Safe Control using Learned Safety Filters and Adaptive Conformal Inference

Sacha Huriot H.SACHA@WUSTL.EDU
Ihab Tabbara I.K.TABBARA@WUSTL.EDU
Hussein Sibai SIBAI@WUSTL.EDU
Computer Science & Engineering, Washington University in St. Louis

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

Abstract

Safety filters have been shown to be effective tools to ensure the safety of control systems with unsafe nominal policies. To address scalability challenges in traditional synthesis methods, learning-based approaches have been proposed for designing safety filters for systems with high-dimensional state and control spaces. However, the inevitable errors in the decisions of these models raise concerns about their reliability and the safety guarantees they offer. This paper presents Adaptive Conformal Filtering (ACoFi), a method that combines learned Hamilton-Jacobi reachability-based safety filters with adaptive conformal inference. Under ACoFi, the filter dynamically adjusts its switching criteria based on the observed errors in its predictions of the safety of actions. The range of possible safety values of the nominal policy's output is used to quantify uncertainty in safety assessment. The filter switches from the nominal policy to the learned safe one when that range suggests it might be unsafe. We show that ACoFi guarantees that the rate of incorrectly quantifying uncertainty in the predicted safety of the nominal policy is asymptotically upper bounded by a user-defined parameter. This gives a soft safety guarantee rather than a hard safety guarantee. We evaluate ACoFi in a Dubins car simulation and a Safety Gymnasium environment, empirically demonstrating that it significantly outperforms the baseline method that uses a fixed switching threshold by achieving higher learned safety values and fewer safety violations, especially in out-of-distribution scenarios.

Keywords: Conformal prediction, safety filters, safe control

1. Introduction

Assuring safety is essential for deploying safety-critical control systems, such as self-driving cars (Chen et al., 2024) and surgical robots (Haidegger, 2019). Safety filters are prominent tools for ensuring their safety. Control barrier functions (CBFs) (Ames et al. (2019)) and Hamilton–Jacobi (HJ) reachability value functions (Bansal et al. (2017)) have been used to design safety filters that guarantee safe operation of control systems by adjusting their unsafe nominal actions to safe ones. However, traditional methods for synthesizing CBFs, such as sum-of-squares programming (Zhang et al. (2023); Clark (2021)), and for computing HJ reachability value functions, such as dynamic programming (Mitchell et al. (2005)), suffer from the curse-of-dimensionality. This motivated data-driven approaches for learning safety filters (Ganai et al. (2024); Tabbara and Sibai (2025); Li et al. (2025); So et al. (2024); Alan et al. (2023)).

In our work, without loss of generality, we focus on designing reliable safety filters relying on a learned HJ reachability value function V_θ. An instance of such filters evaluates the safety of the nominal control action at every state reached, and if it considers it unsafe, it switches to the learned safe policy that optimizes V_θ. Importantly, this backup is not assumed to be a perfect safe policy for

the true system, but rather the safest policy induced by the current learned approximation of the safety value function. Our goal is therefore not to construct a perfect safety filter, but to determine when to switch from task execution to using this safest policy available. Existing methods that rely on such safety filters use fixed thresholds for the value functions evaluating the safety of proposed actions to switch between the nominal and learned safe policies. However, when a HJ value function is learned from data, it is not guaranteed to be correct, and it is expected to be more erroneous in regions of the state space that are poorly represented during training, making fixed thresholds unreliable (Chen et al. (2018); Fisac et al. (2019); Lin et al. (2024); Tabbara et al. (2025a)).

In order to quantify the uncertainties of black-box predictors, conformal prediction (Gammerman et al. (1998); Vovk and Bendtsen (2018)) has emerged as a statistical framework for generating confidence regions called *conformal sets*. Given a desired miscoverage rate α, calibration data, and a predictor input x, the corresponding real output y will belong to the conformal set generated by conformal prediction with at least $1-\alpha$ probability. This method relies on the exchangeability assumption, i.e., that the joint distribution of the calibration data and the new test point is invariant under any permutation. However, the states and actions in trajectories are not exchangeable. Adaptive Conformal Inference (ACI) extends the application of conformal prediction to time-dependent data (Gibbs and Candes (2021)). In ACI settings, time series data, such as trajectories of dynamical systems, are considered. At each time step, the black-box predictor predicts the data point in the next time step and then the true data point is observed at that time step, i.e., ground-truth is observed in a delayed manner. ACI results in time-dependent conformal sets which guarantee that the average rate of miscoverage over time is bounded by a user-defined parameter α.

To address the failure of existing data-driven safety filters in accounting for their prediction errors, we propose Adaptive Conformal Filtering (ACoFi), a method that dynamically adjusts the criteria according to which these filters switch from the nominal policies to the learned safe ones. By monitoring the difference between the learned safety value at the current state and the updated one after receiving the observation at the next state, ACoFi adapts the threshold for switching from the nominal policy to the learned safe one corresponding to the HJ value function, providing probabilistic guarantees on the average rate of actions taken over time that are deemed unsafe by the learned safety value function, while minimizing unnecessary switching. We evaluate ACoFi in two vision-based navigation tasks. We show that ACoFi outperforms fixed threshold-based switching baselines by achieving higher safety values and performing fewer unsafe actions without excessive switching to the learned safe policy.

Our contributions are: (1) we introduce ACoFi, a method that uses ACI to account for the prediction errors of learned safety filters and provides formal guarantees, and (2) we empirically demonstrate ACoFi's effectiveness in high-dimensional control settings.

