertainty estimation methods that must respond to evolving error statistics.

### E.2.5 Evaluation Policy

All experiments use a fixed, non-adaptive random motion policy intended to emulate surveillance-style exploration. At each time step, acceleration inputs are sampled uniformly within bounded limits. Velocity magnitudes are clipped to a maximum of $1.0$ m/s, and accelerations are bounded by $0.5$ m/s$^2$.

The policy is independent of RSSI observations, predictor estimates, and uncertainty regions. The same policy is used across all methods, ensuring that observed differences in coverage and region geometry arise solely from the uncertainty estimation method.

### E.2.6 Evaluation on Abrupt Shifts

We performed an abrupt-shift experiment in the indoor-localization environment. At timestep $t = 3050$, the RSSI measurement from one access point is instantaneously attenuated producing a discontinuous change in the residual distribution. We compare AdaptNC with and without replay over $50$ random seeds over three different attenuation levels: 10 (mild), 30 (strong), and 50 (very strong) dB, reporting global performance and transition performance during the first $100$ timesteps following the shift.

Table 3: Global performance under abrupt attenuation.

| Attenuation | Method | Coverage (%) | Local coverage (%) | Mean volume |
|---|---|---|---|---|
| 10 dB | Replay | $90.78 \pm 0.33$ | $90.39 \pm 0.32$ | $34.70 \pm 3.23$ |
| | No replay | $94.21 \pm 1.02$ | $93.81 \pm 1.02$ | $46.53 \pm 4.96$ |
| 30 dB | Replay | $90.92 \pm 0.35$ | $90.53 \pm 0.34$ | $44.21 \pm 5.92$ |
| | No replay | $94.03 \pm 0.86$ | $93.63 \pm 0.86$ | $57.50 \pm 8.17$ |
| 50 dB | Replay | $90.78 \pm 0.32$ | $90.38 \pm 0.32$ | $39.70 \pm 4.79$ |
| | No replay | $94.18 \pm 0.94$ | $93.79 \pm 0.94$ | $52.09 \pm 7.17$ |

Table 4: Transition performance following the abrupt attenuation.

| Attenuation | Method | Coverage (%) | Local coverage (%) | Mean volume |
|---|---|---|---|---|
| 10 dB | Replay | $90.06 \pm 3.39$ | $90.17 \pm 3.22$ | $35.44 \pm 15.12$ |
| | No replay | $94.26 \pm 4.30$ | $94.08 \pm 4.32$ | $48.41 \pm 16.55$ |
| 30 dB | Replay | $89.82 \pm 3.60$ | $89.88 \pm 3.37$ | $40.80 \pm 24.07$ |
| | No replay | $93.13 \pm 4.75$ | $92.95 \pm 4.69$ | $56.25 \pm 33.15$ |
| 50 dB | Replay | $89.35 \pm 3.12$ | $89.46 \pm 3.17$ | $35.91 \pm 16.68$ |
| | No replay | $93.41 \pm 4.71$ | $93.28 \pm 4.75$ | $50.09 \pm 19.71$ |

The behavior under abrupt shifts is consistent with the trends observed for gradual distribution shifts. Replay maintains coverage close to the nominal $90\%$ target while producing tighter prediction regions than the no-replay variant. The same behavior is observed during the critical 100-timestep transition period immediately following the shift, indicating rapid recovery even after sudden distribution changes.

