Risk-Calibrated Human-Robot Interaction via Set-Valued Intent Prediction

Justin Lidard, Hang Pham, Ariel Bachman, Bryan Boateng, Anirudha Majumdar
Department of Mechanical and Aerospace Engineering
Princeton University, Princeton, New Jersey 08540
Email: jlidard@princeton.edu

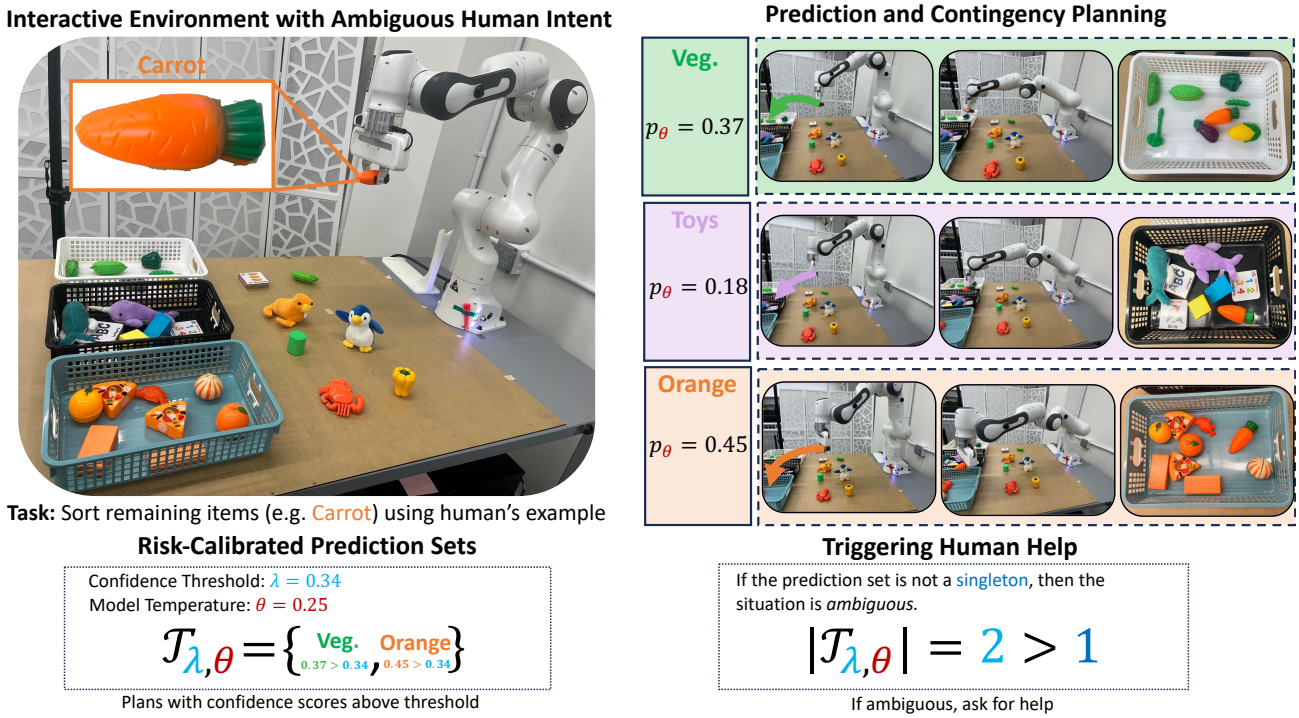

Fig. 1: Risk-Calibrated Interactive Planning (RCIP) statistically calibrates risk for human-robot interaction. Given a set of possible human intents and confidence scores, a planner generates a weighted set of actions. The set of actions from each plan are collected in a set according to a threshold on the predicted intents. If there is more than one action in the set, the robot asks for help.

***Abstract*— Tasks where robots must cooperate with humans, such as navigating around a cluttered home or sorting everyday items, are challenging because they exhibit a wide range of valid actions that lead to similar outcomes. Moreover, *zero-shot* cooperation between human-robot partners is an especially challenging problem because it requires the robot to infer and adapt *on the fly* to a latent human intent, which could vary significantly from human to human. Recently, deep learned motion prediction models have shown promising results in predicting human intent but are prone to being *confidently incorrect*. In this work, we present Risk-Calibrated Interactive Planning (RCIP), which is a framework for measuring and calibrating risk associated with uncertain action selection in human-robot cooperation, with the fundamental idea that the robot should ask for human clarification when unertainty in the human's intent may adversely affect task performance. RCIP builds on the theory of *set-valued risk calibration* to provide a finite-sample statistical guarantee on the cumulative loss incurred by the robot while minimizing the cost of human clarification in complex multi-step settings. Our main insight is to frame the risk control problem as a *sequence-level* multi-hypothesis testing problem, allowing efficient calibration using a low-dimensional parameter that controls a pre-trained risk-aware policy. Experiments across a variety of simulated and real-world environments demonstrate RCIP's ability to predict and adapt to a diverse set of dynamic human intents.**[1]

I. Introduction

Predicting and understanding human intent is a critical task for robotics, specifically for safe interaction with humans in cluttered, close-quarters environments. However, human intent prediction is challenging because no two humans may have the same preferences, and intents may differ depending on the specific environment. As an example, a robot is tasked with sorting items into three bins based on an example provided by the human (see Fig. 1). While the bins have a ground-truth sorting criterion known by the human (vegetables, children's toys, and miscellaneous orange items), the robot must infer the human's intent in order to sort new items. Given the provided context, the robot should be able to sort some unambiguous items (e.g. the crab) autonomously, while other items (e.g. the carrot) may be placed into multiple bins, resulting in *situational ambiguity*. If asked to

[1]Website with additional information, videos, and code: https://risk-calibrated-planning.github.io/

operate fully autonomously, the robot must take a *risk* and guess the correct placement for the carrot. However, the robot may also *ask for help* if it is unsure, guaranteeing the correct action but potentially burdening the human. In this work, we study the tradeoff between risk and autonomy governing optimal action selection in the face of situational ambiguity.

Recently, calibrated predict-then-plan (also known as contingency planning) [1, 2] approaches have demonstrated the ability to generate provably safe plans by first using confidence-aware prediction models to generate a set of possible futures and then constructing a safe plan that accommodates for the future uncertainty. These approaches enable synthesis of large amounts of scene-specific context (such as image or map information) while simultaneously providing a guarantee on the plan success rate by calibrating the coverage of the prediction. However, one of the major challenges of predict-then-plan approaches comes from *multi-modal human behavior*: if the distribution of human actions contains multiple high-level behaviors, a single robot plan may become overly conservative in trying to accommodate all possible human intents. Moreover, environments themselves may generate additional sources of ambiguity that may result in unsafe behavior from the robot if misinterpreted. In such cases, if possible, the robot should ask for help in order to clarify the human's intent instead of committing to a potentially unsafe action.

Our approach utilizes deep-learned human intent prediction models (e.g. [3, 4]) for understanding interactivity, and rigorously quantifies the uncertainty of these models in order to decide when to ask for help. As shown in Fig. 1 (middle), we produce a limited set of human intents based on the prediction model's confidence scores. For each predicted intent, we plan a sequence of actions that satisfy an environment objective, such as placing the item in the correct bin. To accommodate different levels of robot autonomy, we assume that the predictor has a small number of highly flexible hyperparameters (such as the temperature), which allow the end-user to specify high-level behaviors (more or less confident predictions). We use a small calibration dataset of human-robot interactions to choose a set of valid hyperparameters that provide a level of risk and autonomy set in advance by the user. By leveraging recent advances in distribution-free risk control [5], we show that the robot's behavior can simultaneously limit several notions of risk. We formalize this challenge via two objectives: (i) *statistical risk calibration*: the robot should seek sufficient help from the human when necessary to ensure a statistically guaranteed level risk specified by the user, and (ii) *flexible autonomy*: the robot should ask for a minimal amount of help as specified by the user by by narrowing down situational ambiguities through planning. We refer to these simultaneous objectives, with help from the human when necessary, as Risk-Calibrated Interactive Planning (RCIP).

Statement of contributions. In this work, we introduce RCIP, a framework for measuring and calibrating risk in situations that involve interactions with humans with potentially ambiguous action choices. By reasoning about the human's desired task outcome in the space of *intents*, we efficiently plan safe actions in the face of diverse, multi-modal human behavior, and ask for help when necessary. We make the following contributions: **(1)** We demonstrate how to use statistical risk control (SCR) to control the planning error rate across a set of model hyper-parameters, allowing flexible but provably safe levels of autonomy. **(2)** We prove theoretical guarantees for multi-dimensional risk control for both single-step and multi-step planning problems: with a set of user-specified risk budgets $(\alpha_1,...,\alpha_K)$ for different measures of risk (e.g., probability of failure and probability that the robot asks for help) and the robot performs the task correctly (with high probability) by asking for help if any of the K risk budgets will be violated. **(3)** We evaluate RCIP in both simulation and hardware with a suite of human-robot interactive planning tasks with various styles of situational ambiguity (spatial, contextual, semantic). Experiments across multiple platforms and human uncertainty showcase the ability of RCIP to provide statistically guaranteed task success rates while providing more flexible autonomy levels than baseline approaches. RCIP reduces the amount of human help by $5-30\%$ versus baseline approaches.