1.1. Related work

Safe control under uncertainty has been widely explored, particularly for high dimensional systems prone to operating in out-of-distribution (OOD) conditions ((Seo et al., 2025; Singletary et al., 2022; Sadigh and Kapoor, 2016; Wang and Wen, 2025; Daş and Burdick, 2025; Tabbara et al., 2025b; Hu et al., 2025; Michaux et al., 2025)). In (Huriot and Sibai (2025)), we used the theory of conformal decision policies (CDPs) (Lekeufack et al. (2024)) to account for the uncertainty in the trajectory predictions of other agents in multi-agent environments while using CBFs to maintain collision avoidance. CDPs offer deterministic guarantees, in contrast with the probabilistic guarantees of

ACI, on the average-over-time of the number of violations of the uncertainty bounds. In that setting, the ground-truth trajectories are observed after one time step, in contrast with the setting of this paper where the state is only partially observed and the ground-truth is not revealed. Another close work to ours is UNISafe (Seo et al. (2025)), which extends traditional latent safety filters by accounting for epistemic uncertainty to avoid regions with OOD dynamics, and thus unseen hazards, and provides safety guarantees using conformal prediction. Before deployment, it calibrates an uncertainty threshold via conformal prediction, defining an OOD region in the latent space that is added to the failure set for which the HJ value function is learned. While this approach enables avoidance of previously unknown failures, it can be overly conservative, preventing entering unseen regions even when that is not safety-violating. In contrast, our method applies adaptive conformal inference at deployment time, not during training. A Hamilton-Jacobi reachability value function is first learned without modeling uncertainty, then the safety filter is designed by dynamically adjusting the threshold value for switching based on observed prediction errors.

Moreover, Kim et al. (2025) train neural control barrier-value functions (CBVFs) and use conformal prediction to expand their level sets. Then, they solve quadratic programs online to compute the closest safe control to the nominal one. On the other hand, Lin and Bansal (2024) use conformal prediction to verify super-level sets of learned BRTs. Other existing works that use HJ value functions as safety filters typically switch to the safe controller whenever the predicted HJ value function evaluated at the next time-step is greater than some user-defined threshold, without formal guidance on how to choose the threshold (Nakamura et al. (2025); Tabbara et al. (2025a)). These methods do not provide formal guarantees on the safety when following the resulting policy that arises from switching between the nominal and the learned HJ-based safe controllers.

2. Preliminaries

Consider a control system, or agent, operating in an environment described by unobserved states $z \in \mathcal{Z}$, relying on high-dimensional observations $Obs(z) \in \mathcal{X}$ to pick control actions in a control space, a compact set $\mathcal{U} \subset \mathbb{R}^m$, in order to accomplish a task while avoiding a set of unsafe states $\mathcal{Z}_{unsafe} := \{z \in \mathcal{Z} \mid l_{\mathcal{Z}}(z) < 0\}$, for some $l_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}$. An encoder E_ϕ can then be used to generate latent states in a low-dimensional space $\mathcal{Y}$. After each observation $x_t = Obs(z_t)$ of the environment, the encoder combines it with the previous latent state y_{t-1} and returns the current one $y_t \sim E_\phi(y_t \mid y_{t-1}, x_t)$. Such an encoder is usually trained as a component of a world model (Bar et al. (2025)).

2.1. Hamilton-Jacobi value function

Given a set $\mathcal{F}$ of failure states, a Hamilton-Jacobi reachability value function V and its associated safe policy π^{safe} define the *Backward Reachable Tube* (BRT) of $\mathcal{F}$ for the control system. The BRT is the set of states starting from which the system inevitably eventually enter $\mathcal{F}$ using any policy. Moreover, when starting from a state in the complement of the BRT and following π^{safe}, the system never reaches $\mathcal{F}$ (Bansal et al. (2017)). Nakamura et al. (2025) and Tabbara et al. (2025a) train a classifier $l : \mathcal{Y} \to \mathbb{R}$ over the latent space that defines the failure set $\mathcal{F} := \{y \in \mathcal{Y} \mid l(y) < 0\}$. Then, they conduct approximate HJ reachability analysis in the latent space to train both a HJ reachability value function $V_\theta : \mathcal{Y} \to \mathbb{R}$ and a corresponding safety-preserving policy $\pi^{safe}_\theta : \mathcal{Y} \to \mathcal{U}$. For $y \in \mathcal{Y}$, the HJ value function is defined as $V_\theta(y) := \max_{u \in \mathcal{U}} Q_\theta(y,u)$, and the policy as $\pi^{safe}_\theta(y) := \arg\max_{u \in \mathcal{U}} Q_\theta(y,u)$, where Q_θ is the associated Q-function. This Q-function Q_θ is learned by employing reinforcement

learning methods such as DDPG (Lillicrap et al. (2015)) and DDQN (Van Hasselt et al. (2016)) to minimize the following loss function:

$$L(\theta) = \mathbb{E}_{(y_t,u_t,y_{t+1})\sim D}\left[(Q_\theta(y_t,u_t) - R(y_t,u_t,y_{t+1}))^2\right], \tag{1}$$

where R is the *target function* and is defined as follows:

$$R(y_t,u_t,y_{t+1}) := (1-\gamma)l(y_t) + \gamma \min\left\{l(y_t),\ \max_{u\in\mathcal{U}} Q_\theta(y_{t+1},u)\right\}, \tag{2}$$

where $\gamma \in (0,1)$ is a discounting parameter. The policy π_θ^{safe} can either be computed at runtime when the action space is finite and small or can be a learned along with the Q-function using actor-critic methods, otherwise. One can then plug π_θ^{safe} in the second argument of Q_θ to compute V_θ.