### E.2.7 Expert Weights

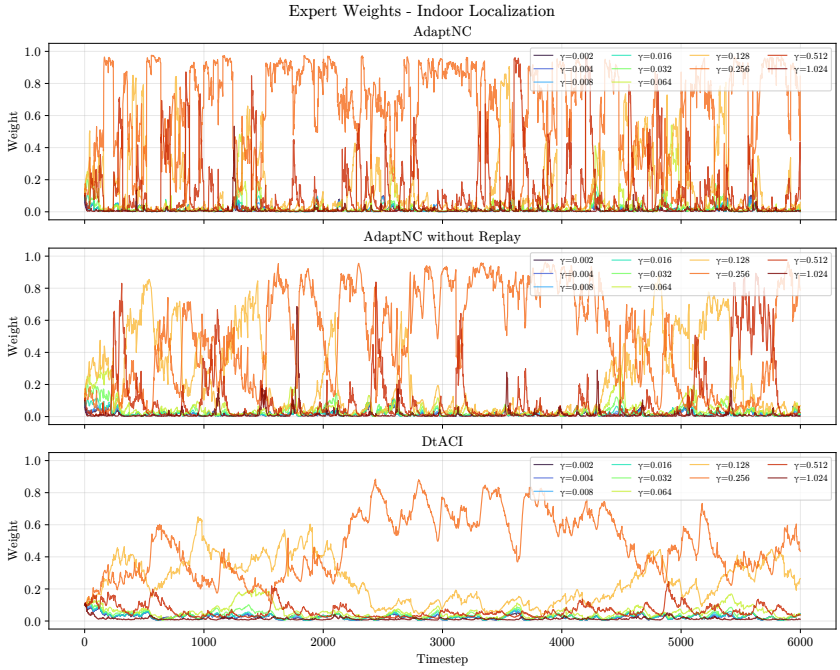



Figure 10: This figure shows the evolution of expert weights in the Indoor Localization setting over the course of the experiment. The pronounced, piecewise variation observed for AdaptNC arises from its adaptive reweighting mechanism, which enables rapid adjustment of expert weights in response to changes in the score function and contributes to maintaining valid coverage. In contrast, DtACI and AdaptNC without replay exhibit slower weight adaptation and consistently select smaller values of $\gamma$ than AdaptNC.

### E.3 Social Navigation

The simulation environment implements a continuous-space multi-agent system based on the Social Force Model described by Helbing and Molnár (1995).

#### E.3.1 Equations of Motion

Let the state of agent $\alpha$ at time $t$ be defined by its position $\vec{r}_\alpha(t)$, actual velocity $\vec{v}_\alpha(t)$, and preferred velocity $\vec{w}_\alpha(t)$. The dynamics are governed by a nonlinearly dynamics equation driving the preferred velocity, subject to a maximum speed constraint.

The evolution of the preferred velocity $\vec{w}_\alpha$ is given by:

$$\frac{d\vec{w}_\alpha}{dt} = \frac{\vec{v}_\alpha^{des} - \vec{v}_\alpha}{\tau} + \sum_{\beta \neq \alpha} \vec{F}_{\alpha\beta} + \sum_B \vec{F}_{\alpha B} + \vec{\xi}(t) \tag{15}$$

where:

- $\tau$ is the relaxation time constant.
- $\vec{v}_\alpha^{des}$ is the desired velocity vector toward the current goal.
- $\vec{F}_{\alpha\beta}$ represents the social repulsive force from agent $\beta$.
- $\vec{F}_{\alpha B}$ represents the repulsive force from boundary $B$.
- $\vec{\xi}(t)$ is a stochastic fluctuation term (Gaussian noise).

The actual velocity $\vec{v}_\alpha(t)$ is derived by clipping the preferred velocity to the agent's maximum speed $v_\alpha^{max}$:

$$\vec{v}_\alpha(t) = \min\left(1, \frac{v_\alpha^{max}}{\|\vec{w}_\alpha(t)\|}\right) \vec{w}_\alpha(t) \tag{16}$$

#### E.3.2 Force Components

**Goal Attraction** The driving term attracts the agent toward its goal $\vec{g}_\alpha$ with a desired speed $v_\alpha^0$:

$$\vec{v}_\alpha^{des} = v_\alpha^0 \frac{\vec{g}_\alpha - \vec{r}_\alpha}{\|\vec{g}_\alpha - \vec{r}_\alpha\|} \tag{17}$$

**Social Repulsion** The interaction force $\vec{F}_{\alpha\beta}$ models the repulsion between agents to maintain personal space. The implementation utilizes an elliptical distance metric approximation. The force is defined as:

$$\vec{F}_{\alpha\beta} = w(\vec{e}_\alpha, \vec{f}_{\alpha\beta}) \cdot \vec{f}_{\alpha\beta} \tag{18}$$

where the raw isotropic force $\vec{f}_{\alpha\beta}$ is:

$$\vec{f}_{\alpha\beta} = \left(\frac{A}{B} \exp\left(-\frac{b}{B}\right) \frac{2d_{\alpha\beta} - s_{ab}}{2b}\right) \vec{n}_{\alpha\beta} \tag{19}$$

with terms defined as follows:

$$\begin{aligned} d_{\alpha\beta} &= \|\vec{r}_\alpha - \vec{r}_\beta\| \quad \text{(Euclidean distance)} \\ \vec{n}_{\alpha\beta} &= \frac{\vec{r}_\alpha - \vec{r}_\beta}{d_{\alpha\beta}} \quad \text{(Normalized direction vector)} \\ s_{ab} &= \|\vec{v}_\beta\| \Delta t_{anticipation} \quad \text{(Anticipated displacement)} \end{aligned}$$

The semi-minor axis $b$ of the repulsive potential is approximated in the implementation as:

$$b = \sqrt{\frac{d_{\alpha\beta}^2 + (d_{\alpha\beta} - s_{ab})^2}{2}} \tag{20}$$

This differs slightly from the exact equipotential formulation in the original work but preserves the qualitative elliptical avoidance behavior .

