

Conformal Decision Theory: Safe Autonomous Decisions from Imperfect Predictions

Jordan Lekeufack^{1,*} Anastasios N. Angelopoulos^{2,*} Andrea Bajcsy^{3,*} Michael I. Jordan^{1,2,**} Jitendra Malik^{2,**}

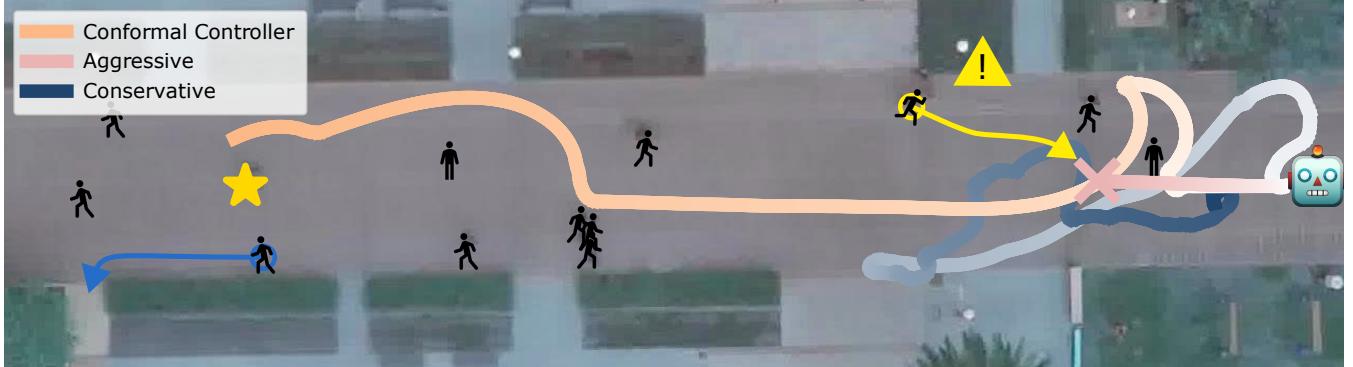


Fig. 1: Robot planner using a conformal controller on the Stanford Drone Dataset [1]. The future trajectories of humans are predicted online by a machine learning algorithm (not visualized). The robot planner finds an optimal spline through the scene and is penalized for being close to humans. This penalty is proportional to a conformal control variable, λ_t , which is adjusted online by the conformal controller so the average distance from a human is no less than 2 meters. The orange, red, and blue curves are the robot trajectory with different planners: the conformal controller, an aggressive planner with $\lambda = 0$ (i.e., no reward for avoiding humans), and a conservative planner with a large negative value of λ (i.e., a large reward for avoiding humans). The darkness of the lines indicates the passage of time. Illustrative pedestrian trajectories are plotted as arrows; only the yellow pedestrians affect the spline planner. Details in Section IV-A and videos on [project website](#).

Abstract— We introduce *Conformal Decision Theory*, a framework for producing safe autonomous decisions despite imperfect machine learning predictions. Examples of such decisions are ubiquitous, from robot planning algorithms that rely on pedestrian predictions, to calibrating autonomous manufacturing to be high throughput but low error, to the choice of trusting a nominal policy versus switching to a safe backup policy at run-time. The decisions produced by our algorithms are safe in the sense that they come with provable statistical guarantees of having low risk without any assumptions on the world model whatsoever; the observations need not be I.I.D. and can even be adversarial. The theory extends results from conformal prediction to calibrate decisions directly, without requiring the construction of prediction sets. Experiments demonstrate the utility of our approach in robot motion planning around humans, automated stock trading, and robot manufacturing.

I. INTRODUCTION

Autonomous systems increasingly rely on complex learned models to make decisions at scale. Self-driving cars rely on deep neural networks [2]–[5] to plan around nearby pedestrians, robotic manipulators leverage learned grasp models [6] to plan high throughput pick-and-place maneuvers in

factories, and AI-enabled stock trading agents optimize the financial future of investors [7]. However, ensuring that these decision-makers make good *decisions* despite imperfect *predictions* remains an open challenge.

One common path forward is to calculate the uncertainty in the predictions independently of their downstream effect on the decision [8]–[12]. For example, one can use conformal prediction [13]–[18] to form uncertainty sets that cover the ground truth outcomes of all predictions uniformly. Then, the robot can pick any decision that is safe with respect to these sets. This is statistically guaranteed to result in safe autonomous behavior, *without any assumption* on the underlying distribution or model. This strategy has been used to provide safety assurances in robot navigation [19]–[21], early warning systems (e.g., collision alerts) [22], out-of-distribution detection [23], [24], probabilistic pose estimation [25], and for large language models [26]. However, this approach decouples prediction uncertainty from decision-making. What if we could solve the problem all-at-once, *directly* control decision-making risk, and bypass the need to construct prediction sets entirely?

This work presents Conformal Decision Theory, a new theoretical and algorithmic framework that unifies predictive uncertainty and safe decision-making. Our key idea is instead of calibrating *prediction sets* for coverage, we directly calibrate *decisions* for low risk.