2.2. Adaptive conformal inference

Consider data points in the form of $(X,Y) \in \mathbf{X}\times\mathbf{Y}$ sampled from an unknown distribution for some sets $\mathbf{X}$ and $\mathbf{Y}$. Given a predictor $\mu : \mathbf{X} \to \mathbf{Y}$, the conformal prediction framework uses a *calibration dataset* $\{(X_n,Y_n)\}_{n\in[N]}$, a score function $s : \mathbf{Y}^2 \to \mathbb{R}$, and a *miscoverage rate* α, to compute the $(1-\alpha)$-quantile q of the set of conformal scores $\{s(\mu(X_n),Y_n)\}_{n\in[N]}$. If the joint distribution from which the calibration data set and fresh data points are sampled is invariant under permutations, it is called exchangeable. In that case, for any freshly sampled data point (X',Y'), Y' is guaranteed to belong to the conformal set $I_{N+1} := \{Y \in \mathbf{Y} \mid s(\mu(X'),Y) \le q\}$ with probability at least $1-\alpha$ over the joint distribution of the calibration set and the fresh data point Angelopoulos and Bates (2023).

Adaptive Conformal Inference (ACI) (Gibbs and Candes (2021)) extends this method to repeated predictions in a time-series $\{(X_t,Y_t)\}_{t\in\mathbb{N}_{\ge 1}}$, even under distribution shift. In ACI, the true output is observed in a delayed fashion, e.g., at the next step. For step $t \ge 1$, the series's history $\{(X_{t'},Y_{t'})\}_{t'<t}$ is considered as the calibration dataset, and an *effective miscoverage rate* α_t is used to define the quantile q_t of the set of conformal scores. This rate adapts to the observed prediction errors using the *update rule* $\alpha_{t+1} := \alpha_t + \lambda(\alpha - err_t)$, with a fixed user-defined *learning rate* λ and *target miscoverage rate* α. The error term is defined as $err_t := 1[Y_t \notin I_t] = 1[s(\mu(X_t),Y_t) > q_t]$. The target miscoverage rate serves as the limit of the average error rate as stated by the following theorem.

Theorem 1 *Long-term error rate bound (Gibbs and Candes (2021)): Fix a user-defined miscoverage rate $\alpha \in [0,1]$ and a learning rate $\lambda \in \mathbb{R}^{>0}$, and consider the update rule for α_t. Then, with probability 1, $\frac{1}{T}\sum_{t=1}^{T} err_t = \alpha + o(1)$, as $T \to \infty$. More precisely, the following holds:*

$$\forall T \in \mathbb{N},\ \left|\frac{1}{T}\sum_{t=1}^{T} err_t - \alpha\right| \le \frac{\max\{\alpha_1, 1-\alpha_1\} + \lambda}{T\lambda} = O\left(\frac{1}{T}\right),$$

where α_1 is the user-initialized value of α_t.

3. Methodology

In this section, we describe ACoFi and discuss its guarantees.

3.1. Safe control while accounting for prediction errors

The safety constraint the agent aims to maintain for the system is $V_\theta(y_t) > 0$. Without uncertainty, the previously described Q_θ and policy π_θ^{safe} can be used for runtime safety filtering by considering the value of $Q_\theta(y, \pi^{task}(y))$. This takes the form of a switching strategy, using π_θ^{safe} at y when $Q_\theta(y, \pi^{task}(y))$ is below a fixed user-defined threshold $\varepsilon > 0$ as follows:

$$\pi^{fixed}(y) = 1[Q_\theta(y, \pi^{task}(y)) \geq \varepsilon] \cdot \pi^{task}(y) + 1[Q_\theta(y, \pi^{task}(y)) < \varepsilon] \cdot \pi_\theta^{safe}(y).$$

However, $Q_\theta(y_t, u_t)$ is not necessarily equal to the target function value $R(y_t, u, y_{t+1})$ because of generalization errors in in-distribution and out-of-distribution states. Our method quantifies how such errors affect safety and accounts for them in the switching strategy. At step $t+1$, when the new latent state y_{t+1} is obtained from the new observation at time $t+1$, the target function value at time t $R_t = R(y_t, u_t, y_{t+1})$ can be computed. Moreover,

$$R_t = (1-\gamma) l(y_t) + \gamma \min\{l(y_t), V_\theta(y_{t+1})\} \leq (1-\gamma) l(y_t) + \gamma V_\theta(y_{t+1}).$$

Hence, if u_t is chosen so that $R_t \geq \gamma\varepsilon + (1-\gamma) l(y_t)$, for some $\varepsilon > 0$, then $V_\theta(y_{t+1}) \geq \frac{1}{\gamma}(R_t - (1-\gamma) l(y_t)) \geq \varepsilon$, satisfying the safety constraint.

3.2. Adaptive conformal filtering

Since we are only concerned with the uncertainty which negatively affects safety, we use the conformal score $S_t = s(Q_\theta(y_t, u_t), R_t) = \max\{Q_\theta(y_t, u_t) - R_t,\ 0\}$. Our safety filter, Algorithm 1, tracks the $(1-\alpha_t)$-quantile of the score history $\{S_{t'}\}_{t' \leq t}$. This defines the following interval:

$$I_t = \{\mathrm{r} \in \mathbb{R} \mid s(Q_\theta(y_t, u_t), \mathrm{r}) \leq q_t\} = \{\mathrm{r} \in \mathbb{R} \mid Q_\theta(y_t, u_t) - \mathrm{r} \leq q_t\} = [Q_\theta(y_t, u_t) - q_t, +\infty).$$