**Boundary Repulsion** Repulsion from static boundaries (walls) decays exponentially with the distance $d_{\alpha B}$ to the nearest point on the boundary:

$$\vec{F}_{\alpha B} = w(\vec{e}_\alpha, \vec{f}_{\alpha B}) \cdot \frac{A_{wall}}{R_{wall}} \exp\left(-\frac{d_{\alpha B}}{R_{wall}}\right) \vec{n}_{\alpha B} \tag{21}$$

where $\vec{n}_{\alpha B}$ is the normal vector pointing away from the boundary.

### E.3.3 Anisotropic Perception

To reflect the limited field of view, forces are scaled by an anisotropic weight $w(\vec{e}, \vec{f})$ dependent on the angle $\phi$ between the agent's desired direction $\vec{e}_\alpha$ and the force vector $\vec{f}$:

$$w(\vec{e}_\alpha, \vec{f}) = \begin{cases} \lambda + (1-\lambda)\frac{1+\cos\phi}{2} & \text{if } \cos\phi \geq 0 \quad \text{(Frontal)} \\ c & \text{if } \cos\phi < 0 \quad \text{(Rear)} \end{cases} \tag{22}$$

where $\cos\phi = \frac{\vec{e}_\alpha \cdot \vec{f}}{\|\vec{e}_\alpha\| \|\vec{f}\|}$.

### E.3.4 Model Parameters

The default parameters used in the simulation are listed in Table 5.

| Parameter | Symbol | Value |
|---|---|---|
| Relaxation time | $\tau$ | 0.5 s |
| Desired speed | $v^0$ | $\max(0.1, \mathcal{N}(1.34, 0.26^2))$ m/s |
| Max speed | $v^{max}$ | $1.3 \times v^0$ m/s |
| Repulsion strength | $A$ | 5.0 m/s$^2$ |
| Repulsion range | $B$ | 2.0 m |
| Anticipation time | $\Delta t_{anticipation}$ | 2.0 s |
| Anisotropy factor | $\lambda$ | 0.5 |
| Rear influence | $c$ | 0.5 |
| Wall repulsion strength | $A_{wall}$ | 10.0 m/s$^2$ |
| Wall repulsion range | $R_{wall}$ | 0.2 m |

Table 5: Simulation parameters corresponding to the Social Navigation experiment

### E.3.5 Trajectory Predictor

The trajectory predictor is implemented as an Long Short-Term Memory (LSTM) to capture temporal dependencies in agent motion. The model is implemented using the Flax/JAX framework.

**Architecture** The model processes a sequence of state observations $X = \{x_1, \ldots, x_T\}$ where each timestep $x_t \in \mathbb{R}^{34}$ contains the ego-agent's state and the relative states of the $k = 7$ nearest neighbors. The specific feature composition is:

- **Ego Features (6)**: Absolute position $(p_x, p_y)$, velocity $(v_x, v_y)$, and goal position $(g_x, g_y)$.
- **Neighbor Features (28)**: Relative position $(\Delta p_x, \Delta p_y)$ and relative velocity $(\Delta v_x, \Delta v_y)$ for each of the 7 nearest neighbors.

The network architecture consists of a single LSTM layer with a hidden dimension of $h = 128$. The final hidden state $h_T$ is passed through a Multi-Layer Perceptron (MLP) head to predict the 2D position $\hat{y} \in \mathbb{R}^2$ of the agent at the prediction horizon $\tau$:

$$h_{1:T} = \text{LSTM}(x_{1:T}) \tag{23}$$

$$z = \text{ReLU}(\text{Dense}_{128}(\text{Dropout}(h_T))) \tag{24}$$

$$\hat{y} = \text{Dense}_2(\text{Dropout}(z)) \tag{25}$$

Dropout with a rate of $0.1$ is applied before the dense layers during training to prevent overfitting .

