COST-AWARE ADAPTIVE CONFORMAL INFERENCE FOR RUNTIME ASSURANCE IN DYNAMIC ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper addresses the problem of providing runtime assurance for systems operating online under unknown and potentially time-varying data distributions. We propose Cost-Aware Adaptive Conformal Inference (ACI), a novel framework that incorporates constraint violation costs directly into the conformal adaptation mechanism. Our key insight is that uncertainty margins should adapt not only to the frequency of constraint violations but also to their severity. We formalize this through a cost-aware loss function that couples the miscoverage indicator with violation costs. By exploiting the structure of this loss, our approach provides a dual statistical guarantee: simultaneously bounding the long-run average violation frequency (reliability) and cumulative violation cost (harm). By weighting prediction failures according to their severity, the algorithm enables the controller to respond proportionally to violation severity, expanding prediction sets aggressively when necessary while maintaining efficiency during nominal operation. We integrate Cost-Aware ACI into a robust control synthesis framework, creating a closed-loop system that balances task performance with runtime risk control without requiring explicit model knowledge. Experiments validate its effectiveness for online risk-aware controller synthesis.

1 INTRODUCTION

Deploying autonomous systems in safety-critical domains such as robotics and autonomous vehicles requires rigorous operational assurances. In formal verification, "safety" is typically defined as strict set invariance, which guarantees that the system never violates constraints (Ames et al., 2019). While this is a gold standard, establishing such absolute guarantees often presupposes accurate models and static environments. In real-world applications characterized by unknown dynamics and distribution shifts (Dixit et al., 2023), strict invariance is often unattainable, as these factors preclude effective offline calibration. The challenge, therefore, lies in proving runtime assurance despite these uncertainties.

To address this, we turn to the paradigm of runtime assurance and risk control. Rather than certifying total absence of failure, we aim to bound the frequency and severity of constraint violations online. Recent progress in distribution-free uncertainty quantification has positioned Conformal Prediction (CP) as a promising tool for such applications (Vovk et al., 2005; Lindemann et al., 2025). Unlike methods that rely on strong distributional assumptions, CP provides rigorous finite-sample coverage guarantees. ACI extends CP to online, non-exchangeable settings by employing a control-theoretic update rule to adjust quantile thresholds, thereby maintaining long-run coverage under distribution shifts (Gibbs & Candes, 2021). This has made ACI an attractive foundation for statistical risk control in dynamic environments. However, standard ACI suffers from a critical limitation we term severity-blindness. Because its update rule is driven by a binary miscoverage indicator, its response is identical for a near-miss and a catastrophic failure. The algorithm adapts to whether a violation occurred, not to its physical severity. In dynamic control systems, this is a significant risk: a series of severe violations can cause the system state to diverge faster than the algorithm can adapt. This limitation of standard ACI motivates the need for a refined framework. We contend that runtime assurance necessitates simultaneous regulation of both statistical reliability (violation frequency) and physical harm (violation intensity).

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

In this paper, we introduce Cost-Aware ACI, a novel framework that integrates violation costs directly into the conformal adaptation process. Our key insight is that assurance margins should adapt not only to the rate of constraint violations but also to the operational severity of resulting constraint violations. By weighting prediction coverage failures according to their costs, our algorithm enables controllers to recover proportionally from dangerous situations, expanding prediction sets aggressively when needed while maintaining efficiency during normal operation. This creates a natural feedback mechanism where the system becomes more conservative precisely when—and only when—the physics demands it. Finally, we validate the efficacy of this framework through a series of experiments. Across all experiments, our method consistently achieves substantially lower long-run average violation costs and frequencies than standard ACI, providing a practical framework for managing risk in unknown environments. Moreover, the proposed method is robust to hyperparameter variations, underscoring its practical applicability. The contributions are summarized below:

• **Cost-Aware Conformal Adaptation:** We introduce a novel update rule based on a cost-aware loss function that couples miscoverage indicators with constraint-violation costs, enabling simultaneous control of both the frequency of constraint violations and their long-run average cumulative cost.

• **Practical Online Control Algorithm:** We develop a data-driven algorithm for online risk-aware controller synthesis, termed *Online Adaptive Conformal Cost Control*. This algorithm integrates our cost-aware update rule into a closed-loop synthesis process that effectively balances task performance with runtime risk control in unknown and dynamic environments.

• **Rigorous Theoretical Guarantees:** We provide rigorous theoretical proofs for the proposed framework. We derive explicit, non-asymptotic bounds on both the violation frequency and the average cumulative cost. Furthermore, we demonstrate the algorithm's adaptive conservatism and formalize the update rule as an online optimization procedure with an explicit regret characterization.

Related Work: A wide array of control techniques falls under the umbrella of constrained control. Prominent model-based methods, including model predictive control (MPC) (Garcia et al., 1989; Morari & Lee, 1999), control barrier functions (Ames et al., 2016; 2019), funnel control (Bechlioulis & Rovithakis, 2008), and their temporal logic extensions (Lindemann & Dimarogonas, 2018; Farahani et al., 2018), can provide strong guarantees, but usually require accurate system models and are hard to apply to complex or learning-enabled systems. Data-driven approaches like reinforcement learning (Ibarz et al., 2021), imitation learning (Torabi et al., 2018), and learned controllers (Dean et al., 2021; Taylor et al., 2020; Srinivasan et al., 2020; Robey et al., 2020; Lindemann et al., 2024), often perform well empirically but typically lack verifiable operational guarantees, creating a critical gap between flexibility and reliability.

CP (Vovk et al., 2005; Shafer & Vovk, 2008) provides a path toward bridging this gap, which is a model-agnostic and distribution-free framework characterized by its rigorous finite-sample marginal coverage guarantee. It has recently been applied to control problems, for instance, in learning uncertainty-informed safety filters (Lindemann et al., 2023; Strawn et al., 2023) and predictive monitors (Cairoli et al., 2023), as surveyed in (Lindemann et al., 2025). However, CP's validity hinges on the assumption of exchangeability, which is violated in non-stationary settings like time-series control. This has motivated adaptive methods for distribution shift, such as weighted conformal approaches (Tibshirani et al., 2019; Barber et al., 2023).

ACI (Gibbs & Candes, 2021) is a powerful control-theoretic solution for non-exchangeable time-series data (Zaffran et al., 2022; Sousa et al., 2024; Szabadváry, 2024). Drawing on online learning (Zinkevich, 2003), ACI adaptively tunes the prediction interval size to regulate the long-run average miscoverage rate to a user-specified level. Subsequent variants, including Conformal PID (Angelopoulos et al., 2023), Bellman Conformal Inference (Yang et al., 2024), and improved step-size adaptation (Gibbs & Candès, 2024; Podkopaev et al., 2024), have improved its stability and responsiveness. While ACI has been integrated into critical domains such as motion planning (Dixit et al., 2023), control barrier functions (Zhou et al., 2024), safe online planning in POMDPs (Sheng et al., 2024), and constrained reinforcement learning (Yao et al., 2024), its standard formulation only controls the frequency of violations, not magnitude. More general adaptive calibration methods have extended ACI beyond binary miscoverage. In particular, (Feldman et al., 2024) develop a general framework for controlling user-specified losses. In contrast, our work addresses runtime assurance for dynamical systems, where both statistical reliability (violation frequency) and physical harm (vi-

olation intensity) must be regulated. Since the framework in (Feldman et al., 2024) is intentionally generic, it does not prescribe how the calibrated loss should encode these safety objectives, and its general theoretical guarantees do not directly imply the separate safety guarantees established in our work. We achieve this through a novel 'Cost-Aware' loss function, which explicitly couples the binary miscoverage error with the physical cost.

Notation: $\mathbb{N}$ denotes the set of natural numbers; $\mathbb{N}_{\geq i}$ denotes the set of natural numbers being larger than or equal to i; $\mathbb{R}$ denotes the set of real numbers; $\mathbb{R}^n$ denotes n-dimensional Euclidean space; $\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers; $|\mathcal{A}|$ denotes the cardinality of a set $\mathcal{A}$; $\mathbb{I}(\cdot)$ denotes the indicator function, which evaluates to 1 if the condition is true, and 0 otherwise.

2 PRELIMINARIES

This section formalizes the problem of online risk-aware controller synthesis for systems with unknown, time-varying dynamics, and reviews ACI together with its limitations in this critical context.

2.1 PROBLEM FORMULATION

System Dynamics. Consider a dynamical system described by the state update equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{1}$$

where $k \in \mathbb{N}$ is the time step, $\boldsymbol{x}_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the system state, and $\boldsymbol{u}_k \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input. We operate under the following assumption.

Assumption 1 (Unknown Dynamics)**.** *The transition function $\boldsymbol{f} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is deterministic but unknown and may vary over time, reflecting a dynamic operational environment.*

Online Control Protocol. The controller synthesis protocol operates under strict causal constraints. At each step k, the control input $\boldsymbol{u}_k$ must be generated using only the current state $\boldsymbol{x}_k$ and historical observations $\{(\boldsymbol{x}_i, \boldsymbol{u}_i)\}_{i=0}^{k-1}$. This online setting requires balancing between *exploration* (to learn the system dynamics) and *exploitation* (to achieve task objectives while maintaining assurance).

Constraint and Cost Formulation. Constraint satisfaction is defined by a known scalar function $h : \mathcal{X} \to \mathbb{R}$, where the *admissible set* (or nominal safe set) is its zero-superlevel set, $\mathcal{S} = \{\boldsymbol{x} \in \mathcal{X} \mid h(\boldsymbol{x}) \geq 0\}$. To quantify the severity of violations, we define a state-dependent cost function:

$$c(\boldsymbol{x}) = \begin{cases} 0, & \text{if } h(\boldsymbol{x}) \geq 0 \quad \text{(Within admissible set)}, \\ \rho(\boldsymbol{x}), & \text{if } h(\boldsymbol{x}) < 0 \quad \text{(Deviation from admissible set)}, \end{cases} \tag{2}$$

where $\rho(\boldsymbol{x}) > 0$ is a known function that penalizes the magnitude of a constraint violation. We assume $\rho(\boldsymbol{x})$ is bounded. For theoretical and numerical stability, it is convenient to normalize this cost to the range $[0, 1]$ by scaling with its known upper bound, without loss of generality.