Then, at the next step, the error term $err_t = 1[R_t \notin I_t]$ is used to update the effective miscoverage level to α_{t+1}. We want to pick a control u_t such that the safety constraint $V_\theta(y_{t+1}) \geq \varepsilon$ is satisfied. Accordingly, our method tests whether u_t satisfies the Q-value constraint. The initialization of the algorithm consists of assigning the values $l(y_1)$, $\emptyset$, and 0, to the variables l_1, $\mathcal{S}$, and q_1, respectively. The algorithm runs until the agent accomplishes the task or the run ends, which is encoded by `Terminating`. Although this is not required, the first control input u_1 should ideally be safe to use in state y_1, since there is no prior history to evaluate the accuracy of Q_θ at the start. The function `StepAndEncode`(y_t, u_t) allows the system to use control u_t, observe $x_{t+1} = Obs(z_{t+1})$, and encode $y_{t+1} \sim E_\phi(y_t, x_{t+1})$. `Insert` updates $\mathcal{S}$ while maintaining it in sorted order for easy quantile computation. The function `Quantile`$(\mathcal{S}, p)$ returns 0 for $p < 0$, $+\infty$ for $p > \frac{|\mathcal{S}|}{|\mathcal{S}|+1}$, and the $\lceil p \cdot (|\mathcal{S}|+1) \rceil$-th element of $\mathcal{S}$, otherwise. This can be interpreted as follows:

- If $\alpha_{t+1} > 1$, then the target values $\{R_{t'}\}_{t' \leq t}$ have satisfied the inequalities $Q_\theta(y_{t'}, u_{t'}) - q_{t'} \leq R_{t'}$ enough times to be confident that the safety estimate is currently accurate, i.e., that R_{t+1} will satisfy the inequality $Q_\theta(y_{t+1}, \pi^{task}(y_{t+1})) \leq R_{t+1}$ with high probability. Hence, the quantile q_{t+1} gets assigned the value 0. That encourages less conservative control, improving task performance.
- If $\alpha_{t+1} < 1/(|\mathcal{S}|+1)$, then the target values $\{R_{t'}\}_{t' \leq t}$ have violated the conformal bounds $Q_\theta(y_{t'}, u_{t'}) - q_{t'} \leq R_{t'}$ enough times to be confident that $Q_\theta(y_{t+1}, \pi^{task}(y_{t+1}))$ is not a good estimate of the safety of following π^{task}. Thus, the algorithm prioritizes safety by setting the quantile q_{t+1} to $+\infty$, ensuring that $err_{t+1} = 0$ and ensuring a switch to π^{safe} at the current step.

Algorithm 1: Adaptive Conformal Filtering (ACoFi) Algorithm

input : Starting latent state y_1, control u_1, and miscoverage rate α_1

1 **while** $\neg$ `Terminating(`y_t`)` **do**
2 $\quad y_{t+1} \leftarrow$ `StepAndEncode(`y_t, u_t`)`
3 $\quad R_t \leftarrow (1-\gamma) l_t + \gamma \min\{l_t,\ V_\theta(y_{t+1})\}$
4 $\quad err_t \leftarrow 1[s(Q_\theta(y_t,u_t),R_t) > q_t]$
5 $\quad \alpha_{t+1} \leftarrow \alpha_t + \lambda(\alpha - err_t)$
6 $\quad$ `Insert(`$s(Q_\theta(y_t,u_t),R_t),\ \mathcal{S}$`)`
7 $\quad q_{t+1} \leftarrow$ `Quantile(`$\mathcal{S}, 1-\alpha_{t+1}$`)`
8 $\quad l_{t+1} \leftarrow l(y_{t+1})$
9 $\quad$ **if** $Q_\theta(y_{t+1}, \pi^{task}(y_{t+1})) \geq q_{t+1} + \gamma\varepsilon + (1-\gamma) l_{t+1}$ **then**
10 $\quad\quad u_{t+1} \leftarrow \pi^{task}(y_{t+1})$
11 $\quad$ **else**
12 $\quad\quad u_{t+1} \leftarrow \pi_\theta^{safe}(y_{t+1})$
13 $\quad$ **end**
14 **end**

Finally, from the test statement at the end, we identify the expression $q_{t+1} + \gamma\varepsilon + (1-\gamma) l_{t+1}$ as the *adaptive threshold* that the Q-value of the task policy must pass to be used.

3.3. Conformal safety guarantee

Our adaptive policy is guaranteed to have its average error rate upper bounded by $\alpha + o(1)$. This means that, in the long term, $\alpha + o(1)$ is an upper bound on the proportion of steps which followed the nominal policy and resulted in a Q-value under the threshold $q_{t+1} + \gamma\varepsilon + (1-\gamma) l_{t+1}$. We formalize this in our main theorem, whose proof is in Appendix C, and its corollary.

Theorem 2 *When using Algorithm 1, with probability one, the computed bound on the next safety value will hold for a proportion of* $1 - \alpha + o(1)$ *of the history. More precisely,*

$$\frac{1}{T} \sum_{t=1}^{T} 1\left[V_\theta(y_{t+1}) \geq B_t\right] \geq 1 - \alpha + O\left(\frac{1}{T}\right),$$

where the lower bound $B_t := \gamma^{-1}(Q_\theta(y_t,u_t) - q_t - (1-\gamma) l_t)$ *can be computed at time step* t.

Corollary 1 *The proportion of violations of the constraint* $V_\theta(y_{t+1}) \geq \varepsilon$ *by task actions over the whole history is at most* $\alpha + o(1)$. *Indeed, line 9 of the algorithm forbids task control* $\pi^{task}(y_{t+1})$ *that would result in* $B_{t+1} \leq \varepsilon$. *Hence, in the long term, a proportion of at least* $1 - \alpha$ *of the steps using* π^{task} *will be guaranteed to satisfy* $V_\theta(y_{t+1}) \geq B_{t+1} \geq \varepsilon > 0$.

ACoFi does not certify that π^{safe} is safe, but it decides when it is preferable to stop following π^{task} and hand control to the former as it is expected to be safer than the latter. In particular, early violations are not ignored by the method. They are the feedback that drives ACoFi to become more conservative than a fixed-threshold switching rule under the same uncertainty. Moreover, observed errors in the value of V_θ at deployment can be assumed to correlate with changes in ground truth safety. In that

setting, ACoFi becomes more conservative when actual safety is inferred to be decreasing, which results in a long-term high probability of preserving safety. The following remarks elaborate on ACoFi's use cases and advantages.

