Conformalized Teleoperation: Confidently Mapping Human Inputs to High-Dimensional Robot Actions

Michelle Zhao, Reid Simmons, Henny Admoni, Andrea Bajcsy
Carnegie Mellon University, Robotics Institute
{mzhao2, rsimmons, hadmoni, abajcsy}@andrew.cmu.edu

Abstract—Assistive robotic arms often have more degrees-of-freedom than a human teleoperator can control with a low-dimensional input, like a joystick. To overcome this challenge, existing approaches use data-driven methods to learn a mapping from low-dimensional human inputs to high-dimensional robot actions. However, determining if such a black-box mapping can *confidently* infer a user's *intended* high-dimensional action from low-dimensional inputs remains an open problem. Our key idea is to adapt the assistive map at training time to additionally estimate high-dimensional action quantiles, and then calibrate these quantiles via rigorous uncertainty quantification methods. Specifically, we leverage adaptive conformal prediction which adjusts the intervals over time, reducing the uncertainty bounds when the mapping is performant and increasing the bounds when the mapping consistently mis-predicts. Furthermore, we propose an uncertainty-interval-based mechanism for detecting high-uncertainty user inputs and robot states. We evaluate the efficacy of our proposed approach in a 2D assistive navigation task and two 7DOF Kinova Jaco tasks involving assistive cup grasping and goal reaching. Our findings demonstrate that conformalized assistive teleoperation manages to detect (but not differentiate between) high uncertainty induced by diverse preferences and induced by low-precision trajectories in the mapping's training dataset. On the whole, we see this work as a key step towards enabling robots to quantify their own uncertainty and proactively seek intervention when needed.

I. INTRODUCTION

Robotic arms designed for assistance typically possess more degrees of freedom (DoF) than can easily be controlled by a human using low-dimensional inputs such as a joystick [53] or a sip-and-puff device [3, 23]. In practice, teleoperating a high-DoF robot can require frequent mode switches that control individual components of the arm during even very basic everyday tasks [15, 48, 52]. For this reason, a core challenge in assistive teleoperation is building mappings that translate low-dimensional human inputs into the human's desired high-dimensional robot actions in an intuitive way [39].

Recently, these low-to-high dimensional mappings have been increasingly learned from human data, wherein annotators provide state, high-DoF action, and sometimes low-DoF input pairs for training [39, 36, 29]. However, quantifying the trustworthiness of these black-box assistive teleoperation systems is an open problem. We find that off-the-shelf, these learned mappings can be highly sensitive to multimodality and suboptimal data in the training distribution. Furthermore, when faced with out-of-distribution human inputs at deployment time, the robot can overconfidently map the human's low-DoF input to an extremely incorrect high-DoF robot action. At best,
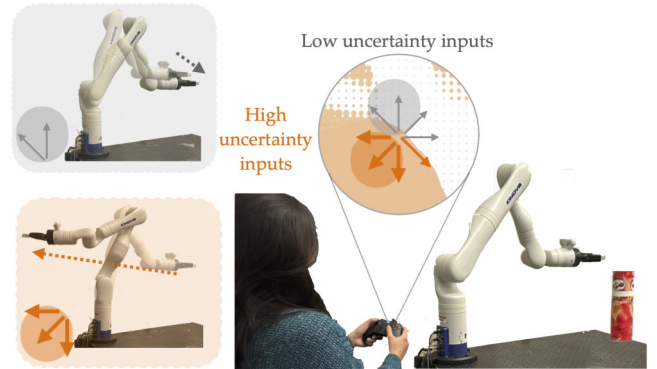

Fig. 1: **Conformalized Teleoperation.** We leverage conformal methods to quantify if the robot's learned controller can reliably lift the human's low-DoF input (joystick) to their desired high-DoF action (7 joint velocities). For any joystick input at the current state, the robot can assess its uncertainty in remapping that input (dot size in the expanded view is proportional to uncertainty at that coordinate). Arrows emphasize directional joystick input. (left, top) If the human pushes *up* or to the *left* on the joystick, the robot has low uncertainty, because it knows with high probability the person wants to go forward towards the object. (left, bottom) If the human pushes *backwards* on the joystick, the robot predicts a large pivot backwards, but is rightfully uncertain this is what the human intended.

this can cause frustration from the end-user; at worst, it can be a safety hazard.

In this work, we seek to rigorously quantify if a learned assistive controller can *confidently* infer a user's *intended* high-dimensional action from only low-dimensional states and inputs. Our key idea is:

> *by training the robot's controller to estimate high-DoF action quantiles, we can calibrate the uncertainty via adaptive conformal prediction techniques.*

Specifically, we leverage the statistical technique of adaptive conformalized quantile regression [17], which is relevant for regression problems and provides asymptotic coverage guarantees (i.e., human's true high-DoF action is included within the uncertainty interval) without strong distributional assumptions. Furthermore, the adaptive nature of this paradigm enables the

calibration procedure to grow and shrink uncertainty intervals over time as the end-user's input is poorly or well-predicted by the robot's learned controller. As such, we call our approach Conformalized Teleoperation. With this rigorous quantification of the robot's uncertainty, we also propose a simple method to detect high-uncertainty states which can be used to alert the end-user or safely stop operation.