**Training Objective** The model is trained to minimize the Mean Squared Error (MSE) between the predicted position $\hat{y}$ and the ground truth position $y$ after $\tau$ simulation steps:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|\hat{y}_i - y_i\|^2 \tag{26}$$

Optimization is performed using the AdamW optimizer. The specific hyperparameters used for training are detailed in Table 6.

| Hyperparameter | Value |
|---|---|
| Hidden Dimension | 128 |
| Dropout Rate | 0.1 |
| Sequence Length ($T$) | 10 |
| Prediction Horizon ($\tau$) | 5 |
| Learning Rate | $1 \times 10^{-4}$ |
| Weight Decay | $1 \times 10^{-5}$ |
| Batch Size | 512 |
| Training Epochs | 200 |

Table 6: Hyperparameters for the LSTM Trajectory Predictor.

### E.3.6 Expert Weights




Figure 11: This figure contains the expert weights for the social navigation setting. The plots show the evolution of the weights over the course of the scenario. The stacatto pattern of the AdaptNC setting is a consequence of the adaptive reweighting scheme, which enables rapid adaptation of the expert weights in response to score function changes, which helps AdaptNC maintain coverage. We can see that the DtACI and AdaptNC without Replay methods do not adapt expert weights as quickly, and choose values for $\gamma$ that are lower than those chosen by AdaptNC.

### E.4 Multirotor Tracking

This section describes the multirotor environment with actuator degradation, the trajectory predictor used for position forecasting, and the control policy employed to generate training, calibration, and evaluation trajectories.

#### E.4.1 Environment with Actuator Degradation

The multirotor state at time $t$ is: $\mathbf{s}_t = [x_t, \dot{x}_t, y_t, \dot{y}_t, z_t, \dot{z}_t, \psi_t, \dot{\psi}_t, \theta_t, \dot{\theta}_t, \delta_t, \dot{\delta}_t]^\top \in \mathbb{R}^{12}$, where $(x, y, z)$ denote position, $(\dot{x}, \dot{y}, \dot{z})$ translational velocity, $\psi$ yaw, $\theta$ pitch, and $\delta$ roll. The system evolves in discrete time with integration step $\Delta t = 0.1$ s using forward Euler integration.

**Control Inputs.** The control input is: $\mathbf{u}_t = [u_{1,t}, u_{2,t}, u_{3,t}, u_{4,t}]^\top$, where $u_1$ corresponds to collective thrust, $u_2$ to pitch torque, $u_3$ to roll torque, and $u_4$ to yaw torque.

**Actuator Degradation.** Each actuator is subject to stochastic degradation modeled by a health variable, $\mathbf{\Gamma}_t \in [0, 1]^4$, where $\Gamma_t^{(i)} = 1$ denotes a healthy actuator. The effective control applied to the system is: $\mathbf{u}_t^{\text{eff}} = \mathbf{\Gamma}_t \odot \mathbf{u}_t$. Actuator health evolves according to an additive Wiener process with negative drift:

$$\mathbf{\Gamma}_{t+1} = \mathbf{\Gamma}_t - \alpha \Delta t + \sigma \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \Delta t I),$$

with $\alpha = 5 \times 10^{-4}$ and $\sigma = 2.5 \times 10^{-4}$. The degradation process is unobserved by the predictor and policy.

**Dynamics.** Translational accelerations are given by :$\ddot{x}_t = g\theta_t,\ \ddot{y}_t = -g\delta_t,\ \ddot{z}_t = u_{1,t}^{\text{eff}} - g$, with gravitational constant $g = 9.81$ m/s$^2$. Rotational accelerations satisfy: $\ddot{\psi}_t = u_{4,t}^{\text{eff}},\ \ddot{\theta}_t = u_{2,t}^{\text{eff}},\ \ddot{\delta}_t = u_{3,t}^{\text{eff}}$. Yaw is wrapped to $[-\pi, \pi]$, and pitch and roll are clipped to $[-0.3, 0.3]$ radians.

#### E.4.2 Trajectory Predictor

To forecast planar motion, we use a hybrid physics learning predictor that combines a simple kinematic prior with a learned residual model.

**Physics Prior.** The prior assumes constant velocity motion in the horizontal plane:

$$\hat{x}_{t+1}^{\text{phys}} = x_t + \Delta t \dot{x}_t, \quad \hat{y}_{t+1}^{\text{phys}} = y_t + \Delta t \dot{y}_t.$$

This prior ignores rotational coupling and actuator degradation and therefore provides only a coarse approximation of the true dynamics.