Remark 1 *The 0-sublevel set of V_θ approximates the set of states starting from which and following the policy π_θ^{safe} leads to reaching $\mathcal{F}$, which in itself is an over-approximation of the BRT of $\mathcal{F}$. While ACoFi does not guarantee preventing actions that lead to $V_\theta(y_{t+1}) < 0$ at some time instances, that does not necessarily imply that following π_θ^{safe} starting from the time step $t+1$ lead to $\mathcal{F}$, i.e., failure is not inevitable and recovery is possible. The reasons are that V_θ is only an approximation of the safety value function of π_θ^{safe}. Moreover, the latent state y_{t+1} might correspond to multiple true states of the system and $V_\theta(y_{t+1})$ is not necessarily the worst value of π_θ^{safe} starting from any such states. Consequently, it is possible that $V_\theta(y_{t+1}) < 0$ and following π_θ^{safe} from $t+1$ onward prevents reaching $\mathcal{F}$.*

Remark 2 *When a sequence of tasks share the same source of uncertainty for the learned HJ function, ACoFi can carry this adaptation across tasks and keep refining the switching threshold for that specific uncertainty pattern. In that regime, early tasks effectively calibrate the filter for later ones, so the practical guarantee of respecting the safety margin $V_\theta(y) > \varepsilon$ approaches the user-chosen threshold over the aggregate deployment horizon rather than only within a single finite-time task. The user-defined threshold $\varepsilon > 0$ acts as a robustness layer to decrease the possibility of instances at which $V_\theta(y_{t+1}) < 0$.*

4. Experiments

We consider two case studies in this work: First, (1) a vision-based Dubins car setup where an agent with Dubins car dynamics must reach a goal while avoiding two obstacles, and (2) Safety Gymnasium's `SafetyCarGoal2-v0` environment (Ji et al. (2023)). Both environments are illustrated in Figure 1. For each case study, we first collect a dataset using a nominal policy. Then, we train a DINO-WM world model using the collected dataset. Next, we derive a HJ value function V from learning the Q-function with (1). Finally, we implement the ACoFi algorithm and compare its performance in completing tasks while maintaining safety against the baseline.

4.1. Dubins car with Dino-WM

We first evaluate our approach in a simulated discrete-time 2D Dubins car environment, a controlled benchmark that enables a clear analysis of how our adaptive safety filtering handles uncertainty. In this experiment, we simulate a car with position (p_x, p_y) and heading θ inside a bounded space populated with one goal zone and two obstacles. During training, the car moves deterministically at a constant speed $v^{ID} = v$ with the only control being one of three steering actions: $\omega_t^{ID} \in \{-\omega, 0, \omega\}$, where $\omega = 0.05$ rad/step, for setting the angular velocity. The observations are bird-eye view pictures of the environment, as seen in Figure 1. During evaluation, π^{task} is a PID controller that steers the agent towards the goal, without any consideration for the obstacles. Each run consists of reaching the goal in the top right of the environment five times before the timeout. When the goal is reached or a wall of the environment is hit (not the obstacles), the agent is placed in a random starting position in the lower left. We discuss the data collection, training the DINO-WM world model, and training the HJ value function in the learned latent space in Appendix A.

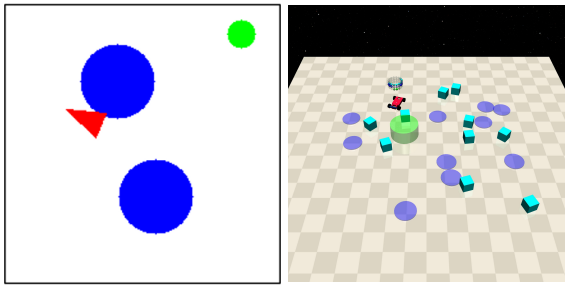

Figure 1: (Left) The Dubins car is depicted with a red triangle, with the heading angle in the direction of the narrower corner, the obstacles as blue circles and the goal as a green circle. Violations only happen when the center point of the triangle is inside the obstacle. (Right) SafetyGymnasium's Car agent needs to navigate to the goal (green circle), while avoiding to collide with fixed obstacles (blue circles) and with movable obstacle cubes (cyan).

Out-of-distribution dynamics Uncertainty is simulated by disturbing the dynamics model's parameters at runtime, making the dynamics stochastic. That is:

$$v_t^{OOD} \sim v^{ID} + U(-1,1) \cdot v \qquad \text{and} \qquad \omega_t^{OOD} \sim \omega_t^{ID} + U(-1,1) \cdot \omega, \tag{3}$$

where $U(-1,1)$ is the uniform distribution over the interval $[-1,1]$. Since the training was done with constant speed v^{ID} and deterministic action ω_t^{ID}, this setup simulates an agent trained on known average values of the dynamics' parameters, and later confronted with large uncertainty on these parameters at runtime due to unmodeled parts of the environment. The HJ value function V_θ is considered the best estimate of safety the agent has access to before deployment in that environment and ACoFi's role is to adjust decision-making to the observed safety-relevant disturbances to the dynamics.

Using the dynamics in (3), the four scenarios we consider are **ID:** using the same dynamics as in the Q-function's training, **VarSpeed:** using v_t^{OOD}, i.e., a perturbation of the speed in $[-v,v]$ at every step, **VarSteer:** using ω_t^{OOD}, i.e., a perturbation of the control input in $[-\omega,\omega]$, and **VarSpeed&Steer:** using v_t^{OOD} and ω_t^{OOD}, i.e., both perturbations at the same time.