II. Related Work

RCIP brings together techniques from contingency planning, human intent prediction, and conformal prediction and empirical risk control.

A. Contingency Planning and Priviledged Learning

Contingency planning [6] is a growing literature on planning for multi-agent interactive scenarios where future outcomes are diverse. Recent approaches [7]–[10] typically favor a predict-then-plan approach, wherein multi-modal motion predictions are first generated and then used to produce a set of safe plans conditioned on each prediction. The authors of [11] formulate a multi-agent contingency planning problem as a generalized Nash equilibrium problem, thereby assuming that agents are non-cooperative. In this work, we assume that the human and robot act in good faith (i.e., they are cooperative). Similar to contingency planning is the *learning under privileged information* paradigm [12]–[14], which provides the learning algorithm with additional information during training to help bootstrap near-optimal behaviors. Privileged learning has shown empirical success in semantic reasoning [15], vision-based robotic manipulation [16], and learning policies that can be deployed in the wild [17]–[19]. In [20], privileged information about the human's trajectory is used to train a policy that most efficiently apprehends a human opponent, and a partially-observed deployment policy is distilled using a teacher-student paradigm. Similarly, in [21], a visuomotor policy for social navigation is trained by using exact pedestrian positions during training, and a model for estimating for the position embedding is distilled from the privileged embedding.

In this work, we provide the robot with additional information about the internal state of the "human" during the planning phase. We eliminate the need for a separate distillation procedure by instead using a set-valued prediction strategy, introduced in the following sections. We use

contingency planning and privileged learning to find (or learn) optimal intent-conditioned policies, which can then be used to predict an optimal action via an upstream predictor. By allowing the robot to ask for help when it is uncertain, we statistically quantify risk associated with the robot acting optimally, even when it is uncertain.

B. Human Intent Prediction

Predicting intent of humans for downstream planning has been widely applied in autonomous driving [22]–[24], social navigation [4, 25, 26], and game theory [27]. Several works [23, 28] use a discrete latent variable to capture qualitative behaviors in human motion. To aid in human goal satisfaction, the authors of [29] show that human actions can be predicted directly in interactive settings, but the prediction model must be retrained whenever the task or human partner changes. Conversely, in our work we leverage recent advances in vision language models (VLMs) [30] for their ability to condition on internet-scale data to predict intuitive human motions in a variety of tasks. In this work, we use intent prediction *to bound directly the risk associated with downstream planning*. We use *set-valued* prediction to compute a set of possible intents, from which a planning module can compute a conditional plan.

C. Conformal Prediction and Empirical Risk Control

Conformal prediction [31]–[33] has recently gained popularity in a variety of machine learning and robotics applications due to its ability to rigorously quantify and calibrate uncertainty. A recent line of works [34]–[36] has extended the theory from prediction of labels (e.g. actions) to sequences (e.g. trajectories). Several works [37, 38] have studied *adaptive* conformal prediction, wherein a robot's predictive conservativeness is dynamically adjusted within a policy rollout by assuming that there always exists a conservative fallback policy. Finally, some recent works [39, 40] have extended conformal prediction theory to handle more general notions of risk. Our work differs in three key ways: (i) we provide a separate calibration stage in which the robot can adjust its parameterization of prediction sets through a modest-size dataset of interactive scenarios, reducing the number of "unrecoverable" scenarios in which the robot exceeds its risk budget early on in a rollout, and (ii) we provide a way to synthesize from scratch risk-averse control policies, and (iii) we reason about human uncertainty in the space of *intents*, permitting a more natural way to capture diverse interactive behaviors than other representations (e.g. trajectories).

III. Problem Formulation

In this section, we pose the problem of human-robot cooperation with intent uncertainty as a partially observable Markov decision process (POMDP). We present a brief overview of the prediction-to-action pipeline and our goals of risk specification and flexible autonomy.

A. Dynamic Programming with Intent Uncertainty

Environment Dynamics. We consider an interaction between a robot R and human H in environment e, governed by a nonlinear dynamical system with time horizon T:

$$x_{t+1} = f_e(x_t, u_t) \quad \forall t \in [T], \tag{1}$$

where $x_t \in \mathcal{S} \subseteq \mathbb{R}^n$ is the joint state of the system and $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the joint (robot-human) control input (u_t^R, u_t^H). We use a superscript for individual agent indexing, and we use bar notation to denote aggregation over time, e.g. $\bar{x}_t = (x_1, \ldots, x_t)$. Let $\pi = (\pi^R, \pi^H)$ denote the joint control policy governing system (1). We permit the human's action to be drawn from a potentially multi-modal distribution π^H.

We present a methodology for learning a policy set Π^R that selects a set of optimal actions contingent on the uncertainty in π^H.

Intent Dynamics. We assume that the human's (potentially unknown) policy π^H is parameterized by a discrete latent variable with the following dynamics:

$$z_{t+1} \sim q(\cdot | x_t, z_t), \tag{2}$$

where $z_t \in \mathcal{Z} = [N]$ characterizes the human's intent at time t, and N is the number of high-level human behaviors. We assume that conditioned on the human's latent intent (which may be stochastic), each agent's action is conditionally deterministic, i.e., $u_t^i = \pi^i(x_t, z_t)$, for $i \in \{R, H\}$.

Planning Objective. Each agent $i \in \{R, H\}$ has the goal to minimize their corresponding cost function J^i in finite-horizon T with running cost l^i. The cumulative cost of a policy π^i starting from initial state x and a *known* human intent z is

$$J^i(x, z, \pi^i) = \mathbb{E}^{\pi} \left[\sum_{t=1}^{T} l^i(x_t, u_t) \middle| x_1 = x, z_1 = z \right]. \tag{3}$$

The objective of agent i is to find a policy π^i that minimizes eqn. (3). To ensure the safety of the human, we add an additional set of inequality constraints h^i that depend on the (time-varying) intent of the human:

$$\begin{aligned} \min_{\pi^i} \quad & J^i(x, z, \pi^i) \\ \text{s.t.} \quad & h^i(x_t, z_t) \leq 0 \quad \forall t \in [T] \\ & (x_1, z_1) = (x, z). \end{aligned} \tag{4}$$

Conditional Action Selection. For each intent z, the value function associated with the latent intent z can be evaluated as a function of the state-intent pair (x, z). That is,

$$V^i(x, z) = \inf_{\pi^i} \left\{ J^i(x, z, \pi^i) \right\}. \tag{5}$$

The Bellman optimality principle states that the optimal policy satisfies Bellman's equation:

$$V^i(x, z) = \inf_{\pi^i} \left\{ \mathbb{E}^{\pi, q} \left[l^i(x, u) + V^i(f_e(x, u), z_{t+1}) \right] \right\}, \tag{6}$$

where z_{t+1} is sampled according to eqn. (2). The action-value function is similarly defined as

$$Q^i(x, u^i, z) = l^i(x, u^i, u^{\neg i}) + \mathbb{E}^{\pi, q}[V^i(x_{t+1}, z_{t+1})], \tag{7}$$

where (x_{t+1}, z_{t+1}) are the next state-intent pair under the state dynamics (1) and intent dynamics (2), $\neg i$ is the other agent, and $u^{\neg i} = \pi^{\neg i}(x, z)$. Eqn. (7) states that since both

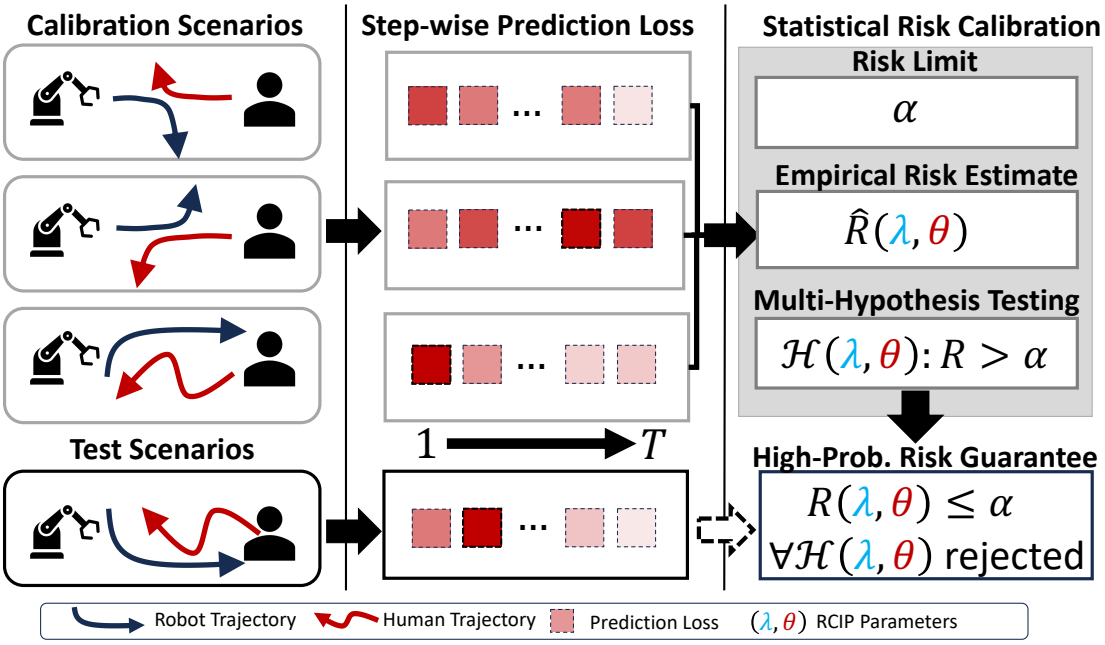

Fig. 2: RCIP formulates interactive planning as a multi-hypothesis risk control problem. Using a small set of calibration scenarios, RCIP computes step-wise prediction losses to form an aggregate emperical risk estimate. Using a risk limit, for each pair (λ,θ) of prediction thresholds and model hyperparameters, RCIP evaluates the hypothesis that the test set risk is above the limit. Thus, for all hypotheses that are rejected, the test set risk satisfies the threshold (with high probability).