**Residual Learning.** Rather than predicting absolute positions, the predictor learns the residual

$$\mathbf{r}_t = \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} - \begin{bmatrix} \hat{x}_{t+1}^{\text{phys}} \\ \hat{y}_{t+1}^{\text{phys}} \end{bmatrix}.$$

The final prediction is obtained as

$$\hat{\mathbf{p}}_{t+1} = \begin{bmatrix} \hat{x}_{t+1}^{\text{phys}} \\ \hat{y}_{t+1}^{\text{phys}} \end{bmatrix} + \hat{\mathbf{r}}_t.$$

**LSTM Architecture and Training.** Residuals are predicted using a single layer long short term memory network that takes as input a fixed length history

$$[\mathbf{s}_{t-H+1}, \ldots, \mathbf{s}_t] \in \mathbb{R}^{H \times 12},$$

with $H = 50$. The LSTM hidden dimension is 128, and the output is a two dimensional residual $(r_x, r_y)$. The network is trained by minimizing mean squared error between predicted and true residuals using supervised data generated from the environment.

**Model Mismatch.** The predictor does not observe actuator health and does not model degradation dynamics. As actuator effectiveness degrades over time, the physics prior becomes increasingly inaccurate. Although the LSTM can partially compensate using temporal context, prediction errors remain time varying and correlated, providing a nonstationary residual process.

### E.4.3 Evaluation Policy

All trajectories are generated using a fixed model predictive path integral control policy. The policy is used solely to excite the system and is not part of the proposed uncertainty estimation method.

**Reference Motion.** The policy tracks a figure eight trajectory in the horizontal plane:

$$x_r(t) = A\sin(\omega t), \quad y_r(t) = A\sin(2\omega t),$$

with amplitude $A = 3.0$ and angular frequency $\omega = 0.25$. A constant altitude reference $z_r = 2.0$ m is enforced using a proportional derivative controller on thrust.

**MPPI Control.** At each time step, the policy samples $N = 30$ control sequences over a horizon of $H = 35$ by perturbing a nominal sequence with Gaussian noise. Each sequence is rolled out using the environment dynamics, and a quadratic cost penalizing position tracking error, velocity error, attitude deviation, and control effort is accumulated. Sampled trajectories are weighted using an exponential transformation of cost, and the nominal sequence is updated accordingly. The first control input is applied, and the horizon is receded.

**Policy Independence.** The policy does not observe actuator health, predictor outputs, or uncertainty regions. The same policy and parameters are used across all experiments. Consequently, differences in coverage and region geometry arise solely from the uncertainty estimation method rather than from changes in control behavior.

### E.4.4 Expert Weights

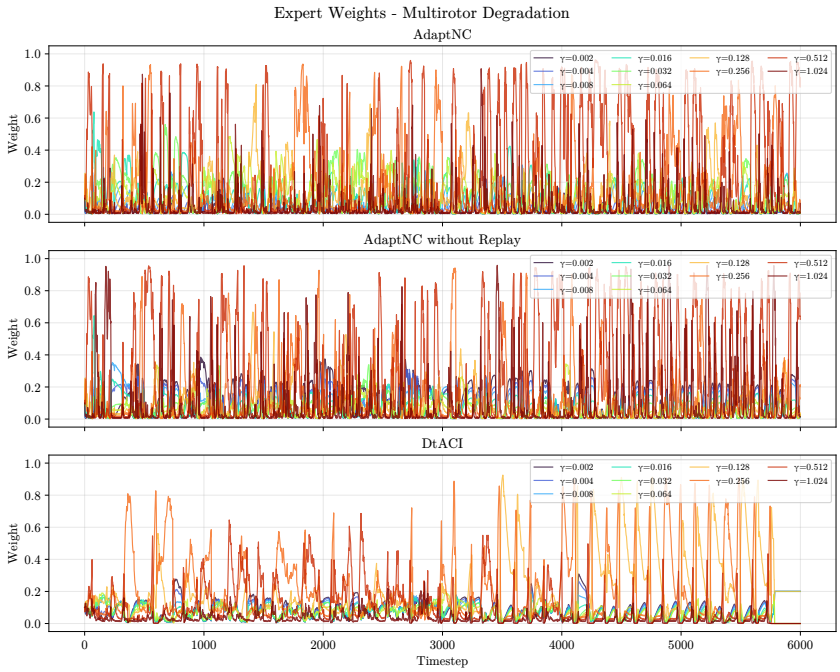



Figure 12: This figure shows the evolution of expert weights in the multirotor navigation setting. The plots illustrate how weights change over time across the different methods. The pronounced oscillatory pattern observed for AdaptNC arises from its adaptive reweighting mechanism, which enables rapid adjustment of expert weights in response to changes in the nonconformity score and contributes to maintaining coverage. In contrast, DtACI and AdaptNC without Replay exhibit slower weight adaptation and assign greater weight to intermediate values of $\gamma$ toward the end of the rollout, resulting in reduced responsiveness compared to AdaptNC.