Control Objectives. The overall objective is to optimize a task performance metric, J_{task}, while satisfying critical operational constraints. The metric J_{task} can take various forms depending on the user-specified performance preferences. As strict, state-wise enforcement of the admissible set ($\boldsymbol{x}_k \in \mathcal{S}, \forall k$) is often infeasible in unknown and dynamic environments, we adopt a more pragmatic, two-part risk definition that constrains both the long-run average violation cost and frequency:

$$J = \limsup_{N\to\infty} \tfrac{1}{N} \textstyle\sum_{k=1}^{N} c(\boldsymbol{x}_k) \leq D_{\text{tol}}, \quad V = \limsup_{N\to\infty} \tfrac{1}{N} \textstyle\sum_{k=1}^{N} \mathbb{I}(c(\boldsymbol{x}_k) > 0) \leq \alpha,$$

where $D_{\text{tol}} \geq 0$ is a user-specified tolerance for the average cost and $\alpha \in (0, 1)$ is the maximum allowable violation frequency. This formulation transforms the rigid invariance requirement into two manageable "risk budgets", permitting sparse violations while ensuring their cumulative harm remains bounded. Crucially, this ensures the control problem remains feasible even if the system briefly exits the admissible set, enabling the controller to minimize damage rather than failing. For many applications, this is the most practical and achievable type of runtime assurance.

2.2 ADAPTIVE CONFORMAL INFERENCE

In this subsection, we briefly review ACI (Gibbs & Candes, 2021), a control-theoretic extension of CP for non-exchangeable data. Consider an online prediction setting with a sequence of covariate–target pairs $(\boldsymbol{x}_0, y_0), (\boldsymbol{x}_1, y_1), \ldots$, where $y_k \in \mathcal{Y}$. At each time step k, given the current covariate

$\boldsymbol{x}_k$ and past observations, the goal is to construct a prediction set $\hat{C}_k \subseteq \mathcal{Y}$ for the unknown target y_k, targeting a long-run miscoverage rate α. Since exchangeability is typically violated in online settings, ACI does not fix the miscoverage level α; instead, it maintains a time-varying parameter α_k, updated by a gradient-like mechanism that decreases after coverage failures and increases otherwise.

At each step k, ACI uses a non-negative conformity score $s_k = S(\boldsymbol{x}_k, y_k; \hat{g}_{k-1})$ based on a predictive model $\hat{g}_{k-1}$. The empirical quantile function, $\hat{Q}_k(\cdot)$, is constructed from a calibration set of past scores, $\mathcal{D}_k^{\text{cal}} = \{s_i\}_{i<k}$:

$$\hat{Q}_k(p) := \inf\left\{s \in \mathbb{R} : \tfrac{1}{|\mathcal{D}_k^{\text{cal}}|}\left(\textstyle\sum_{s_i \in \mathcal{D}_k^{\text{cal}}} \mathbb{I}(s_i \le s)\right) \ge 1 - p\right\},$$

where $p \in [0, 1]$. ACI uses an adaptive internal variable, α_k, to construct the prediction set $\hat{C}_k(1 - \alpha_k)$ for a target variable y_k:

$$\hat{C}_k(1-\alpha_k) := \begin{cases} \mathcal{Y} & \text{if } \alpha_k \le 0, \\ \emptyset & \text{if } \alpha_k \ge 1, \\ \{y \in \mathcal{Y} : S(\boldsymbol{x}_k, y; \hat{g}_{k-1}) \le \hat{Q}_k(\alpha_k)\} & \text{if } 0 < \alpha_k < 1. \end{cases} \tag{3}$$

The variable α_k is updated via a linear rule, starting with $\alpha_0 = \alpha$:

$$\alpha_{k+1} := \alpha_k + \gamma(\alpha - e_k), \tag{4}$$

where $e_k = \mathbb{I}(y_k \notin \hat{C}_k(1 - \alpha_k))$ is the miscoverage indicator and $\gamma > 0$ is a step-size. This update ensures a powerful long-run average miscoverage guarantee.

Proposition 1 (Long-run Average Miscoverage (Gibbs & Candes, 2021)). *For the update rule in (4), the cumulative miscoverage rate converges to the target level α, i.e., $\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} e_k = \alpha$.*

However, while ACI can provide a powerful tool to enforce the frequency constraint $V \le \alpha$ as shown in Theorem 1 below, its standard form is insufficient for our dual-objective risk control problem. The core issue is "severity-blindness": the binary nature of e_k means the system's response is identical for a near-miss and a catastrophic failure. This decoupling from the physical cost $c(\boldsymbol{x}_k)$ means ACI makes no effort to satisfy the cost constraint $J \le D_{\text{tol}}$. Although prior work (Feldman et al., 2024) offers a general online risk-control framework to regulate user-specified losses, they do not prescribe how such losses should be constructed to capture operational safety in dynamical systems. Our objective is to simultaneously characterize two concrete safety quantities: constraint violation frequency and cumulative violation cost. This critical gap necessitates a new, cost-aware framework.

3 Adaptive Conformal Cost Control

This section introduces Cost-Aware ACI, a framework designed to overcome the severity-blindness of standard ACI in runtime assurance. It explicitly integrates the cost of constraint violations into the adaptive mechanism, ensuring that uncertainty bounds adapt not only to a miscoverage event but also in proportion to its severity. We embed this framework into an online algorithm to synthesize a controller with dual statistical guarantees. It simultaneously bounds the long-run constraint violation frequency and average cumulative cost, facilitating rapid recovery from dangerous transients.

3.1 Cost-Aware ACI

We define a nonconformity score that quantifies prediction uncertainty in terms of constraint satisfaction. Let $\Phi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ be the composite function that maps a state-action pair to the assurance value of the resulting state, i.e., $\Phi(\boldsymbol{x}, \boldsymbol{u}) := h(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}))$. We learn a surrogate model $\widehat{h}(\boldsymbol{x}, \boldsymbol{u}) \approx \Phi(\boldsymbol{x}, \boldsymbol{u})$ from online data. The prediction error at time k, which results from applying control input $\boldsymbol{u}_{k-1}$ at the previous state $\boldsymbol{x}_{k-1}$, is the absolute residual:

$$s_k := \left|\Phi(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) - \widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})\right| = \left|h(\boldsymbol{x}_k) - \widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})\right|, \tag{5}$$

where $k \in \mathbb{N}_{\ge 1}$. This residual, s_k, serves as our nonconformity score, measuring the discrepancy between predicted and realized runtime assurance. A large score indicates a significant prediction error that could compromise operational reliability. To guarantee the well-posedness of the online controller synthesis optimization ((8) in Section 3.2), the prediction residuals must remain finite.

Specifically, unbounded residuals could lead to infinitely large uncertainty margins, rendering the robust constraint unsatisfiable. This assumption is mild and practically justified for physical control systems. Since system states and control inputs are subject to inherent physical limits, both the true assurance function and its learned surrogate are naturally bounded, guaranteeing finite residuals.

Assumption 2 (Bounded Nonconformity)**.** *There exists a constant $M > 0$ such that $s_k \leq M$ for all $k \in \mathbb{N}$.*

Building on the nonconformity score, we define a loss function that incorporates the cost of a constraint violation. First, we define the empirical $(1-\delta)$-quantile function, $\hat{Q}_k$, of the historical scores $\mathcal{C} = \{s_1, \ldots, s_{k-1}\}$ and a value $\epsilon > 0$:

$$\hat{Q}_k(\delta) := \begin{cases} M, & \text{if } \delta < 0, \\ -\epsilon, & \text{if } \delta > 1, \\ \inf\left\{c \in \mathbb{R} : \frac{1}{|\mathcal{C}|}\sum_{r\in\mathcal{C}} \mathbb{I}(r \leq c) \geq 1-\delta\right\}, & \text{if } 0 \leq \delta \leq 1. \end{cases}$$

Standard ACI uses a binary miscoverage indicator, $e_k = \mathbb{I}(s_k > \hat{Q}_k(\delta_k))$, as feedback. To introduce severity awareness, we augment this indicator with the constraint violation cost, $c(\boldsymbol{x}_k)$, defining the **boosted loss**:

$$L_k := e_k \cdot (1 + \beta c(\boldsymbol{x}_k)). \tag{6}$$

Equivalently, since $c(\boldsymbol{x}_k) = 0$ when $h(\boldsymbol{x}_k) \geq 0$ and $c(\boldsymbol{x}_k) = \rho(\boldsymbol{x}_k) > 0$ when $h(\boldsymbol{x}_k) < 0$, we have

$$L_k = \begin{cases} 0 & \text{if } s_k \leq \hat{Q}_k(\delta_k), \\ 1 & \text{if } s_k > \hat{Q}_k(\delta_k) \text{ and } h(\boldsymbol{x}_k) \geq 0, \\ 1 + \beta\rho(\boldsymbol{x}_k) & \text{if } s_k > \hat{Q}_k(\delta_k) \text{ and } h(\boldsymbol{x}_k) < 0. \end{cases}$$

The parameter $\beta \geq 0$ is a severity-sensitivity parameter that controls the trade-off between frequency and severity of constraint violations:

• $\beta = 0$: Recovers standard ACI, where all failures are weighted equally regardless of severity.

• $\beta > 0$: Introduces cost awareness. A miscoverage event that causes constraint violations ($h(\boldsymbol{x}_k) < 0$) is penalized more heavily, in proportion to $\rho(\boldsymbol{x}_k)$.

With the boosted loss defined, we introduce the update mechanism for the adaptive miscoverage level δ_{k+1}. This parameter is adjusted via online gradient descent on the loss L_k:

$$\delta_{k+1} = \delta_k + \gamma(\alpha - L_k), \tag{7}$$

where $\alpha \in (0, 1)$ is the target risk budget and $\gamma > 0$ is the learning rate. This update rule creates three distinct operational regimes based on the outcome at time step k:

1. Within Admissible Set ($L_k = 0$): When the prediction is accurate ($s_k \leq \hat{Q}_k$), the loss is zero. The change in δ, denoted $\Delta\delta_{k+1} = \delta_{k+1} - \delta_k$, is positive: $\Delta\delta_{k+1} = \gamma\alpha$. This gradually increases δ_k, tightening the uncertainty bounds to improve control performance.