agents' policies are conditionally deterministic, the optimal cost-to-go is a deterministic function of the state-intent pair (x, z). If the latent intent z were known, then the expected cost-to-go is a deterministic function of the current observation x, intent z, and the joint control u. Without knowledge of z, the state of the system would not be fully observable, and the cost-to-go would vary due to uncertain action selection at the current state. The optimal action for both agents is given by the minimizer of Eqn. (7),

$$u^{i*}(z) = \underset{u}{\operatorname{argmin}}\, Q^i(x,u,z) \quad \forall i \in \{R,H\}. \tag{8}$$

Intent Prediction. During deployment, the true human intent z is not observed. However, we assume access to a model g_θ with hyperparameters θ that *predicts* human intent based on a sequence $\bar{x}_t$ of prior states. Specifically, let $g_\theta(\bar{x}_t,z)$ produce a confidence score in $[0,1]$ that estimates the probability of each possible intent z in $\mathcal{Z}$. In general, g_θ will output heuristic and *uncalibrated* confidence scores, and need not be trained on the same distribution q.

B. Risk-Calibrated Interactive Planning

Predicted Action Set. We aggregate confidence scores from g_θ into a set $\mathcal{S}_{\lambda,\theta} \subseteq \mathcal{Z}$ of predicted intents via the rule

$$\mathcal{S}_{\lambda,\theta}(\bar{x}_t) = \{z \in \mathcal{Z} : g_\theta(\bar{x}_t,z) \geq \lambda\}, \tag{9}$$

where λ is a confidence threshold (cf. Section IV). Since human intent uncertainty alone may not alter the optimal robot plan, we compute a set of predicted actions from the set of predicted intents as

$$\mathcal{T}_{\lambda,\theta}(\bar{x}_t) = \{u \in \mathcal{U} : \exists z \in \mathcal{S}_{\lambda,\theta} \text{ s.t. } u = u^*(z) \text{ and } g^*_\theta(\bar{x}_t,z) \geq \lambda\}, \tag{10}$$

where we define g^*_θ as the sum of all intent-based confidence scores that lead to the same action, i.e., $g^*_\theta(\bar{x}_t, z) := \sum_{z' \in \mathcal{Z}} g_\theta(\bar{x}_t, z') \mathbb{1}\{u^{R*}(z) = u^{R*}(z')\}$. To simplify notation, we define $g^*_\theta(z) := g^*_\theta(\bar{x}_t,z)$.

Policy Deployment. We now define our overall robot policy Π^R. Given the predicted action set $\mathcal{T}_{\lambda,\theta}(\bar{x}_t)$ defined in Eqn. (10), the robot has two behaviors:

1) **Autonomy.** If $\mathcal{T}_{\lambda,\theta}(\bar{x}_t)$ is a singleton, then the robot is confident in the predicted action, and the action is executed.
2) **Triggering Help.** If $\mathcal{T}_{\lambda,\theta}(\bar{x}_t)$ is not a singleton, then the robot triggers human help, and the human reveals their true intent, z^*. The robot executes the action $u^{R*}(z^*)$.

If λ and θ are chosen such that $\mathcal{T}_{\lambda,\theta}(\bar{x}_t)$ is empty, the task is failed.

C. Goal: Certifiable Autonomy

Situational ambiguity results in many potentially correct robot actions arising with potentially no safe external resolution, save for direct human intervention (see e.g. Fig. 1). Our goal in this work is to address certifiable autonomy: selecting a set of model hyper-parameters (θ,λ) that achieves multiple user-specified levels of risk. As shown in Fig. 2, we formalize this problem by considering a joint distribution $\mathcal{D}$ over scenarios $\xi := (e,q,l)$, where e is an environment (a POMDP) with dynamics f_e, q is a stochastic function describing the human's intent and partially observed through f, and l is

a cost function encoding the robot's goal, which is assumed to be known *a-priori* by both the human and robot. We do not assume knowledge of $\mathcal{D}$ except for the availability of a modestly-sized calibration dataset $\mathcal{C}$ containing 500 samples from $\mathcal{D}$. We formalize certifiable autonomy in our context as (i) risk calibration: the robot must meet a set of user-specified risk levels $(R_1,...,R_K)$ with user-specified probability over new scenarios $\xi \sim \mathcal{D}$, and (ii) flexible autonomy: the policy Π^R should return a set of model hyper-parameters that control each risk but allow different high-level behaviors.

IV. Approach

In this section we present a procedure for guaranteeing optimal action selection while controlling a user-specified notion of risk. We introduce statistical risk calibration below, then present the different practical settings we consider (single-step, multi-step, and multi-risk).

A. Background: Statistical Risk Calibration

What is a risk? We now present an approach for controlling the *risk* of the robot's multi-modal policy Π^R by calibrating when the robot should ask for help at inference time according to a user-specified notion of risk. Our approach builds on the Learn-then-Test framework for distribution-free statistical risk control [5]. Let $\mathcal{D}$ be an unknown distribution over i.i.d. scenarios such that $\xi \sim \mathcal{D}$. If we fix a policy for the human and the robot and assume that the robot has ground-truth knowledge of z, then the distribution over scenarios induces a distribution over the context-label pairs $(\bar{x},z)$, where the context contains a history of the previous states up to and including the current state at time t.

Assume that we are given a risk signal R that we wish to control, where $R \in [0,1]$ measures an expected loss as a function of the prediction threshold λ and model hyperparameters θ. Here, we let $\phi := (\lambda,\theta) \in \Phi$ be the pair of prediction parameters we wish to optimize, where $\Phi = \Lambda \times \Theta$. For fixed parameters ϕ, the expected loss is itself a function of the context, prediction sets, and true labels over the unknown distribution $\mathcal{D}$, i.e.

$$R(\phi) = \mathbb{E}^{(\bar{x},z)\sim\mathcal{D}}\Big[L(\bar{x},\mathcal{T}_\phi(\bar{x}),z)\Big], \tag{11}$$

and the loss L is similarly defined on $[0,1]$.

Bounding the Probability of Suboptimal Actions. As an example, L could be miscoverage, i.e., $L = 0$ if the optimal action is in the prediction set, and $L = 1$ otherwise. In expectation over $\mathcal{D}$, the risk associated with suboptimal action selection is the *miscoverage risk*, i.e

$$R_{\text{cov}}(\phi) = \mathbb{P}^{(\bar{x},z)\sim\mathcal{D}}\big(u^{R*}(z^*) \notin \mathcal{T}_\phi(\bar{x})\big). \tag{12}$$

Remark. *Equation (12) is identical to the typical conformal prediction (CP) setting [41, 42] in which the risk is miscoverage of the true label in the prediction set. However, the standard CP framework only allows one to choose λ to bound the miscoverage rate. In contrast, the formulation we consider permits the modification of other model parameters θ (e.g., the temperature of the softmax outputs) in order to have more fine-grained control of the prediction sets and bound risks beyond miscoverage. We will demonstrate the benefits of this flexibility empirically in later sections.*

Calibrating the Predicted Action Set. We will assume access to a calibration set $\mathcal{C} = \{(\bar{x}_i,z_i)\}_{i=1}^M$ of i.i.d. random variables drawn from $\mathcal{D}$, which we will use to estimate the risks. We seek to take the (uncalibrated) prediction model $g_\theta^* : \mathcal{X}^t \times \mathcal{Z} \to [0,1]$ that produces softmax scores for each intent-conditioned action $u^{R*}(z)$. As described in Sec. III-B, we post-process the raw model outputs in $[0,1]$ to generate a prediction set $\mathcal{T}_{\lambda,\theta}(\bar{x})$ containing actions; this set is parameterized by a low-dimensional set of parameters $\lambda \in \Lambda$ and $\theta \in \Theta$, where Λ is a finite set of prediction thresholds values one wishes to test, and Θ is a finite set of model hyperparameters, such as temperature. Then, we use the calibration set in order to choose the parameter pair (λ,θ) to control a user-specified risk, regardless of the quality of the predictor g_θ^*.

Ahead of calibration, we set a desired risk threshold α. Our goal is to identify a set $\hat{\Phi} \subseteq \Phi$ such that for any $\phi \in \hat{\Phi}$, $R(\phi) \leq \alpha$ with some user-defined probability δ. In particular, the probability δ is with respect to the randomness in sampling over the calibration dataset $\mathcal{C}$, which itself is randomly sampled from the unknown distribution $\mathcal{D}$.

