Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

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`

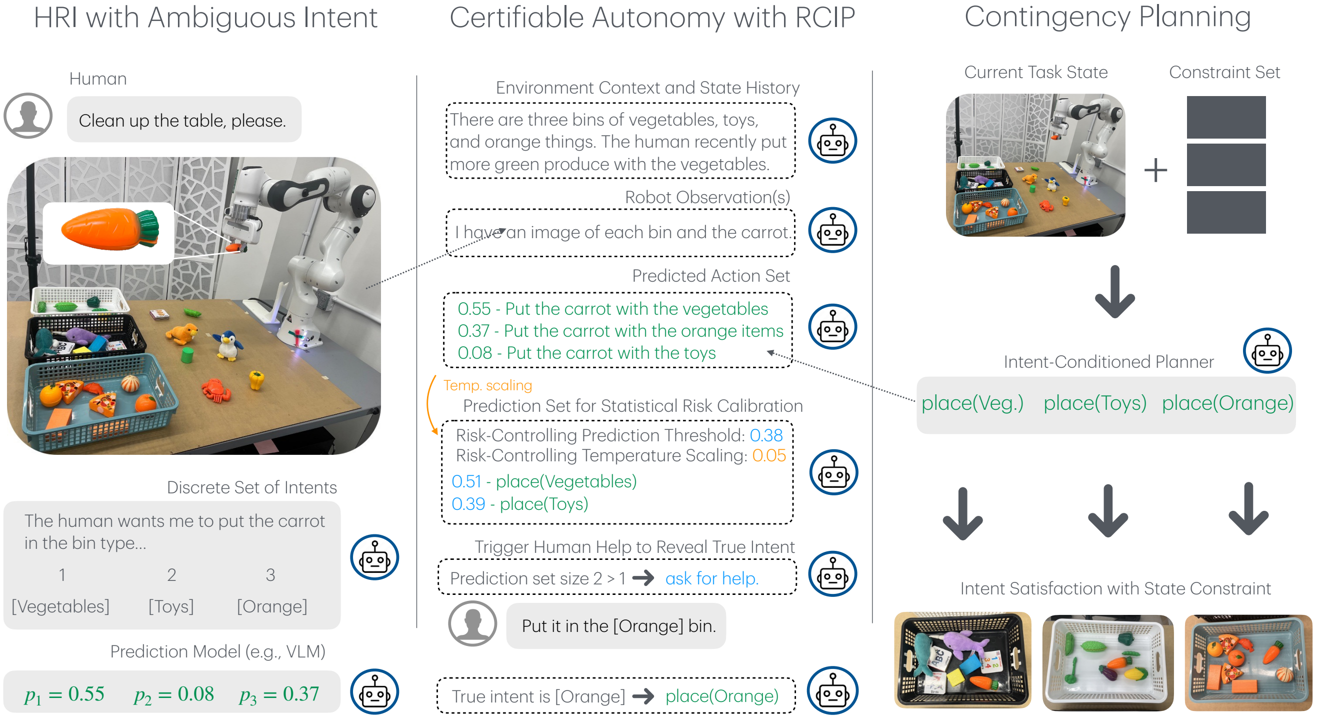

HRI with Ambiguous Intent
Certifiable Autonomy with RCIP
Contingency Planning
Human
Clean up the table, please.
Discrete Set of Intents
The human wants me to put the carrot in the bin type...
1
2
3
[Vegetables]
[Toys]
[Orange]
Prediction Model (e.g., VLM)
$p_1 = 0.55$
$p_2 = 0.08$
$p_3 = 0.37$
Environment Context and State History
There are three bins of vegetables, toys, and orange things. The human recently put more green produce with the vegetables.
Robot Observation(s)
I have an image of each bin and the carrot.
Predicted Action Set
0.55 - Put the carrot with the vegetables
0.37 - Put the carrot with the orange items
0.08 - Put the carrot with the toys
Temp. scaling
Prediction Set for Statistical Risk Calibration
Risk-Controlling Prediction Threshold: 0.38
Risk-Controlling Temperature Scaling: 0.05
0.51 - place(Vegetables)
0.39 - place(Toys)
Trigger Human Help to Reveal True Intent
Prediction set size 2 > 1 → ask for help.
Put it in the [Orange] bin.
True intent is [Orange] → place(Orange)
Current Task State
Constraint Set
+
Intent-Conditioned Planner
place(Veg.)
place(Toys)
place(Orange)
Intent Satisfaction with State Constraint

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 anticipate human intent, 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* intent-prediction in human-robot cooperative scenarios 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 uncertainty 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 risk 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

---

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

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 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. Depending on the robot's confidence level and the human's preferred level of autonomy, the robot can either take a risk or ask for help. To allow the human to specific different levels of robot autonomy (more or less confident predictions), we assume that the predictor has a small number of tunable model parameters (such as the temperature used in softmax scoring). We use a small calibration dataset of human-robot interactions to choose a set of valid parameters 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 (SRC)*: 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 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 SRC 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 K 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 $8-58\%$ versus baseline approaches.

II. RELATED WORK

RCIP brings together techniques from contingency planning, human intent prediction, and conformal prediction and empirical risk control. We discuss related work in each area below.

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 (for example, their preference for a particular bin as shown in Fig. 1). 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) an intent-conditioned policy, 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. In practice, we show that intent-condition planning permits a modular approach to planning in uncertain environments: the intent predictor allows the robot to discern high-level uncertainly in human preferences for task completion, while a low-level intent-conditioned planner allows the robot to ensures that an acceptable intent-conditioned policy exists.

B. Human Intent Prediction

Predicting intent of humans for downstream planning has been widely applied in autonomous driving [22]–[24], social navigation [3, 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 trained on a set of representative interactions. To showcase RCIP's ability to enable *zero-shot* intent-conditioned planning we leverage recent advances in vision language models (VLMs) [30] to predict human intent by extracting task-relevant semantic features through vision, *without fine-tuning on specific interactions.*

In this work, we show that both task-specific (e.g., [3]) and zero-shot intent predictors (e.g., [4]) can be used to achieve a variety of robot autonomy levels when combined with RCIP. We enable efficient transfer to downstream planning by restricting the support of the predictor to a small set of human intents, allowing the model to focus on high-level uncertainties.

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:

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

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 $[T]$ notation to denote the set of natural numbers up to T. We use a superscript for individual agent indexing, and we use bar notation to denote aggregation over time, e.g. $\bar{x}_t = (x_1,\, ...,\, x_t)$. Let $\pi = (\pi^R,\, \pi^H)$ denote the joint policy governing system (1). We permit the human's action to be drawn from a potentially multi-modal distribution π^H. We model the modes of π^H as corresponding to different human intents, which may be time-varying, and their dynamics is described next.

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

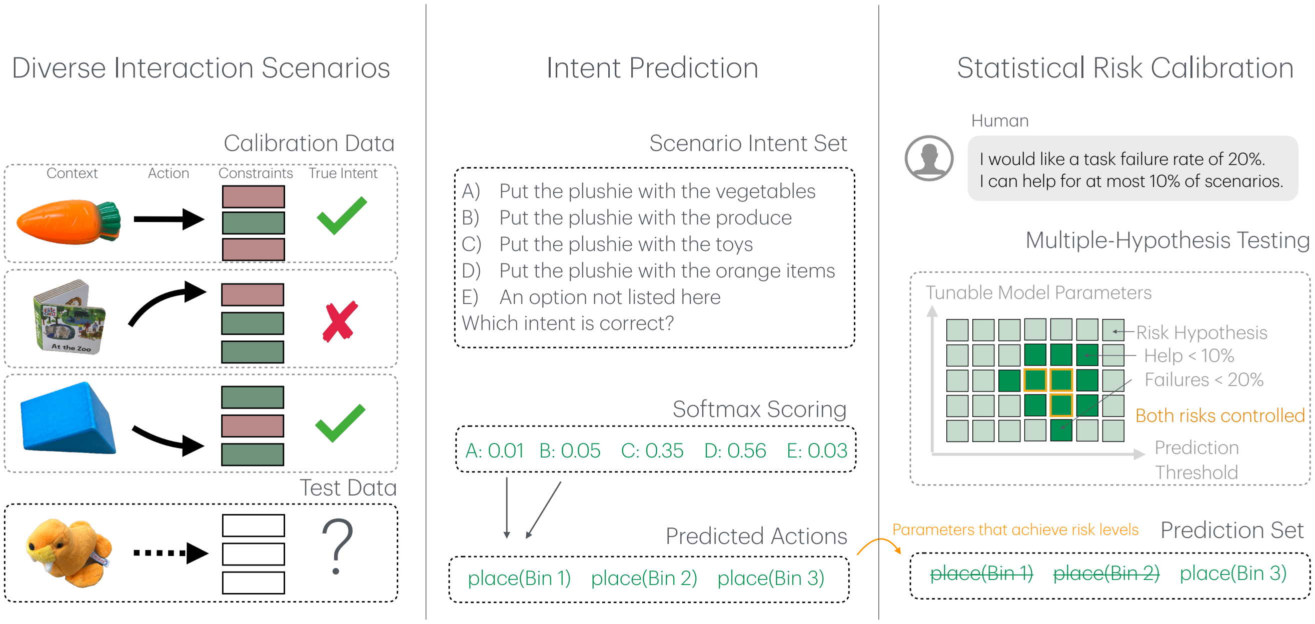
$$z_{t+1} \sim q_t(\cdot | x_t, z_t;\, H) \quad \forall t \in [T], \tag{2}$$