We evaluate how well our approach calibrates in a 2D assistive navigation task, and two 7DOF Kinova Jaco tasks involving assistive cup grasping and goal reaching. Our experiments indicate that naturally-occurring diversity in the training dataset, such as multiple preferences for how a task should be completed or low-precision behaviors, can significantly impact the robot's learned assistive control confidence. Beyond this, we find that our approach is capable of detecting critical high-uncertainty states and inputs for both in- and out-of-distribution end user behavior. Overall, we see this work as an important first step towards robots that rigorously quantify their own uncertainty in the assistive teleoperation domain. Code is available at https://conformalized-assistive-teleoperation.github.io/.

II. RELATED WORK

Assistive Teleoperation. Assistive robots enable users with physical impairments to perform a variety of everyday tasks [8, 43]. Teleoperating a high-DoF robot can be difficult and often requires frequent mode switches to accomplish simple tasks [15, 48, 52]. One of the key challenges is the low-to-high-dimensional mapping induced by low-DoF controllers like joysticks [53], sip-and-puff devices [3, 23], or eye gaze [2]. Prior works on dimensionality reduction have shown increasingly sophisticated methods for designing or learning a mapping from the user's input to the robot's high-DOF control action [39, 24, 40, 19]. Alternatively, the shared control paradigm combines low-DoF human input with autonomous robot policies in order to make teleoperation of the robot more fluent [7, 6]. Other works considers the inverse approach of developing high-dimensional interfaces to control high-DoF robots [35]. In this work, we draw upon the data-driven approaches [39, 40] but focus on getting statistical performance assurances on the output of the model [1, 17] when the assistive mapping is a neural network.

Uncertainty-Aware Shared Control. In shared control paradigms, robots must distinguish intents of the human operator [7]. Uncertainty-aware shared control paradigms can elect to prioritize human control, handing off control to the human operator, while constraining robot inputs to maintain safety [6], or suggesting a control mode in which human input maximally disambiguates their intent [19]. Prior works maintain an estimate of intent confidence for low-dimensional hand-designed intent spaces [59, 14] or, for data-driven estimators, when relying on data-driven based methods like learned encoder-decoder architectures, use reconstruction error as a signal of intent inference quality [26]. When intent inference is detected to be poor, control is handed over to the human. In

the latter category, methods assume known human intent but focus on preventing collisions during robot assistance planning [54, 57]. In a similar vein as our work, adaptive shared control approaches predict human actions online and adjust robot guidance as a result of uncertainty in the robot's inferences [20]. However, we translate ideas from conformal prediction to get rigorous uncertainty bounds for learned robot assistive controllers.

Uncertainty Quantification seeks to estimate the confidence a model has in its predictions. Bayesian approaches such as Bayes filters [10, 11] and Bayesian neural networks [18, 34, 32] quantify uncertainty via posterior updates. In deep learning other popular uncertainty quantification methods include Monte Carlo dropout [16, 58, 49] and deep ensemble approaches [42, 33, 25]. Other ensemble model disagreements can also be used such as random forests [5] and bagging [4]. We ground our approach in conformal prediction [1], specifically adaptive conformal inference [17], an increasingly popular method which provides coverage guarantees without assumptions on the data distribution nor model assumptions.

Conformal Prediction for Robotics. Conformal prediction is a paradigm for constructing rigorous prediction intervals for both classification and regression problems [1, 46, 47, 55]. The method has been explored in a variety of robotics contexts for providing collision-avoidance assurances [9, 38, 13, 44, 51, 12, 37], calibrating early warning systems [41] and large language models [45]. We take inspiration from these recent successes of conformal prediction techniques applied to robotics, and propose how to incorporate these advances into the assistive teleoperation domain.

III. PROBLEM FORMULATION

In this work, we seek to rigorously quantify how *confidently* a learned assistive controller can infer a user's *intended* high-dimensional action from low-dimensional inputs and states. Here, we formalize the key components of our problem setup.

Robot States & Actions. We model the state as $s \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$, consisting of the robot's joint angles and relevant object poses in the scene, and the robot's high-dimensional action as $a \in \mathcal{A} \subseteq \mathbb{R}^{n_a}$ (e.g., joint velocities). The robot's state evolves via the discrete-time dynamics $s^{t+1} = f(s^t, a^t)$.

Human Teleoperation. We model the human teleoperator as desiring to execute a specific high-DoF action on the robot: let $a_{\mathcal{H}} \in \mathcal{A}$ be the human's desired high-dimensional action. However, the human teleoperates the robot via a *low-dimensional* interface, $u_{\mathcal{H}} \in \mathcal{H} \subseteq \mathbb{R}^{n_u}$. For example, if $n_u = 2$ then this could be a 2-dimensional joystick that the human uses to control a robot's 7 joint velocities, $n_a = 7$.