Multi-Hypothesis Testing for a Single Risk. Since the prediction set $\mathcal{T}_\phi$ is controlled by low-dimensional hyperparamters ϕ drawn from the set $\hat{\Phi}$, controlling a single risk is a multiple-hypothesis testing problem [5]. For each $j \in \{1,...,|\Phi|\}$, we consider the hypothesis $\mathcal{H}^j$ such that the risk $R(\phi^j)$ is not controlled, where $\phi^j \in \Phi$. Therefore, rejecting $\mathcal{H}^j$ is equivalent to certifying that the risk is controlled. For a calibration set size M, define the empirical risk estimate on the calibration set:

$$\hat{R}^j = M^{-1}\sum_{i=1}^{M} L(\bar{x}_i,\mathcal{T}_{\phi^j}(\bar{x}_i),z_i). \tag{13}$$

Using $\hat{R}^j$, the Hoeffding-Bentkus inequality [39] gives the jth p-value as

$$p^j = \min\Big(\exp\big(-Mh_1(\max(\hat{R}^j,\alpha),\alpha)\big), e\hat{\Phi}^{\text{Bin}}_{\alpha,n}(\lceil n\hat{R}^j\rceil)\Big), \tag{14}$$

where $h_1(a,b) = a\log(a/b) + (1-a)\log((1-a)/(1-b))$ and $\hat{\Phi}^{\text{Bin}}_{\alpha,n}(y)$ is the cumulative distribution function of the binomial distribution with parameter α and number of trials n.

We now have left to construct our set $\hat{\Phi}$ of low-dimensional parameters ϕ^j that reject $\mathcal{H}^j$ and control the risk R. Bounding (11) for all $\phi \in \hat{\Phi}$ requires that the p-values hold simultaneously; any nontrivial subset $\hat{\Phi} \subseteq \Phi$ that controls the risk is said to control the family-wise error rate (FWER). A simple but powerful approach, which we use in the following analysis, is to apply a union bound over a coarse grid $\mathcal{J}$ of initializations (e.g. each item in $\mathcal{J}$ is an equally spaced grid of indices of Φ) in an iterative procedure called *fixed-sequence testing* [5, 43]. In fixed-sequence testing, for each $j \in \mathcal{J}$, the set $\hat{\Phi}$ of valid prediction thresholds is

initialized as the empty set and grown according to the rule

$$\hat{\Phi} \leftarrow \begin{cases} \hat{\Phi} \cup \{\phi^l\} & \phi^l \notin \hat{\Phi} \text{ and } p^l \leq \delta/|\mathcal{J}|, \quad l \geq j \quad j \in [\mathcal{J}] \\ \hat{\Phi} & o.w. \end{cases} \tag{15}$$

That is, parameters ϕ^j are only added to $\hat{\Phi}$ if $\mathcal{H}^j$ is rejected, eliminating the need for a union bound over a large set of parameters. The set of parameters that satisfy the risk bound is given by

$$\hat{\Phi} := \{\phi^j : p^j \leq \delta/|\mathcal{J}|\}. \tag{16}$$

Thus,

$$\mathbb{P}^{(\bar{x},z)\sim D^M}\left(\sup_{\phi \in \hat{\Phi}}\{R(\phi)\} \leq \alpha\right) \geq 1-\delta, \tag{17}$$

where the supremum over the empty set is defined as $-\infty$. The calibration procedure thus yields $\hat{\Phi}$, which is a set of values ϕ^j that each control the risk $R(\phi^j)$ to the desired level α (with probability $1-\delta$ over the randomness in the calibration dataset).

B. Single-Step, Single-Risk Control

We now state our first proposition, which bounds the action miscoverage rate for single-step settings.

Proposition 1. *Consider a single-step setting ($T = 1$) where we use risk calibration parameters $\phi \in \hat{\Phi}$ to generate predicted action sets and seek help whenever the prediction set is not a singleton (cf. Sec. III-B). If the FWER-controlling parameter set $\hat{\Phi}$ is non-empty, then with probability $1-\delta$ over the sampling of the calibration set, the new scenarios drawn from $\mathcal{D}$ incur at most α_1 rate of optimal action miscoverage.*

Proof. The proof follows immediately from application of fixed-sequence testing to the p-values obtained from the Hoeffding-Bentkus inequality, as given in [5], and is identical to the conformal prediction setting [36]. □

C. Single-Step, Multi-Risk Control

We now introduce two key risks that will play a significant role in determining the robot's level of autonomy. The first relates to suboptimal action selection and is defined in Eqn. (12), and the second relates to the level of human help.

While typical conformal prediction guarantees a minimal *average* prediction set size, we are interested in minimizing the human help rate, introduced here.

Bounding the Human Help Rate. We now seek to provide an additional bound on the probability of asking for human help, i.e.,

$$R_{\text{help}}(\phi) = \mathbb{P}^{(\bar{x},z)\sim\mathcal{D}}\left(|\mathcal{T}_\phi(\bar{x})| > 1\right). \tag{18}$$

Eqn. (18) is the fraction of scenarios where the prediction set is not a singleton, which is exactly the fraction of scenarios where help is needed. Hence, optimizing for action miscoverage alone may result in the robot asking for help an excessive amount of times and over-burdening the human. Instead, we apply the risk control procedure again to the help-rate risk. As before, define risk thresholds α_1 and α_2 and null hypotheses

$$\mathcal{H}_k^j: \quad R_k(\phi^j) \geq \alpha_k \quad k \in \{\text{cov, help}\} \tag{19}$$

for $j \in [|\hat{\Phi}|]$. We now present a bound on the probability that both risks are controlled simultaneously by using the p-value $p^j := \max_k p_{j,k}$.

Proposition 2. *Consider a single-step setting where we use risk calibration parameters $\phi \in \hat{\Phi}$ to generate prediction sets and seek help whenever the prediction set is not a singleton. Let the upper bound on the help rate (18) be set to α_2. If the FWER-controlling parameter set $\hat{\Phi}$ is non-empty, then with probability $1-\delta$ over the sampling of the calibration set, the new scenarios drawn from $\mathcal{D}$ incur at most α_1 rate of optimal action miscoverage and at most α_2 rate of human help.*

Proof. Follows directly from Proposition 6 of [5]. □

We provide in the Appendix an extension of our approach for the multi-step, multi-risk setting.

V. Experiments

Environments. We demonstrate RCIP in three multi-step, interactive domains, which exhibit three ways in which a robot planner can be integrated with an intent predictor. First, we consider a multi-hallway setting in which an autonomous vehicle and a human-driven vehicle must coordinate to reach opposite ends of a room by navigating a set of hallways that are only one vehicle-width wide (see Fig. 4). The human vehicle randomly selects one of the hallways in advance but does not communicate the hallway to the robot. Next, we investigate human-robot cooperative navigation in close-quarters, cluttered household settings in the Habitat 3.0 [44] simulator (see Fig. 3). Finally, we show simulation and hardware experiments for zero-shot cooperative manipulation, in which the robot aids the human in sorting common household objects (e.g. books, toys, and fruit) by a mixture of size, shape, and color (see Fig. 1). Since the environment dynamics (1) may evolve at a much faster time scale than the human's intent dynamics (2), the human's intent is updated once every T_z timesteps and is constant otherwise.

Scenario Distribution and Calibration Dataset. RCIP can be used to obtain risk guarantees for an *unknown* scenario distribution — that is, of environments and human partners — if can can collect i.i.d. samples from it for calibration. We envision that RCIP will enable a robot to interact with an end user (or set of users) through interactive data collection. Then, using the set of FWER-controlling parameters obtained from calibration (cf. section IV), the user may set a level of autonomy for the robot depending on the risk limits of the task. The scenario distribution for each environment is described in the following subsections. Each calibration dataset is generated by random sampling from the environment distribution and from the distribution over human intents. For the simulation environments, we use a pre-trained prediction model using $10k$ random scenarios. For the hallway and cooperative navigation environments, the prediction models are trained on a single NVIDIA GeForce RTX 2080 Ti GPU. Pre-training the prediction model takes about 4 hours per environment. For calibration on hardware, data collection takes about 8 hours. For all environments, we

fix $\delta = 0.01$ and use a calibration dataset of size $M = 500$. In all experiments, we evaluate thresholds $\lambda \in [0,1]$ with a step size of 0.001 and model temperature $\theta \in \{0.2, 0.4, 0.6, 0.8, 1\}$.

Baselines. We compare RCIP against similar set-valued prediction approaches. A simple but naive approach for approximated $1-\alpha_1$ coverage of optimal actions is **Simple Set**, which ranks actions according to a $1-\alpha_1$ threshold using the predictor's raw confidence scores. Actions are sorted by greatest to least confidence, and actions are added to the prediction set in order of the sorted action set until the threshold is reached. To measure the effect of *overall uncertainty* rather than individual scores, we compare against **Entropy Set**, which includes the highest overall prediction if the entropy of the distribution predicted actions is below a threshold; if not, then all actions are included in the prediction set, and the robot must ask for help. To evaluate the performance of vanilla conformal prediction against the richer hypothesis space of RCIP, we report results for **KnowNo** [36]. Similar in spirit but different from our work, KnowNo seeks to maximize coverage of optimal actions but without any guarantees on the human help rate, and assumes model parameters are fixed. Instead of maximizing coverage outright, RCIP balances prediction of optimal actions with limits on the human help rate, providing flexible performance guarantees depending on model parameters. Lastly, we consider **No Help** as an option, where the predicted action set always contains the predictor's most-confident action, and the human help rate is identically zero.