¹Department of Statistics, UC Berkeley. ²Department of Electrical Engineering and Computer Science, UC Berkeley. Emails: {jordan.lekeufack, angelopoulos, michael_jordan, malik}@berkeley.edu ³Robotics Institute, Carnegie Mellon University. Email: {fabajcsy}@cmu.edu * Equal contribution. ** Equal senior authorship.

Our main algorithmic innovation is a class of algorithms called *conformal controllers*. A conformal controller starts with a conformal control variable, λ_t , which determines the decision-maker's conservatism or aggressiveness. Then, it dynamically adjusts λ_t to balance risk and performance in such a way that guarantees a low risk. The main practical benefit of this approach is its *emergent ability to ignore irrelevant uncertainty*, only accounting for that which *affects decisions*. This can be much less conservative than the prediction-set strategy. For example, in Figure 1, the planner only considers humans that pose a collision risk.

The contributions of this paper are threefold:

- We introduce Conformal Decision Theory, the idea of directly calibrating decisions with conformal controllers. This extends the line of work in online adversarial conformal prediction [15], [18], [27], [28] to the decision-making setting. To our knowledge, all previous conformal works calibrate predictions, not decisions.
- We prove finite-time risk bounds for conformal controllers. Even when applied to prediction sets, these results are stronger than any previously known results for online adversarial conformal prediction.
- We show the utility of the framework in three simulation examples of Conformal Decision Theory applied to robot navigation in the Stanford Drone Dataset [1], a stock trading simulation, and a robot manufacturing example.

The main potential impact of this work is to broaden the scope of conformal prediction. Our methods are more appropriate for disciplines that focus on decision-making, such as controls, reinforcement learning, and operations research domains. In these disciplines, algorithms are ultimately evaluated by the decisions, not the predictions, that they make. Furthermore, there are many settings where it does not make sense to construct prediction sets, and our technique would provide a totally new distribution-free outlook for such problems (see, e.g., Section IV-B).

II. CONFORMAL DECISION THEORY

Conformal Decision Theory (CDT) is an approach for calibrating an agent's decisions to achieve statistical guarantees on the realized average loss of those decisions. Consider a decision-making agent whose input space is \mathcal{X} and action space is \mathcal{U} . In our running example of robot navigation, $x_t \in \mathcal{X}$ captures the current state of the robot, the current scene information (e.g., environment geometry), and the agent information (e.g., pedestrian predictions) while $u_t \in \mathcal{U}$ is the action that the ego vehicle plans at the current time t . At time t , the agent has access to a family of *decision functions*

$$\mathcal{D}_t := \{D_t^\lambda : \mathcal{X} \rightarrow \mathcal{U}, \lambda \in \mathbb{R}\} \quad (1)$$

parameterized by λ , which we call a *conformal control variable*. One should think of λ as indexing the decisions from least to most conservative. In Figure 1, \mathcal{D}_t is the set of dynamically feasible splines at time t , λ is the coefficient of

the reward term for avoiding humans, and D_t^λ is the spline maximizing the total reward given λ .

Assessing the quality of an agent's decision depends on a space of *targets* \mathcal{Y} . Importantly, the realizations of these targets are *unknown* at the time of the decision; the agent only observes them at deployment time, after decisions are made, and in an online fashion. For example, the robot in Figure 1 does not know the true future state of nearby pedestrians; at any current time t , it only knows the (potentially erroneous) pedestrian predictions. In this example, \mathcal{Y} is the space of pedestrian states (e.g., 2D positions) and $y_t \in \mathcal{Y}$ is the *true state* where the pedestrian moves to at time t .

Mathematically, the quality of the decision-making is quantified by a *loss function* $\mathcal{L} : \mathcal{U} \times \mathcal{Y} \rightarrow [0, 1]$ ¹. Often, the loss is more likely to be large when aggressive decisions are taken — i.e., when λ is large. For example, \mathcal{L} may be the distance from the planned spline D_t^λ to the nearest human y_t . Aggressive decisions can be unsafe, but taking λ too small yields conservative and under-performing decisions.

We seek an algorithm for adapting λ_t (and thus the corresponding decision D_t^λ) at each time step such that the average loss is controlled in hindsight for *any* realization of an input-target sequence $\{(x_t, y_t)\}_{t=1}^T$. This is commonly known as the *adversarial sequence model* [15], [29]. In this setting, our goal is to set $\lambda_{1:T}$ to achieve a *long-term risk bound*:

$$\text{find } \lambda_{1:T} \text{ s.t. } \hat{R}_T(\mathcal{D}, \lambda_{1:T}) \leq \varepsilon + \frac{C}{T}, \quad (2)$$

where ε is a pre-defined risk level in $[0, 1]$, C is a (small) constant, and

$$\hat{R}_T(\mathcal{D}_{1:T}, \lambda_{1:T}) := \frac{1}{T} \sum_{t=1}^T \mathcal{L}(D_t^{\lambda_t}(x_t), y_t) \quad \text{and} \quad \hat{R}_0 = 0. \quad (3)$$

The bound in (2) can be trivially extended to

$$\hat{R}_T(\mathcal{D}, \lambda_{1:T}) \leq \varepsilon + \frac{C \cdot h(T)}{T} \quad (4)$$

where $h(T)$ is any sub-linear function, i.e one where $h(T)/T \rightarrow 0$ as $T \rightarrow \infty$.