2. Benign Failure ($L_k = 1$): When a miscoverage event occurs but the system remains within the admissible set ($h(\boldsymbol{x}_k) \geq 0$), the loss is 1. The change is negative: $\Delta\delta_{k+1} = -\gamma(1-\alpha)$. This penalizes the prediction error by making the uncertainty bounds more conservative.

3. Critical Failure ($L_k > 1$): When miscoverage coincides with a constraint violation ($h(\boldsymbol{x}_k) < 0$), the loss is $1 + \beta\rho(\boldsymbol{x}_k)$. The change becomes sharply negative: $\Delta\delta_{k+1} = -\gamma(1 - \alpha + \beta\rho(\boldsymbol{x}_k))$. The decrease is now proportional to the violation's severity, triggering a strong corrective response.

The key innovation is this cost-sensitive adaptation during critical failures. While standard ACI uses a fixed correction step, Cost-Aware ACI employs a cost-scaled update. This enables proportional recovery: severe violations trigger stronger corrections, reducing the risk of prolonged violations.

3.2 Runtime Assurance Control Synthesis

Having established the adaptive uncertainty quantification, we now integrate it into online control synthesis. At each step k, the newly calibrated parameters ensure the next action $\boldsymbol{u}_k$ is admissible. The calibrated quantile $\hat{Q}_{k+1}(\delta_{k+1})$ provides a robust lower bound on the true assurance value:

$$\Phi(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq \widehat{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \hat{Q}_{k+1}(\delta_{k+1}).$$

To provide runtime assurance, we require the predicted assurance value to be non-negative, leading to the robust constraint $\widehat{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq \hat{Q}_{k+1}(\delta_{k+1})$. Consequently, at each time step k, we determine the optimal control input $\boldsymbol{u}_k^*$ by solving the following MPC optimization problem:

$$
\begin{aligned}
(\boldsymbol{u}_k^*, \{\boldsymbol{u}_i^{**}\}_{i=k+1}^{k+N-1}) = & \underset{\{\boldsymbol{u}_i\}_{i=k}^{k+N-1}}{\arg\min} \; J_{\text{task}}(\{\boldsymbol{x}_i\}_{i=k+1}^{k+N}, \{\boldsymbol{u}_i\}_{i=k}^{k+N-1}) \\
& s.t. \begin{cases} \widehat{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq \hat{Q}_{k+1}, \\ \boldsymbol{x}_{i+1} = \widehat{\boldsymbol{f}}(\boldsymbol{x}_i, \boldsymbol{u}_i), i = k, \ldots, k+N-1. \end{cases}
\end{aligned} \tag{8}
$$

Here, $\widehat{\boldsymbol{f}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ serves as a surrogate model for the transition function $\boldsymbol{f} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$. It is worth noting that we employ a separate estimator $\widehat{h}$ to approximate Φ, rather than relying on the composition $h(\widehat{\boldsymbol{f}}(\cdot))$. This design decouples assurance prediction errors from state dynamics errors. By directly calibrating the scalar residual of $\widehat{h}$, we would avoid the excessive conservatism often caused by propagating high-dimensional state uncertainty through nonlinear functions. The constraint $\widehat{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq \hat{Q}_{k+1}$ enforces a reliability level that is dynamically adjusted by the uncertainty margin $\hat{Q}_{k+1}$. Meanwhile, state evolution within the planning horizon is projected using the surrogate dynamics model $\widehat{\boldsymbol{f}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$. Both surrogate models, $\widehat{h}$ and $\widehat{\boldsymbol{f}}$, can be continuously updated online to accurately reflect the current environment.

This optimization establishes a critical feedback loop that balances task performance, prediction reliability, and control conservatism. By adaptively tightening the margin $\hat{Q}_{k+1}$ only when necessary, the system secures reliability during critical environmental shifts while minimizing conservatism during stable operation to maximize task performance.

The complete online adaptive conformal cost control algorithm integrates these components into a cohesive online control framework, outlined in Algorithm 1, with further details in Appendix C.

Algorithm 1 Online Adaptive Conformal Cost Control Algorithm

Require: Target risk budget α, learning rate γ, sensitivity β;
Initialize $\delta_1 = \alpha, \mathcal{C} = \{\}$, models $\widehat{h}$ and $\widehat{\boldsymbol{f}}$; Observe initial state $\boldsymbol{x}_0$;
for $k = 0, 1, \ldots$ **do**
  if $k > 0$ **then**
    Scoring: $s_k = |h(\boldsymbol{x}_k) - \widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}^*)|$;
    Severity Evaluation: $e_k = \mathbb{I}(s_k > \hat{Q}_k(\delta_k))$;
      $c(\boldsymbol{x}_k) = \rho(\boldsymbol{x}_k) \cdot \mathbb{I}(h(\boldsymbol{x}_k) < 0)$;
      $L_k = e_k(1 + \beta c(\boldsymbol{x}_k))$;
    Update: $\delta_{k+1} = \delta_k + \gamma(\alpha - L_k)$;
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{s_k\}$;
      Update $\widehat{h}$ and $\widehat{\boldsymbol{f}}$ (online learning);
  end if
  Calibration: $\hat{Q}_{k+1} \leftarrow$ Quantile$(\{-\epsilon, M\}, 1 - \delta_{k+1})$ if $k = 0$, else Quantile$(\mathcal{C}, 1 - \delta_{k+1})$;
  Control Synthesis: Solve optimization (8) using $\hat{Q}_{k+1}$ to obtain $\boldsymbol{u}_k^*$;
  Apply $\boldsymbol{u}_k^*$ and observe $\boldsymbol{x}_{k+1}$;
end for

4 Theoretical Analysis

This section establishes the theoretical properties of the proposed algorithm. We first introduce mild structural assumptions, then prove the stability of the algorithm by showing that its adaptive parameter remains bounded (Lemma 1). Third, we establish its core convergence property: the long-run average boosted loss converges to the target level α (Lemma 2). Building on this convergence, we derive our main performance guarantees: rigorous upper bounds on both long-run constraint violation frequency (Theorem 1) and average cumulative constraint violation cost (Theorem 2). Finally, we analyze the algorithm's adaptive nature, showing it dynamically adjusts to environmental hostility (Proposition 2) and corresponds to a principled online optimization procedure (Proposition 3). All proofs are shown in Appendix A. We begin by stating the assumptions required for our analysis.

Assumption 3 (Bounded Constraint Violation Costs)**.** *The constraint violation cost is normalized and bounded, i.e.,* $0 \leq \rho(\boldsymbol{x}) \leq 1$ *for all* $\boldsymbol{x} \in \mathcal{X}$*. Consequently, the boosted loss is bounded by* $L_{\max} = 1 + \beta$*.*

Assumption 4 (Feasibility of the Robust Constraint)**.** *For every state* $\boldsymbol{x}$ *reachable by the system, optimization (8) is feasible. That is, the set of control inputs satisfying the constraint* $\widehat{h}(\boldsymbol{x}, \boldsymbol{u}) \geq \hat{Q}_{k+1}$ *is non-empty.*

Assumption 3 is a practical condition reflecting that physical damage is typically finite. Assumption 4 is a standard condition ensuring the control synthesis is well-defined, similar to its use in MPC (Scokaert et al., 2002). In practice, the feasibility required by Assumption 4 is structurally achieved by the availability of a robust recovery policy (e.g., emergency braking or hovering). This hierarchical architecture, which prioritizes a high-performance optimizer while retaining a conservative backup, parallels standard shielding and recovery mechanisms in Safe Reinforcement Learning (Alshiekh et al., 2018; Thananjeyan et al., 2021). Since this recovery policy is designed to satisfy constraints under worst-case uncertainty, its presence ensures that the feasible set for the optimization (8) is never empty. While this guarantees solvability, our Cost-Aware algorithm is designed to proactively adjust margins to prioritize high-performance actions, thereby minimizing the frequency with which the controller is forced to select this conservative recovery option. With these assumptions, we first prove that the adaptive parameter δ_k remains stable.

Lemma 1 (Parameter Boundedness)**.** *Let* $\{\delta_k\}$ *be generated by the update rule in (7). Under Assumptions 2 and 3, if* δ_1 *is initialized within the interval* $[-\gamma(L_{\max} - \alpha), 1 + \gamma\alpha]$*, then* δ_k *remains within this interval for all* $k \in \mathbb{N}$*.*

This stability is crucial, as it prevents erratic behavior and is a prerequisite for proving convergence. We now prove the fundamental convergence property of Cost-Aware ACI: the long-run average cumulative boosted loss converges exactly to the target risk budget α.

Lemma 2 (Long-Run Average Boosted Loss Convergence)**.** *Under the same conditions as Lemma 1, the long-run time-average of the boosted loss converges to the target level* α*, i.e.,* $\lim_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} L_k = \alpha$*.*

This convergence of the boosted loss is the key to unlocking our main theoretical results: explicit upper bounds on the long-run constraint violation frequency V and average cost J.

Theorem 1 (Upper Bound on Constraint Violation Frequency)**.** *Let* $v_k := \mathbb{I}(h(\boldsymbol{x}_k) < 0)$ *be the indicator of a constraint violation at time* k*. Under Assumptions 2, 3 and 4, the violation frequency* V *is upper-bounded by* α*, i.e.,* $V = \limsup_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} v_k \leq \alpha$*.*

Theorem 1 shows that our cost-aware algorithm bounds the long-run constraint violation frequency V. Next, we establish a bound on the cumulative cost J.

Theorem 2 (Upper Bounds on Average Cumulative Cost J)**.** *Under Assumptions 2, 3 and 4, for any risk sensitivity parameter* $\beta > 0$*, long-run average cumulative constraint violation cost* J *satisfies* $J = \limsup_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} c(\boldsymbol{x}_k) \leq \frac{\alpha}{\beta}$*.*

Theorems 1 and 2 provide a direct method for online control design with runtime assurance. A designer can first set the maximum tolerable constraint violation frequency by choosing α (since $V \leq \alpha$). Subsequently, to ensure the average cost J remains below a specified tolerance D_{tol}, the sensitivity parameter β is chosen to satisfy the condition $\alpha/\beta \leq D_{\text{tol}}$.