## E.5 Hyperparameter Sensitivity Analysis

| $W$ | $t_s$ | Global Cov. (%) | Mean Local Cov. (%) | Finite Vol. ($\mathrm{m}^2$) | Vacuous (%) |
|---|---|---|---|---|---|
| | | | **Indoor Localization** | | |
| 25 | 40 | 92.67 ± 0.27 | 92.67 ± 0.23 | 31.19 ± 1.00 | 34.20 ± 0.67 |
| 50 | 40 | 91.82 ± 0.25 | 91.80 ± 0.24 | 31.57 ± 0.84 | 28.48 ± 0.54 |
| 75 | 40 | 91.39 ± 0.54 | 91.39 ± 0.54 | 33.59 ± 0.73 | 23.32 ± 0.51 |
| 100 | 40 | 90.54 ± 0.22 | 90.50 ± 0.21 | 35.13 ± 1.33 | 18.49 ± 0.62 |
| 125 | 20 | 86.85 ± 0.11 | 86.80 ± 0.12 | 27.30 ± 0.13 | 9.85 ± 0.52 |
| **125** | **40** | **90.05 ± 0.12** | **90.00 ± 0.12** | **34.88 ± 1.37** | **16.22 ± 0.37** |
| 125 | 60 | 91.32 ± 0.21 | 91.26 ± 0.23 | 40.72 ± 1.45 | 18.53 ± 0.57 |
| 125 | 80 | 92.32 ± 0.09 | 92.27 ± 0.10 | 43.54 ± 3.99 | 20.45 ± 1.74 |
| 125 | 100 | 92.48 ± 0.22 | 92.41 ± 0.22 | 45.46 ± 2.31 | 19.87 ± 0.49 |
| 125 | 120 | 92.43 ± 0.16 | 92.34 ± 0.16 | 46.02 ± 4.09 | 21.22 ± 1.34 |
| 125 | 150 | 91.28 ± 0.91 | 91.21 ± 0.90 | 47.31 ± 1.79 | 19.15 ± 0.47 |
| 150 | 40 | 89.56 ± 0.24 | 89.50 ± 0.24 | 36.67 ± 1.66 | 13.59 ± 0.15 |
| | | | **Social Navigation** | | |
| 25 | 80 | 88.72 ± 0.64 | 88.64 ± 0.57 | 43.01 ± 3.30 | 45.79 ± 3.84 |
| 50 | 80 | 87.77 ± 0.14 | 87.77 ± 0.10 | 43.76 ± 2.19 | 37.54 ± 3.83 |
| 75 | 20 | 93.16 ± 0.38 | 93.16 ± 0.37 | 25.19 ± 0.41 | 12.58 ± 0.33 |
| 75 | 40 | 94.22 ± 0.29 | 94.19 ± 0.28 | 34.53 ± 2.44 | 26.98 ± 2.77 |
| 75 | 60 | 93.79 ± 0.35 | 93.76 ± 0.39 | 40.81 ± 1.75 | 33.81 ± 3.30 |
| **75** | **80** | **90.41 ± 1.06** | **90.40 ± 1.03** | **46.26 ± 3.23** | **32.31 ± 5.26** |
| 75 | 100 | 88.42 ± 1.57 | 88.36 ± 1.67 | 47.22 ± 3.70 | 30.97 ± 3.81 |
| 75 | 120 | 86.34 ± 1.11 | 86.19 ± 1.11 | 51.86 ± 1.42 | 28.63 ± 3.19 |
| 75 | 150 | 83.65 ± 0.77 | 83.75 ± 0.80 | 51.03 ± 2.82 | 26.72 ± 3.96 |
| 100 | 80 | 93.65 ± 0.34 | 93.64 ± 0.29 | 52.04 ± 6.12 | 26.70 ± 3.82 |
| 125 | 80 | 94.02 ± 0.49 | 93.98 ± 0.50 | 53.56 ± 5.84 | 22.32 ± 3.36 |
| 150 | 80 | 94.14 ± 0.33 | 94.15 ± 0.33 | 58.55 ± 7.29 | 17.97 ± 1.55 |
| | | | **Multirotor Tracking** | | |
| 25 | 20 | 95.99 ± 0.31 | 96.00 ± 0.32 | 0.0120 ± 0.0018 | 50.08 ± 0.83 |
| 25 | 40 | 94.81 ± 0.23 | 94.77 ± 0.23 | 0.0199 ± 0.0031 | 51.76 ± 1.49 |
| 25 | 60 | 92.84 ± 1.04 | 92.78 ± 1.06 | 0.0265 ± 0.0024 | 47.49 ± 1.76 |
| **25** | **80** | **89.87 ± 0.40** | **89.89 ± 0.52** | **0.0292 ± 0.0046** | **44.09 ± 1.17** |
| 25 | 100 | 86.84 ± 0.44 | 86.82 ± 0.41 | 0.0339 ± 0.0040 | 41.73 ± 0.57 |
| 25 | 120 | 86.89 ± 0.32 | 87.02 ± 0.42 | 0.0344 ± 0.0010 | 38.87 ± 0.92 |
| 25 | 150 | 86.94 ± 0.49 | 86.84 ± 0.51 | 0.0389 ± 0.0059 | 38.47 ± 1.31 |
| 50 | 80 | 90.69 ± 0.32 | 90.71 ± 0.36 | 0.0267 ± 0.0056 | 36.64 ± 1.08 |
| 75 | 80 | 92.47 ± 0.38 | 92.61 ± 0.34 | 0.0249 ± 0.0053 | 31.44 ± 1.64 |
| 100 | 80 | 94.06 ± 0.63 | 93.99 ± 0.63 | 0.0232 ± 0.0044 | 25.71 ± 0.73 |
| 125 | 80 | 90.52 ± 0.91 | 90.40 ± 0.92 | 0.0216 ± 0.0043 | 20.80 ± 0.25 |
| 150 | 80 | 90.40 ± 1.24 | 90.33 ± 1.27 | 0.0223 ± 0.0049 | 18.69 ± 1.45 |