III. THEORY & CONFORMAL CONTROLLER ALGORITHM

In this section, we prove the core theoretical results behind Conformal Decision Theory. Specifically, we show that any sequence of families of decision functions $\mathcal{D}_{1:T}$ that are *eventually safe* can be calibrated online to achieve bounded long-term risk. We then introduce an algorithm called **CONFORMALCONTROLLER** which solves Equation (2) under the assumption of eventual safety.

Definition 1 (Eventually Safe). *In the setting above, we say that $\mathcal{D}_{1:T}$ is eventually safe if $\exists \varepsilon^{\text{safe}} \in [0, 1], \lambda^{\text{safe}} \in \mathbb{R}$ and a time horizon $K > 0$ such that uniformly over all sequences*

¹The framework works for any bounded loss, but we assume the loss to be in $[0, 1]$ for simplicity.

$\lambda_{1:K}$ and $\{(x_1, y_1), \dots, (x_k, y_k)\} \in \mathcal{X} \times \mathcal{Y}$,

$$\begin{aligned} & \{\forall k \in [K], \lambda_k \leq \lambda^{\text{safe}}\} \\ & \implies \frac{1}{K} \sum_{k=1}^K \mathcal{L}(D_k^{\lambda_k}(x_k), y_k) \leq \varepsilon^{\text{safe}} \end{aligned} \quad . \quad (5)$$

Intuitively, this condition says that there exists a safe value λ^{safe} such that if the conformal control variable lands below that value, it will incur a low risk $\varepsilon^{\text{safe}}$ after no more than K time steps. For example, even the most conservative robot planner may not be able to change its trajectory fast enough *in a single time-step*, but it could possibly do so in K time steps. Note that this is a strictly weaker assumption than that used for the proofs in other works, such as [15], [28], [30], which require $K = 1$. Conformal controllers are simple yet efficient algorithms that solves the Conformal Decision Theory problem stated in Equation (2). An example is below.

Theorem 1 (Conformal Controller). *In the above setting, consider the following update rule for λ :*

$$\lambda_{t+1} = \lambda_t + \eta(\varepsilon - \ell_t), \quad (6)$$

where $\eta > 0$. If $\lambda_1 \geq \lambda^{\text{safe}} - \eta$ and $\mathcal{D}_{1:T}$ satisfies Definition 1 for a given $K \geq 1$ and $\varepsilon^{\text{safe}} \leq \varepsilon$, then for any realization of the data, $\forall t \in [T]$ the empirical risk is bounded:

$$\hat{R}_t(\lambda_{1:t}) \leq \varepsilon + \frac{(\lambda_1 - \lambda^{\text{safe}})/\eta + K}{t} \quad (7)$$

The update in (6) resembles ACI [15] and is a hybrid between the RollingRC update [18], and the P-controller update [28]. The difference is that the update is applied to λ and not the conformal quantile or quantile level.

Proof of Theorem 1. By the definition of the update rule,

$$\lambda_t = \lambda_1 + \eta \sum_{s=1}^t (\varepsilon - \ell_s). \quad (8)$$

Dividing both sides by $-\eta T$ yields

$$\hat{R}_t(\lambda_{1:t}) = \frac{1}{t} \sum_{s=1}^t \ell_s = \varepsilon + \frac{\lambda_1 - \lambda_t}{\eta t}$$

To conclude, we just need to show that $\lambda_t \geq O(K)$. That's exactly what is proved in the following Lemma 1.1. \square

Lemma 1.1. *For the sequence in Equation 6, with probability one, we have that $\forall t \in [T]$ the parameter λ_t is bounded below by $\lambda_t \geq \lambda^{\text{safe}} - K\eta$.*

Proof. First note that the maximal change in the parameter is $\sup_{t \in [T]} |\lambda_{t+1} - \lambda_t| < \eta$ because $\ell_t \in [0, 1]$ and $\varepsilon \in [0, 1]$. We will then proceed by contradiction: Assume that with non-zero probability, $\inf_{u \in [T]} \lambda_u < \lambda^{\text{safe}} - K\eta$. Denote $t = \arg \min_{u \in [T]} \{u, \lambda_u < \lambda^{\text{safe}} - K\eta\}$ in other words t is the first instant when the parameter goes below that lower bound. Because the max difference between successive steps is η , we can easily prove recursively that $\forall k \in \{0, \dots, K\}, \lambda_{t-k} < \lambda^{\text{safe}} - (K - k)\eta$. Note that, from those inequalities, we can

$$\begin{aligned} & \text{deduce that } t > K \text{ since } \lambda_1 \geq \lambda^{\text{safe}} - \eta. \text{ by recursively applying the update rule } \lambda_t = \lambda_{t-K} + K\eta(\varepsilon - \frac{1}{K} \sum_{k=1}^K \ell_{t-k}), \\ & (\forall k \in \{0, \dots, K-1\}, \lambda_{t-k} < \lambda^{\text{safe}}) \\ & \implies \frac{1}{K} \sum_{k=1}^K \ell_{t-k} \leq \varepsilon^{\text{safe}} \\ & \implies \lambda_t = \lambda_{t-K} + K\eta \left(\varepsilon - \frac{1}{K} \sum_{k=1}^K \ell_{t-k} \right) \geq \lambda_{t-K} + K\eta(\varepsilon - \varepsilon^{\text{safe}}) \\ & \implies \lambda_t \geq \lambda_{t-K} \end{aligned}$$