The preceding theorems provide long-run guarantees. Such finite-time upper bounds for both the constraint violation frequency and the average cumulative cost can be directly obtained by leveraging the results of Lemma 1.

Corollary 1 (Finite-Time Violation Frequency and Cost Bounds)**.** *Let* $V_T := \frac{1}{T} \sum_{k=1}^{T} v_k$ *be the violation frequency and* $J_T := \frac{1}{T} \sum_{k=1}^{T} c(\boldsymbol{x}_k)$ *be the average cumulative cost over the first* T *time steps. The following bounds hold:*

1. Violation Frequency: $V_T \leq \alpha + \frac{\delta_1 - \delta_{T+1}}{T\gamma} = \alpha + O(1/T)$*.*

2. Average Cost: $J_T \leq \frac{\alpha}{\beta} + \frac{\delta_1 - \delta_{T+1}}{\beta T\gamma} - \frac{1}{\beta T} \sum_{k=1}^{T} e_k \leq \frac{\alpha}{\beta} + O(1/T)$*.*

This corollary provides a stronger, more practical guarantee. It shows that the violation frequency and average cost are tightly concentrated around their long-run targets, α and α/β, respectively. The deviation from these targets is controlled by a transient term that diminishes at a rate of $O(1/T)$, ensuring that the system's performance rapidly converges to its desired steady-state behavior.

The next propositions offer deeper insight into the algorithm's adaptive behavior, framing it first as a rational economic agent and second as a principled online learning procedure.

Proposition 2 (Conservation of Risk Budget)**.** *The algorithm dynamically trades off miscoverage frequency against violation severity. Let* $\hat{\alpha}_T := \frac{1}{T}\sum_{k=1}^{T} e_k$ *be the empirical miscoverage rate and* $\bar{c}_T := \frac{\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k)}{\sum_{k=1}^{T} e_k}$ *be the average cost conditional on a miscoverage event. In the long run, these quantities adhere to the conservation law:* $\lim_{T\to\infty} \hat{\alpha}_T(1+\beta\bar{c}_T) = \alpha$.

This proposition reveals the algorithm's economic rationality. It treats α not as a simple frequency target, but as a fixed risk budget. The convergence condition can be rewritten as $\hat{\alpha}_T \longrightarrow \frac{\alpha}{1+\beta\bar{c}_T}$, showing how this budget is spent. This expression demonstrates a dynamic trade-off. In benign environments where the average violation cost $\bar{c}_T$ is low, the denominator approaches 1, and the miscoverage rate $\hat{\alpha}_T$ converges toward the full budget α. However, in hostile environments where $\bar{c}_T$ is high, the term $(1+\beta\bar{c}_T)$ acts as an inflation factor. The "price" of each miscoverage event increases, compelling the algorithm to automatically reduce its miscoverage frequency $\hat{\alpha}_T$ to stay within its fixed budget α. This mechanism enforces harm regulation by shifting from a purely statistical guarantee (bounded failure rate) to an operational one (bounded average cost).

The proposed update rule also admits an online optimization interpretation, while the corresponding regret analysis is provided in Appendix B.

5 Experiments

In this section, we validate the effectiveness of the proposed Online Adaptive Conformal Cost Control Algorithm through the numerical experiments on four challenging control benchmarks: Vanderpol (Henrion & Korda, 2013), Pendulum (Towers et al., 2024), MountainCar (Towers et al., 2024), and Lorenz (Lorenz, 1996). Our primary objective is to demonstrate that our method provides superior runtime assurance in terms of both violation frequency and cumulative cost when compared to representative adaptive calibration baselines. In Appendix D.4, we present a **sensitivity analysis** of the proposed algorithm with respect to γ, β, and the sliding-window size.

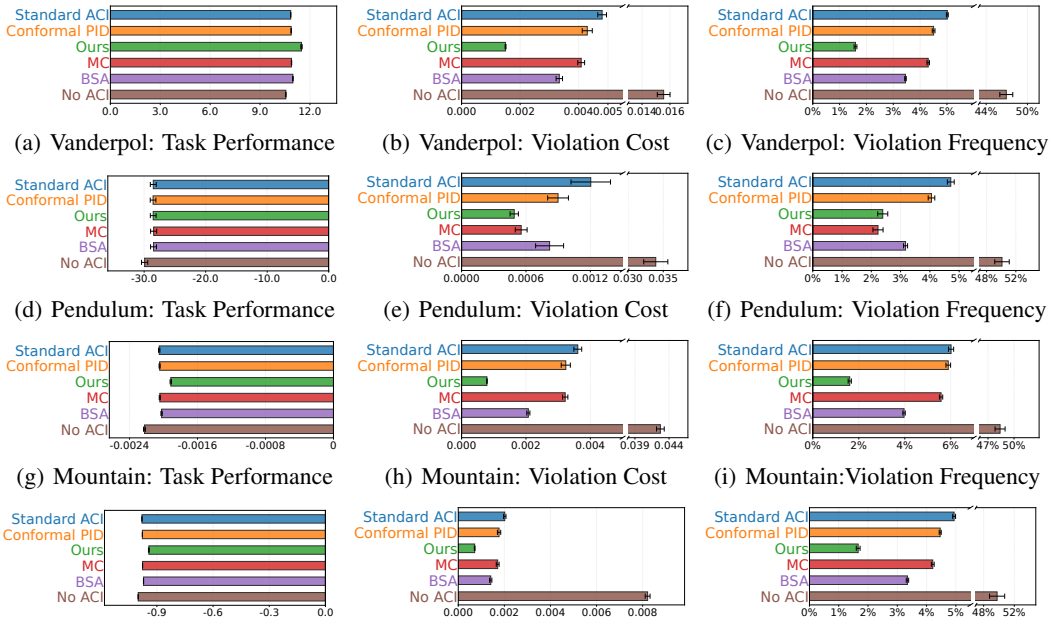
Comparison Methods: We evaluate the proposed Cost-Aware ACI in Alg. 1 against several baselines by replacing the Cost-Aware ACI component in the closed-loop control framework with the following alternatives. 1. ***Standard ACI***: This corresponds to the conventional ACI approach, equivalently obtained by setting $\beta = 0$ in our method. 2. ***Conformal PID***: The ACI update is replaced by the conformal PID method of Angelopoulos et al. (2023). 3. ***Rolling RC (Feldman et al., 2024)***: We consider two representative loss choices within its generalized risk-control framework. ***MC***(Miscoverage Counter) uses the miscoverage-counter loss employed in the experiments of Feldman et al. (2024), which tracks consecutive miscoverage events. ***BSA*** (Binary Safety-Aware loss) uses $L_k^{\mathrm{BSA}} = e_k + \mathbb{I}(h(x_k) < 0)$ as the loss to additionally incorporate binary safety information without distinguishing violation severity. 4. ***No ACI***: The adaptive calibration module is removed entirely by fixing the threshold $\hat{Q}_k \equiv 0$.

To create a rigorous evaluation, the task performance objective J_{task} in each benchmark is intentionally designed to be antagonistic to constraint satisfaction, meaning that achieving better task performance incentivizes the system to leave the designated admissible set. This creates a realistic and challenging trade-off between performance and runtime assurance, stress-testing each algorithm's ability to balance competing objectives. Each simulation is run for $10,000$ time steps. Further implementation details are provided in Appendix C. Fig. 1 compares the six methods across 10 random seeds with $\alpha = 0.1$, whereas Table 1 reports the results on MountainCar under different choices of α. Our analysis focuses on two key comparisons:

The Benefit of ACI: Comparing all ACI-based methods to the baseline without ACI shows the value of adaptive assurance constraints. The "No ACI" baseline suffers from catastrophic failure rates across all benchmarks. In contrast, all ACI methods successfully regulate this risk. This confirms that Alg. 1 provides a robust foundation for enhancing runtime assurance.

Superiority of Cost-Aware ACI: The central claim of our work is that Cost-Aware ACI provides superior runtime assurance. The results in Fig. 1 and Table 1 (in Appendix D.3) strongly support this. Across nearly all benchmarks and all values of α, our method consistently and substantially outperforms the competing baselines on both metrics (J and V). This performance gap is a direct result of our cost-sensitive update rule, which penalizes severe violations more aggressively, compelling the controller to adopt a more cautious policy. The comparisons with MC and BSA further show that, although the framework in Feldman et al. (2024) accommodates different user-specified loss functions, the choice of calibrated loss can substantially affect closed-loop safety behavior. By explicitly encoding violation severity in the calibration loss, our method achieves lower J and V than these representative loss designs in nearly all settings. Furthermore, from Table 1, as predicted by our theory, this advantage becomes more pronounced for larger α, where the algorithm has a larger "risk budget" that the standard ACI is more likely to misuse on severe violations.

The improved runtime assurance comes at a slight cost to task performance ($\bar{J}_{task}$). This trade-off is expected and even desirable, as our benchmarks were designed to make constraint satisfaction and performance compete. The fact that our method achieves substantial reductions in violation frequency and cost for only a minor performance loss highlights its efficiency.



(a) Vanderpol: Task Performance (b) Vanderpol: Violation Cost (c) Vanderpol: Violation Frequency

(d) Pendulum: Task Performance (e) Pendulum: Violation Cost (f) Pendulum: Violation Frequency

(g) Mountain: Task Performance (h) Mountain: Violation Cost (i) Mountain:Violation Frequency

(j) Lorenz: Task Performance (k) Lorenz: Violation Cost (l) Lorenz: Violation Frequency

Figure 1: Comparison of experimental results (**lower is better for all three metrics**)

Fig. 2 (in Appendix D.3) shows excerpts of system trajectories for Vanderpol under standard ACI and our method with $\alpha = 0.5$. When the system dynamics vary over time, both methods initially exit the admissible set. Our method accounts for violation severity and quickly recovers after a single violation, whereas standard ACI ignores severity, leading to slower recovery and ten violations.