Learned Low-to-High-Dimensional Controller. To enable easier teleoperation, we study learned assistive controllers [36, 39, 27, 29] which predict a human's desired high-dimensional action from the current state and low-dimensional input. Let this controller be:

$$\hat{a}_{\mathcal{H}}^t = f_\theta(u_{\mathcal{H}}^t, s^t) \tag{1}$$

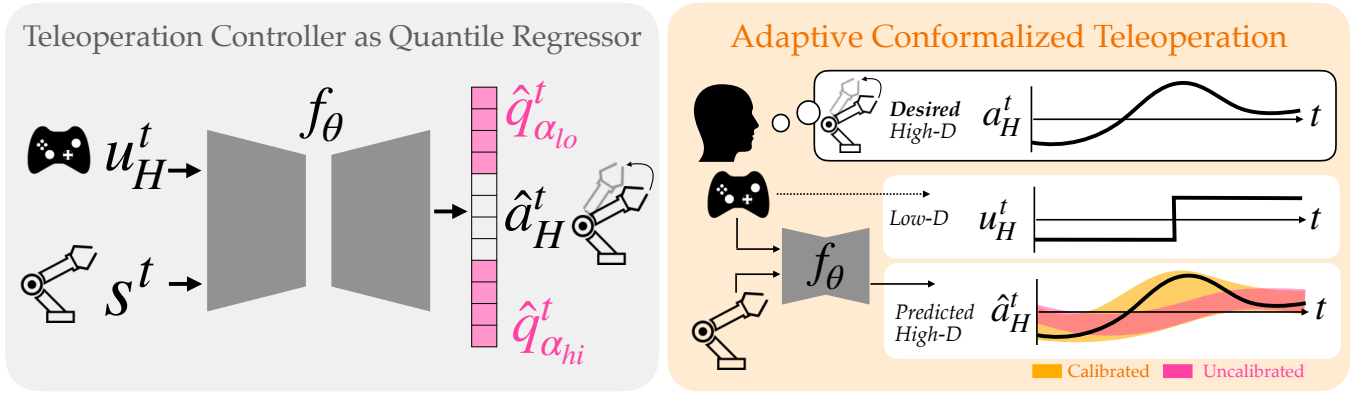Fig. 2: **Approach for Conformalized Teleoperation** (left) During training, we modify the teleoperation controller f_θ to regress both the high-dimensional action corresponding with the low-dimensional human input, but also the empirical quantiles. (right) When deployed around a new user, we can calibrate the model to the user's new dataset distribution. Adaptive Conformal Quantile Regression enables us to enlarge or shrink the predicted quantiles to get coverage of the user's *desired* high-dimensional action.

In our setting of interest, this controller f_θ is represented as a neural network with parameters θ and is learned from a dataset $(s^t, u_{\mathcal{H}}^t, s^{t+1}) \in \mathcal{D}_{\text{train}}$ where s^t is the current state and s^{t+1} is the intended next state, and $u_{\mathcal{H}}^t$ is the corresponding low-dimensional input the human would give to yield the desired next state. In practice, to generate a sufficiently large and diverse training dataset, a variety of strategies can be employed: for example, using heuristics to deterministically generate $u_{\mathcal{H}}^t$ for consecutive state pairs [36], showing human annotators current and next-state robot pairs (e.g., robot is near an object, then the robot is closer to the object) and asking them to label them with the corresponding low-dimensional action (e.g., $u_{\mathcal{H}} = \text{left}$ on the joystick) [36], or querying large-language models as a proxy for human labels [56].

After training f_θ, the robot is deployed to interact with a target user, and executes the predicted high-dimensional action $\hat{a}_{\mathcal{H}}^t$ output by the learned controller conditioned on the current state s^t and the human's low-dimensional action $u_{\mathcal{H}}^t$.

Goal: Calibrated Low-to-High Dimensional Control. Our goal is to calibrate the controller f_θ trained on $\mathcal{D}_{\text{train}}$ to a target end-user. Intuitively, we want a principled measure of the learned controller's *uncertainty* at different states and low-dimensional end-user inputs. More formally, we seek to ensure that the human's *intended* high-DoF action, $a_{\mathcal{H}}$, is contained within the predicted action bound on f_θ's output with designer-specified probability $1 - \alpha$.

IV. METHOD: CONFORMALIZED TELEOPERATION

Our approach to quantifying the confidence of a black-box, learned assistive controller on target user data is grounded in adaptive conformal prediction [17]. While there are many variants of conformal prediction (CP) (see [1] for an overview), we focus on Adaptive Conformalized Quantile Regression (ACQR) [46] which frames the problem of constructing high-confidence prediction intervals on temporally-correlated data as an online learning problem, is relevant to regression

problems, and yields asymptotic coverage guarantees without strong distributional assumptions [17, 46]. Additional background on relevant conformal prediction concepts can be found in the Appendix, Section **??**.

In this section, we break down our Conformalized Teleoperation approach into three key steps. First, we train the robot's initial assistive mapping with paired low-DoF inputs and high-dimensional-DoF outputs, but force the model to approximate its own uncertainty by predicting empirical quantiles (Section IV-A). Next, we use ACQR to calibrate this base model to data from a target user. Given some miscoverage rate $\alpha \in [0, 1]$, our goal in conformal prediction to construct prediction intervals such that the probability the interval contains the correct label (i.e., action) is almost exactly $1 - \alpha$. A target user's calibration data—drawn from the same distribution that the model will be deployed with at test time (from the target user) but unseen during training—is what enables the calibration procedure to produce prediction intervals guaranteed to contain the user's desired high-DoF action with a specified high probability (Section IV-B). Finally, we use the size of the calibrated prediction intervals as a calibrated monitor indicating the model's uncertainty on a target user (Section IV-C).