Metrics. For all environments, we report the task-level risks of (i) plan success rate and (ii) human help rate, on the test set. We also report the instantaneous risks — measured as an average over time — of plan success and human help.

A. Simulation: Cooperative Navigation in Habitat

Habitat [44] is a photo-realistic simulator containing a diverse set of scenes, objects, and humans models for human robotics tasks. In this experiment, a Boston Dynamics Spot robot and human are jointly tasked with navigating to a set of goal objects in sequence, to simulate cleaning up a house (i.e., grabbing various items, such as crackers, cans of soup, etc. as shown in Fig. 3). Each scene contains $5-10$ objects of interest. Although the human may initially be out of view of the robot, the robot must find the human and maintain a safe distance of one meter at all times. We simulate the human's decision making by choosing a high-level intent from the set of objects; here, $\mathcal{Z} = [N_o]$, where N_o is the number of objects in the scene. The confidence scores for each intended object are computed by taking the temperature-weighted softmax scores for each goal object. The final action probabilities are computed according to Eqn. (10). The robot interacts with the human over $T = 600$ environment time steps and selects a new goal object every $T_z = 100$ time steps.

Since the human's goal object is not observed by the robot, one naive strategy is to navigate to the human first, then follow the human around the house. However, since the scene is cluttered, remaining too close to the human could impede their progress (e.g. getting in the way) or block the robot, resulting in suboptimal, unsafe behavior. By predicting the human's motion, the robot is able to better accommodate the human's task while remaining safe (with high probability) with respect to unsafe interactions. We present results for cooperative navigation in Table II.

Method	$1-\alpha_1$	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
RCIP	0.85	0.85	**0.95**	**0.98**	**0.65**
KnowNo [36]	0.85	0.85	0.96	**0.98**	0.66
Simple Set	0.97	0.85	1.00	1.00	1.00
Entropy Set	–	0.66	0.46	0.45	0.06
No Help	–	0.62	0	0.94	0

TABLE I: Results for **Cooperative Manipulation**. The optimal action miscoverage rate is held fixed between RCIP, KnowNo, and Simple Set for comparing the other metrics.

B. Hardware: Cooperative Manipulation

In this example (Fig. 1), each scenario tasks the robot with helping a human to sort a set of objects by inferring the sorting category for each object. Since the human may have a preference for how the robot sorts the objects, the robot is additionally tasked with exactly matching the human's sorting preferences (e.g. if the human wants the objects sorted by color, then the robot cannot add a conflicting color to the pile). We assume that the human's intent set $\mathcal{Z}$ is represented in the (high-dimensional) space of natural language descriptions, such as "the color orange", "children's toys", and "vegetables", and that intents and actions are one-to-one. The robot interacts with the human over $T = 8$ environment time steps, and the human selects a new sorting plan every $T_z = 1$ time step.

To simplify the planning task, we assume that the robot takes in an image observation of the table and has access to a (vision) language model to process the semantic features of the image. We first use GPT-4V (`gpt-4-vision-preview`) [3] to process the image by asking for a description of each bin and the item to be sorted (e.g. the carrot in Fig. 1), commenting on possible sorting criteria for each bin. Then, using the bin descriptions, we prompt a language-only model (`gpt-3.5-turbo`) to rank a set of possible plans via multiple-choice question and answering (MCQA) [45, 46]. The temperature-weighted softmax scores for each bin give the final action probabilities.

For safety, we restrict the robot and human to work in separate workspaces, such that the human only places objects inside the human workspace and the robot only places block in the robot workspace (i.e., there is no shared workspace). To warm-start the predictor, we allow the human to initially place $3-10$ objects, with eight more to place, for a total of up to 30 objects per task. We show in Table I that RCIP reduces the plan-wise help rate by 5% and the step-wise help rate by 35% in cooperative manipulation. We use a Franka Emika Panda arm for the robotic manipulation portion of the task. Images of the scene (for both perception and planning) are obtained from an Azure Kinect RGB-D camera.

VI. Limitations and Future Work

The primary limitation of our work is a lack of guarantee on the low-level execution of the controller. Concretely,

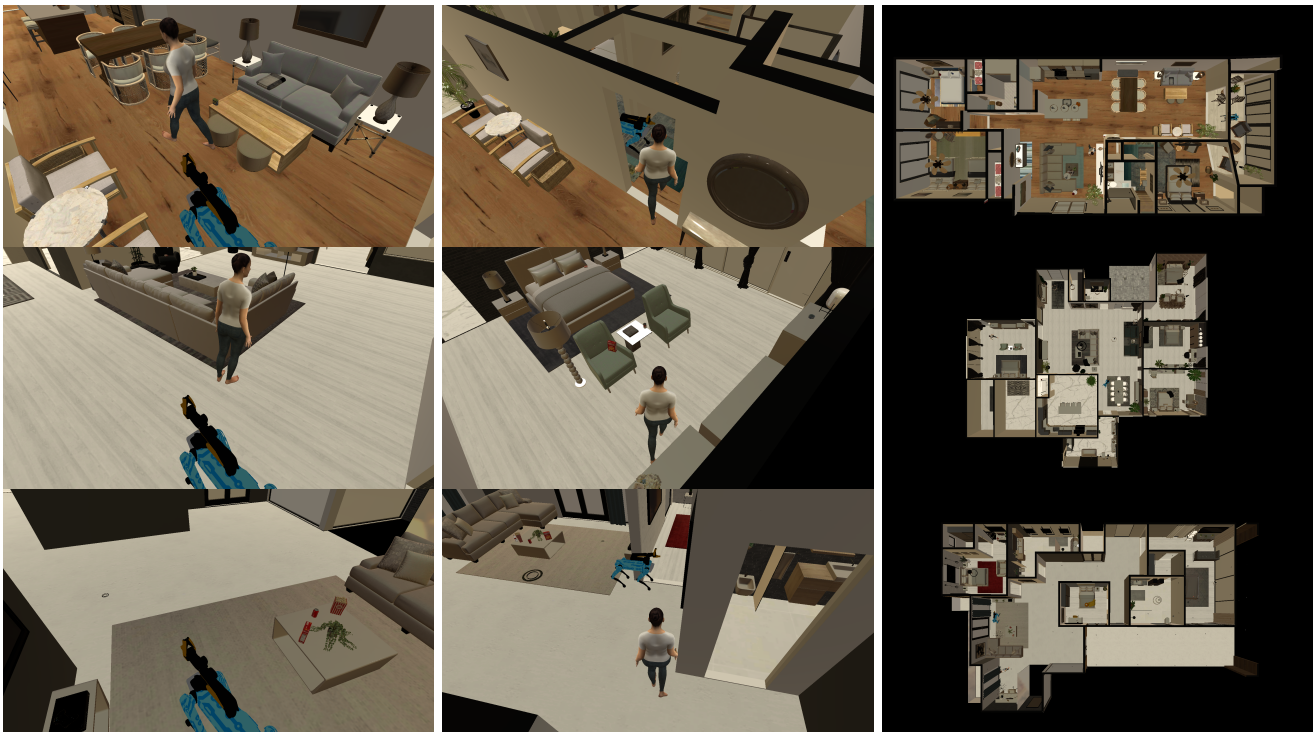

Fig. 3: Multi-step RCIP is applied in **Social Navigation**. The human and robot are tasked with finding and collecting a series of objects (e.g. cans of soup, crackers) around a close-quarters living space. The robot must recognize the human's intent and either follow or evade the human depending on the human's desired object. The robot must minimize action miscoverage across a variety of human intents and environments.

if the correct optimal action is predicted by the robot, but the controller fails to execute the computed command, then the robot will execute a suboptimal action and encounter a distribution shift, invalidating the results from RCIP. In the future, we are looking to incorporate low-level control failures as part of the risk calibration procedure. Additionally, our work fundamentally assumes that the human's intent is drawn from a finite set, and moreover that the human's intent is verbalizable or clarifiable (i.e. the human is able to provide meaningful clarifications when the robot asks for help).

In the future, we hope that RCIP can be combined with active preference learning [47]–[49] to better incorporate the human's preferences in determining the appropriate level of robot autonomy (e.g. choosing from the valid set of RCIP parameters). We also plan to study RCIP's ability to capture higher levels of interactivity in a system, such as when the robot must operate around more than one human, or when some humans are non-cooperative.

VII. Conclusion

We propose Risk-Calibrated Interactive Planning (RCIP), a framework that applies statistical multi-hypothesis risk control to address the problem of risk calibration for interactive robot tasks. We formalize RCIP as providing a statistical guarantee on an arbitrary number of user-specified risks, such as prediction failures and the amount of human help, subject to a bound on the rate at which the robot fails to predict the optimal actions. By optimizing preferences over a small number of model parameters, RCIP is able to achieve higher flexibility in aligning to user preferences than fixed-paramter methods. Experiments across a variety of simulated and hardware setups demonstrate that RCIP does not exceed user-specified risk levels. Moreover, RCIP reduces user help $5-30\%$ when compared to baseline approaches that lack formal assurances.