6 Conclusion and Limitations

In this work, we introduced Cost-Aware ACI, a framework that overcomes the severity-blindness of standard ACI by integrating violation cost directly into a boosted loss function, and proposed a closed-loop method for online risk-aware controller synthesis in unknown and dynamic environments. Our theoretical analysis proved that this method provides simultaneous, non-asymptotic upper bounds on both long-run constraint violation frequency and average cumulative violation cost. Experiments show that our method substantially reduces both violation frequency and cost compared with standard ACI, with only a modest trade-off in task performance. This work provides a principled, verifiable way to manage not only the rate of failures but also severity, moving toward more meaningful runtime assurance for learning-based control. However, our method still has several limitations. It does not support complex temporal logic specifications, and similar to standard ACI, may lead to less smooth system behavior. Addressing these issues remains part of future work.

AI USE STATEMENT

In this work, generative AI tools were used solely to assist with language polishing and improving the clarity of written expressions. No generative AI tools were used for developing the research ideas, deriving the theoretical results, or conducting the experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All assumptions required for the theoretical results are explicitly stated in the main text, and complete proofs of the corresponding theoretical results are provided in the appendix. To facilitate reproducibility of the experimental results, the implementation code and brief instructions are publicly available in an anonymous repository https://github.com/tempt555anonymous/CostAwareACI.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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, 2016.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. Ieee, 2019.

Joel Andersson, Joris Gillis, Greg Horn, Jim Rawlings, and Moritz Diehl. Casadi—a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2018.

Anastasios Angelopoulos, Emmanuel Candes, and Ryan J Tibshirani. Conformal pid control for time series prediction. *Advances in neural information processing systems*, 36:23047–23074, 2023.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Charalampos P Bechlioulis and George A Rovithakis. Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *IEEE transactions on Automatic Control*, 53(9):2090–2099, 2008.

Francesca Cairoli, Nicola Paoletti, Luca Bortolussi, et al. Conformal quantitative predictive monitoring of stl requirements for stochastic processes. In *HSCC'23: Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, volume 1, pp. 1–11. ACM, 2023.

Sarah Dean, Andrew Taylor, Ryan Cosner, Benjamin Recht, and Aaron Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning*, pp. 654–670. PMLR, 2021.

Anushri Dixit, Lars Lindemann, Skylar X Wei, Matthew Cleaveland, George J Pappas, and Joel W Burdick. Adaptive conformal prediction for motion planning among dynamic agents. In *Learning for Dynamics and Control Conference*, pp. 300–314. PMLR, 2023.

Samira S Farahani, Rupak Majumdar, Vinayak S Prabhu, and Sadegh Soudjani. Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances. *IEEE Transactions on Automatic Control*, 64(8):3324–3331, 2018.

Shai Feldman, Liran Ringel, Stephen Bates, and Yaniv Romano. Achieving risk control in online learning settings. *Transactions on Machine Learning Research*, 2024.

Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

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.

Didier Henrion and Milan Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312, 2013.

Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

Lars Lindemann and Dimos V Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE control systems letters*, 3(1):96–101, 2018.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8): 5116–5123, 2023.

Lars Lindemann, Alexander Robey, Lejun Jiang, Satyajeet Das, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *IEEE Open Journal of Control Systems*, 3:158–172, 2024.

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.

Lennart Ljung and Torsten Söderström. *Theory and practice of recursive identification*. 1983.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pp. 1–18. Reading, 1996.

Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & chemical engineering*, 23(4-5):667–682, 1999.

Aleksandr Podkopaev, Darren Xu, and Kuang-chih Lee. Adaptive conformal inference by betting. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 40886–40907, 2024.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724. Ieee, 2020.

Pierre OM Scokaert, David Q Mayne, and James B Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 2002.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Shili Sheng, Pian Yu, David Parker, Marta Kwiatkowska, and Lu Feng. Safe pomdp online planning among dynamic agents via adaptive conformal prediction. *IEEE Robotics and Automation Letters*, 2024.

Martim Sousa, Ana Maria Tomé, and José Moreira. A general framework for multi-step ahead adaptive conformal heteroscedastic time series forecasting. *Neurocomputing*, 608:128434, 2024.

Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio A Vela. Synthesis of control barrier functions using a supervised machine learning approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7139–7145. Ieee, 2020.

Kegan J Strawn, Nora Ayanian, and Lars Lindemann. Conformal predictive safety filter for rl controllers in dynamic environments. *IEEE Robotics and Automation Letters*, 8(11):7833–7840, 2023.

Johan Hallberg Szabadváry. Adaptive conformal inference for multi-step ahead time-series forecasting online. *arXiv preprint arXiv:2409.14792*, 2024.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for dynamics and control*, pp. 708–717. PMLR, 2020.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1): 25–57, 2006.

Zitong Yang, Emmanuel Candès, and Lihua Lei. Bellman conformal inference: Calibrating prediction intervals for time series. *arXiv preprint arXiv:2402.05203*, 2024.

Jianpeng Yao, Xiaopan Zhang, Yu Xia, Zejin Wang, Amit K Roy-Chowdhury, and Jiachen Li. Sonic: Safe social navigation with adaptive conformal inference and constrained reinforcement learning. *arXiv preprint arXiv:2407.17460*, 2024.

Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pp. 25834–25866. PMLR, 2022.

Hao Zhou, Yanze Zhang, and Wenhao Luo. Safety-critical control with uncertainty quantification using adaptive conformal prediction. In *2024 American Control Conference (ACC)*, pp. 574–580. IEEE, 2024.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

APPENDIX

A PROOF

A.1 PROOF OF LEMMA 1

Lemma 1 (Parameter Boundedness). *Let $\{\delta_k\}$ be generated by the update rule (7). Under Assumptions 2 and 3, if δ_1 is initialized within the interval $[-\gamma(L_{\max}-\alpha), 1+\gamma\alpha]$, then δ_k remains within this interval for all $k \in \mathbb{N}$.*

Proof. **Lower bound:** We prove the lower bound by contradiction. Assume there is a first time k_0 such that $\delta_{k_0} < -\gamma(L_{\max}-\alpha)$. This implies $\delta_{k_0-1} \geq -\gamma(L_{\max}-\alpha)$ and that the loss L_{k_0-1} must have been large enough to cause the drop. Specifically, $L_{k_0-1} > \alpha$.

1. If $\delta_{k_0-1} < 0$, then by definition $\hat{Q}_{k_0-1}(\delta_{k_0-1}) = M$. By Assumption 2, $s_{k_0-1} \leq M = \hat{Q}_{k_0-1}$, leading to $e_{k_0-1} = 0$ and $L_{k_0-1} = 0$. This contradicts $L_{k_0-1} > \alpha$.
2. If $\delta_{k_0-1} \geq 0$, the minimum possible δ_{k_0} is $0+\gamma(\alpha-L_{\max}) = -\gamma(L_{\max}-\alpha)$, establishing the contradiction.

Upper bound: The argument follows symmetrically. Assume there is a first time k_0 such that $\delta_{k_0} > 1+\gamma\alpha$. Then $\delta_{k_0-1} \leq 1+\gamma\alpha$. If $\delta_{k_0-1} > 1$, then $\hat{Q}_{k_0-1} = -\epsilon$. Since $s_{k_0-1} \geq 0 > -\epsilon$, we have $e_{k_0-1} = 1$ and hence $L_{k_0-1} \geq 1 > \alpha$. Therefore,

$$\delta_{k_0} = \delta_{k_0-1} + \gamma(\alpha - L_{k_0-1}) < \delta_{k_0-1} \leq 1+\gamma\alpha,$$

which is a contradiction. If $\delta_{k_0-1} \leq 1$, then, since $L_{k_0-1} \geq 0$,

$$\delta_{k_0} = \delta_{k_0-1} + \gamma(\alpha - L_{k_0-1}) \leq 1+\gamma\alpha,$$

again yielding a contradiction. □

A.2 PROOF OF LEMMA 2

Lemma 2 (Long-Run Average Boosted Loss Convergence). *Under the same conditions as Lemma 1, the long-run time-average of the boosted loss converges to the target level α, i.e., $\lim_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} L_k = \alpha$.*

Proof. Expanding the recursion, we obtain $\delta_{T+1} = \delta_1 + \gamma\sum_{k=1}^{T}(\alpha - L_k)$. Rearranging and dividing by $T\gamma$, we obtain $\frac{1}{T}\sum_{k=1}^{T} L_k - \alpha = \frac{\delta_1 - \delta_{T+1}}{T\gamma}$. From Lemma 1, both δ_1 and δ_{T+1} are bounded. Let $B = 1+\gamma\alpha+\gamma(L_{\max}-\alpha)$ be the interval width. Then, we have

$$\left|\frac{1}{T}\sum_{k=1}^{T} L_k - \alpha\right| \leq \frac{B}{T\gamma} \to 0 \quad \text{as } T \to \infty.$$

The proof is completed. □

A.3 PROOF OF THEOREM 1

Theorem 1 (Upper Bound on Constraint Violation Frequency). *Let $v_k := \mathbb{I}(h(\boldsymbol{x}_k) < 0)$ be the indicator of a constraint violation at time k. Under Assumptions 2, 3 and 4, the violation frequency V is upper-bounded by α:*

$$V = \limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} v_k \leq \alpha.$$

Proof. A constraint violation at time k ($v_k = 1$, so $h(\boldsymbol{x}_k) < 0$) implies a miscoverage event ($e_k = 1$). This is because the controller enforces $\widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) \geq \hat{Q}_k \geq 0$. Thus, a violation implies:

$$s_k = |h(\boldsymbol{x}_k) - \widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})| = \widehat{h}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) - h(\boldsymbol{x}_k) > \hat{Q}_k - 0 = \hat{Q}_k.$$

This means $v_k = 1 \implies e_k = 1$, so $v_k \leq e_k$. Furthermore, by definition, $L_k = e_k(1 + \beta c(\boldsymbol{x}_k)) \geq e_k$. Combining these inequalities, we have $\frac{1}{T}\sum_{k=1}^{T} v_k \leq \frac{1}{T}\sum_{k=1}^{T} e_k \leq \frac{1}{T}\sum_{k=1}^{T} L_k$. Taking the limit superior as $T \to \infty$ and applying Lemma 2, we obtain $\limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} v_k \leq \limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} L_k = \alpha$. The proof is completed. □