A. Training Teleoperation Controller as a Quantile Regressor

We start by training the robot's assistive teleoperation controller via a neural network, and force the model to predict its own uncertainty by predicting the empirical quantiles of the predicted high-DoF action.

Training Data. Let $\mathcal{D}_{\text{train}}$ be a dataset consisting of tuples $(s^t, u_{\mathcal{H}}^t, s^{t+1})$, where s^t is the current state, s^{t+1} is the intended next state, and $u_{\mathcal{H}}^t$ a low-dimensional input expected to move the robot from s^t to s^{t+1}. Note that in our primary problem setting, the states are the 7DOF joint angles of the robot ($s := q$) and actions are joint velocities ($a := \dot{q}$), so we

can simply subtract s^t from s^{t+1} to determine the action $a_{\mathcal{H}}^t$ intended by $u_{\mathcal{H}}^t$.

Architecture. The assistive controller f_θ is a neural network parameterized by θ that takes as as input $(s^t, u_{\mathcal{H}}^t)$ pairs and estimates $a_{\mathcal{H}}^t$. The network architecture is a simple feed-forward neural network with GELU activation [21], and an information bottleneck layer of dimension 6. The information bottleneck layer is followed by a tanh activation (see Figure 2). To construct a controller suitable for conformal inference, we train f_θ to additionally estimate quantiles $\hat{q}_{\alpha_{lo}} = \frac{\alpha}{2}$ and $\hat{q}_{\alpha_{hi}} = 1 - \frac{\alpha}{2}$ of the predicted action $\hat{a}_{\mathcal{H}}$ given $(s, u_{\mathcal{H}})$, using quantile regression [31]. More formally, the α-th conditional quantile function is

$$q_\alpha := \inf\{a \in \mathbb{R}^m : P(Y \geq a \mid X = (s, u_{\mathcal{H}})) \geq \alpha\} \quad (2)$$

The uncertainty in the prediction of the human's intended high-dimensional action is reflected in the size of the interval.

Loss Function. We train f_θ to minimize:

$$\mathcal{L} = \mathcal{L}_{MSE} + \mathcal{L}_{p(\alpha_{hi})} + \mathcal{L}_{p(\alpha_{lo})}, \quad (3)$$

where $\mathcal{L}_{MSE}$ measures error between the controller's predicted high-DoF action, $\hat{a}_{\mathcal{H}}$, and the desired high-DoF action, $a_{\mathcal{H}}$. The losses, $\mathcal{L}_{p(\alpha_{hi})}$ and $\mathcal{L}_{p(\alpha_{lo})}$, are two pinball losses (where $p(\cdot)$ stands for pinball) that are standard within conformal literature (as defined in [31, 46, 50]). Thus, in addition to the mean prediction, the model also outputs the empirical upper $1 - \frac{\alpha}{2}$ and lower $\frac{\alpha}{2}$-quantile:

$$\hat{C}(s^t, u_{\mathcal{H}}^t) = [\hat{q}_{\alpha_{lo}}(s^t, u_{\mathcal{H}}^t), \hat{q}_{\alpha_{hi}}(s^t, u_{\mathcal{H}}^t)]. \quad (4)$$

For training, we use a learning rate of 0.01.

B. Adaptive Conformalized Quantile Regression (ACQR)

Although the base model in Section IV-A is trained to estimate its own uncertainty, the estimated uncertainty interval $\hat{C}(s^t, u_{\mathcal{H}}^t)$ is not necessarily calibrated. Here, we use a calibration dataset consisting of a target user's desired low-DoF and high-DoF mappings and conformal prediction to correct $\hat{C}$ to ensure our desired coverage.

Measuring Misprediction. The critical step in ACQR [17] is to, for any input $(s^t, u_{\mathcal{H}}^t)$, compute the *conformity score* that quantifies the error made by the initial prediction interval. One design choice when applying adaptive conformal prediction to our robotics domain is how to scale uncertainty across the high-dimensional output vector. In general, each dimension of the high-dimensional output may not have the same scale; to account for this, we select a multiplicative (rather than additive) factor as the nonconformity score. Specifically, we look for the smallest multiplicative factor $\rho \in \mathbb{R}$ for which we get coverage across all dimensions of the output vector.