Baseline ACoFi is evaluated against the fixed threshold switching policy π^{fixed}. The range of HJ values after training is $[-7.5,+27.5]$, the chosen safety value threshold is $\varepsilon = 0.1$. We observe that this ε best balanced safety and goal reaching performance when using π^{fixed}. The metrics for comparing adaptive and fixed-threshold switching are the averages over 16 runs of the following measures: the *success rate* of reaching the goal r_{goal}, the *minimum learned safety value* encountered $\min_t V_\theta(y_t)$, the *proportion of steps of violations* p_{unsafe} of the constraint $V_\theta(y) > \varepsilon$, and the *proportion of steps using the safe policy* $p_{\pi_\theta^{safe}}$. The runs were capped at 1000 steps and are seeded so that the agents using π^{fixed} and ACoFi experience the same OOD disturbances on their speed/steering for fair comparison. The experiments use a target miscoverage rate of $\alpha = 0.2$ and a conformal learning rate of $\lambda = 0.05$.

4.2. Safety-Gymnasium's CarGoal environment

ACoFi is evaluated in the CarGoal environment from the Safety-Gymnasium benchmark suite, which provides standardized tests for safe reinforcement learning. In the `SafetyCarGoal2-v0` environment, the car agent must reach a designated goal region while navigating around two kinds of obstacles: fixed collision regions, and movable cubes (see Figure 1). CarGoal introduces high dimensional state and observation spaces with partial observability, and more complex dynamics compared to the Dubins car task. The goal region, hazard position, and the agent's initial state are randomly sampled at the beginning of every run. The agent is not given access to lidar measurements

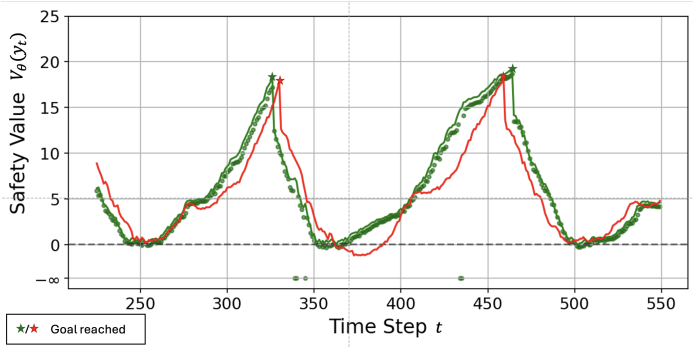

Figure 2: Graphs of V_θ for Dubins car agents using π^{fixed} (red) and ACoFi (green), under the same **VarSpeed&Steer** OOD scenario, with safety threshold $\varepsilon = 0.1$ (gray). The selected time frame shows both agents completing two goal-reaching tasks and being put back in a starting position afterwards. The circle markers plot the lower bound B_t, which is sometimes set to $-\infty$ forcing a switch to π_θ^{safe} in the case of ACoFi.

given by the simulator, which can be used to calculate the distance between the agent and obstacles. We use image observations instead, which makes the task harder. Within the same run, the goal region is resampled once it is reached. The action space is $[-1,1]^2$, representing the force (N) acting on the two independent front wheels. Details on data collection, DINO-WM and HJ value function training are in Appendix B.

ID			
Policy	$\min_t V_\theta(y_t)$	p_{unsafe}	$p_{\pi_\theta^{safe}}$
π^{task}	-1.311	15.0/603.4	-
π^{fixed}	0.040	1.2/603.9	**10.2/603.9**
ACoFi	**0.194**	**0.0/605.4**	22.3/605.4

VarSteer			
Policy	$\min_t V_\theta(y_t)$	p_{unsafe}	$p_{\pi_\theta^{safe}}$
π^{task}	-1.625	19.9/605.1	-
π^{fixed}	-0.277	4.8/606.3	**18.4/606.3**
ACoFi	**0.000**	**1.7/607.7**	25.7/607.7

VarSpeed			
π^{task}	-1.346	15.2/603.3	-
π^{fixed}	-0.087	3.8/608.2	**13.9/608.2**
ACoFi	**0.158**	**2.0/605.1**	22.2/605.1

VarSpeed&Steer			
π^{task}	-1.801	17.8/607.3	-
π^{fixed}	-0.196	3.4/598.8	**18.2/598.8**
ACoFi	**0.001**	**2.4/611.6**	25.7/611.6

Table 1: Results for the Dubins car environment with $\varepsilon = 0.1$. For both the minimum learned safety value encountered and number of violations, ACoFi ($\alpha = 0.2$) is safer than π^{fixed}, and the latter is safer than π^{task}.

Value function learning inaccuracy Since the world model and the HJ value function are trained on pre-collected trajectories produced by the Dreamerv3 policy (π^{task}), the distribution of states reached during deployment under a different policy (ACoFi or π^{fixed}) is likely to be different than the one reached during training. Thus, the learned HJ value function is likely to be erroneous at some states during evaluation, which is what we observe in our experiments.

Baselines During evaluation, the Dreamerv3 controller serves as the unsafe task policy π^{task}. We use it as a baseline against the following policies: π^{fixed} which switches from the task policy to the learned safe policy π_θ^{safe} whenever the predicted safety value drops below a fixed threshold ε, and ACoFi, our proposed method, which replaces that fixed threshold with an adaptive one. We chose ε to be equal to 0.01 as we observed it best balances safety and task completion when using π^{fixed}. The agent is allowed to navigate in the environment for 1000 steps. The evaluation uses the same metrics as the previous experiment but we replace the success rate r_{goal} with the *number of times the task is achieved* M_{goal}, averaged over all runs like previously. For the target miscoverage α, the range of values from 0.1 to 0.5 are tested to illustrate its role in our approach. We ran each baseline 25 times with shared seeds, so that they are confronted to the same obstacle and goal locations.