Table 7: Slices of the joint hyperparameter grid through each environment's selected configuration (bold), reporting mean ± population standard deviation across three pilot seeds. Each slice holds one parameter fixed while varying the other. Coverage and volume respond differently across environments. Finite volume averages all finite predictions; Table 1 additionally conditions on coverage. These are validation results, distinct from the 50-seed evaluation.

| Method | Indoor Localization (%) | Social Navigation (%) | Multirotor Tracking (%) |
|---|---|---|---|
| **ANC** | $15.55 \pm 0.90$ | $33.12 \pm 3.50$ | $44.30 \pm 1.49$ |
| ANC NR | $14.26 \pm 0.58$ | $20.27 \pm 2.14$ | $36.21 \pm 1.79$ |
| DtACI | $14.00 \pm 0.72$ | $17.74 \pm 1.53$ | $33.88 \pm 1.46$ |
| Sp. CP | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| Sh. CP | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| ANC uniform | $15.23 \pm 0.74$ | $19.33 \pm 1.80$ | $35.42 \pm 1.76$ |
| SF-OGD | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| SAOCP | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |

Table 8: Percentage of timesteps with vacuous (unbounded) prediction regions, mean $\pm$ population standard deviation across the same 50 evaluation seeds as Table 1.

| | Indoor Localization | | | Social Navigation | | | Multirotor Tracking | | |
|---|---|---|---|---|---|---|---|---|---|
| ACI $\gamma$ | Global (%) | Vol. ($\mathrm{m}^2$) | Vacuous (%) | Global (%) | Vol. ($\mathrm{m}^2$) | Vacuous (%) | Global (%) | Vol. ($\mathrm{m}^2$) | Vacuous (%) |
| 0.002 | $89.76 \pm 0.13$ | $67.38 \pm 7.57$ | $0.00 \pm 0.00$ | $89.40 \pm 0.16$ | $107.87 \pm 12.98$ | $1.93 \pm 1.54$ | $89.24 \pm 0.01$ | $0.3158 \pm 0.0097$ | $20.97 \pm 1.68$ |
| 0.004 | $89.89 \pm 0.11$ | $68.41 \pm 7.43$ | $0.02 \pm 0.06$ | $89.75 \pm 0.11$ | $105.54 \pm 12.74$ | $3.33 \pm 1.69$ | $89.66 \pm 0.01$ | $0.3152 \pm 0.0098$ | $21.62 \pm 1.63$ |
| 0.008 | $89.96 \pm 0.10$ | $67.22 \pm 7.06$ | $0.34 \pm 0.22$ | $89.92 \pm 0.09$ | $101.34 \pm 12.17$ | $5.11 \pm 1.75$ | $89.87 \pm 0.01$ | $0.3145 \pm 0.0097$ | $22.58 \pm 1.43$ |
| 0.016 | $90.01 \pm 0.07$ | $61.72 \pm 5.90$ | $1.58 \pm 0.43$ | $89.99 \pm 0.08$ | $94.53 \pm 11.37$ | $7.11 \pm 1.65$ | $89.97 \pm 0.01$ | $0.3128 \pm 0.0096$ | $23.90 \pm 1.49$ |
| 0.032 | $90.02 \pm 0.05$ | $54.05 \pm 4.86$ | $3.60 \pm 0.56$ | $90.03 \pm 0.06$ | $84.49 \pm 9.89$ | $9.20 \pm 1.65$ | $90.02 \pm 0.01$ | $0.3098 \pm 0.0097$ | $25.69 \pm 1.43$ |
| 0.064 | $90.03 \pm 0.04$ | $46.27 \pm 4.06$ | $6.52 \pm 0.65$ | $90.03 \pm 0.05$ | $70.22 \pm 7.87$ | $11.69 \pm 1.79$ | $90.05 \pm 0.01$ | $0.3058 \pm 0.0097$ | $28.33 \pm 1.44$ |
| 0.128 | $90.02 \pm 0.02$ | $39.90 \pm 3.37$ | $10.84 \pm 0.79$ | $90.03 \pm 0.03$ | $55.31 \pm 5.82$ | $14.18 \pm 1.62$ | $90.06 \pm 0.01$ | $0.2988 \pm 0.0102$ | $30.82 \pm 1.32$ |
| 0.256 | $90.01 \pm 0.01$ | $35.21 \pm 3.06$ | $17.44 \pm 0.77$ | $90.02 \pm 0.02$ | $46.69 \pm 4.63$ | $18.89 \pm 1.43$ | $90.07 \pm 0.01$ | $0.2903 \pm 0.0107$ | $34.33 \pm 1.13$ |