Since t is the first ever timestep to go below $\lambda^{\text{safe}} - K\eta$, this is contradictory. \square

IV. EXPERIMENTS

We demonstrate Conformal Decision Theory in three autonomous decision-making domains, each of which exhibit a different way that a conformal controller can be instantiated. First, we consider a robot navigation around humans example in the Stanford Drone Dataset [1] where CDT tunes the robot's reward function online to be safe but efficient. Next, we model a manufacturing setting where CDT directly calibrates the speed of the conveyor belt under a robot to achieve successful but high-throughput robot grasps. Finally, we study an automated high-frequency trading example where CDT must optimize the buying and selling of stocks.

A. Robot Navigation in Stanford Drone Dataset

Robot navigation around people must balance safety (i.e., not colliding with humans) and efficiency (i.e., robot making progress towards goal). To ensure that the risk of collision is low while still making progress to the goal, the robot will constantly calibrate its cost function at run-time using a conformal controller.

Decision Function & Parameterization. The robot plans via model predictive control and at each timestep fits a minimum-cost spline subject to the robot's dynamics constraints, which is a non-linear Dubins car [31]. Let $g := [g_x, g_y] \in \mathbb{R}^2$ be the robot's goal location. Let t be the current time, $H < T$ to be the planning horizon, and $u_{t:t+H} \in \mathbb{R}^{H \times 3}$ be a spline consisting of the robot's planar position and orientation. The robot also gets as input the current set of short-horizon predictions of each human's state, $x_{t:t+H} \in \mathcal{P}_t$, generated by an autoregressive predictive model [32]. Note, that this set \mathcal{P}_t can include predictions for *multiple* humans in the scene (as shown in Figure 1). The robot's planning objective is

$$J(u_{t:t+H}; \mathcal{P}_t, \lambda) := \sum_{\tau=t}^{t+H} \underbrace{\|u_\tau^{\text{pos}} - g\|}_{\text{Goal distance}} + \lambda \cdot \underbrace{\inf_{x_\tau \in \mathcal{P}_t} \|u_\tau^{\text{pos}} - x_\tau\|_2^2}_{\text{Human avoidance}} \quad (9)$$

where the notation $u_\tau^{\text{pos}} \in \mathbb{R}^2$ indicates the *xy*-positional entries of the robot's state at time τ . Note that the conformal control variable λ scales the cost of staying far away from predicted human states: if $\lambda = 0$ the the robot only cares about goal reaching; if $\lambda > 0$ then the robot is increasingly penalized for planning to intersect with predicted human

TABLE I: Stanford Drone Dataset: Quantitative Results. Results on the `nexus_4` scenario from SDD [1]. Robot’s goal is to cross the nexus while avoiding pedestrians. Safety was violated if the robot collided with a human. At all learning rates η , the conformal controller is more efficient at navigation than ACI in terms of time. It remains safe so long as the learning rate is set high enough so that the robot planner can quickly adapt to nearby humans; when the learning rate is set too low, (near zero), proximity to humans is effectively not penalized, leading to collisions.

		Metrics								
Method	η	success	time (s)	safe	min dist (m)	avg dist (m)	5% dist (m)	10% dist (m)	25% dist (m)	50% dist (m)
Aggressive	n/a	✓	8.567	✗	0.1595	4.058	1.253	1.546	2.495	4.021
ACI ($\alpha = 0.01$)	0	✓	27.17	✗	0.07612	5.201	1.842	2.415	3.9	5.614
	0.01	✓	26.67	✗	0.8026	4.575	2.261	3.014	3.507	4.574
	0.1	✓	24.73	✗	0.7906	4.771	2.284	2.825	3.561	4.78
Conformal Controller ($\varepsilon = 2m$)	50	✓	20.03	✗	0.6122	3.299	0.8688	1.426	2.022	2.978
	100	✓	17.4	✓	1.142	3.794	1.678	1.811	2.378	3.262
	500	✓	17.33	✓	1.116	3.989	1.69	1.812	2.452	3.795
	1000	✓	16.17	✓	1.265	3.599	1.698	1.81	2.282	3.303
Conservative	n/a	✗	∞	✓	2.268	6.291	3.801	3.982	4.982	5.993