4.3. Dubins car results

Figure 2 illustrates direct comparison between the two switching policies under the same OOD conditions: ACoFi switches to π_θ^{safe} when the high probability lower bound B_t goes below ε, while π^{fixed} switches when $Q_\theta(y_t, \pi^{task}(y_t))$ does. Over the 16 runs, no agent collided with a wall, and they all reached the goal five times before the timeout, hence r_{goal} is 100% for all three baselines. Table 1 shows the results for the other metrics for all four scenarios. For the two metrics quantifying safety, ACoFi performs better, with a higher minimum learned safety value and fewer violations than π^{fixed}, which itself improves on π^{task}. ACoFi was able to maintain $\min V_\theta(y) > 0$ on ID observations and avoid any safety violation over the 16 runs. ACoFi only incurs a minor slow down in goal completion sometimes. In fact, under **VarSpeed**, ACoFi resulted in faster goal reach than π^{fixed} on average. ACoFi results in switches to the safe policy for over twice as many steps as π^{fixed} does in the **ID** scenarios. However, for the OOD scenarios, using ACoFi does not incur a stronger reliance on the learned safe policy as much as using π^{fixed} does.

Policy	$\min_t V_\theta(y_t)$	p_{unsafe}	$p_{\pi_\theta^{safe}}$	M_{goal}
π^{task}	-0.294	287.2	-	5.92
π^{fixed}	-0.107	140.1	435.0	1.16
$ACoFi_{\alpha=.5}$	-0.090	54.1	438.3	0.96
$ACoFi_{\alpha=.4}$	-0.075	44.0	**415.8**	**1.32**
$ACoFi_{\alpha=.3}$	-0.066	40.2	420.1	0.92
$ACoFi_{\alpha=.2}$	-0.077	36.0	424.4	1.20
$ACoFi_{\alpha=.1}$	**-0.060**	**19.9**	463.4	0.80

Table 2: Results for the policies tested in the CarGoal environment. As in the Dubins Car experiment, ACoFi (with $\alpha \leq 0.5$) is safer than π^{fixed}, and the latter is safer than π^{task}, in terms of both the minimum learned safety value encountered and the number of violations of the constraint $V_\theta(y) \geq 0$. Agents using ACoFi see higher safety values and less unsafe states as α decreases, while the average number of goal reaching, which quantifies task completion, is not conclusively decreasing. The total number of steps is omitted since it is 1000 across experiments.

4.4. CarGoal results

ACoFi significantly improves upon the baselines, as Table 2 shows. Specifically, it maintained a higher minimum learned safety value and committed up to 7 times fewer safety violations than π^{fixed} on average. It also did not use the safe policy significantly more than π^{fixed}. Finally, it reached fewer goals on average, but only by 30% at most. Decreasing the target miscoverage rate α steadily improves the minimum learned safety value and reduces the number of unsafe steps, with little effect on the number of steps using the learned safe policy. The tradeoff between safety and task completion seems positive, as M_{goal} stays close to what π^{fixed} achieved.

5. Conclusion

In this work, we aim to enhance the safety of control systems with high-dimensional observations and relying on learned latent safety filters based on HJ reachability under potential distribution shifts. We propose Adaptive Conformal Filtering, a dynamic extension of the traditional fixed threshold-based switching policy applied within the latent space. Our method was evaluated in two vision-based environments, where it outperformed safety filters which use a fixed switching threshold. Specifically, our adaptive safety filter maintained higher safety values and allowed fewer unsafe actions in out-of-distribution scenarios without significantly increasing reliance on the learned safe policy. Future research should evaluate the effectiveness of this approach for multi-step predictions, and explore an extension to continuous-time control tasks.

Acknowledgments

This project was partially supported by the NSF CPS award No. 2403758.

References

Anil Alan, Andrew J. Taylor, Chaozhe R. He, Aaron D. Ames, and Gábor Orosz. Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Transactions on Control Systems Technology*, 31(6):2744–2759, 2023. doi: 10.1109/TCST.2023.3286090.

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)*, pages 3420–3431. Ieee, 2019.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591, March 2023. ISSN 1935-8237. doi: 10.1561/2200000101. URL https://doi.org/10.1561/2200000101.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Mo Chen, Qizhan Tam, Scott C Livingston, and Marco Pavone. Signal temporal logic meets reachability: Connections and applications. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 581–601. Springer, 2018.

Andrew Clark. Verification and synthesis of control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6105–6112. Ieee, 2021.

Ersin Daş and Joel W Burdick. Robust control barrier functions using uncertainty estimation with application to mobile robots. *IEEE Transactions on Automatic Control*, 2025.

Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019. doi: 10.1109/TAC.2018.2876389.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Milan Ganai, Sicun Gao, and Sylvia L Herbert. Hamilton-jacobi reachability in reinforcement learning: A survey. *IEEE Open Journal of Control Systems*, 3:310–324, 2024.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6vaActvpcp3.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Tamás Haidegger. Autonomy for surgical robots: Concepts and paradigms. *IEEE Transactions on Medical Robotics and Bionics*, 1(2):65–76, 2019.

Peiyan Hu, Xiaowei Qian, Wenhao Deng, Rui Wang, Haodong Feng, Ruiqi Feng, Tao Zhang, Long Wei, Yue Wang, Zhi-Ming Ma, et al. From uncertain to safe: Conformal fine-tuning of diffusion models for safe pde control. *arXiv preprint arXiv:2502.02205*, 2025.

Sacha Huriot and Hussein Sibai. Safe decentralized multi-agent control using black-box predictors, conformal decision policies, and control barrier functions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7445–7451, 2025. doi: 10.1109/ICRA55743.2025.11128015.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=WZmlxIuIGR.

Matthew Kim, William Sharpless, Hyun Joe Jeong, Sander Tonkens, Somil Bansal, and Sylvia Herbert. Reachability barrier networks: Learning hamilton-jacobi solutions for smooth and flexible control barrier functions. *arXiv preprint arXiv:2505.11755*, 2025.