A.4 Proof of Theorem 2

Theorem 2 (Upper Bounds on Average Cumulative Cost J)**.** *Under Assumptions 2, 3 and 4, for any risk sensitivity parameter $\beta > 0$, the long-run average cumulative constraint violation cost J satisfies:*

$$J = \limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} c(\boldsymbol{x}_k) \leq \frac{\alpha}{\beta}.$$

Proof. From Lemma 2 and the definition of L_k, we have $\alpha = \lim_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} L_k = \lim_{T\to\infty} \left(\frac{1}{T}\sum_{k=1}^{T} e_k + \beta \frac{1}{T}\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k) \right)$. Since both terms are nonnegative and $c(\boldsymbol{x}_k) = 0$ when $e_k = 0$, we have $\beta J \leq \beta \limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k) \leq \alpha$. Dividing both sides by β, we obtain

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k) \leq \frac{\alpha}{\beta}.$$

Since $c(\boldsymbol{x}_k) = 0$ whenever $e_k = 0$, it follows that

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} c(\boldsymbol{x}_k) \leq \frac{\alpha}{\beta}.$$

This completes the proof. □

A.5 Proof of Corollary 1

Corollary 1 (Finite-Time Violation Frequency and Cost Bounds)**.** *Let $V_T := \frac{1}{T}\sum_{k=1}^{T} v_k$ be the violation frequency and $J_T := \frac{1}{T}\sum_{k=1}^{T} c(\boldsymbol{x}_k)$ be the average cumulative cost over the first T time steps. The following bounds hold:*

1. ***Violation Frequency:*** $V_T \leq \alpha + \frac{\delta_1 - \delta_{T+1}}{T\gamma} = \alpha + O(1/T)$.
2. ***Average Cost:*** $J_T \leq \frac{\alpha}{\beta} + \frac{\delta_1 - \delta_{T+1}}{\beta T\gamma} - \frac{1}{\beta T}\sum_{k=1}^{T} e_k \leq \frac{\alpha}{\beta} + O(1/T)$.

Proof. The proof follows directly from the relationships $v_k \leq e_k \leq L_k$ and $c(\boldsymbol{x}_k) \leq e_k c(\boldsymbol{x}_k)$. Summing the core equality $\frac{1}{T}\sum L_k = \alpha + \frac{\delta_1 - \delta_{T+1}}{T\gamma}$ and rearranging terms yield the desired bounds. The $O(1/T)$ rates are a consequence of the boundedness of δ_k established in Lemma 1. □

A.6 Proof of Proposition 2

Proposition 2 (Conservation of Risk Budget)**.** *The algorithm dynamically trades off miscoverage frequency against violation severity. Let $\hat{\alpha}_T := \frac{1}{T}\sum_{k=1}^{T} e_k$ be the empirical miscoverage rate and $\bar{c}_T := \frac{\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k)}{\sum_{k=1}^{T} e_k}$ be the average cost conditional on a miscoverage event. In the long run, these quantities adhere to the conservation law: $\lim_{T\to\infty} \hat{\alpha}_T(1 + \beta\bar{c}_T) = \alpha$.*

Proof. Decomposing the sum of the boosted loss gives $\sum_{k=1}^{T} L_k = \sum_{k=1}^{T} e_k + \beta \sum_{k=1}^{T} e_k c(\boldsymbol{x}_k)$. We can factor out the term $\sum e_k$:

$$\sum_{k=1}^{T} L_k = \left(\sum_{k=1}^{T} e_k\right)\left(1 + \beta \frac{\sum_{k=1}^{T} e_k c(\boldsymbol{x}_k)}{\sum_{k=1}^{T} e_k}\right) = T\hat{\alpha}_T(1 + \beta \bar{c}_T).$$

Dividing by T and applying Lemma 2 yield the result. □

B Online Optimization Interpretation

We further provide an online optimization interpretation of the proposed adaptive update. In particular, the update rule can be viewed as an instance of online gradient descent on a sequence of linearized loss functions, which yields the following regret characterization.

Proposition 3 (Regret Minimization Perspective)**.** *The update rule in (7) is an instance of Online Gradient Descent (OGD) on the sequence of linearized loss functions $\mathcal{L}_k(\delta) = (L_k - \alpha)\delta$. Consequently, the regret against the best fixed parameter $\delta^\star$ in hindsight over the feasible interval satisfies:*

$$\textit{Regret}_T = \sum_{k=1}^{T} \mathcal{L}_k(\delta_k) - \min_{\delta^\star \in [-\gamma(L_{\max}-\alpha),\, 1+\gamma\alpha]} \sum_{k=1}^{T} \mathcal{L}_k(\delta^\star) \le O\left(\frac{1}{\gamma} + \gamma T\right)$$

Proof. For each round k, define the linearized online loss

$$\mathcal{L}_k(\delta) = (L_k - \alpha)\delta,$$

where L_k is the realized boosted loss at round k and is treated as fixed when defining $\mathcal{L}_k$. The loss $\mathcal{L}_k$ is linear and hence convex, with gradient

$$g_k := \nabla \mathcal{L}_k(\delta) = L_k - \alpha.$$

By Assumption 3,

$$0 \le L_k \le L_{\max},$$

and therefore

$$|g_k| = |L_k - \alpha| \le L_{\max} =: G.$$

Furthermore, Lemma 1 guarantees that all iterates remain in the invariant set

$$\delta_k \in \mathcal{K}, \qquad \mathcal{K} = [-\gamma(L_{\max} - \alpha),\, 1 + \gamma\alpha].$$

Thus, although the update is written without an explicit projection, the iterates remain in $\mathcal{K}$ and no additional projection is required. Let

$$\delta^\star \in \arg\min_{\delta \in \mathcal{K}} \sum_{k=1}^{T} \mathcal{L}_k(\delta)$$

be a best fixed decision in hindsight. The diameter of the feasible set is

$$D := \operatorname{diam}(\mathcal{K}) = 1 + \gamma L_{\max},$$

which implies

$$|\delta_k - \delta^\star| \le D, \qquad \forall k.$$

The update rule can be written as

$$\delta_{k+1} = \delta_k - \gamma g_k.$$

Thus,

$$\begin{aligned}(\delta_{k+1} - \delta^\star)^2 &= (\delta_k - \delta^\star - \gamma g_k)^2 \\ &= (\delta_k - \delta^\star)^2 - 2\gamma g_k(\delta_k - \delta^\star) + \gamma^2 g_k^2.\end{aligned}$$

Rearranging gives

$$g_k(\delta_k - \delta^\star) = \frac{(\delta_k - \delta^\star)^2 - (\delta_{k+1} - \delta^\star)^2}{2\gamma} + \frac{\gamma}{2} g_k^2.$$

Since the loss is linear,

$$\mathcal{L}_k(\delta_k) - \mathcal{L}_k(\delta^\star) = g_k(\delta_k - \delta^\star).$$

Therefore,

$$\begin{aligned}\mathrm{Regret}_T &= \sum_{k=1}^{T} \left(\mathcal{L}_k(\delta_k) - \mathcal{L}_k(\delta^\star)\right) \\ &= \frac{(\delta_1 - \delta^\star)^2 - (\delta_{T+1} - \delta^\star)^2}{2\gamma} + \frac{\gamma}{2}\sum_{k=1}^{T} g_k^2.\end{aligned}$$

Since

$$(\delta_{T+1} - \delta^\star)^2 \geq 0,$$

we obtain

$$\mathrm{Regret}_T \leq \frac{(\delta_1 - \delta^\star)^2}{2\gamma} + \frac{\gamma}{2}\sum_{k=1}^{T} g_k^2.$$

Using the boundedness of the decision set and gradients,

$$(\delta_1 - \delta^\star)^2 \leq D^2, \qquad g_k^2 \leq G^2,$$

we have

$$\mathrm{Regret}_T \leq \frac{D^2}{2\gamma} + \frac{\gamma G^2}{2} T.$$

Substituting

$$D = 1 + \gamma L_{\max}, \qquad G = L_{\max},$$

yields

$$\begin{aligned}\mathrm{Regret}_T &\leq \frac{(1 + \gamma L_{\max})^2}{2\gamma} + \frac{\gamma L_{\max}^2}{2} T \\ &= O\left(\frac{1}{\gamma} + \gamma T\right).\end{aligned}$$

In particular, a fixed step size $\gamma > 0$ independent of T gives $\mathrm{Regret}_T = O(T)$, whereas choosing $\gamma = \Theta(T^{-1/2})$ yields the standard $O(\sqrt{T})$ regret bound. □

This regret-minimization perspective provides a principled theoretical foundation for the proposed adaptive mechanism. By interpreting the update rule through the lens of online optimization, it shows that the adaptation is not merely heuristic, but follows a well-established optimization principle with an explicit regret characterization. In particular, the regret bound provides a clear interpretation of the role of the step size γ: a larger γ enables more responsive adaptation to newly observed losses, whereas a smaller γ yields more gradual and conservative updates. This trade-off is especially relevant in non-stationary deployment environments, where maintaining responsiveness to changing conditions is essential. Thus, the proposed update is grounded in a well-understood online optimization framework, while the regret analysis offers additional theoretical insight and practical guidance for selecting the adaptation rate.

C Practical Implementation Strategies

In this subsection, we provide further intuition and implementation details for Algorithm 1. We begin by outlining its main workflow. Algorithm 1 operates in a continuous loop:

1. **Feedback and Adaptation** ($k > 0$): Upon observing state $\boldsymbol{x}_k$, the algorithm computes the nonconformity score s_k and the cost-aware loss L_k. This loss is used to update the adaptive parameter to δ_{k+1}.

2. **Calibration**: Using the updated history of scores $\mathcal{C}$ and the new parameter δ_{k+1}, the algorithm re-computes the assurance threshold $\hat{Q}_{k+1}$.

3. **Admissible Control Synthesis**: The algorithm solves the optimization problem (8) using the new threshold $\hat{Q}_{k+1}$ to determine the optimal, admissible control action $\boldsymbol{u}_k^*$. This action is then applied to the system.