Mathematically, for each datapoint $(s^t, u_{\mathcal{H}}^t, a_{\mathcal{H}}^t)$, we compute the similar upper and lower prediction error:

$$\Delta_t^+ = \max\{\hat{q}_{\alpha_{hi}} - \hat{a}_{\mathcal{H}}, \epsilon\}, \quad \Delta_t^- = \max\{\hat{a}_{\mathcal{H}} - \hat{q}_{\alpha_{lo}}, \epsilon\}, \quad (5)$$

for a small $\epsilon = 0.001$. With this, we calculate the nonconformity score as the minimum multiplicative factor for which we get coverage in all dimensions of the output [1]:

$$S(s^t, u_{\mathcal{H}}^i, a_{\mathcal{H}}^i) = \inf\{\rho : a_{\mathcal{H}}^t \in (\hat{a}_{\mathcal{H}}^t - \rho\Delta_t^-, \hat{a}_t + \rho\Delta_t^+)\}. \quad (6)$$

Calibrating Uncertainty to Target User. At test time, we receive sequential data, referred to as $\mathcal{D}_{\text{calib}}$, from a target end-user. We want to quantify how uncertain is our learned controller when mapping inputs from this target user. Let $\mathcal{D}_{\text{calib}} = \{(s^t, u_{\mathcal{H}}^t, a_{\mathcal{H}}^t)\}_{t=0}^T$ contain low-dimensional inputs labeled by our target user.

To perform adaptive conformal quantile regression (ACQR), we initialize our desired miscoverage rate $\alpha_1 = \alpha$ and then seek to compute the empirical quantiles that conformalize the prediction interval. Define time-dependent set $S_t = \{S(s^i, u_{\mathcal{H}}^i, a_{\mathcal{H}}^i)\}_{i=1}^t$ as the set of conformal scores for all datapoints in $\mathcal{D}_{\text{calib}}$ up until the current time t, where the score function S itself is not time-dependent.

Given a new input X_t, ACQR constructs a prediction interval for Y_t by leveraging the conformity scores obtained via Equation (6) on the calibration dataset. Finally, for any new human inputs $(s^t, u_{\mathcal{H}}^t)$, the calibrated prediction interval is obtained via

$$C(s^t, u_{\mathcal{H}}^t) = [\hat{a}_{\mathcal{H}}^t - Q_{1-\alpha_t}(S_t)\Delta_t^-, \hat{a}_{\mathcal{H}}^t + Q_{1-\alpha_t}(S_t)\Delta_t^+]. \quad (7)$$

where $\lambda_t = Q_{1-\alpha_t}(S_t) := (1-\alpha_t)(1+1/|\mathcal{D}_{\text{calib}}|)$th empirical quantile of set S_t. To regulate the level of conservativism, at each timestep, we also adjust α_t via the online update

$$\alpha_{t+1} := \alpha_t + \gamma(\alpha - \text{err}_t) \quad (8)$$

where err_t is defined

$$\text{err}_t := \begin{cases} 1, & \text{if } Y_t \notin C_t(X_t). \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In practice, we take $\gamma = 0.005$, a value found by [17] to give relatively stable trajectories while being large enough to make meaningful changes to α_t.

Coverage Guarantee. Our final guarantees are inherited from ACQR, which ensures asymptotic α coverage [17]. Mathematically,

$$\left| \frac{1}{T} \sum_{t=1}^T \text{err}_t - \alpha \right| \leq \frac{\max\{\alpha_1, 1 - \alpha_1\} + \gamma}{T\gamma} \quad (10)$$

At $\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^T \text{err}_t - \alpha$ approaches α. This guarantees ACQR gives the $1-\alpha$ long-term empirical coverage frequency regardless of the underlying data generation process. While these asymptotic coverage guarantees are sound in theory, we believe it is important to acknowledge that typical robot deployment conditions are finite-horizon. Nevertheless our empirical findings in Section V indicate the practical utility of using ACQR to calibrate black-box assistive controllers compared to naively running the base model.

C. Monitoring Uncertainty

With our conformalized teleoperation approach, the robot can be confident that any user's low-DoF input to f_θ can be associated with an interval of high-DoF actions that is guaranteed to include the human's *desired* high-DoF action (under the assurances provided by ACQR). Note that if we had left the regressed quantiles as they are output by f_θ then we have no rigorous notion of uncertainty with respect to the specific end-user. In fact, as we see in our experiments in Section VI, QR is often overly confident even when it lifts the user's inputs into erroneous high-DoF action spaces.

With this calibrated notion of uncertainty, we further propose a simple mechanism for detecting high uncertainty in the user inputs seen at test time. Recall that our ACQR method yields a high-dimensional (7-DoF) calibrated upper and lower interval bound denoted by $C(\cdot)$ in Equation 7. We distill the high-dimensional uncertainty into a single scalar measure by considering the radius of the sphere formed by the L2 distance between the upper and lower interval bounds. Mathematically, our *scalar uncertainty score* is:

$$U(s^t, u_{\mathcal{H}}^t) \coloneqq ||(\hat{a}_{\mathcal{H}}^t + Q_{1-\alpha_t}(s^t)\Delta_t^+)) - (\hat{a}_{\mathcal{H}}^t - Q_{1-\alpha_t}(s^t)\Delta_t^-)||_2. \tag{11}$$

By choosing a threshold β of high uncertainty scores, we have a simple switching mechanism which flags high uncertainty at inputs where $U(s^t, u_{\mathcal{H}}^t) > \beta$, and low-uncertainty elsewhere. This can be used to stop robot operation, ask for more clarification on the human's intents, or collect more data to improve the model. While in this work we do not study exactly *how* this detection mechanism could be used, we do investigate if it can, with statistical significance, distinguish between low and high-prediction error states in Section VI.