References

[1] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 250–265, 2020.

[2] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] 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.

[5] A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei, "Learn then test: Calibrating predictive algorithms to achieve risk control," *arXiv preprint arXiv:2110.01052*, 2021.

[6] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013.

[7] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 459–464.

[8] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5365–5372, 2022.

[9] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic mpc with multi-modal predictions for traffic intersections," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 635–640.

[10] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "LookOut: Diverse multi-future prediction and planning for self-driving," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021.

[11] L. Peters, A. Bajcsy, C.-Y. Chiu, D. Fridovich-Keil, F. Laine, L. Ferranti, and J. Alonso-Mora, "Contingency games for multi-agent interaction," *arXiv preprint arXiv:2304.05483*, 2023.

[12] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural networks*, vol. 22, no. 5-6, pp. 544–557, 2009.

[13] D. Pechyony and V. Vapnik, "On the theory of learnining with privileged information," *Advances in neural information processing systems*, vol. 23, 2010.

[14] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

[15] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 825–832.

[16] S. James and A. J. Davison, "Q-attention: Enabling efficient learning for vision-based robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1612–1619, 2022.

[17] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[18] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[19] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[20] A. Bajcsy, A. Loquercio, A. Kumar, and J. Malik, "Learning vision-based pursuit-evasion robot policies," *arXiv preprint arXiv:2308.16185*, 2023.

[21] G. Monaci, M. Aractingi, and T. Silander, "Dipcan: Distilling privileged information for crowd-aware navigation," *Robotics: Science and Systems (RSS) XVIII*, 2022.

[22] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

[23] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "Hyper: Learned hybrid trajectory prediction via factored inference and adaptive sampling," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2906–2912.

[24] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.

[25] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 015–12 021.

[26] P. Agand, M. Taherahmadi, A. Lim, and M. Chen, "Human navigational intent inference with probabilistic and optimal approaches," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8562–8568.

[27] M. F. A. R. D. T. (FAIR)†, A. Bakhtin, N. Brown, E. Dinan, G. Farina, C. Flaherty, D. Fried, A. Goff, J. Gray, H. Hu, *et al.*, "Human-level play in the game of diplomacy by combining language models with strategic reasoning," *Science*, vol. 378, no. 6624, pp. 1067–1074, 2022.

[28] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[29] J. Z.-Y. He, Z. Erickson, D. S. Brown, A. Raghunathan, and A. Dragan, "Learning representations that enable generalization in assistive tasks," in *Conference on Robot Learning*. PMLR, 2023, pp. 2105–2114.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[31] V. Vovk, V. Fedorova, I. Nouretdinov, and A. Gammerman, "Criteria of efficiency for conformal prediction," in *Conformal and Probabilistic Prediction with Applications: 5th International Symposium, COPA 2016, Madrid, Spain, April 20-22, 2016, Proceedings 5*. Springer, 2016, pp. 23–39.

[32] V. Vovk, "Conditional validity of inductive conformal predictors," in *Asian conference on machine learning*. PMLR, 2012, pp. 475–490.

[33] M. Sadinle, J. Lei, and L. Wasserman, "Least ambiguous set-valued classifiers with bounded error levels," *Journal of the American Statistical Association*, vol. 114, no. 525, pp. 223–234, 2019.

[34] K. Stankeviciute, A. M Alaa, and M. van der Schaar, "Conformal time-series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 6216–6228, 2021.

[35] K. J. Strawn, N. Ayanian, and L. Lindemann, "Conformal predictive safety filter for rl controllers in dynamic environments," *arXiv preprint arXiv:2306.02551*, 2023.

[36] 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," *arXiv preprint arXiv:2307.01928*, 2023.

[37] 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.

[38] I. Gibbs and E. Candes, "Adaptive conformal inference under distribution shift," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1660–1672, 2021.

[39] S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. Jordan, "Distribution-free, risk-controlling prediction sets," *Journal of the ACM (JACM)*, vol. 68, no. 6, pp. 1–34, 2021.

[40] J. Lekeufack, A. A. Angelopoulos, A. Bajcsy, M. I. Jordan, and J. Malik, "Conformal decision theory: Safe autonomous decisions from imperfect predictions," *arXiv preprint arXiv:2310.05921*, 2023.

[41] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005, vol. 29.

[42] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[43] P. Bauer, "Multiple testing in clinical trials," *Statistics in medicine*, vol. 10, no. 6, pp. 871–890, 1991.

[44] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," *arXiv preprint arXiv:2310.13724*, 2023.

[45] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, *et al.*, "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," *arXiv preprint arXiv:2206.04615*, 2022.

[46] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.

[47] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, *Active preference-based learning of reward functions*, 2017.

[48] B. Eric, N. Freitas, and A. Ghosh, "Active preference learning with discrete choice data," *Advances in neural information processing systems*, vol. 20, 2007.

[49] N. Wilde, D. Kulić, and S. L. Smith, "Active preference learning using maximum regret," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 952–10 959.

[50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[51] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[52] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[53] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

APPENDIX

A. Multi-Step, Single-Risk Control

We now extend SRC to settings where a robot applies set-valued prediction in multiple time steps. This setting is useful for settings where the robot receives feedback from the human between steps. However, we cannot directly apply the above procedure because the help from the human changes the distribution $\mathcal{D}$ of state-intent pairs, and the i.i.d. assumption is no longer valid. We give an extension of the Learn-then-Test procedure to multi-step settings.

Sequence-Level Risk Calibration. Similar to [36], the key idea is to (i) lift the data to sequences and (ii) perform the LTT procedure using a carefully designed score function that allows for causal reconstruction of the prediction set at test time. We now consider a distribution $\bar{\mathcal{D}}$ of *lifted* contexts induced by $\mathcal{D}$, where each lifted context contains a state-intent pair $(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}$. The lifted state-intent pairs are given as $\tilde{x}=(\bar{x}_1,...,\bar{x}_T)$ and $\tilde{z}=(z_1,...,z_T)$ respectively. Here, $\tilde{x}$ arises from the robot having performed the *correct* action in previous steps. Using the robot policy specified in Section IIIB, there are three cases to consider: the robot will (i) take the only available (and optimal) action if $\mathcal{T}_\phi(\tilde{x})$ is a singleton, or (ii) ask for clarification of the human's intent z if the action set $\mathcal{T}_\phi(\tilde{x})$ is not a singleton. We bound the risk associated with case (iii): the optimal action is not in the prediction set as follows. Let $\bar{u}^{R*}(\tilde{z}):=(u^{R*}(z_1),...,u^{R*}(z_T))$ denote the *sequence* of optimal robot actions. Let the sequence-level confidence be given as the lowest confidence over the timesteps

$$\bar{g}^*_\theta(\tilde{x},\tilde{z})=\min_{t\in[T]} g^*_\theta(\bar{x}_t,z), \tag{20}$$

where the corresponding *sequence-level* prediction set is given as $\bar{\mathcal{T}}_\phi(\tilde{x})=\{\bar{u}\in\mathcal{U}^T:\exists\tilde{z}\in\mathcal{Z}^T \text{ s.t. } \bar{u}=\bar{u}^{R*}(\tilde{z}) \text{ and } \bar{g}^*_\theta(\tilde{x},\tilde{z})\geq\lambda)\}$.

Causal Reconstruction of Sequence-Level $\bar{\mathcal{T}}_\phi$. The sequence-level prediction set $\bar{\mathcal{T}}_\phi$ is constructed with the full sequence $\tilde{z}$ as labels, which depend causally on the sequence $\bar{x}$. Hence, we do not have the entire sequence $\tilde{z}$ *a-priori*; the robot must instead construct the prediction set at each time-step in a *causal* manner (i.e., relying only on current and past observations). Let $\mathcal{T}^t_\phi(\bar{x}_t):=\{u\in\mathcal{U}:\exists z\in\mathcal{Z} \text{ s.t. } u=u^{R*}(z) \text{ and } g^*_\theta(\bar{x}_t,z_t)\geq\lambda)\}$ be the *instantaneous* action prediction set at time t. We construct $\bar{\mathcal{T}}_\phi$ in a causal manner using

$$\mathcal{T}_\phi(\tilde{x}):=\mathcal{T}^1_\phi(\bar{x}_1)\times\mathcal{T}^2_\phi(\bar{x}_2)\times...\times\mathcal{T}^{T-1}_\phi(\bar{x}_{T-1}). \tag{21}$$

Proposition 3. *Consider a multi-step setting where we use risk calibration parameters $\phi\in\hat{\Phi}$ and the sequence-level confidence (20) to generate sequence-level prediction sets and seek help whenever the prediction set is not a singleton. If the FWER-controlling parameter set $\hat{\Phi}$ is non-empty, then with probability $1-\delta$ over the sampling of the calibration set, the new scenarios drawn from $\bar{\mathcal{D}}$ under Π^R and using the causally reconstructed predicted action set (21) incur at most α_1 rate of action miscoverage.*