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 drawn from a (potentially time-varying) probability distribution, i.e., $u_t^i = \pi_t^i(\cdot | x_t, z_t)$, 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

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

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



A) Put the plushie with the vegetables
B) Put the plushie with the produce
C) Put the plushie with the toys
D) Put the plushie with the orange items
E) An option not listed here
Which intent is correct?

I would like a task failure rate of 20%.
I can help for at most 10% of scenarios.

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 tunable model parameters, 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).

additional set of inequality constraints h^i that depend on the (time-varying) intent of the human:

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

Intent-Conditioned 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:

$$V^i(x,z)=\inf_{\pi^i}\left\{\mathbb{E}^{(u,z')\sim(\pi,q)}\left[l^i(x,u)+V^i(f(x,u;e),z')\right]\right\}, \tag{6}$$

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

$$Q^i(x,z,u^i)=l^i(x,u)+\mathbb{E}^{(u,z')\sim(\pi,q)}[V^i(f(x,u;e),z')]. \tag{7}$$

Eqn. (7) states that since both agents' policies are conditioned on the human's intent (interpreted as a state constraint), the optimal action can be approximated with a variety of existing policy optimization or dynamic programming techniques, e.g., PPO [41], A-star graph search [42], or Rapidly Exploring Random Trees (RRT) [43]. In particular, the optimal action can be approximated for both agents as the minimizer of the Q-function, i.e.,

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

We emphasize that, in reality, the robot need not *exactly* actuate the optimal action as long as the intent constraint in (4) is satisfied.

Intent Prediction. During deployment, the true human intent z is not observed. However, we assume access to a frozen model g_θ with a small number of tunable parameters θ that *predicts* human intent based on a sequence $\bar{x}_t$ of prior states. Specifically, $g_\theta(\bar{x}_t,z)$ seeks to estimate the probability mass $q(z|x_{t-1},z_{t-1})$ of each intent $z\in\mathcal{Z}$ by producing a set of softmax confidence scores on the simplex $\Delta^{|\mathcal{Z}|-1}$. That is,

$$g_\theta(\bar{x}_t,z)=\frac{\exp(G_\theta(\bar{x}_t,z))}{\sum_{z'\in\mathcal{Z}}\exp(G_\theta(\bar{x}_t,z'))} \quad \forall z\in[\mathcal{Z}], \tag{9}$$

where G_θ outputs the logit scores for each intent. In general, g_θ will output heuristic and *uncalibrated* confidence scores, and the approximation of the true intent dynamics q may be poor, since g_θ need not be trained on the same human intent dynamics as given by (2). However, we show in the following analysis (c.f. Section IV) that by asking for help, the robot is able to overcome incorrect predictions through strategic selection of human interventions with SRC.

B. Risk-Calibrated Interactive Planning

RCIP seeks to distill a weighted set of actions using the scores from g_θ and the actions from the intent-conditioned planner. By using an action *set* during action prediction, RCIP ensures that a plan exists for each possible human intent. We now describe how the action set is obtained.

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

$$\mathcal{S}_{\lambda,\theta}(\bar{x}_t) = \{z \in \mathcal{Z} : g_\theta(\bar{x}_t, z) \geq \lambda\}, \tag{10}$$

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

$$\mathcal{T}_{\lambda,\theta}(\bar{x}_t) = \{u \in \mathcal{U} : \exists z \in \mathcal{S}_{\lambda,\theta} \text{ s.t. } u = u^{R*}(z) \text{ and } g^*_\theta(\bar{x}_t, z) \geq \lambda\}, \tag{11}$$

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

Remark. *Eqns.* (10) *and* (11) *convey the fact that the mapping from intents to actions can be many-to-one. Taking Fig. 1 as an example, if the predicted intent set is augmented to include {Vegetables, Produce, Toys, Orange}, the predicted intent set will have* $|\mathcal{S}_{\lambda,\theta}| = 4$ *intents for sorting, but the predicted action set will only have* $|\mathcal{T}_{\lambda,\theta}| = 3$ *actions, since Vegetables and Produce are the same bin and the action is the same. Quantifying uncertainty in action space allows RCIP to ask for help only when the predicted actions are different.*

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. (11), 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 robot ceases operation and 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 parameters (θ, λ) that achieve multiple user-specified levels of risk. As shown in Fig. 2, we formalize this problem by considering a joint distribution $\mathcal{D}$ over scenarios $\xi := (e, H, l, h)$, where e is an environment (a POMDP) with dynamics f are parameterized by e, H is a particular human with stochastic intent dynamics q, and l and h are respectively a cost and constraint 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 D_{cal} containing 400 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 $(\alpha_1, \dots, \alpha_K)$ 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 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 parameters θ. For fixed parameters (λ, θ), the expected loss is itself a function of the context, prediction sets, and true labels over the unknown distribution $\mathcal{D}$, i.e.

$$R(\lambda, \theta) = \mathbb{E}^{(\bar{x}, z) \sim \mathcal{D}} \Big[L(\bar{x}, \mathcal{T}_{\lambda,\theta}(\bar{x}), z) \Big], \tag{12}$$

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

$$R_{\text{cov}}(\lambda, \theta) = \mathbb{P}^{(\bar{x}, z) \sim \mathcal{D}} \big(u^{R*}(z^*) \notin \mathcal{T}_{\lambda,\theta}(\bar{x}) \big). \tag{13}$$

Remark. *Equation* (13) *is identical to the typical conformal prediction (CP) setting [44, 45] 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 $D_{\text{cal}} = \{(\bar{x}_i, z_i)\}_{i=1}^M$ 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 parameters, such as temperature. Then, we use the calibration set in order to choose the parameter pair $(\lambda,\theta) \in \Lambda \times \Theta$ to control a user-specified risk, regardless of the quality of the predictor g_θ^*. In future sections, we let Φ denote this product search space.

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 $(\lambda,\theta) \in \hat{\Phi}$, $R(\lambda,\theta) > \alpha$ (i.e., the risk is not controlled) with some user-defined probability δ. In particular, the probability δ is with respect to the randomness in sampling over the calibration dataset D_{cal}, which itself is randomly sampled from the unknown distribution $\mathcal{D}$.