V. Evaluation Setup

To evaluate the efficacy of our proposed approach, we run a series of quantitative and qualitative experiments to study how different training data distributions $\mathcal{D}_{\text{train}}$ and user's inputs ($\mathcal{D}_{\text{calib}}$) impact the learned mapping's uncertainty.

Types of Uncertainty. In the context of assistive teleoperation, we study three sources of real-world uncertainty present in teleoperation data: *latent preferences* (at the trajectory level), *low control precision* (at the trajectory level), and *low-dimensional input schemes* (at the input level).

Trajectory level: Latent preferences. First, we consider the scenario where the training distribution $\mathcal{D}_{\text{train}}$ contains demonstrations from a population of users with varied latent preferences about the task. We assume all users have the same low-dimensional input scheme (e.g., they all agree on which low-DoF input corresponds to which high-DoF action), but the learning problem is under-specified: a single low-DoF input could correspond to multiple different desired high-DoF actions under two different latent preferences. Here, the *latent preferences* we consider are which object (i.e., goal) the user wants to move the robot to, and which grasp pose the user wants to pick up a coffee mug with.

Trajectory level: Low control precision. Second, we study scenarios where the training demonstrations exhibit periods of *low control precision* or inconsistent task execution. We assume there is a single goal but the users exhibit noisy behavior on the way towards a goal (such as grasping a cup), in the training dataset. Both training and calibration users use the same low-dimensional input labeling scheme, and the calibration users also exhibit low-precision task execution.

Input level: Low-dimensional input schemes. Third, we consider uncertainty that stems from *diverse low-dimensional input schemes*, i.e., the different ways that humans choose low-DoF inputs for the same high-DoF robot trajectory. Here, uncertainty arises because of natural differences between the low-dimensional input schemes seen at training time and the deployment-time users' intuitive understanding of low-dimensional inputs. We obtain calibration data in this scenario by showing the same robot trajectory to three human users, and collecting their low-dimensional input sequence labels.

Environments. We study uncertainty in the *latent preferences* and *low control precision* contexts through experiments in three controlled environments: in a toy assistive navigation **GridWorld** environment for building intuition, and a Kinova robotic manipulator **7DOF goal-reaching** and **7DOF cup grasping** setting. The Gridworld environment is a 25x25 gridworld in which the robot must navigate to achieve a goal state. In the 7DOF settings, expert demonstrators are tasked with either kinesthetically moving the robot towards one of the two objects on the table, or moving the robot to grasp a mug from either the lip or the handle. The uncertainty context informs the construction of the training dataset for each domain. We study uncertainty in the *low-dimensional input schemes* context through an experiment in the **7DOF goal-reaching** setting. To evaluate our method on diverse user input schemes at calibration time, we collected low-dimensional input sequences from a mixture of simulated users and novice human operators (more details in Section V).

Baselines. We compare our method **ACQR** to vanilla Quantile Regression (**QR**), where we train our teleoperation controller but do not calibrate the intervals on the target user online. We additionally compare our method to an ensemble uncertainty quantification approach **Ensemble** [33]. For the **Ensemble** baseline, we train $M = 5$ neural networks with the same encoder-decoder structure as in our teleoperation controller design. Each model outputs a predicted mean $\mu_\theta(u_{\mathcal{H}}, s) \in \mathbb{R}^{n_a}$ and variance $\sigma_\theta^2(u_{\mathcal{H}}, s) \in \mathbb{R}^{n_a}$ for the prediction of the high-DoF robot action a intended by the user input $u_{\mathcal{H}}$ at state s. We randomly initialize the model weights and data order. We take the mixture of the multivariate Gaussians as the model prediction, and the first standard deviation from the mean as the prediction interval $C_t(u_{\mathcal{H}}, s)$.

Conformal Hyperparameters. Our implementation of ACQR uses a step size $\gamma = 0.005$ [17], target mis-coverage level of $\alpha = 0.1$, and an initial $\alpha_1 = 0.1$. Additionally, our proposed detection mechanism (Section IV-C) uses a threshold β. In the *latent preferences* setup, $\beta_{grid} = 1.5$, $\beta_{goal} = 0.05$, and

$\beta_{cup} = 0.15$. In the *control precision* context, $\beta_{grid} = 1.0$, $\beta_{goal} = 0.05$, and $\beta_{cup} = 0.12$.

Metrics. We focus on measures of coverage and prediction interval length. Coverage is defined as

$$Cov := \frac{1}{|\mathcal{D}_{\text{calib}}|} \sum_{(u_{\mathcal{H}}^t, s^t, a_{\mathcal{H}}^t) \in \mathcal{D}_{\text{calib}}} \mathbb{1}[a_{\mathcal{H}}^t \in C_t(u_{\mathcal{H}}^t, s^t)]. \quad (12)$$

We also seek to evaluate the size of the prediction interval, as there exists a tradeoff between coverage and interval sizes. An ideal model has tight intervals but good coverage. For **ACQR** and **QR**, interval length for each datapoint is defined as the distance between the upper bound C_t^+ and lower bound C_t^- of the standard deviation of the prediction interval, averaged across each dimension $d \in n_a$:

$$I_L(u_{\mathcal{H}}, s) := \frac{1}{n_a} \sum_{d=1}^{n_a} C_t^+(u_{\mathcal{H}}^t, s^t)_d - C_t^-(u_{\mathcal{H}}^t, s^t)_d. \quad (13)$$

We average final metrics over all datapoints in $\mathcal{D}_{\text{calib}}$.