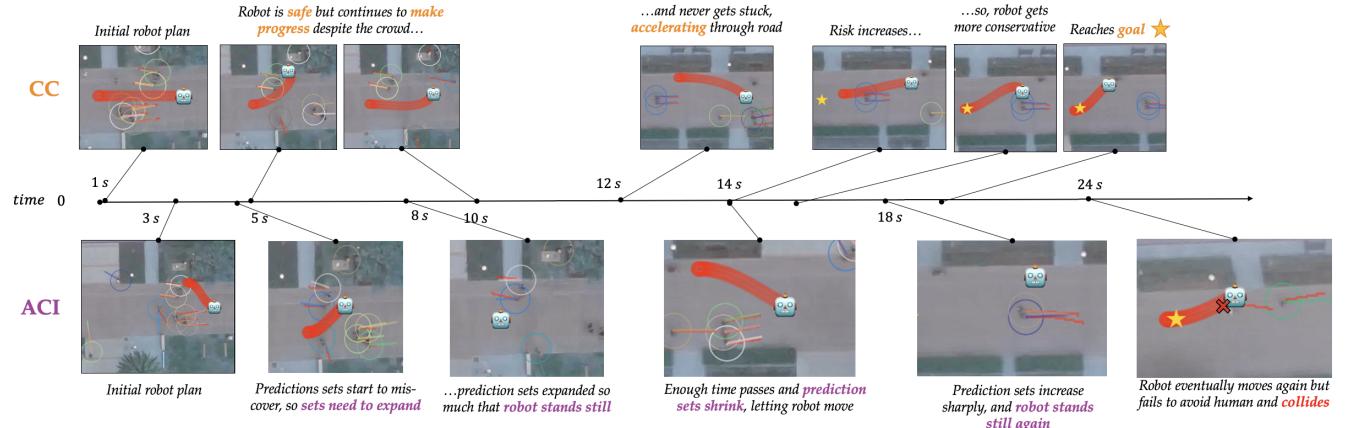


Fig. 3: Stanford Drone Dataset: Qualitative Results. Visualization of interaction over time (left to right). (top) With our conformal controller (CC), the robot always makes progress towards its goal while remaining safe, even when blocked by crowds of people. (bottom) The ACI baseline calibrates the prediction sets. As soon as a mis-prediction happens, ACI expands the prediction sets to obtain coverage, but this frequently blocks the robot from moving anywhere (see $t = 10s$) even though the mis-predictions happened for a pedestrian very far away that wasn’t interfering with the robot’s plan.

trajectories. The decision function outputs the minimum-cost trajectory for the robot

$$D_t^\lambda := \arg \min_{u_{t:t+H} \in \mathcal{U}} J(u_{t:t+H}; \mathcal{P}_t, \lambda), \quad (10)$$

where \mathcal{U} is the set of feasible splines (ones that are dynamically feasible for the robot and also do not intersect with environment obstacles). At the next timestep, the robot re-predicts the human trajectory (i.e., generates \mathcal{P}_{t+1}) and re-plans the decision D_{t+1}^λ .

Loss Function. Let $\mathcal{Y} \subset \mathbb{R}^2$ and the targets $y_t^1, \dots, y_t^M \in \mathcal{Y}$ be the actual xy positions of each of the M humans that the robot observes at time t . The loss function is defined as the negative distance to the nearest human,

$$\mathcal{L} := - \inf_{i \in [M]} \|y_t^i - u_t^{\text{pos}}\|_2, \quad (11)$$

where u_t^{pos} is the robot’s current position. To make it bounded, we clip the loss to the size of the video.

Metrics. We measure a boolean *safe* variable indicating if the robot never collided with a human. We also measure a

boolean *success* variable if the robot reached the goal location by the end of the interaction episode (i.e., length of video in the SDD). We also measure the time to reach the goal location and the minimum, mean, and $\{5\%, 10\%, 25\%, 50\%\}$ quantiles of the distance to the nearest human.

Experimental Setup. All methods are evaluated on interactions from the `nexus_4` video in the Stanford Drone Dataset (SDD) [1]. The risk threshold is $\varepsilon = 2m$ (i.e., radius around human). The robot always starts from the same initial condition and moves to the same goal. This scenario has a high density of pedestrians, making the risk-performance tradeoff for the robot nontrivial. Our approach (CC) adapts the reward weight λ_t on the human collision cost based on Equation 6 so that the decision risk is calibrated. Our baseline robot planners: **conservative** which always uses the safe decision function $D_t^{\lambda=1}$, **aggressive** which uses $D_t^{\lambda=0}$, and **ACI** [21] which first uses adaptive conformal prediction to calibrate prediction sets and then plans to avoid these sets.

Results. Quantitative results shown in Table I and qualita-

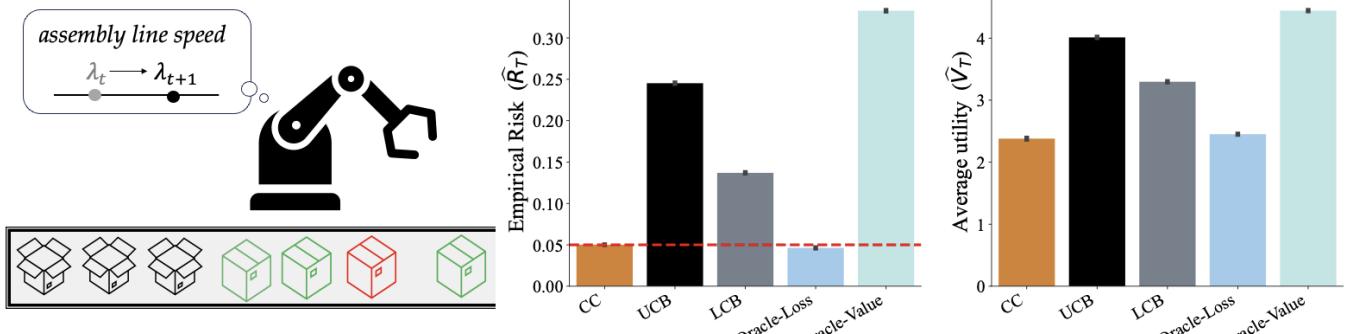


Fig. 4: **Manufacturing Assembly Line Robot: Quantitative Results.** (Left) Illustrative example: Robot must adjust the speed so that it grasps the most items while minimizing grasp failure. (Right) Empirical risk, \hat{R}_T , and average utility (i.e., successful grasps), \hat{V}_T on 1000 runs. Our method is denoted by (CC). Dashed red line is target risk $\varepsilon = 0.05$.