Multi-Hypothesis Testing for a Single Risk. Since the prediction set $\mathcal{T}_{\lambda,\theta}$ is controlled by low-dimensional hyperparamters (λ,θ) drawn from the set $\hat{\Phi}$, controlling a single risk is a multiple-hypothesis testing problem [5]. For each $j \in \{1,\ldots,|\Phi|\}$, we consider the hypothesis $\mathcal{H}^j$ such that the risk $R(\lambda^j,\theta^j)$ is not controlled, where $\lambda^j,\theta^j \in \Phi$. 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:

$$\hat{R}^j = \frac{1}{M}\sum_{i=1}^{M} L(\bar{x}_i, \mathcal{T}_{\lambda^j,\theta^j}(\bar{x}_i), z_i). \tag{14}$$

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

$$p^j = \min\Big(\exp\big(-Mh_1(\max(\hat{R}^j,\alpha),\alpha)\big), e\hat{\Phi}^{\text{Bin}}_{\alpha,n}(\lceil n\hat{R}^j\rceil)\Big), \tag{15}$$

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 (λ^j,θ^j) that reject $\mathcal{H}^j$ and control the risk R. Bounding (12) for all $(\lambda,\theta) \in \hat{\Phi}$ 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, 46]. 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

$$\hat{\Phi} \leftarrow \begin{cases} \hat{\Phi} \cup \{(\lambda^l,\theta^l)\} & (\lambda^l,\theta^l) \notin \hat{\Phi} \text{ and } p^l \le \delta/|\mathcal{J}|, \\ \hat{\Phi} & o.w. \end{cases} \tag{16}$$

for $l \ge j$ and $j \in [\mathcal{J}]$. That is, parameters (λ^j,θ^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

$$\hat{\Phi} := \{\lambda^j,\theta^j : p^j \le \delta/|\mathcal{J}|\}. \tag{17}$$

Thus,

$$\mathbb{P}^{(\bar{x},z)\sim D^M}\left(\sup_{(\lambda,\theta)\in\hat{\Phi}} \big\{R(\lambda,\theta)\big\} \le \alpha\right) \ge 1-\delta, \tag{18}$$

where the supremum over the empty set is defined as $-\infty$. The calibration procedure thus yields $\hat{\Phi}$, which is a set of values λ^j,θ^j that each control the risk $R(\lambda^j,\theta^j)$ 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 $(\lambda,\theta) \in \hat{\Phi}$ 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 nearly identical to the conformal prediction setting [36] up to the choice of concentration inequality. ☐

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. (13), 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.,

$$R_{\text{help}}(\lambda,\theta) = \mathbb{P}^{(\bar{x},z)\sim\mathcal{D}}\big(|\mathcal{T}_{\lambda,\theta}(\bar{x})| > 1\big). \tag{19}$$

Eqn. (19) 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

$$\mathcal{H}^j_k: \quad R_k(\lambda^j,\theta^j) \ge \alpha_k \quad k \in \{\text{cov, help}\} \tag{20}$$

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 $(\lambda,\theta) \in \hat{\Phi}$ to generate prediction sets and seek help whenever the prediction set is not a singleton. Let the upper bound on the help rate (19) 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]. ☐

D. 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 $(\tilde{x},\bar{z}) \sim \bar{\mathcal{D}}$. The lifted state-intent pairs are given as $\tilde{x} = (\bar{x}_1,\ldots,\bar{x}_T)$ and $\bar{z} = (z_1,\ldots,z_T)$ 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 $\mathcal{T}_{\lambda,\theta}(\tilde{x})$ is a singleton, or (ii) ask for clarification of the human's intent z if the action set $\mathcal{T}_{\lambda,\theta}(\tilde{x})$ is not a singleton. We bound the risk associated with case (iii): the optimal action is not in the prediction set as follows. Let $\bar{u}^{R*}(\bar{z}) := (u^{R*}(z_1),\ldots,u^{R*}(z_T))$ denote the *sequence* of optimal robot actions. Let the sequence-level confidence be given as the lowest confidence over the timesteps

$$\bar{g}^*_\theta(\tilde{x},\bar{z}) = \min_{t\in[T]} g^*_\theta(\bar{x}_t,z), \tag{21}$$

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

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

$$\mathcal{T}_{\lambda,\theta}(\tilde{x}) := \mathcal{T}^1_{\lambda,\theta}(\bar{x}_1) \times \mathcal{T}^2_{\lambda,\theta}(\bar{x}_2) \times \ldots \times \mathcal{T}^{T-1}_{\lambda,\theta}(\bar{x}_{T-1}). \tag{22}$$

Proposition 3. *Consider a multi-step setting where we use risk calibration parameters $(\lambda,\theta) \in \hat{\Phi}$ and the sequence-level confidence (21) 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 (22) incur at most α_1 rate of action miscoverage.*

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

$$\begin{aligned}
\bar{u}^{R*}(\bar{z}) \in \bar{\mathcal{T}}_{\lambda,\theta}(\tilde{x}) &\iff \min_{t\in[T]} g^*_\theta(\bar{x}_t,z_t) \geq \lambda \\
&\iff g^*_\theta(\bar{x}_t,z_t) \geq \lambda \quad \forall t \\
&\iff u^{R*}(z_t) \in \mathcal{T}^t_{\lambda,\theta}(\bar{x}_t) \quad \forall t \\
&\iff \bar{u}^{R*}(\bar{z}) \in \mathcal{T}_{\lambda,\theta}(\tilde{x}).
\end{aligned} \tag{23}$$

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 (12) 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.

E. 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,

$$\mathbb{P}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}(\text{OOD}) = \mathbb{P}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}\Big(u^{R*}(z) \notin \mathcal{T}_{\lambda,\theta}(\bar{x}) \wedge |\mathcal{T}_{\lambda,\theta}(\bar{x})| \leq 1\Big), \tag{24}$$

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. (24) 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 $(R_1,\ldots,R_k)$ for contexts in $\bar{\mathcal{D}}$ bounded at nominal levels $(\alpha_1,\ldots,\alpha_k)$ by all $(\lambda,\theta) \in \hat{\Phi}$ 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* $(\lambda,\theta)\in\hat{\Phi}$ *and the sequence-level score function* (21) *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* (22) *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

$$\begin{aligned}\alpha_1+\alpha_k &\geq R_1(\lambda,\theta)+R_k(\lambda,\theta)\\ &=\mathbb{E}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}\Big[L_1(\tilde{x},\mathcal{T}_{\lambda,\theta}(\tilde{x}),\bar{z})+L_k(\tilde{x},\mathcal{T}_{\lambda,\theta}(\tilde{x}),\bar{z})\Big]\\ &=\mathbb{E}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}\Big[\mathbb{1}[E_1]+\mathbb{1}[E_k]\Big]\\ &=\mathbb{E}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}\Big[\mathbb{1}[E_1]\Big]+E^{(\tilde{x},\bar{z})\sim\mathcal{D}}\Big[\mathbb{1}[E_k]\Big]\\ &=\mathbb{P}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}(E_1)+\mathbb{P}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}(E_k)\\ &\geq\mathbb{P}^{(\tilde{x},\bar{z})\sim\bar{\mathcal{D}}}(E_1\vee E_k).\end{aligned}\tag{25}$$

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.* (25)*, if one wishes to calibrate risks other than the optimal action miscoverage rate, such as the user help rate* (19)*, 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* $[0,1]$.

V. Experiments

Environments. We demonstrate single- and multi-step RCIP in four interactive domains, which exhibit four ways in which a robot planner can be integrated with an intent predictor. First, we consider a multi-hallway setting in which two vehicles must coordinate to reach opposite ends of a room by navigating a set of hallways that are only one vehicle-width wide (see Fig. 3). One vehicle is driven by simulated human that has diverse intent. At the start of the episode, the human vehicle selects one of the hallways at random but does not communicate the hallway to the robot. Next, we investigate human-robot Social Navigation in close-quarters, cluttered household settings in the Habitat 3.0 [47] simulator (see Fig. 4(left)). Third, we show hardware experiments for zero-shot Open-Category Sorting, 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). Finally, we show hardware experiments for a bimanual sorting setting in which the robot must take an object from the human and place it in the human's intended bin when multiple bins may be appropriate (see Fig. 4(right)). In all settings, since the environment dynamics (1) may evolve at a 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.

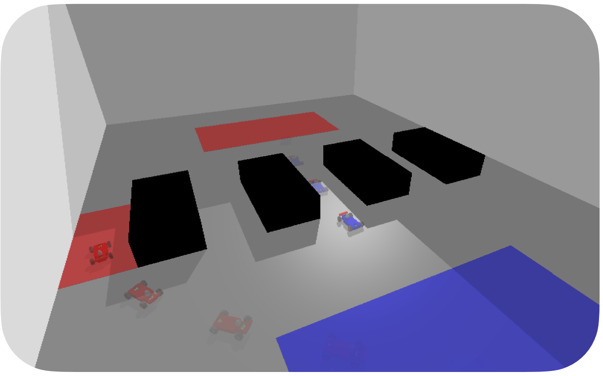


Fig. 3: Multi-step RCIP is applied in **Hallway Navigation**. The robot car (blue) and human car (red) are tasked with navigating to their respective goal states (large blue and red rectangles). The human car is constrained via its intent to pass through one of the five hallways (highlighted in red). The blue car does not observe the human's intent during evaluation.