Proof. Let $\phi\in\hat{\Phi}$, where $\hat{\Phi}$ controls the sequence-level FWER for the non-causal set $\bar{\mathcal{T}}_\phi(\tilde{x})$ at level α_1. We first show that $\bar{u}^{R*}(\tilde{z})\in\bar{\mathcal{T}}_\phi(\tilde{x})\iff\bar{u}^{R*}(\tilde{z})\in\mathcal{T}_\phi(\tilde{x})$. For any $\tilde{z}\in\bar{\mathcal{T}}_\phi(\tilde{x})$,

$$\begin{aligned}\bar{u}^{R*}(\tilde{z})\in\bar{\mathcal{T}}_\phi(\tilde{x}) &\iff \min_{t\in[T]} g^*_\theta(\bar{x}_t,z_t)\geq\lambda\\ &\iff g^*_\theta(\bar{x}_t,z_t)\geq\lambda \quad \forall t\\ &\iff u^{R*}(z_t)\in\mathcal{T}^t_\phi(\bar{x}_t) \quad \forall t\\ &\iff \bar{u}^{R*}(\tilde{z})\in\mathcal{T}_\phi(\tilde{x}).\end{aligned} \tag{22}$$

Since the causally constructed prediction set is the same as the sequence-level prediction set, and since bound the risk associated with the sequence-level sets, we also bound the risk for the causally constructed sets. Applying the expectation definition of the risk (11) shows that the risks are the same. Applying fixed-sequence testing to the Hoeffding-Bentkus p-values completes the proof. □

We now state our most general proposition for the multi-risk, multi-step setting.

B. Multi-Step, Multi-Risk Control

In the multi-step, multi-risk setting, we seek to bound multiple risks simultaneously over the rollout of the robot policy Π^R over $\bar{\mathcal{D}}$. However, the risk guarantee only holds for the lifted contexts in $\bar{\mathcal{D}}$ and are invalid if any distribution shift occurs from taking the wrong action. In RCIP, distribution shift from $\bar{\mathcal{D}}$ (to some other distribution induced by taking suboptimal actions) may occur with the following probability,

$$\mathbb{P}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}(\text{OOD})=\mathbb{P}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}\Big(u^{R*}(z)\notin\mathcal{T}_\phi(\bar{x})\wedge|\mathcal{T}_\phi(\bar{x})|\leq 1\Big), \tag{23}$$

i.e., when the optimal action is not covered by the prediction set and the prediction set is a singleton or empty, and thus the robot takes a non-optimal action. Here, we assume that the robot cannot take a suboptimal action if it asks for help. Eqn. (23) may be upper bounded by the action miscoverage rate R_{cov} because it is the union of two events, but when R_{cov} is large, distribution shift could be frequent.

In the multi-step, multi-risk setting, we consider a set of sequence-level risk signals $(R_1,...,R_k)$ for contexts in $\bar{\mathcal{D}}$ bounded at nominal levels $(\alpha_1,...,\alpha_k)$ by all $\phi\in\hat{\Phi}$ as before. We assume that each risk models an event E_k, and the loss for each risk L_k is an indicator function $\mathbb{1}[E_k]$. We assume that $R_1=R_{\text{cov}}$. In addition, since any OOD sequence incurs a task failure, we seek to bound the probability of E_k occurring subject to an R_1 probability of distribution shift (in which case E_k can also happen).

Proposition 4. *Consider a multi-step setting where we use risk calibration with threshold level $\phi\in\hat{\Phi}$ and the sequence-level score function (20) to generate sequence-level prediction sets and seek help whenever the prediction set is not a singleton. Consider a set of sequence-level risks $(R_1,...,R_k)$ bounded at nominal levels $(\alpha_1,...,\alpha_k)$, where R_1 is the miscoverage risk R_{cov}. If the action miscoverage rate*

is bounded at level α_1 over the sampling of the calibration set, the new scenarios drawn from $\bar{\mathcal{D}}$ under Π^R and using the causally reconstructed predicted action set (21) incur at most α_1 and $\alpha_k+\alpha_1$ rate of risk for $k\geq 2$ with failure rate $1-\delta$ over the sampling of the calibration set.

Proof. For $k=1$, risk R_{cov} already provides a bound on the OOD rate. For $k\geq 2$, the remainder of the proof follows a union bound argument. If the OOD rate is large, then the OOD-aware bound α_k will be much larger than the nominal bound. Therefore, the OOD rate must be controlled to have a non-trivial limit on the other risks. Using the definition of each risk and the linearity of expectation, we have that

$$\begin{aligned}\alpha_1+\alpha_k &\geq R_1(\phi)+R_k(\phi)\\ &=\mathbb{E}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}\Big[L_1(\tilde{x},\mathcal{T}_\phi(\tilde{x}),\tilde{z})+L_k(\tilde{x},\mathcal{T}_\phi(\tilde{x}),\tilde{z})\Big]\\ &=\mathbb{E}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}\Big[\mathbb{1}[E_1]+\mathbb{1}[E_k]\Big]\\ &=\mathbb{E}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}\Big[\mathbb{1}[E_1]\Big]+E^{(\tilde{x},\tilde{z})\sim\mathcal{D}}\Big[\mathbb{1}[E_k]\Big]\\ &=\mathbb{P}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}(E_1)+\mathbb{P}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}(E_k)\\ &\geq\mathbb{P}^{(\tilde{x},\tilde{z})\sim\bar{\mathcal{D}}}(E_1\vee E_k).\end{aligned}\tag{24}$$

Then, either event E_k or the event of distribution shift E_1 occurs at a rate no more than $\alpha_1+\alpha_k$. □

Corollary 1. *As a direct consequence of Eqn. (24), if one wishes to calibrate risks other than the optimal action miscoverage rate, such as the user help rate (18), then it is sufficient to calibrate at level $\alpha_k=\max(\alpha_k'-\alpha_1,0)$, where α_k' is the desired overall risk that incorporates distribution shift and the maximum is due to the constraint that the risk be in $[0,1]$.*

C. Additional Experiments: Hallway Navigation

Autonomous navigation around other autonomous decision-making agents, including humans, requires the robot to recognize scenario uncertainty (whether another agent will turn right or left) with task efficiency (energy spent braking or taking detours). While safety can almost always be guaranteed if each vehicle declares their intent at all times, such communication can be costly, especially if human prompting is involved. In this example (Fig. 4), the robot is asked to navigate to the human vehicle's initial condition without colliding while the human does the same. The set of intents is $\mathcal{Z}=\{1,2,3,4,5\}$, where each intent corresponds to one of the five hallways. The confidence scores for each intent are computed by taking the temperature-weighted softmax scores for each hallway. The final action probabilities are computed according to Eqn. (10). The robot interacts with the human over $T=200$ environment time steps and predicts the human's intent every $T_z=20$ time steps.

To ensure that the robot reaches its goal state in a minimal amount of time, we permit the robot to prompt the human for their chosen hallway if its optimal action set is not a singleton. We jointly learn the robot and human policies using proximal policy optimizatio (PPO) [50, 51]. The human and robot PPO policies are trained jointly using 256 environments and take about 4 hours to train.

Method	$1-\alpha_1$	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
RCIP	0.85	0.86	**0.34**	**0.95**	**0.24**
KnowNo [36]	0.85	0.86	0.48	0.92	0.42
Simple Set	0.98	0.85	0.48	0.92	0.42
Entropy Set	–	0.75	0.07	0.86	0.02
No Help	–	0.73	0	0.86	0

TABLE II: Results for **Cooperative Navigation**. The optimal action miscoverage rate is held fixed between RCIP, KnowNo, and Simple Set for comparing the other metrics.

Fig. 5 provides a comparison between RCIP and other baseline approaches that employ set-valued prediction. While entropy and simple-set can be used to provide (respectively) static and dynamic thresholds for heuristic uncertainty quantification, these uncalibrated methods often ask for too much help and scale poorly as the desired plan success rate increases.

Fig. 6 provides an ablation study on the effect of the bounds on miscoverage and the human help rate on the size of the FWER-controlling parameter set $|\hat{\Phi}|$ in the *multi-risk, multi-step* setting. As the miscoverage rate bound becomes lower, lowering the human help rate provides fewer valid parameters, until $|\hat{\Phi}|=0$, and controlling both risks is infeasible.

D. Additional Task Details for Hallway Navigation

1) Environment: As shown in Fig. 4, the blue car ("robot") and red car("human") are tasked with navigating to opposite ends of a room 16 meters long and 9 meters wide. Each car is controlled by a two-dimensional vector that sets the desired velocity and turning rate. Each hallway is one meter wide, and two cars cannot pass in a single hallway. The human car's intent is selected from a uniform random distribution over each of the five hallways and does not change over time. The interaction is constrained such that if the robot car collides with the walls, boundary, or human car, the episode automatically terminates. Each car's goal region is 2 meters long and 4 meters wide. Each car's goal is to maximize the forward progress at each time set, and the loss in Eqn. (4) is the negative forward progress. Agent's initial positions are drawn randomly from the other agent's goal set.

2) Policy: We train a PPO policy to maximize the forward progress jointly for both cars. To ensure satisfaction of the collision constraints, we terminate the episode for both cars if either car violates a collision constraint. We train a three-layer PPO policy using 256 parallel environments and a hidden dimension of 64, learning rate of 0.0001, batch size of 4096, and 32 gradient steps per rollout.