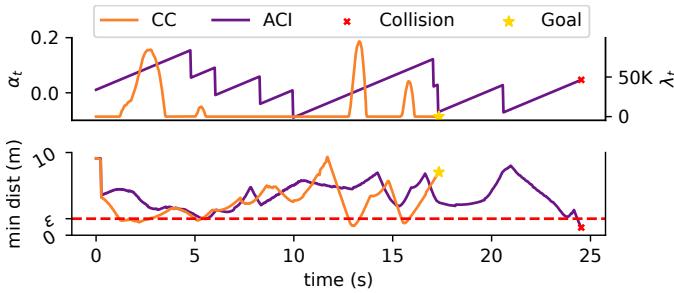


Fig. 5: **Stanford Drone Dataset.** (top) λ_t (calibrated by CC) and α_t (used by ACI to calibrate prediction sets) over time. When $\alpha_t \leq 0$, ACI returns infinite set and the robot stops. (Bottom) Distance to the nearest human over time. λ_t is large when the robot is close to a human, while α_t is unrelated. λ_t trajectory is shorter because it reaches the goal faster.

tive results in Figure 1. Because the **conformal controller** calibrates the robot’s decisions directly, it is substantially (~29%) faster at reaching the goal than the **ACI** algorithm (see visualization over time in Figure 3). While the **aggressive** baseline reaches the goal fastest, it consistently violates the safety threshold. On the other hand, the **conservative** baseline never completes the task, getting stuck far away from the crowds of pedestrians. The **conformal controller** ensures safety so long as the learning rate is fast enough for the robot planner to quickly adapt to changes in nearby human behavior (see Figure 5). Note that **ACI** can result in collisions in two reasons: 1) the prediction sets do not adapt fast enough for the spline planner to react and swerve out of the way of the pedestrian, 2) if the prediction sets become so large that there is no feasible spline and the robot must stand in place, the pedestrians sometimes run into the robot. This issue was independently observed in [21].

B. Manufacturing Assembly Line Robot

Consider a factory assembly line where a robot has to grab items from a conveyor belt (left, Figure 4). As the speed increases, the throughput of items increases but so does the ratio of robot grasp failures. The agent must calibrate the speed so that the ratio of failures over time stays below ε .

Decision Function & Parameterization. The agent will

directly modify the speed, thus the action $u_t := \lambda_t$. Here we take $\lambda_t \in [0, 1]$.

Risk Function. For a given conveyor belt speed λ , the robot will attempt to grab $n(\lambda)$ items, among which $d(\lambda)$ are failed grasps. The loss received by the robot will be $\mathcal{L}(\lambda) := d(\lambda)/n(\lambda)$.

Metrics. We measure average utility (i.e., # of successful grasps), $\hat{V}_T := \frac{1}{T} \sum_{t=1}^T V(\lambda_t)$, and empirical risk, $\hat{R}_T(\lambda_{1:T})$.

Experimental Setup. To simulate our setting, we assume that the number of items $n(\lambda)$ the robot attempts to grab is drawn as $\text{Pois}(C \cdot \sqrt{\lambda})$. The number of failed grasps conditioned on the total number of items is $d(\lambda)|n \sim \text{Bin}(n, C' \cdot \lambda)$. Importantly, the distributions of n, d , and the parameters C, C' are all *unknown to the agent*. Our conformal controller method (CC) adjusts the speed λ_t based on the update rule from Equation 6. In addition to the risk function, we also track a utility function which is the number of successful grasps $V(\lambda) := n(\lambda) - d(\lambda)$. We compare our method with two baselines: A bandit algorithm running the upper confidence bound algorithm (**UCB**) [33] to maximize the utility V and another algorithm running the lower confidence bound algorithm (**LCB**) to minimize the loss \mathcal{L} . We also add two methods with oracle access to the otherwise unknown parameters: **Oracle-Value** selects the best speed to maximize grasp success $\lambda_V^* := \arg \max_{\lambda} \mathbb{E}[V(\lambda)]$ and **Oracle-Loss** selects the best speed λ_L^* such that $\mathbb{E}[\mathcal{L}(\lambda_L^*)] := \varepsilon$. The values selected for the parameters are in Figure 4. We run all methods for a horizon $T = 2000$, set $C = 10$, $C' = 0.2$, and the target risk is $\varepsilon = 0.05$ (i.e., $\leq 5\%$ failed grasp).