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 Social 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 $M=400$. We fix $\alpha_{\text{cov}}=0.15$ for the Hallway and Social Navigation tasks and We fix $\alpha_{\text{cov}}=0.25$ for zero-shot prediction in the Open-Category Sorting and Bimanual Sorting tasks. In all

Method	$1-\alpha_{\text{cov}}$	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
RCIP	0.85	0.86	**0.27**	0.98	**0.27**
KnowNo [36]	0.85	0.87	0.58	0.98	0.58
Simple Set	0.94	0.86	0.54	0.98	0.54
Entropy Set	–	0.61	0.03	0.94	0.00
No Help	–	0.59	0	0.94	0

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

experiments, we evaluate thresholds $\lambda \in [0,1]$ using an evenly spaced grid with $|\hat{\Lambda}| = 2000$. We evaluate the model temperature $\theta \in [0.001, 10]$ using a log-spaced grid with $|\Theta| = 5$. For more details on the scenario distribution, please see Appendix A.

Baselines. We compare RCIP against similar set-valued prediction approaches. A simple but naive approach for approximated $1-\alpha_{\text{cov}}$ coverage of optimal actions is **Simple Set**, which ranks actions according to a $1-\alpha_{\text{cov}}$ 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: Hallway Navigation

Autonomous navigation around other autonomous decision-making agents, including humans, requires a robot to recognize scenario uncertainty (whether another agent will turn right or left) with task efficiency (energy spent braking or taking detours). While human prompting can reduce uncertainty in multi-agent coordination, such communication can be costly or burdensome for the human. In this example (Fig. 3), the robot is asked to navigate to the initial condition of the human's vehicle without colliding. 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. (11). The robot interacts with the human over $T = 200$ environment time steps and predicts the human's intent every $T_z = 20$ time steps. Additional details on training the intent prediction model are deferred to Appendix B.

Model	Method	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
1	RCIP	0.86	0.27	0.98	0.27
	No Help	0.59	0	0.94	0
2	RCIP	0.87	0.04	0.98	0.04
	No Help	0.68	0	0.94	0
3	RCIP	1.00	0.00	1.00	0.00
	No Help	1.00	0	1.00	0

TABLE II: Help rate results for **Hallway Navigation** as the prediction model improves. RCIP is able to keep the plan success rate relatively constant even for poorly performing predictors. As the no-help predictions become better, RCIP also asks for less help.

Method	$1-\alpha_{\text{cov}}$	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
RCIP	0.85	0.86	**0.00**	0.90	**0.00**
KnowNo [36]	0.85	0.88	0.58	0.95	0.58
Simple Set	0.83	0.86	0.56	0.94	0.56
Entropy Set	–	0.79	0.38	0.93	0.21
No Help	–	0.61	0	0.81	0

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

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 optimization (PPO) [41, 48]. The human and robot PPO policies are trained jointly using 256 environments and take about 4 hours to train.

We present results for Hallway Navigation in Table I at the fixed coverage level $\alpha_{\text{cov}} = 0.85$. Overall, the plans generated using RCIP satisfy the target success level of 85% while needing 31% less help than the KnowNo [36] baseline. We additionally study various model calibration levels and their effect on the RCIP help rate. We present results in Table II.

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.

B. Simulation: Social Navigation in Habitat

Habitat [47] 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

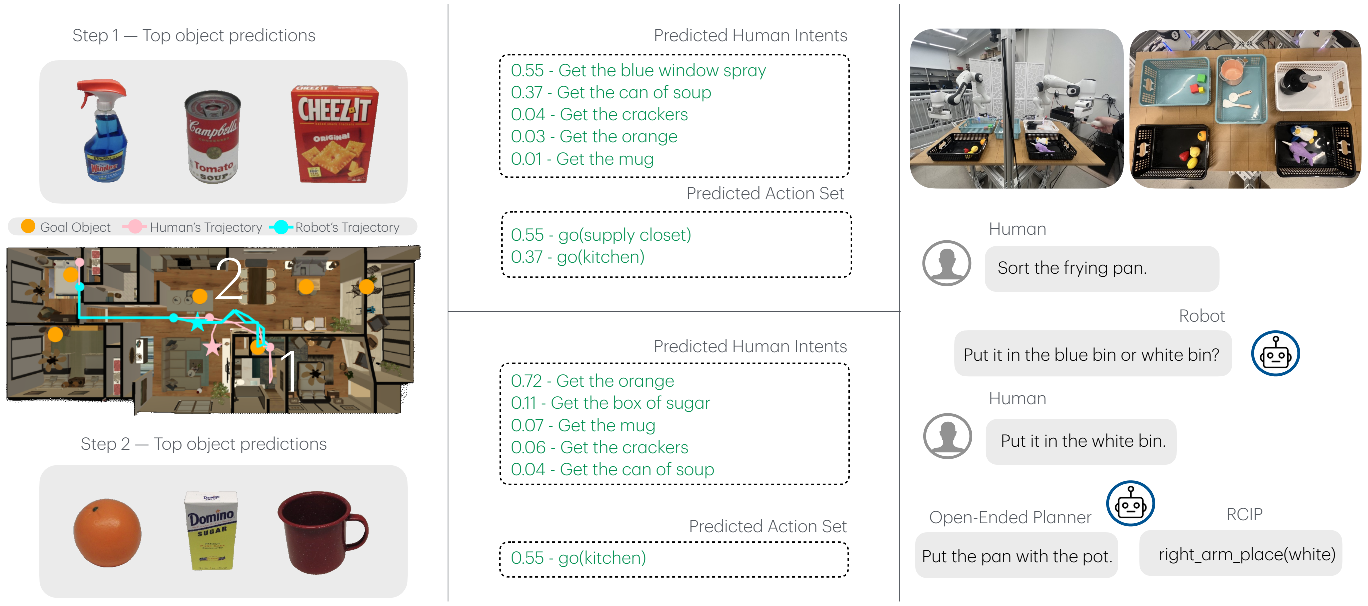


Fig. 4: (Left, Center) Multi-step RCIP is applied in **Social Navigation**. The human's trajectory is shown in pink, and the robot's trajectory is shown in blue. The human's possible goal objects are shown in orange. (Right) Single-step RCIP is applied in **Bimanual Sorting**. Open-ended language planners, such as the large language model employed by KnowNo [36], may generate plans that are technically correct, but ambiguous to execute for a language-conditioned policy (e.g., both the blue and white bin have a pot). RCIP instead guarantees that the human's intent is satisfied with the intent-conditioned planner.

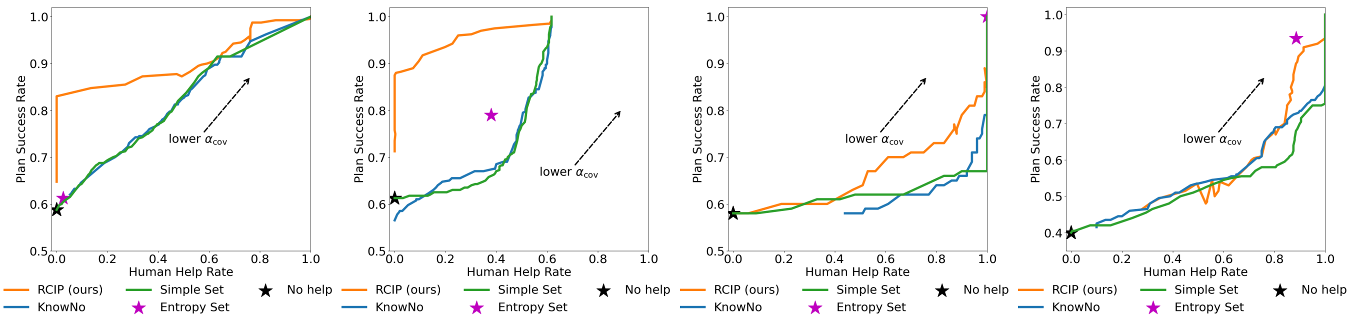


Fig. 5: Baseline comparison for RCIP versus other set-valued predictors for all tasks. RCIP consistently requires less help to achieve a specified plan success rate than other baseline methods. RCIP provides a framework for tuning model parameters to achieve risk control, versus other methods that assume that model parameters are held fixed: KnowNo [36], Simple Set, Entropy Set, and No Help.