| 0.512 | $90.01 \pm 0.01$ | $32.40 \pm 2.87$ | $28.61 \pm 1.02$ | $90.01 \pm 0.01$ | $46.26 \pm 4.57$ | $27.86 \pm 1.24$ | $90.07 \pm 0.01$ | $0.2802 \pm 0.0110$ | $39.59 \pm 1.00$ |
| 1.024 | $90.00 \pm 0.00$ | $31.82 \pm 2.81$ | $48.17 \pm 0.69$ | $90.00 \pm 0.00$ | $49.24 \pm 5.42$ | $45.44 \pm 0.95$ | $90.07 \pm 0.00$ | $0.2668 \pm 0.0100$ | $48.71 \pm 0.90$ |

Table 9: Fixed-step ACI comparisons (mean $\pm$ population standard deviation across 50 evaluation seeds). Volume conditions on finite, covered predictions, as in Table 1. Small finite-region volumes at large step sizes must be interpreted together with the vacuous prediction rate.

| Metric | Indoor Localization | Social Navigation | Multirotor Tracking |
|---|---|---|---|
| Global coverage (%) | $90.06 \pm 0.11$ | $89.13 \pm 0.25$ | $90.13 \pm 0.01$ |
| Mean local coverage (%) | $90.04 \pm 0.07$ | $89.42 \pm 0.29$ | $90.24 \pm 0.00$ |
| Finite covered volume ($\mathrm{m}^2$) | $56.02 \pm 5.23$ | $96.06 \pm 13.97$ | $0.3121 \pm 0.0061$ |
| Mean runtime (ms) | $0.004 \pm 0.000$ | $0.005 \pm 0.002$ | $0.004 \pm 0.000$ |

Table 10: SF-OGD [16] performance and runtime across 50 evaluation seeds (mean $\pm$ population standard deviation). Volume averages finite regions at covered timesteps; vacuous-region rates are reported in Table 8.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly states the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have pointed out the limitation of our work in the Appendix A of the paper.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have included the assumptions in all theorems and their proofs in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:: We have included all the experimental details in the Appendix. The code is publicly available at `https://github.com/nandantumu/conformal_region_designer/`.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/nandantumu/conformal_region_designer/`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details are available in the Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have performed each of our experiments for 50 random seeds and reported the mean and std deviation of the evaluation metrics aggregated across them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included all the details on compute resources in the Runtime Analysis of Section 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics, in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The research in the paper does not harm or raise technology to harm the society.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The experiments conducted in the paper only uses simulation-based data of dynamical systems, therefore, it poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited and given credits to all the creators/original owners of the assets, in the paper wherever it is applicable.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have provided the code along with a comprehensively documented Readme.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.