VI. EVALUATION RESULTS

We break down our results into five major takeaways, focusing on in-distribution (ID) and out-of-distribution (OOD) calibration users, comparison of our various uncertainty quantification methods, and the performance of our proposed detection mechanism.

Takeaway 1: *Even when users operate with an in-distribution input scheme on in-distribution high-DoF trajectories, an uncalibrated mapping f_θ (**QR**) miscovers the human's desired high-DoF action more than **ACQR**.*

We highlight this takeaway in the setting of **7DOF Cup-grasping** with diverse **latent preferences**, but further results can be found in the supplementary. Recall that in this setting the demonstrators may pick up a cup from the handle, others from the lip. Thus, $\mathcal{D}_{\text{train}}$ consists of 14 expert demonstration trajectories, where half pick up the cup from the handle, and half pick up the cup from the lip (shown in left of Figure 3). We calibrate f_θ on an unseen user, referred to as Alice and denoted $\mathcal{D}_{\text{calib}}^A$, who gives 3 demonstrations of picking up the cup from the lip. In this case, the calibration demonstrations of target user, Alice, were provided by one of the researchers who also provided expert demonstrations in the training data. Both the demonstrators and target user employ a heuristic strategy to deterministically annotate $u_{\mathcal{H}}^t$ for consecutive state pairs, where $u_{\mathcal{H}}^t$ is the change in x-direction and y-direction of the end effector from s^t to s^{t+1}. We calibrate to each held-out trajectory, simulating the inputs to the assistive teleoperation controller over time as though the user was controlling the robot in real time.

Using **ACQR**, we see that uncertainty is highest at the *start* and *end* of the interaction when the human has to give the final inputs to orient the robot arm to face downward to achieve their desired cup grasp (shown in right, Figure 3). Intuitively, $\mathcal{D}_{\text{train}}$ contains higher disagreement amongst the training data generators as they position the robot for
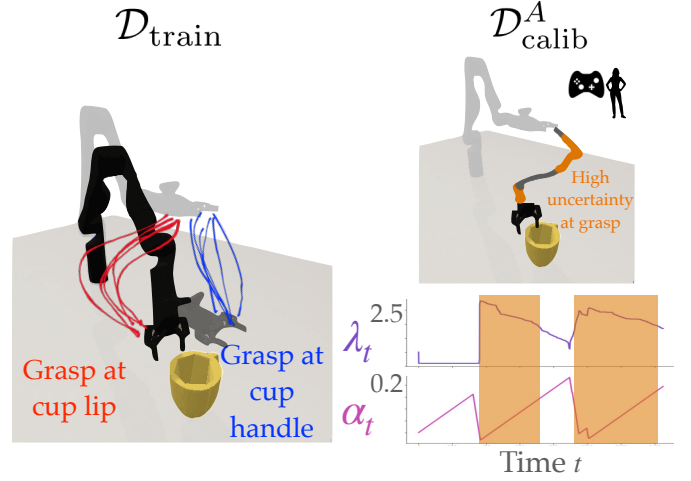

Fig. 3: **7DOF cup-grasping with latent preferences.** (left) $\mathcal{D}_{\text{train}}$ exhibits multimodality in grasp. (right) When ACQR calibrates to $\mathcal{D}_{\text{calib}}^A$. Orange dots indicate timesteps with high uncertainty ($U > \beta_{cup}$). λ_t represents the multiplicative factor by which the quantile intervals are expanded. While α_t increases initially, it decreases later as the robot orients itself towards the cup and as it approaches the cup, and uncertainty in the desired high-dimensional action increases. Correspondingly, λ_t increases at the points of uncertainty. The orange highlights denote timesteps where the uncertainty is greater than threshold β_{cup} and aligns temporally with the orange points on the trajectory. For grasps on the lip, there is higher uncertainty at the start of the trajectory and at the grasp location, due to the variance in lip grasp demonstrations in the training dataset.

grasping. These critical states [22] are informed by the specific user's preferences. Without additional context and due to the underspecified input, the robot cannot be certain about the correct way to map the user's low-DoF input to a high-DoF action. Quantitatively, **QR** achieves 52.7% coverage on $\mathcal{D}_{\text{calib}}^A$, while **ACQR** achieves 92.6% coverage (where target coverage is 90%). This result demonstrates that even for an end-user providing in-distribution demonstrations, adaptively calibrating to unseen data is necessary for achieving informative uncertainty bounds.