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. 4). Each scene contains $5-10$ objects of interest from the YCB dataset [49]. 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, the set of intents is $\mathcal{Z}=[N_o]$, 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. (11). 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

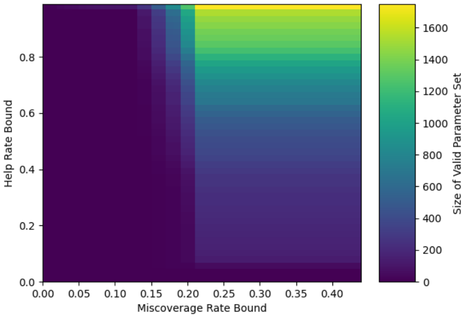


Fig. 6: Ablation study on the effect of action miscoverage and help rate risk limits versus FWER-controlling parameter set size for RCIP on **Hallway Navigation** using $\alpha_{\text{cov}} \in [0, 0.45]$ and $\alpha_{\text{help}} \in [0, 1]$. The color denotes the size of the set of FWER-controlling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

Method	$1-\alpha_{\text{cov}}$	Plan Succ.↑	Plan Help↓	Step Succ.↑	Step Help↓
RCIP	0.75	0.76	**0.87**	0.94	**0.93**
KnowNo [36]	0.75	0.75	0.98	0.95	0.98
Simple Set	0.59	0.76	1.00	0.96	1.00
Entropy Set	–	1.00	1.00	1.00	1.00
No Help	–	0.58	0	0.88	0

TABLE IV: Results for **Open-Category Sorting**. The optimal action miscoverage rate is held fixed between RCIP, KnowNo, and Simple Set for comparing the other metrics.

Method	$1-\alpha_{\text{cov}}$	Plan Success ↑	Execution Success ↑	Plan Help ↓
RCIP	0.75	0.75	0.67	**0.85**
KnowNo [36]	0.75	0.75	0.60	0.93
Simple Set	0.75	0.75	0.67	0.95
Entropy Set	–	0.94	0.83	0.88
No Help	–	0.40	0.27	0

TABLE V: Results for **Bimanual Sorting**. The optimal action miscoverage rate is held fixed between RCIP, KnowNo, and Simple Set for comparing the other metrics. We use a random set of 30 scenarios from the test set to evaluate the execution success rate for each method.

Social Navigation in Table III. Notably, due to the effect of temperature scaling and the many-to-one mapping from intents to actions, RCIP reduces the the help rate by 58% versus baseline methods. We defer additional discussion, such as intuition on temperature scaling, to Appendix C.

C. Hardware: Open-Category Sorting

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's trajectory is constrained such that the object must be placed in the correct bin. The human's intent space $\mathcal{Z} = \{1, 2, 3\}$, where each number corresponds to one of the bins. The robot interacts with the human over $T = 6$ environment time steps, and the human selects a new sorting plan every $T_z = 1$ time step.

To predict the human's intent, 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 use GPT-4V (`gpt-4-turbo-2024-04-09`) [4] to process the image and predict (e.g. the carrot in Fig. 1) a confidence score for each bin via multiple-choice question and answering (MCQA) [50, 51]. 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 IV that RCIP reduces the plan-wise help rate by 11% and the step-wise help rate by 5% in Open-Category Sorting. 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. For more hardware implementation details, please see Appendix D.

D. Hardware: Bimanual Sorting

In this experiment (Fig. 4), a bimanual setup with two Franka Panda arms is asked to help the human sort a common household item: frying pan, plushie, wooden block, carrot, or grape. We initialize each bin to contain similar objects; for example, bins can contain other cookware, plushies, wooden blocks, vegetables, or fruit. There are five bins total: two bins are on each side of the table, and one bin is in the middle of the table. The workspace of both arms is limited such that each arm cannot reach the two bins at the other end of the table; thus, there is ambiguity arising from which arm the human wants to help sort the item. Additionally, duplicate categories provide intent ambiguity: each of the initializations is selected at random and may appear up to twice in the five bins. The robot must then interpret a human instruction and infer the set of bins that it can place the object into (Bin 1, Bin 1 or Bin 2, Bin 1, Bin 3, or Bin 5, etc). Therefore, the human's intent space $\mathcal{Z}$ is the *power set* of possible bins, i.e., $\mathcal{Z} = \mathcal{P}(\{1, 2, 3, 4, 5\})$. Hence, there are $|\mathcal{Z}| = 31$ possible intents for this task (when excluding the empty set). The robot interacts over a single time step, so $T = T_z = 1$. We show in Table V that RCIP reduces the plan-wise help rate by 8%.

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, 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 [52]–[54] 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 $8-58\%$ 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] 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.

[4] 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.

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

[42] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[43] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

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

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

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

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

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

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

[54] 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.

[55] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[56] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Schacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva, "Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation," *arXiv preprint arXiv:2306.11290*, 2023.

[57] 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.

[58] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[59] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[60] H. Xi, J. Huang, L. Feng, and H. Wei, "Does confidence calibration help conformal prediction?" *arXiv preprint arXiv:2402.04344*, 2024.

[61] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[62] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[63] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

APPENDIX

A. Additional Experiment Details

Environments. In addition to Fig. 1, Fig. 3, Fig. 4 (left) and Fig. 4(bimanual). Fig. 7 shows the map and object set from the Social Navigation task. Fig. 8 shows the full object set from the hardware Open-Category Sorting and Bimanual Sorting.

Scenario Distribution and Calibration Dataset. Here, we provide more details on the on the parameterization of the scenario distribution for the four experiment settings. For each setting, we state the time horizon (single or multi-step), the environment details, the human intent distribution, and the robot's goal. With each distribution defined, the calibration datasets are then generated by sampling 400 i.i.d. scenarios from the distributions over scenarios. Specific to the Open-Category Sorting and Bimanual Sorting tasks, we augment the RCIP context with a language instruction since these tasks require querying a vision-language model. Each setting is as follows:

Simulation: Hallway Navigation.

- Horizon: multi-step (10 steps – 1 step per 20 environment timesteps).
- Environment: two cars placed in opposite ends of an area that is 16 meters long and 9 meters wide. The arena is bottlenecked by five 1-meter wide parallel hallways spaced evenly throughout the arena. Each car is 0.5 meters wide and has a safety radius of 0.5 meters, and therefore two cars cannot take the same hallway without violating safety specifications. Each car is initialized with a random position and heading within their respective sides of the arena.
- Intents: Each of the 5 hallways, with equal spacing between each hallway. The intent set Z is {1,2,3,4,5}. At each episode, the intent is sampled from Z with uniform probability.
- Task: reach the other end of the arena without colliding with the human or with the walls.

Simulation: Social Navigation.

- Horizon: multi-step (6 steps – 1 step per 100 environment timesteps).
- Environment: Our Habitat simulation environment consists of 77 objects from the YCB dataset [55], up to 10 of which are randomly sampled per scene. Scenes are randomly sampled from the Habitat Synthetic Scenes Dataset [56], which is a dataset of 211 high-quality 3D scenes containing over 18k individual models of real-world objects. The human and robot are placed in one of three randomly selected, feature-rich environments shown in Figure Each environment contains a variety of objects that the human can interact with, shown in Figure . The robot has a safety radius of 1 meter such the task is failed if the safety radius is violated through contact with the human. There are several hallways, doors, and passageways that the robot and human cannot simultaneously pass through without colliding.
- Intents: Each of the N_o objects in the scene. The intent set $\mathcal{Z} = \{1,2,3,...,N_o\}$. At each episode, the initial intent is sampled from $\mathcal{Z}$ with uniform probability. Then, every 100 timesteps, the human chooses a new object.
- Task: reach a series of human goal objects without colliding with the human.

Hardware: Open-Category Sorting

- Horizon: multi-step (6 steps).
- Environment: there are eight items to be sorted placed randomly on the table, and there are three bins: white, black, and blue. The human initially fills each bin with 5-10 objects from the object set shown in Fig. 1. Each bin satisfies a single grouping from the set of 10 groupings shown in Fig. 1. The groupings are: fruit, vegetables, toys (set 1), toys (set 2), red, orange, cookware, tableware, sauces, and blocks.
- Intents: The human's intent is represented as a choice of one of the 3 bins. The intent set Z is {1, 2, 3}. There is considerable ambiguity between the categories for many scenarios; for example, a carrot is both orange and a vegetable (see Fig. 1). GPT-4V is asked to score the human's intent based on an image of the object to be sorted and three images of the goal bins.
- Task: place the eight items in the bin that the human thinks is the best match for the object.
- Language context: To augment the RCIP context, GPT-4V is given the following prompt, "The first three images are different bins sorted by a human. Describe the items in each of the bins." We additionally provide the prompt, "Here is a description of three bins and an object we want to sort. VLM description Which bin should we place the object in?" The 3 choices are then listed via MCQA.