Results. We run the simulation $N = 1000$ times, and calculate the average empirical risk and the average number of successful grasps. In Figure 4, we find that our method performs as well as the **Oracle-Loss**, ensuring that the empirical risk of grasps never exceeds $\varepsilon = 0.05$, while still ensuring high throughput of successfully grasped items. **UCB** and **LCB** both violate the empirical risk threshold: UCB incurs this risk but achieves higher number of successful grasps, while LCB is slow to learn its target, resulting in a higher risk over the time horizon.

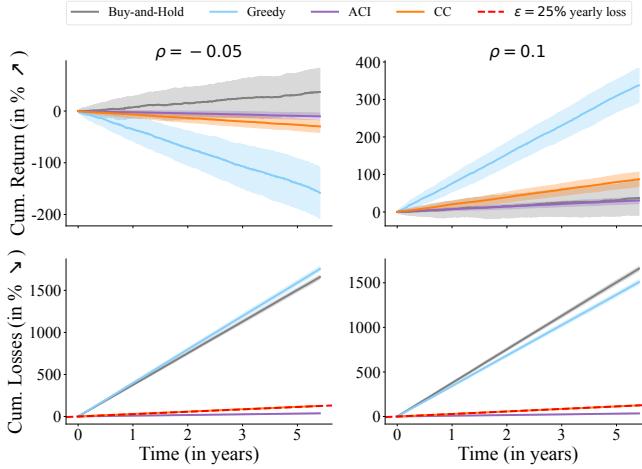


Fig. 6: Stock Trading: Quantitative Results. All results over 5 year period. The yearly loss threshold $\varepsilon = 25\%$. (left) Despite a poor prediction model of return (negative correlation), the CC achieves bounded loss at the user’s threshold (bottom, dashed red line overlaps with orange CC line) but is not the best at keeping the return the highest. (right) With a strong prediction model on the return (positive correlation), the CC is able to achieve high yearly returns (second only to Greedy) while simultaneously respecting the loss threshold (which the Greedy violates).

C. Stock Trading Agent

An automated trading agent is trading a stock at high-frequency (e.g every five minutes). We model the agent as able to either buy or short-sell the stock, with no trading cost. When buying the stock at time t , the agent receives return r_t . When short-selling it, the agent receives a return $-r_t$. The agent must calibrate its trading decisions so the annualized loss is at or beneath the investor’s loss threshold of $\varepsilon\%$.

Decision Function & Parameterization. At every timestep, t , the agent has access to the past history of returns and it’s own actions. The agent can use it to construct a confidence set \hat{C}_λ where λ is the conformal control variable. Given a predicted set, the agent can decide to either buy if the entire set is above 0, short-sell if the entire set is below 0, and not do anything if 0 is in the set:

$$D_t^\lambda := \begin{cases} 1 & \text{if } \min(\hat{C}_\lambda) > 0 \\ -1 & \text{if } \max(\hat{C}_\lambda) < 0 \\ 0 & \text{o.w.} \end{cases} \quad (12)$$

Risk Function. The agent’s action is $u \in \{-1, 0, 1\}$ which incurs a loss $\mathcal{L}(u, r) := -u \cdot r \cdot \mathbf{1}\{u \cdot r < 0\}$, i.e., the agent suffers a loss equal to the amount of money lost by that decision. We clip the loss to make it bounded.

Experimental Setup. We simulate stock returns using a geometric Brownian motion. We assume that we observe returns every hour, so we have $n = 252(\text{days}) \times 7(\text{hours per day})$ steps per year

$$r_t := \mu\Delta + \sigma\sqrt{\Delta}Z_t \quad \text{where } \Delta = 1/n \quad (13)$$

We assume that at $t - 1$, the agent has access to a prediction \hat{r}_t and we assume that the correlation $\text{corr}(r_t, \hat{r}_t) := \rho$. The higher ρ the better the predicted returns \hat{r}_t . The predicted interval is

$$\hat{C}_\lambda(\hat{r}_t) := [\hat{r}_t - \sigma\sqrt{\Delta}z_{\lambda/2}, \hat{r}_t + \sigma\sqrt{\Delta}z_{1-\lambda/2}],$$

where z_λ is the quantile of level λ of the normal distribution.

Metrics. In addition to the loss, we also measure return $V(u, r) := u \cdot r$ which for the agent’s action u .

Results. We run $N = 100$ simulations over 5 years. We set $\mu = 0.08, \sigma = 0.2$, which are approximately the historical values for the S&P 500. We compare our **CC** method with: the **Buy-and-Hold** strategy that simply buys the stock at each timestep, the **Greedy** strategy that buys the stock whenever the prediction is above 0 and short-sells it when the prediction is below 0 (equivalent to $D(\lambda = 1)$), and **ACI** that adjusts λ online using the ACI algorithm. We set the target coverage for ACI at 90% and our annualized loss threshold to be less than $\varepsilon = 25\%$ (the threshold per time-step is therefore ε/n). For the prediction of returns, we simulate another geometric Brownian motion,

$$\hat{r}_t := \mu\Delta + \sigma\sqrt{\Delta}W_t \quad \text{where } \text{corr}(W_t, Z_t) = \rho. \quad (14)$$

The results for the different methods are in Figure 6. We plot the cumulative return and cumulative loss for all methods and for two models: $\rho = 0.1$ (good model) and $\rho = -0.05$ (bad model). In both cases, our **CC** quickly adapts the parameter to stay below the loss threshold, while having good returns when the predictive model is good ($\rho = 0.1$). **Greedy** approaches has more extreme returns (negative when the model is bad, positive when the model is good) with high level of losses. **ACI** is highly conservative, resulting in smaller loss, way below the threshold. By being too conservative, it limits the potential gain when the predictive model is actually good. **Buy-and-hold** also has high cumulative loss as it moves with the stock, and has a more consistent return, as it is independent of the model quality.