Next, we discuss several practical implementation strategies. To ensure real-time feasibility in high-frequency control loops, two modifications can be applied to Alg. 1:

Sparse Model Updates: The surrogate models $\widehat{h}$ and $\widehat{\boldsymbol{f}}$ need not be updated at every time step k. In practice, they are often updated via *batch learning* (e.g., every N steps) or *event-triggered learning* (e.g., only when prediction error exceeds a threshold) to reduce computational overhead without significantly degrading prediction accuracy.

Sliding Window Calibration: To ensure adaptivity and computational tractability, the calibration set $\mathcal{C}$ is implemented as a *sliding window* of fixed size W. This method bounds the complexity of the quantile computation and, more importantly, enhances responsiveness to non-stationarity. As the surrogate model $\widehat{h}$ is updated online, the window naturally discards "stale" residuals from prior model versions. This ensures the uncertainty threshold $\hat{Q}_k$ adapts to local distribution shifts and always reflects the performance of the *current* model.

D Additional Experimental Results and Implementation Details

The implementation code and brief instructions are publicly available in an anonymous repository https://github.com/tempt555anonymous/CostAwareACI.

All computations are performed on a machine equipped with an Intel i7-13700H 2.10 GHz CPU and 32 GB of RAM. The MPC optimization problem is solved using CasADi Andersson et al. (2018) in conjunction with the Ipopt solver Wächter & Biegler (2006). The 10 random seeds used in Fig. 1 are $0, 1, \ldots, 9$. For all other experiments, the random seed is fixed to 0.

D.1 Benchmark and Experimental Setup

Vanderpol. Consider the Vanderpol oscillator adapted from Henrion & Korda (2013),

$$\begin{cases} x_{k+1} = x_k + 0.01(-2y_k + u_k + p_k), \\ y_{k+1} = y_k + 0.01(0.8x_k - 10(x_k^2 - 0.21)y_k + v_k + q_k), \end{cases}$$

where the state space $\mathcal{X} = \{(x, y) \mid 2 - x^2 - y^2 \geq 0\}$, the admissible set $\mathcal{S} = \{(x, y) \mid 1 - x^2 - y^2 \geq 0\}$, and the control set $\mathcal{U} = \{(u, v) \mid -40 \leq u, v \leq 40\}$. The parameter variation p_k and q_k are modeled as piecewise-constant, time-varying signals. Every 10 time steps, each parameter variation is independently resampled from $14 \times \text{Beta}(0.1, 0.1) - 7$ and kept constant until the next update. This setting captures slowly varying dynamics in the system model. In this benchmark, the MPC task objective J_{task} encourages the system state to remain as close as possible to $(0.8, 0.8)$, which directly conflicts with the constraints.

Pendulum. Consider the *Pendulum* benchmark adapted from GYM Towers et al. (2024),

$$\begin{cases} \dot{\theta}_{k+1} = \dot{\theta}_k + 0.05(15 \sin(\theta_k) + 3u_k + r_k + q_k), \\ \theta_{k+1} = \theta_k + 0.05\dot{\theta}_{k+1}, \end{cases}$$

where the state space $\mathcal{X} = \{(\dot{\theta}, \theta) \mid 2 - (\frac{\dot{\theta}}{6})^4 - (\frac{\theta}{1.5\pi})^4 \geq 0\}$, the admissible set $\mathcal{S} = \{(\dot{\theta}, \theta) \mid 1 - (\frac{\dot{\theta}}{6})^4 - (\frac{\theta}{1.5\pi})^4 \geq 0\}$, and the control set $\mathcal{U} = \{u \mid -80 \leq u \leq 80\}$. The terms r_k and q_k jointly model time-varying effects in the system dynamics. Specifically, r_k represents a piecewise-constant parameter variation: every 50 time steps, r_k is independently resampled from a uniform distribution over $[-5, 5]$ and kept constant until the next update. In addition, q_k captures a drifting bias, which increases by 0.01 every 50 time steps, with $q_0 = 0$. The MPC task objective J_{task} encourages the magnitude of the state $\dot{\theta}_k$ to be as large as possible.

MountainCar. Consider the *MountainCarContinuous* benchmark adapted from GYM Towers et al. (2024),

$$\begin{cases} v_{k+1} = v_k + (0.0015 + m_k)u_k - 0.0025\cos(3p_k) + r_k, \\ p_{k+1} = p_k + v_{k+1} + q_k, \end{cases}$$

where the state space $\mathcal{X} = \{(v, p) \mid 2 - (\frac{v}{0.05})^4 - (\frac{p+0.3}{0.7})^4 \geq 0\}$, the admissible set $\mathcal{S} = \{(v, p) \mid 1 - (\frac{v}{0.05})^4 - (\frac{p+0.3}{0.7})^4 \geq 0\}$, and the control set $\mathcal{U} = \{u \mid -200 \leq u \leq 200\}$. Here, m_k gradually decreases over time. Specifically, every five time steps, m_k decreases by 5×10^{-8}. In addition, every five time steps, the parameter variation r_k and q_k are independently resampled from the distribution $0.004 \times \text{Beta}(0.1, 0.1) - 0.002$ and then held constant until the next update. The MPC task objective J_{task} encourages the magnitude of the state v_k to be as large as possible.

Lorenz. Consider a 6-dimensional Lorenz model adapted from Lorenz (1996),

$$\begin{cases} \boldsymbol{x}_{k+1}[0] = \boldsymbol{x}_k[0] + 0.01((\boldsymbol{x}_k[1] - \boldsymbol{x}_k[4])\boldsymbol{x}_k[5] - \boldsymbol{x}_k[0] + \boldsymbol{u}_k[0] + p_k), \\ \boldsymbol{x}_{k+1}[1] = \boldsymbol{x}_k[1] + 0.01((\boldsymbol{x}_k[2] - \boldsymbol{x}_k[5])\boldsymbol{x}_k[0] - \boldsymbol{x}_k[1] + \boldsymbol{u}_k[1]), \\ \boldsymbol{x}_{k+1}[2] = \boldsymbol{x}_k[2] + 0.01((\boldsymbol{x}_k[3] - \boldsymbol{x}_k[0])\boldsymbol{x}_k[1] - \boldsymbol{x}_k[2] + \boldsymbol{u}_k[2]), \\ \boldsymbol{x}_{k+1}[3] = \boldsymbol{x}_k[3] + 0.01((\boldsymbol{x}_k[4] - \boldsymbol{x}_k[1])\boldsymbol{x}_k[2] - \boldsymbol{x}_k[3] + \boldsymbol{u}_k[3]), \\ \boldsymbol{x}_{k+1}[4] = \boldsymbol{x}_k[4] + 0.01((\boldsymbol{x}_k[5] - \boldsymbol{x}_k[2])\boldsymbol{x}_k[3] - \boldsymbol{x}_k[4] + \boldsymbol{u}_k[4]), \\ \boldsymbol{x}_{k+1}[5] = \boldsymbol{x}_k[5] + 0.01((\boldsymbol{x}_k[0] - \boldsymbol{x}_k[3])\boldsymbol{x}_k[4] - \boldsymbol{x}_k[5] + \boldsymbol{u}_k[5]), \end{cases}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are six-dimensional vectors, and $\boldsymbol{x}_k[i]$, for $i = 0, 1, \ldots, 5$, denotes the i-th component of the state $\boldsymbol{x}$ at time step k. The state space is $\mathcal{X} = \{\boldsymbol{x} \mid 12 - \sum_{i=0}^{5} \boldsymbol{x}^2[i] \geq 0\}$, the admissible set $\mathcal{S} = \{\boldsymbol{x} \mid 1 - \sum_{i=0}^{5} \boldsymbol{x}^2[i] \geq 0\}$, and the control set $\mathcal{U} = \{\boldsymbol{u} \mid \bigwedge_{i=0}^{5} -100 \leq \boldsymbol{u}[i] \leq 100\}$. Specifically, every five time steps, the parameter variation p_k is independently resampled from a uniform distribution over $[-5, 5]$ and kept constant until the next update. The MPC task objective J_{task} encourages the value of the state $\boldsymbol{x}[0]$ to be as small as possible.

Experimental Setup: We employ a physics-informed offline learning procedure to initialize $\widehat{h}$ and $\widehat{\boldsymbol{f}}$. Specifically, we obtain the initial surrogate models by interpolation-based fitting over a compact dictionary of standard basis functions, such as polynomial and trigonometric features. These models serve as warm-start surrogates in Algorithm 1. To account for unmodeled time-varying effects, we further augment the nominal predictors with lightweight residual corrections learned from online one-step prediction errors Ljung & Söderström (1983). The residual components are updated only when the corresponding prediction error exceeds a prescribed threshold, following the sparse online update strategy described in Appendix C. For example, in the Vanderpol benchmark, $\hat{h}$ is updated when $|h - \widehat{h}| > 0.01$.

In the experiments reported in Figure 1 and Table 1, the ACI learning rate is set to $\gamma = 0.01$, and the quantile estimation uses a sliding window of the 200 most recent scores (see Appendix C). The sensitivity parameter β is set to 50 for Vanderpol and 100 for Pendulum, MountainCar, and Lorenz. Across all experiments, the violation cost is defined as $\rho(\boldsymbol{x}) = -\frac{h(\boldsymbol{x})}{M_h}$, where M_h normalizes the cost to satisfy Assumption 3. The horizon in the MPC optimization (8) is set to $N = 20$.

D.2 Details of the Rolling RC Baselines

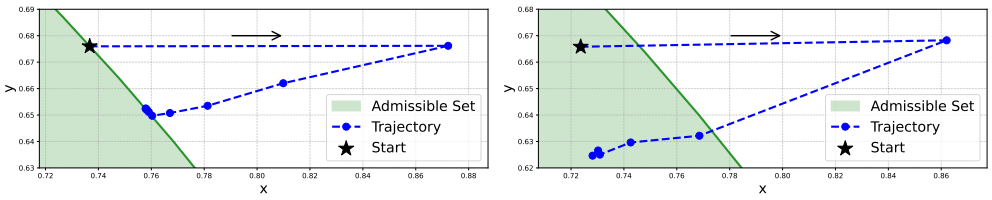
In this subsection, we provide additional details on the two loss functions used to instantiate the Rolling RC framework Feldman et al. (2024) in our experiments. To ensure a consistent comparison, we retain the same closed-loop control framework and replace only the loss used in the adaptive calibration module.