Hardware: Bimanual Sorting

- Horizon: single-step.
- Environment: there are five objects that the human can ask the robot to sort: frying pan, carrot, grape, block, and blue plushie. One of the objects is selected uniformly at random. There are five bins placed on the table. Each bin contains objects from the following categories: cooking, fruits, vegetables, blocks, and plushies. Three bins are fixed (fruits, cooking, and plushies), and two bins are drawn uniformly at random (possibly with duplication) from the set of categories. There is a human standing at the left, middle, or right side of the table, with their position being drawn uniformly at random.

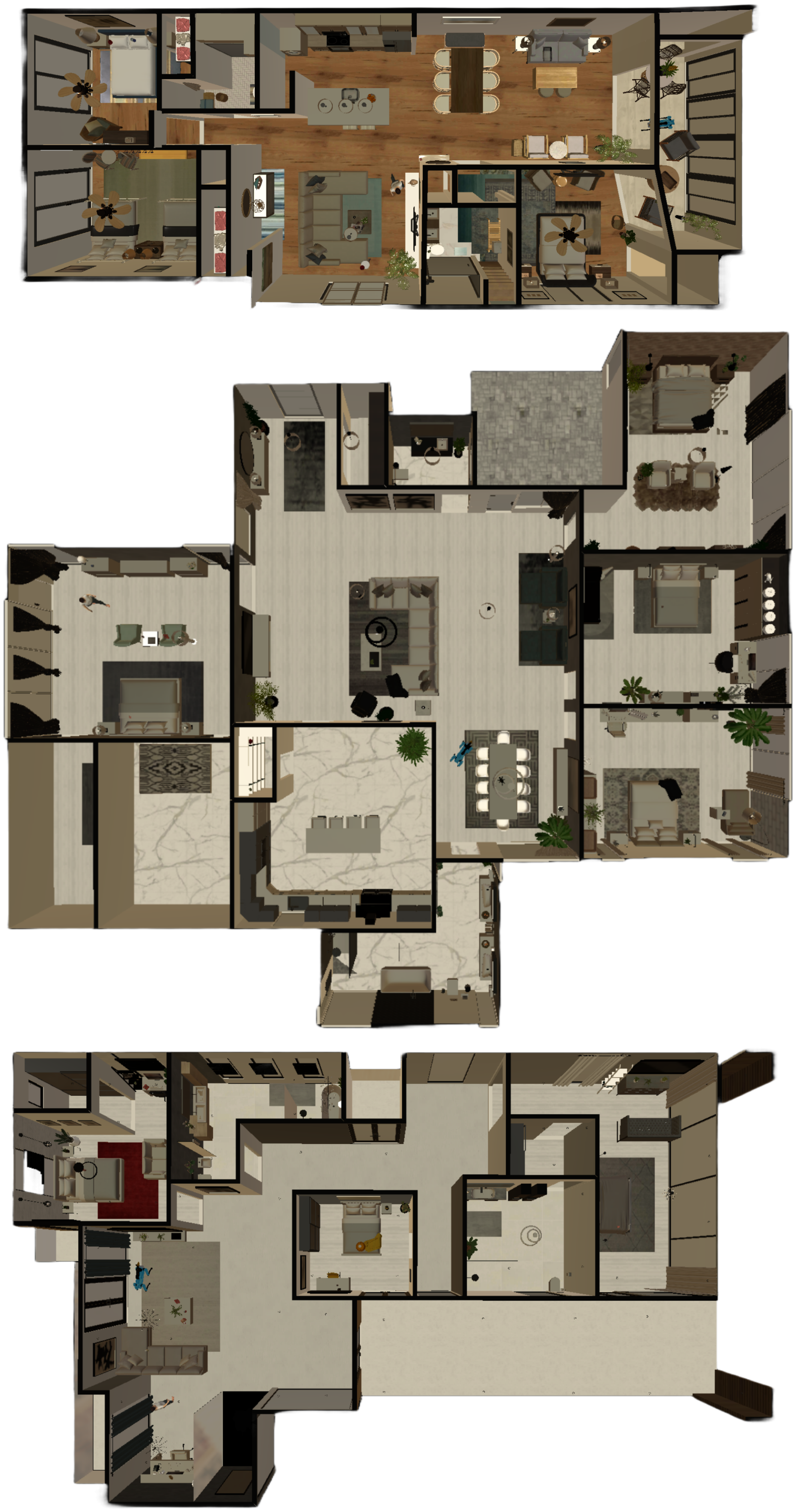


Fig. 7: Three sample maps from **Social Navigation**. The human and robot are tasked with finding and collecting a series of objects (e.g. cans of soup, crackers) around a close-quarters living space. The robot must recognize the human's intent and either follow or evade the human depending on the human's desired object. The robot must minimize action miscoverage across a variety of human intents and environments.

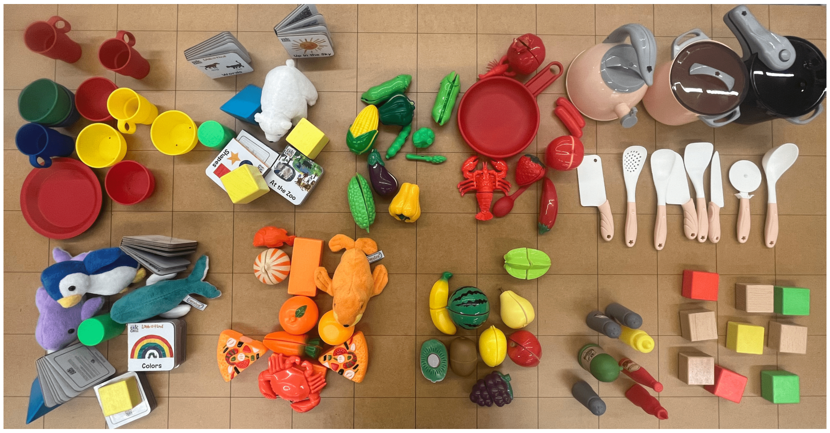


Fig. 8: Full object set for Open-Category Sorting and Bimanual Sorting

Transformer Encoder Layers	6
Transformer Decoder Layers	6
MLP Layers	256
Hidden Dimension	256

TABLE VI: Intent prediction architecture for Hallway and Social Navigation.

Initial Learning rate	1e-4
Learning rate decay epochs	[50,80,90,100]
Learning rate decay factor	0.5
Total Epochs	200
Batch size	64
Prediction Horizon	100
λ	1

TABLE VII: Intent prediction training details for Hallway and Social Navigation.

- Intents: power set of the five bins for the object to be placed, $\mathcal{P}(\{1,2,3,4,5\})$. For example, if there are two vegetable bins 1, 2 on the left and right side, the human could instruct the robot "place the carrot with the vegetables", and could mean either vegetable bin 1, 2. Thus, there are 31 intents total that each represent a combination of possible placements (the empty set is excluded).
- Task: move the object to the specified location that satisfies the human intent. Minimize the distance between the object and the closest intended bin.
- Language instruction (part of RCIP context): To augment the RCIP context, GPT-4V is given the following prompt: "This is a bimanual robot setup with a human standing on the other side of the table. Describe the human's relative position on the table, the object the human is holding, the five bins, and their contents. The human has instructed the robot instruction. What could be the human's intent with the object? " The 31 choices are then listed and scored via MCQA. The instruction is chosen at random from the following list:
  1) put it in the bin
  2) put it with the others of the same color
  3) put it in the bin closest to me
  4) sort the object
  5) place it with the similar objects
  6) put it in any of the bins
  7) sort it with the left arm
  8) sort it with the right arm

B. Additional Model Details

We now give additional predictor and policy details for each setting. We detail the architecture, training procedure, input representation, and output representation for each learned model.