V. DISCUSSION & CONCLUSION

In this paper, we introduce *Conformal Decision Theory*, a new theoretical and algorithmic framework for producing safe autonomous decisions despite imperfect machine learning predictions. Our method is described in the online, adversarial setting, but in other decision-making domains, you may have batch or offline datasets from which you learn how to make decisions (e.g., offline RL [34]). Thus, it may be worth asking how these ideas might apply in the batch setting. Future work may additionally consider optimizing the conformal control variable to maximize utility subject to the constraint of risk control.

REFERENCES

- [1] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII* 14. Springer, 2016, pp. 549–565.

- [2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.
- [3] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-rnn: Deep learning on spatio-temporal graphs,” in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.
- [4] A. Vemula, K. Muelling, and J. Oh, “Social attention: Modeling attention in human crowds,” in *2018 IEEE international Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4601–4607.
- [5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.
- [6] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv preprint arXiv:1703.09312*, 2017.
- [7] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, and N. Zhang, “Graph-based stock correlation and prediction for high-frequency trading systems,” *Pattern Recognition*, vol. 122, p. 108209, 2022.
- [8] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
- [9] J. Lampinen and A. Vehtari, “Bayesian approach for neural networks—review and case studies,” *Neural networks*, vol. 14, no. 3, pp. 257–274, 2001.
- [10] E. Goan and C. Fookees, “Bayesian neural networks: An introduction and survey,” *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pp. 45–87, 2020.
- [11] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Advances in neural information processing systems*, vol. 30, 2017.
- [12] R. Koenker and G. Bassett Jr, “Regression quantiles,” *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.
- [13] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005, vol. 29.
- [14] V. Vovk and C. Bendtsen, “Conformal predictive decision making,” in *Conformal and Probabilistic Prediction and Applications*. PMLR, 2018, pp. 52–62.
- [15] I. Gibbs and E. Candes, “Adaptive Conformal Inference Under Distribution Shift,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 1660–1672. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html>
- [16] M. Zaffran, O. Féron, Y. Goude, J. Josse, and A. Dieuleveut, “Adaptive conformal predictions for time series,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 834–25 866.
- [17] A. N. Angelopoulos and S. Bates, “A gentle introduction to conformal prediction and distribution-free uncertainty quantification,” *arXiv preprint arXiv:2107.07511*, 2021.
- [18] S. Feldman, L. Ringel, S. Bates, and Y. Romano, “Achieving risk control in online learning settings,” 2023.
- [19] Y. Chen, U. Rosolia, C. Fan, A. Ames, and R. Murray, “Reactive motion planning with probabilistic safety guarantees,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1958–1970.
- [20] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, “Safe planning in dynamic environments using conformal prediction,” *IEEE Robotics and Automation Letters*, 2023.
- [21] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, “Adaptive conformal prediction for motion planning among dynamic agents,” in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 300–314.
- [22] R. Luo, S. Zhao, J. Kuck, B. Ivanovic, S. Savarese, E. Schmerling, and M. Pavone, “Sample-efficient safety assurances using conformal prediction,” in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2022, pp. 149–169.
- [23] F. Cai and X. Koutsoukos, “Real-time out-of-distribution detection in learning-enabled cyber-physical systems,” in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2020, pp. 174–183.
- [24] R. Sinha, E. Schmerling, and M. Pavone, “Closing the loop on runtime monitors with fallback-safe mpc,” *Conference on Decision and Control*, 2023.
- [25] H. Yang and M. Pavone, “Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8947–8958.
- [26] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley *et al.*, “Robots that ask for help: Uncertainty alignment for large language model planners,” *Conference on Robot Learning*, 2023.
- [27] O. Bastani, V. Gupta, C. Jung, G. Noarov, R. Ramalingam, and A. Roth, “Practical adversarial multivalid conformal prediction,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 362–29 373, 2022.
- [28] A. N. Angelopoulos, E. J. Candès, and R. J. Tibshirani, “Conformal pid control for time series prediction,” *arXiv preprint arXiv:2307.16895*, 2023.
- [29] A. P. Dawid, “The well-calibrated bayesian,” *Journal of the American Statistical Association*, vol. 77, no. 379, pp. 605–610, 1982.
- [30] A. Bhatnagar, H. Wang, C. Xiong, and Y. Bai, “Improved online conformal prediction via strongly adaptive online learning,” *arXiv preprint arXiv:2302.07869*, 2023.
- [31] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, “Optimal trajectory generation for car-type mobile robot using spline interpolation,” *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.
- [32] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.
- [33] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.
- [34] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline reinforcement learning: Tutorial, review, and perspectives on open problems,” *arXiv preprint arXiv:2005.01643*, 2020.