Following Feldman et al. (2024), Rolling RC updates a calibration parameter according to

$$\delta_{k+1} = \delta_k + \gamma(r - L_k). \tag{9}$$

where L_k denotes the user-specified loss and r is its target risk level. For both baselines, δ_1 is initialized as α, consistent with the other adaptive calibration methods considered in our experiments. The learning rate and sliding-window size are also kept unchanged, with $\gamma = 0.01$ and $W = 200$ in the main experiments.

Miscoverage Counter (MC). Our first Rolling RC baseline directly adopts the miscoverage-counter loss introduced and used in the experiments of Feldman et al. (2024). Recall that $e_k = \mathbb{I}(s_k > \hat{Q}_k(\delta_k))$ denotes the miscoverage indicator. The miscoverage counter records the number of



(a) Standard ACI (b) Ours

Figure 2: Comparison of selected trajectories on Vanderpol

consecutive miscoverage events up to time step k and is recursively defined as

$$\mathrm{MC}_0 = 0, \qquad \mathrm{MC}_k = \begin{cases} \mathrm{MC}_{k-1} + 1, & e_k = 1, \\ 0, & e_k = 0. \end{cases}$$

The corresponding calibration loss is $L_k^{\mathrm{MC}} = \mathrm{MC}_k$. Therefore, an isolated miscoverage event incurs a loss of one, whereas consecutive miscoverage events lead to progressively larger losses. Following Feldman et al. (2024), when the nominal miscoverage level is α, we set the target MC risk to

$$r_{\mathrm{MC}} = \frac{\alpha}{1-\alpha}.$$

This choice corresponds to the expected MC value of an ideal predictor whose coverage events occur with probability $1-\alpha$. The theoretical analysis of Rolling RC assumes a bounded loss, whereas the raw MC in (D.2) is formally unbounded. As discussed in Feldman et al. (2024), a bounded alternative can be defined as

$$L_k^{\mathrm{MC},B} = \min\{\mathrm{MC}_k, B\},$$

for some sufficiently large $B > 0$. Following the experimental implementation of Feldman et al. (2024), however, all reported results use the original uncapped MC loss.

Binary Safety-Aware Loss (BSA). The second baseline considers a representative loss within the generalized framework of Feldman et al. (2024) that explicitly incorporates binary information about physical constraint violations. We define

$$L_k^{\mathrm{BSA}} = e_k + \mathbb{I}\big(h(x_k) < 0\big),$$

where $\mathbb{I}(h(x_k) < 0)$ indicates whether the realized state lies outside the admissible set. Thus, in addition to penalizing prediction miscoverage through e_k, BSA assigns an additional unit of loss whenever a physical constraint violation occurs. Unlike our Cost-Aware loss, however, this additional penalty is binary and therefore does not distinguish constraint violations according to their severity.

For BSA, we use the target risk level $r_{\mathrm{BSA}} = \alpha$, and update the adaptive parameter according to (9), i.e.,

$$\delta_{k+1} = \delta_k + \gamma\left(\alpha - L_k^{\mathrm{BSA}}\right).$$

For MC, the same update instead becomes

$$\delta_{k+1} = \delta_k + \gamma\left(\frac{\alpha}{1-\alpha} - L_k^{\mathrm{MC}}\right).$$

D.3 Additional Results

This section provides additional experimental results, including the quantitative comparison under different values of α in Table 1 and representative closed-loop trajectories on the Vanderpol benchmark in Fig. 2.

Table 1: Performance comparison under different α on the MountainCar benchmark

α	Standard ACI			Conformal PID			Ours		
	$\bar{J}_{\text{task}}$ ($\times 10^{-3}$)	J ($\times 10^{-2}$)	V	$\bar{J}_{\text{task}}$ ($\times 10^{-3}$)	J ($\times 10^{-2}$)	V	$\bar{J}_{\text{task}}$ ($\times 10^{-3}$)	J ($\times 10^{-2}$)	V
0.5	-2.17	2.081	23.37%	-2.17	2.003	22.77%	-2.02	0.410	4.85%
0.4	-2.15	1.675	18.94%	-2.15	1.625	18.30%	-1.98	0.327	4.21%
0.3	-2.12	1.260	14.67%	-2.12	1.209	13.91%	-1.92	0.244	3.29%
0.2	-2.10	0.815	10.45%	-2.11	0.749	9.81%	-1.83	0.162	2.41%
0.1	-2.06	0.376	5.98%	-2.05	0.341	5.90%	-1.91	0.079	1.53%
0.01	-1.94	0.023	0.89%	-1.94	0.012	0.71%	-1.88	0.007	0.30%

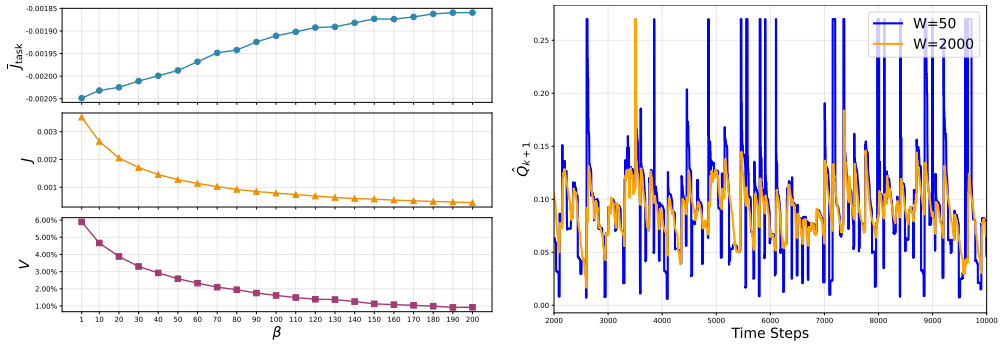
α	MC			BSA		
	$\bar{J}_{\text{task}}$ ($\times 10^{-3}$)	J ($\times 10^{-2}$)	V	$\bar{J}_{\text{task}}$ ($\times 10^{-3}$)	J ($\times 10^{-2}$)	V
0.5	-2.14	1.625	18.32%	-2.13	1.426	16.11%
0.4	-2.12	1.303	15.10%	-2.11	1.142	13.17%
0.3	-2.11	1.002	12.07%	-2.09	0.829	10.24%
0.2	-2.08	0.677	9.03%	-2.07	0.521	7.28%
0.1	-2.04	0.328	5.54%	-2.03	0.207	4.03%
0.01	-1.95	0.022	0.95%	-1.90	0.015	0.48%

D.4 Sensitivity Analysis

In this section, we further analyze the algorithm's sensitivity to its key hyperparameters, γ, β, and the sliding window size W, to verify the robustness of the performance gains and to confirm that these parameters behave as theoretically expected. We analyze each parameter in a representative setting where its influence is most evident for clarity: γ and β on MountainCar with $\alpha = 0.1$, and the sliding window size on Pendulum with $\alpha = 0.3$.

Table 2: Performance comparison under different values of γ

γ	Standard ACI			Ours		
	$\bar{J}_{\text{task}}$	J	V	$\bar{J}_{\text{task}}$	J	V
0.2	-2.03×10^{-3}	5.00×10^{-3}	4.92%	-1.84×10^{-3}	0.83×10^{-3}	0.84%
0.1	-2.04×10^{-3}	4.58×10^{-3}	5.45%	-1.85×10^{-3}	0.83×10^{-3}	1.05%
0.01	-2.06×10^{-3}	3.76×10^{-3}	5.98%	-1.91×10^{-3}	0.79×10^{-3}	1.53%
0.001	-2.05×10^{-3}	3.53×10^{-3}	6.11%	-1.98×10^{-3}	0.79×10^{-3}	2.38%



(a) Performance Variation Under Different Values of β

(b) Comparison of $\hat{Q}_{k+1}$ Under Different Sliding Window Size W

Figure 3: Effect of Sensitivity β and W

Effect of Learning Rate γ: Table 2 shows that our method's superiority over standard ACI is robust across different learning rates. For all tested values of γ, our approach maintains a significant advantage in reducing both cost J and frequency V. The choice of γ itself involves a trade-off related to the system's temporal dynamics, and as noted in Gibbs & Candès (2024), it can also be adapted online. We will investigate this in future work.

Effect of Sensitivity β: Fig. 3(a) shows that the parameter β acts as a tuning knob for runtime assurance. As β increases, the algorithm penalizes violations more heavily, making it more conservative. This directly improves assurance, causing both the average cost J and violation frequency V to decrease. In exchange, the task cost J_{task} slightly increases, which is expected since our benchmarks are designed to oppose assurance. This confirms that β provides a clear and predictable way to trade task performance for a higher level of assurance.

Effect of the sliding window size: The sliding window size, W, is a trade-off between stability and responsiveness. As shown in Fig. 3(b), a small window ($W = 50$) causes a volatile quantile $\hat{Q}_{k+1}$ that closely tracks recent events, while a large window ($W = 2000$) provides a much smoother, more stable estimate. Despite this difference in behavior, Table 3 shows that our method's superior assurance performance is robust. Across all tested window sizes, our approach consistently outperforms standard ACI by achieving lower violation costs (J) and violation frequencies (V). This is a key practical advantage, as it shows our algorithm's effectiveness does not depend on carefully tuning this parameter.

Table 3: Performance comparison under different sliding-window sizes

W	Standard ACI			Ours		
	$\bar{J}_{\text{task}}$	J	V	$\bar{J}_{\text{task}}$	J	V
50	-28.93	2.71×10^{-3}	14.54%	-28.82	1.09×10^{-3}	9.28%
500	-28.82	3.82×10^{-3}	14.89%	-28.85	1.30×10^{-3}	8.23%
2000	-28.86	3.67×10^{-3}	14.43%	-28.81	1.23×10^{-3}	8.11%