Simulation: Hallway Navigation

- Intent Predictor: To predict the human car's intent, we train a transformer-based prediction model similar to [3, 57] to output a probability distribution over the hallway in addition to the future position of the human car. We encode the

Learning rate	1e-4
Actor MLP Layers	3
Critic MLP Layers	3
Hidden Dimension	64
Discounting factor γ	0.99
Environments	256
Batch size	4096
Total Epochs	800

TABLE VIII: PPO Training Parameters for learning the intent-conditioned policy in Hallway.

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. The full list of paramters are in Table VI.

- Predictor training: To train the model, we use the following loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{MSE} \tag{26}$$

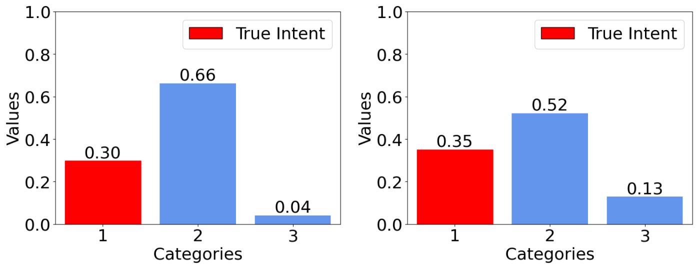
  where $\mathcal{L}_{CE}$ is the cross entropy of the predicted intent distribution versus the true ground truth label, $\mathcal{L}_{MSE}$ is the mean square error from the human car's ground truth state, and λ is a scalar that controls the relative weight of the state prediction task. We find the training 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. The full set of training parameters are in Table VII.
- Input representation: The predictor takes as input a matrix $X = (X_1, \ldots, X_5) \in R^{T \times 120}$ containing the concatenation of the observation histories for each possible intent. Each intent-conditioned observation $X_z \in R^{24}$ contains the joint position and heading history of both cars, their relative heading and position, the distance from each car to each hallway, and the distance from each car to their respective goals.
- Output representation: The predictor outputs log-softmax scores corresponding to each of the five hallways that the human could possibly take.
- Policy: We train an intent-conditioned 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. The full set of PPO training parameters are shown in Table VIII.

Simulation: Social Navigation

- Intent Predictor: We use an identical prediction architecture and training procedure as that of the Hallway Navigation task. Since the number of objects may change across the environments, we use the object locations as anchors for the transformer decoder similar to the strategy used by [3].
- The training setup is identical to the Hallway Navigation task.
- Input representation: The predictor takes as input a matrix $X = (X_1, \ldots, X_{N_o}) \in R^{T \times 350}$ containing the concatenation of the (zero-padded) observation histories for each possible intent. Each intent-conditioned observation $X_z \in R^{35}$ contains the joint position and heading history of both agents, their relative heading and position, the distance and heading from each human goal object. The maximum possible objects in a given scene N_o is 10.
- Output representation: the predictor outputs log-softmax scores corresponding to each of the N_o goal objects in the scene.
- Policy: To accommodate the close-quarters, constrained motion planning needed for social navigation, we use Habitat's [47] 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.

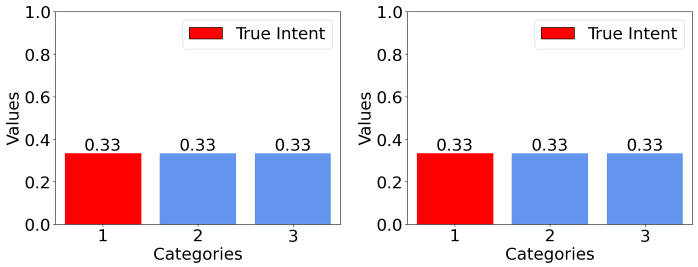
Hardware: Open-Category Sorting

- Intent Predictor: We use GPT-4V to select the intent using MCQA (labeling the bins as A: Blue Bin, B: White Bin, C: Gray Bin, or D: Not sure). Specifically, we obtain log-softmax scores from GPT-4V for the four MCQA options. We multiply the log-softmax scores by the optimal temperature obtained by RCIP before taking the softmax probabilities for each intent.
- Training: We do not perform any training or fine tuning for this setup.
- Input representation: The input to the VLM captioner is a set of four images: one image of the object to be sorted, and three images of the bins.
- Output representation: The output of the prediction model is a set of softmax scores over the three bins (choices A, B, C), and an additional choice for the model not being sure (D).
- Policy: We use a pick-and-place primitive to grasp the object of interest and place it into the human's intended bin. Since the policy is not learned, there is no training involved.



(a) Temperature = 2 (KnowNo, before rescaling) (b) Temperature = 1 (RCIP, after rescaling)

Fig. 9: Hypothetical predictor outputs for scenario A with intent space $\mathcal{Z} = \{1,2,3\}$. Since the confidence for the true intent is less than the maximum confidence intent, the prediction is *incorrect*. However, after temperature rescaling, the predictor is more confident in the true intent. The ground-truth intent is shown in red.



(a) Temperature = 2 (before rescaling) (b) Temperature = 1 (after rescaling)

Fig. 10: Hypothetical predictor outputs for scenario B with intent space $\mathcal{Z} = \{1,2,3\}$. The ground-truth intent is shown in red. Assume that the first intent is chosen in the event of a tie; thus, the prediction is *correct*. Since the predictor is maximally *uncertain*, temperature rescaling does not affect the confidence scores.

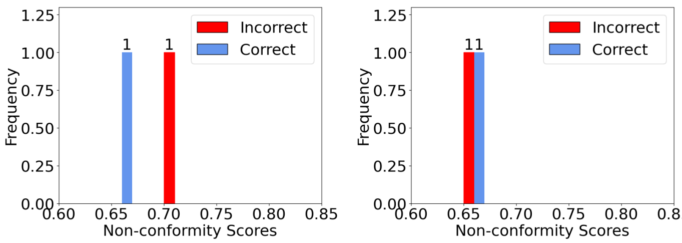
Hardware: Bimanual Sorting

- Intent Predictor: We use GPT-4V to rank the possible intents. We then select the final intent using MCQA on the top 20 intents. Specifically, we obtain log-softmax scores from GPT-4V for the 20 MCQA options. We multiply the log-softmax scores by the optimal temperature obtained by RCIP before taking the softmax probabilities for each intent.
- Training: We do not perform any training or fine tuning for this setup.
- Input representation: The input to the VLM predictor is a single image of the scene.
- Output representation: The output of the prediction model is a set of softmax scores over the top-5 intents from the 31 possible bin combinations.
- Policy: We use a pick-and-place primitive to hand over the object from the human and place it in one of the intended bins. Since the policy is not learned, there is no training involved.

C. *Additional Discussion*

Optimizing model temperature with RCIP enables better coverage than conformal prediction alone. Temperature scaling is an effective ad-hoc calibration technique in neural network prediction [58]–[60]. However, achieving optimal temperature calibration is usually a tedious process that requires hand-tuning. Instead, RCIP provides a lightweight procedure for achieving *automatic* temperature calibration with performance guarantee on downstream risk.

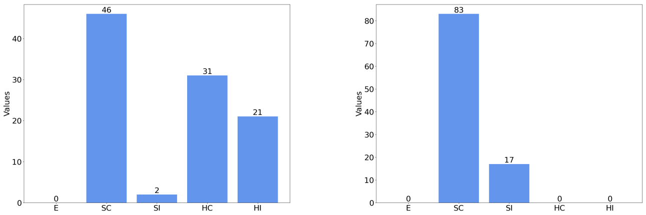
Temperature scaling can be used to increase the rate of singleton sets less-confident, but correct, predictors, resulting in a reduced rate of human intervention for a given coverage level. Conversely, temperature scaling can also be used to reduce the rate of incorrect prediction sets for overconfident predictors. To illustrate this point, we present a toy example where a prediction model is tasked with predicting a distribution over three intents over two contexts. Consider the confidence scores for a distribution of the two scenarios: Scenario A in Fig. 9 and Scenario B in Fig. 10. Depending on the *true intent*, the predictor's raw (Temperature equals 1) confidence scores can be scaled to be more or less confident in the true intent. Here, the raw scores provide better calibration than a higher temperature (Temperature equals 2).



(a) Temperature $=2$ non-conformity scores (before rescaling)

(b) Temperature $=1$ non-conformity scores (after rescaling)

Fig. 11: Non-conformity scores before and after temperature rescaling.