3) Prediction Model: To predict the human car's intent, we train a transformer-based prediction model similar to [4, 52] to predict a probability distribution over the hallway in addition to the future position of the human car. We encode the position histories for both agents using a three layer MLP with hidden dimension 256. We process the encoded input using six transformer encoder layers and six transformer decoder layers, each with a hidden dimension of 256. For the state prediction task, we predict with a time horizon of up to 100 time steps.

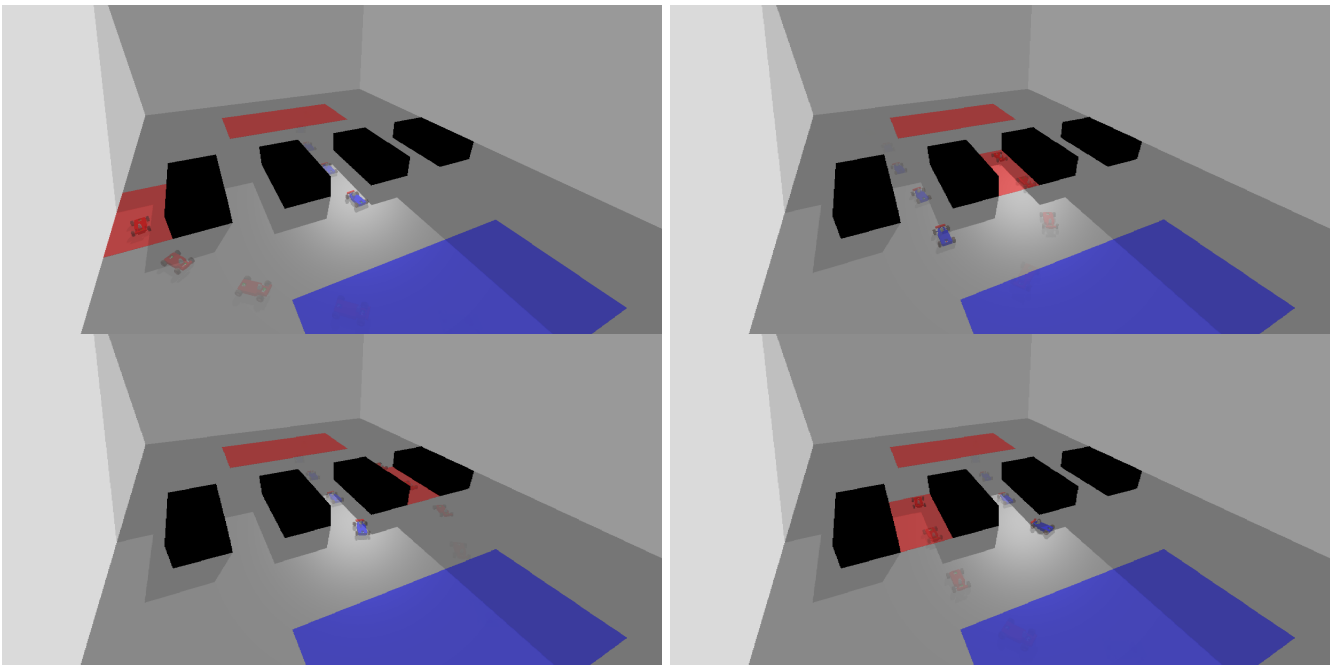

Fig. 4: Multi-step RCIP is applied in **Hallway Navigation**. The robot car (blue) and human car (red) are tasked with navigating to their respective goal states (large blue and red rectangles). The human car is constrained via its intent to pass through one of the five hallways (highlighted in red). The blue car does not observe the human's intent during evaluation.

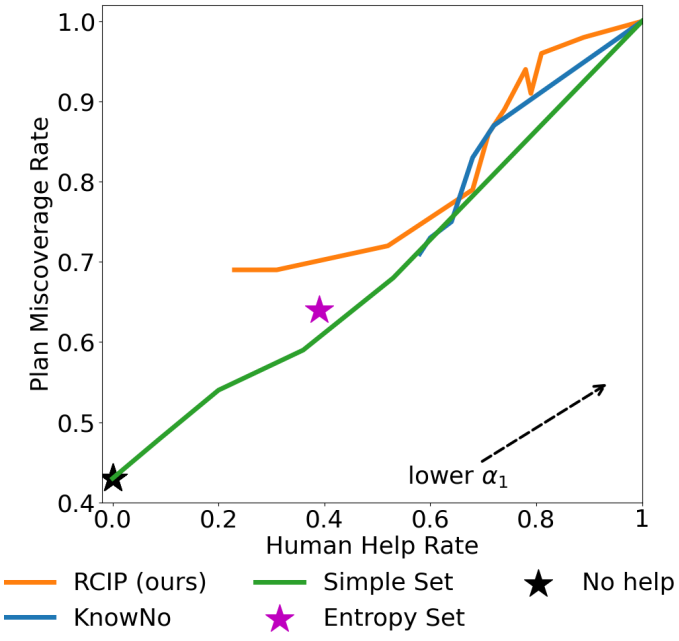

Fig. 5: Baseline comparison for RCIP verus other set-valued predictors for **Hallway Navigation**. RCIP provides a framework for tuning model parameters to achieve risk control, versus other methods that assume that model parameters are held fixed: KnowNo [36], Simple Set, Entropy Set, and No Help.

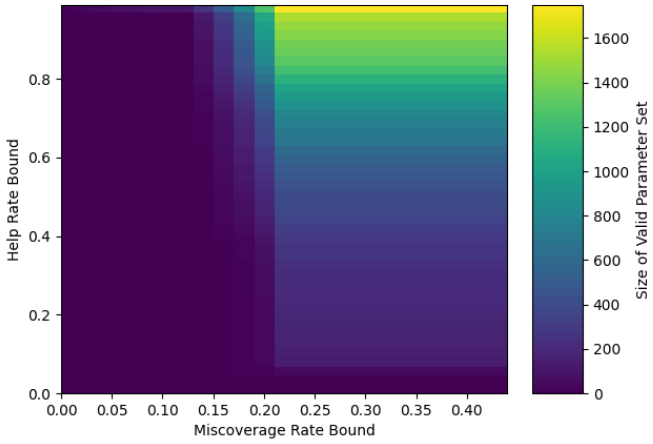

Fig. 6: Ablation study on the effect of action miscoverage and help rate risk limits versus FWER-controlling parameter set size for RCIP on **Hallway Navigation** using $\alpha_{\text{cov}} \in [0, 0.45]$ and $\alpha_{\text{help}} \in [0, 1]$. The color denotes the size of the set of FWER-controlling parameters $\hat{\Phi}$, with empty (infeasible) sets taking a size of zero.

To train the model, we use the following loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{MSE} \tag{25}$$

where $\mathcal{L}_{CE}$ is the cross entropy of the predicted intent distribution versus the true ground truth label, $\mathcal{L}_{MSE}$ is the mean square error from the human car's ground truth state, and λ is a scalar that controls the relative weight of the state

Fig. 7: Full object set for cooperative manipulation.

prediction task. We find the training to be relatively insensitive to λ, but that $\lambda = 1$ works well in practice. We do not use knowledge of the human's static intent in the prediction task.

E. Additional Task Details for Social Navigation

1) Environment: As shown in Fig. 3, the Spot robot must navigate around a shared living space with the human. The Spot robot is tasked with meeting the human at a series of goal objects (e.g. box of crackers, can of soup) while avoiding the human. Each environment requires the robot to make roughly 15 meters of forward progress to meet the human while navigating between tight walls. Before the environment is initialized, a random set of up to 10 objects is sampled from the YCB dataset [53]. The human's intended object changes every 100 time steps. We use a total episode length of 600 time steps (i.e. 5 goal objects per episode). To constrain the robot to maintain a safe distance of at least 1 meter from the human at all times and terminate the episode if this distance is violated.

2) Policy: To accommodate the close-quarters, constrained motion planning needed for social navigation, we use Habitat's [44] built-in shortest-path navigation algorithm. During the contingency planning phase, the robot plans a shortest path for each of possible objects intended by the human. To ensure that the human and robot do not collide, we augment the robot's path planning dynamics with an additional artificial potential term that repels the robot from the human with force proportional to the inverse distance to the human.

3) Prediction Model: We use an identical prediction architecture and training procedure as that of the social navigation task. Since the number of objects may change across the environments, we use the object locations as anchors for the transformer decoder task.

F. Additional Task Details for Cooperative Manipulation

1) Environment: As shown in Fig. 1, the robot is tasked with sorting a set of objects using a limited demonstration from the human. The full set of objects is shown in Fig. 7. Eight objects are sorted per rollout. We preselect 10 bins of objects sorted with a preset criteria and randomly select 3 bins for each trial, giving various levels of situational ambiguity. After the fourth time step, two additional objects are added for additional context; in practice, we find that this addition has little effect on prediction accuracy.

2) Policy: Since the human's intent is evaluated per-object at each time step, no dynamic programming is needed; the optimal policy is to place the object in the correct bin.

3) Prediction Model: At each time step, we use the GPT-4V vision language model to compare four images: the three bins containing the robot-human pair's example, and the object to be sorted. We use GPT-3.5 to generate the softmax scores using a MCQA to select one of the three bins or "not sure".