Jordan Lekeufack, Anastasios N. Angelopoulos, Andrea Bajcsy, Michael I. Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions, 2024.

Jingqi Li, Donggun Lee, Jaewon Lee, Kris Shengjun Dong, Somayeh Sojoudi, and Claire Tomlin. Certifiable reachability learning using a new lipschitz continuous value function. *IEEE Robotics and Automation Letters*, 2025.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. In *6th Annual Learning for Dynamics & Control Conference*, pages 719–731. PMLR, 2024.

Albert Lin, Shuang Peng, and Somil Bansal. One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments, 2024. URL https://arxiv.org/abs/2412.09989.

Jonathan Michaux, Patrick Holmes, Bohao Zhang, Che Chen, Baiyue Wang, Shrey Sahgal, Tiancheng Zhang, Sidhartha Dey, Shreyas Kousik, and Ram Vasudevan. Can't touch this: Real-time, safe motion planning and control for manipulators under uncertainty. *IEEE Transactions on Robotics*, 2025.

Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

Kensuke Nakamura, Lasse Peters, and Andrea Bajcsy. Generalizing safety beyond collision-avoidance via latent-space reachability analysis, 2025. URL https://arxiv.org/abs/2502.00935.

Dorsa Sadigh and Ashish Kapoor. Safe control under uncertainty with probabilistic signal temporal logic. In *Proceedings of Robotics: Science and Systems XII*, 2016.

Junwon Seo, Kensuke Nakamura, and Andrea Bajcsy. Uncertainty-aware latent safety filters for avoiding out-of-distribution failures, 2025. URL https://arxiv.org/abs/2505.00779.

Andrew Singletary, Mohamadreza Ahmadi, and Aaron D Ames. Safe control for nonlinear systems with stochastic uncertainty via risk control barrier functions. *IEEE Control Systems Letters*, 7: 349–354, 2022.

Oswin So, Zachary Serlin, Makai Mann, Jake Gonzales, Kwesi Rutledge, Nicholas Roy, and Chuchu Fan. How to train your neural control barrier function: Learning safety filters for complex input-constrained systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11532–11539. IEEE, 2024.

Ihab Tabbara and Hussein Sibai. Learning conservative neural control barrier functions from offline data. *arXiv preprint arXiv:2505.00908*, 2025.

Ihab Tabbara, Yuxuan Yang, Ahmad Hamzeh, Maxwell Astafyev, and Hussein Sibai. Designing latent safety filters using pre-trained vision models. *arXiv preprint arXiv:2509.14758*, 2025a.

Ihab Tabbara, Yuxuan Yang, and Hussein Sibai. Statistically assuring safety of control systems using ensembles of safety filters and conformal prediction. *arXiv preprint arXiv:2511.07899*, 2025b.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Vladimir Vovk and Claus Bendtsen. Conformal predictive decision making. In *Conformal and Probabilistic Prediction and Applications*, pages 52–62. PMLR, 2018.

Shengbo Wang and Shiping Wen. Safe control against uncertainty: A comprehensive review of control barrier function strategies. *IEEE Systems, Man, and Cybernetics Magazine*, 11(1):34–47, 2025.

Hongchao Zhang, Zhouchi Li, Hongkai Dai, and Andrew Clark. Efficient sum of squares-based verification and construction of control barrier functions by sampling on algebraic varieties. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 5384–5391. IEEE, 2023.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2025. URL https://arxiv.org/abs/2411.04983.

Appendix A. Dubins task training

The dataset of trajectories consisting of the RGB images and actions was collected by simulating the agent using a random policy. After collecting this dataset, we train DINO-WM (Zhou et al. (2025)). We note that the dataset we used to train DINO-WM does not contain p_x, p_y, θ as they can be inferred from the image observation. Next, we train the HJ value function in the latent space by optimizing the loss in (1) using DDQN. We use $\gamma = 0.98$ and define $l(y)$ as the ground truth distance between the agent and the closest obstacle, which we have access to from the simulator.

Appendix B. Cargoal task training

We collect a dataset of 2000 trajectories using a reward-driven unsafe nominal policy trained with Dreamerv3 (Hafner et al. (2024)). We use this dataset to train the DINO-WM. Then, we train the HJ value function by minimizing (1) using DDPG. We use $\gamma = 0.98$ and assume that $l(y)$ is the ground truth distance function between the agent and the closest obstacle, which we have access to from the LIDAR measurements given by the environment.

Appendix C. Proof of Theorem 2

Here is the proof of Theorem 2.
Proof: For time step $t \geq 1$, using the fact that $q_t \geq 0$, we have:

$$\begin{aligned} err_t &= 1[s(Q_\theta(y_t,u_t),R_t) > q_t] = 1[\max\{Q_\theta(y_t,u_t) - R_t,\ 0\} > q_t] = 1[Q_\theta(y_t,u_t) - R_t > q_t] \\ &= 1[Q_\theta(y_t,u_t) - q_t > (1-\gamma)l_t + \gamma \min\{l_t, V_\theta(y_{t+1})\}], \end{aligned}$$

which implies $err_t \geq 1[Q_\theta(y_t,u_t) - q_t > (1-\gamma)l_t + \gamma V_\theta(y_{t+1})] = 1[B_t > V_\theta(y_{t+1})]$. Thus, we have $1[V_\theta(y_{t+1}) \geq B_t] \geq 1 - err_t$, and, by averaging over $1 \leq t \leq T$ and using Theorem 1, we get:

$$\frac{1}{T}\sum_{t=1}^{T} 1[V_\theta(y_{t+1}) \geq B_t] \geq 1 - \frac{1}{T}\sum_{t=1}^{T} err_t = 1 - \alpha + O\left(\frac{1}{T}\right).$$

■