(a) Temperature $=1$ non-conformity scores (before rescaling)

(b) Temperature $=0.01$ non-conformity scores (after rescaling)

Fig. 12: Distribution of help rates and correctness of prediction sets on the test set of **Hallway Navigation** before and after rescaling. E = Empty, SC = Singleton (Correct), SI = Singleton (Incorrect), HC = Help (Correct), SI = Singleton (Incorrect).

Now, consider the case where we want to ensure maximal coverage ($\alpha_{\text{cov}}=0$) with minimal help. To ensure coverage of scenario A, the robot needs to ask for help with a prediction threshold $\lambda \geq 0.30$ when the temperature is 2. Thus, this results in a 100% success rate, since the prediction for Scenario B is correct. However, Scenario B has a confidence in the ground-truth intent of 0.33. Therefore, taking $\lambda=0.3$ forces the predictor to ask for help in Scenario B even when it is correct. The help rate for this temperature is 100%. Fig. 11 (left) shows the non-conformity scores for this temperature, defined as $1-\texttt{confidence}(z_{\text{true}})$. To ensure coverage of the right-most (least confident) intents, the predictor has to ask for help for every intent to the left.

After the temperature is rescaled to 1, the robot now has 0.35 confidence in the true intent for Scenario A. Since this confidence now exceeds that of the true intent for Scenario B, we can set a prediction threshold of $\lambda \geq 0.35$. This ensures 100% coverage with only 50% help. In Fig. 11 (right), the non-conformity score for Scenario A is now to the left of the non-conformity score for Scenario B, meaning that the predictor's incorrect predictions are those that it is least confident in. To illustrate this phenomenon on a larger scale, we show a histograms for empty, singleton, and help-needed prediction sets for Hallway Navigation in Fig. 12.

RCIP enables better coverage of intents than open-ended language planning. In the Experiments (c.f. Section V), we assume that the KnowNo [36] baseline has access to the intent set. However, in the framework original proposed in [36], the robot plans are generated by an LLM and need to be actuated using a language-conditioned policy such as RT-2 [61], which may present challenges for downstream intent satisfaction. To illustrate this challenge, we show an example scenario from the Bimanual Manipulation experiment. The prompt image is show in Fig. 13 and the generated plans are shown in Fig. 14. Several of the open-ended plans (marked in orange, red) are either ambiguous (resulting in the language-conditioned policy to confuse the human's intent, and causing the human's help to be ineffective) or fictitious (e.g., several descriptive objects proposed by GPT-4V do not exist). We envision that RCIP can be a useful alternative to open-ended language planning in situations where generated plans are not semantically meaningful.

D. Additional Implementation Details

Perception. In the Hallway and Habitat experiment, we assume that the robot has full observability of the human's position. In the Open Category Sorting Experiment, we apply several filters to extract the pose of the target for sorting. We first apply create a depth mask using the RGB-D image to filter out foreground (e.g., the Franka Panda end effector) and background (e.g., the floor). For segmentation, we use the Segment Anything [62] model with uniform sampling over the image to obtain masks for any potentially relevant objects. Using the depth filter, masks for irrelevant objects (e.g., each grid square on the

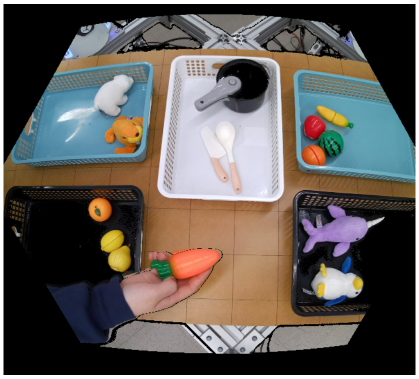


Fig. 13: Example image prompt from the Bimanual Sorting scenario distribution.

[redacted]

4. **Add the carrot to the bin with the plush toys (next to the plush dinosaur).**
5. **Place the carrot in the empty blue bin, starting a new category for root vegetables.**

Here are five different sorting plans for the carrot:

1. **Place the carrot in the bin with the other vegetables (the one with the pepper, corn, and possibly a tomato).**
2. Put the carrot in the bin with the cooking utensils, suggesting it's ready to be cooked.
3. **Sort the carrot into the bin with the plush toys, making it a play food item for a child.**
4. Include the carrot in the bin with the orange and lemons, grouping it by color.
5. **Set the carrot in the empty bin, indicating it's a start for a new category or type of items.**

Here are five different plans for sorting the carrot:

1. **Place the carrot in the bin with the other vegetables, alongside the pepper.**
2. **Sort the carrot into the bin with the fruits, if considering color coordination with the orange.**
3. Put the carrot in the bin with the cooking utensils, as it can be used as an ingredient in cooking.
4. **Choose the bin with the plush animals for the carrot, grouping it by texture since the plush toys and the carrot might have a similar softness.**
5. **Finally, you could place the carrot in the empty bin, dedicating it to additional items that may be found later.**

Fig. 14: Five example open-ended language plans generated by GPT-4V for the example image shown in Fig. 13. Non-existent objects are marked in red. Ambiguous objects are marked in orange. Open-ended language plans used in the original KnowNo [36] framework may pose a significant challenge for language-conditioned policies (e.g., with RT2 [61]) due to semantic mismatch. RCIP avoids open-ended language planning by pre-specifying a fixed set of intents and using an intent-condtioned planner, ensuring consistency with the human's intent.

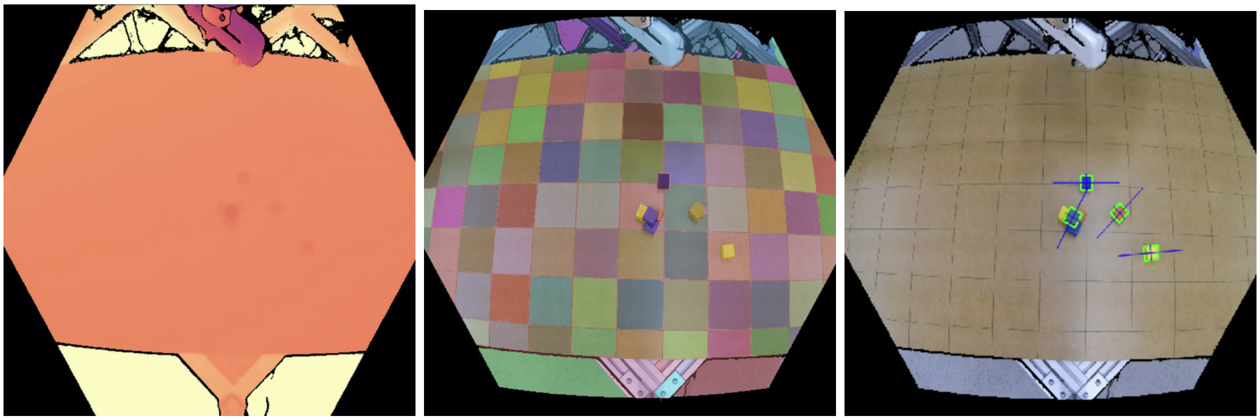


Fig. 15: Perception pipeline for manipulation tasks: (left) depth masking, (middle) semantic segmentation with Segment Anything [62], (right) pose inference using OpenCV [63].

table) are removed. For each Segment Anything mask, a rectangle is fitted to the contour that defines the mask's boundary using OpenCV [63]. For each object, the position and yaw in the robot's coordinate frame is determined using the centroid and rotation for the rectangle approximation. The bimanual task does not require precise object localization due to the transfer between the human and the robot, so the Azure Kinect's RGB image is instead passed directly to GPT-4V.

Low-level control. For the Hallway and Habitat setups, we directly learn a low-level policy using kinematic state information. The Hallway car follows a 4D kinematic bicycle model with a two dimensional control input corresponding to target velocity and steering angle. The Habitat spot robot follows a 2D kinematic unicycle model with direct velocity and steering angle commands. For the sorting and bimanual experiments, we solve trajectories for pick-and-place maneuvers using inverse kinematics. Since the bimanual task requires object transfer between the human and robot at variable heights, we use simple scripted primatives to take the object from the human at various positions around the table (left, center, and middle) and drop the object in each of the five bins.

Human Feedback. In RCIP, we directly prompt the human for their intent by giving a numbered list for each intent. The robot picks the low-level action conditioned on this intent, allowing the human to give high-level information while the robot distinguishes between low-level action. Since intents can be many-to-one, clarification in the intent space is equivalent to clarification in the action space as long as the robot can compute the action corresponding to the human's true intent.