Takeaway 2: *When users provide in-distribution low-dimensional inputs on out-of-distribution calibration trajectories, **ACQR** can expand uncertainty when necessary but also contract it for inputs that align with its training distribution.*

We highlight this takeaway in the setting of **7DOF Goal-reaching** with diverse **latent preferences**. Here, $\mathcal{D}_{\text{train}}$ consists of 120 expert demonstrated trajectories, where half of the demonstrators prefer the blue goal and half prefer the red goal (see left, Figure 4). As in the cup-grasping domain, demonstrators and target users employ a heuristic strategy to deterministically generate $u_{\mathcal{H}}^t$ for consecutive state pairs, using the change in x-direction and y-direction of the end
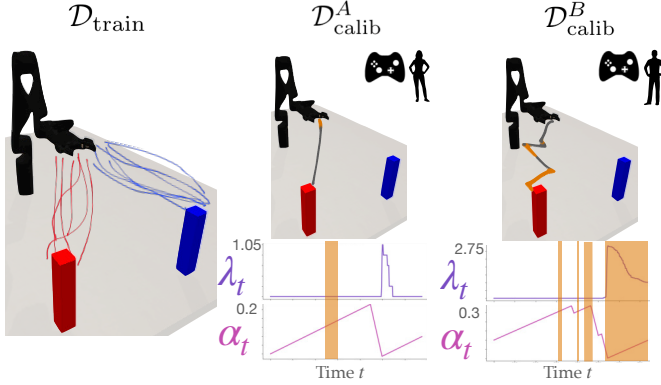
Fig. 4: **7DOF goal-reaching with latent preferences.** (left) $\mathcal{D}_{\text{train}}$ contains teleoperation trajectories towards red and blue goals. (center) Alice's demonstration contains high uncertainty only at the beginning of the task. (right) Bob's indirect path flags high uncertainty throughout. ACQR maintains lower λ_t and higher α_t throughout $\mathcal{D}_{\text{calib}}^A$ demonstration than throughout $\mathcal{D}_{\text{calib}}^B$.
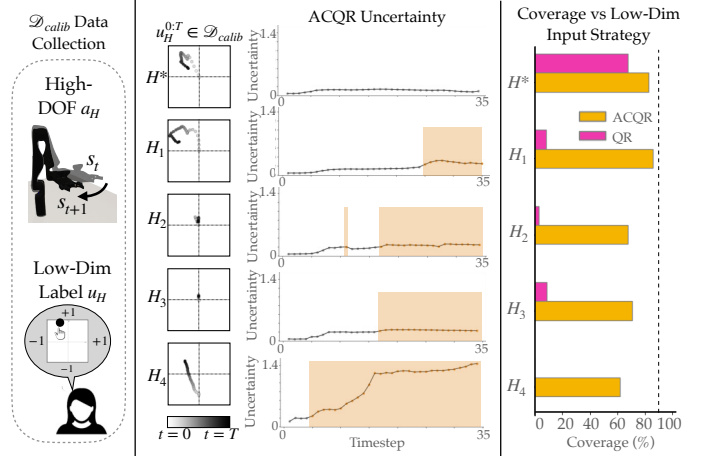

Fig. 5: **7DOF goal-reaching (blue) with out-of-distribution low-dimensional input schemes.** (Left) The calibration data is collected by querying a mixture of simulated and human users for their desired low-dimensional input for a given high-DOF robot action. (Middle) Our uncertainty monitoring mechanism, at the threshold $\beta_{goal,human} = 0.2$, flags high uncertainty for out-of-distribution input schemes $H_1 - H_4$. (Right) Calibration using **ACQR** increases coverage over **QR** on the calibration data from all in- and out-of-distribution users, increasing coverage towards, but not yet at, the desired level of 90%.

effector. We calibrate on two target user profiles, Alice and Bob. Alice represents an in-distribution user: she demonstrates 6 near-expert trajectories to each of the two goals, taking direct paths to the objects. Her data is denoted $\mathcal{D}_{\text{calib}}^A$. The second user, Bob, takes 6 extremely indirect paths towards each goal, representing our out-of-distribution user; his data is denoted $\mathcal{D}_{\text{calib}}^B$. In this case, calibration data for Alice and Bob was provided by a researcher who teleoperated in accordance with the particular user profile.

In the center and right of Figure 4, we see qualitatively that **ACQR** run on demonstrations in $\mathcal{D}_{\text{calib}}^A$ contain fewer instances of high uncertainty, only flagging the start states as uncertain since the assistive mapping isn't confident in which initial high-DOF actions the human wants to take. However, input data from $\mathcal{D}_{\text{calib}}^B$ are frequently flagged as high uncertainty. Interestingly, there are several key portions of the trajectory wherein even the out-of-distribution user appear to be well-predicted by the model. Quantitatively on $\mathcal{D}_{\text{calib}}^A$, **QR** achieves only 82.1% coverage on $\mathcal{D}_{\text{calib}}^A$ while **ACQR** achieves 91.9% coverage. More surpisingly, on the out-of-distribution user from $\mathcal{D}_{\text{calib}}^B$, **QR** is overly confident, achieving only 68.8% coverage. Instead, **ACQR** achieves 90.2% coverage. This result demonstrates that f_θ can confidently mispredict the intended actions of an out-of-distribution user, but ACQR can adjust the prediction intervals to handle such interaction data.

Takeaway 3: As users' low-dimensional input schemes become more out-of-distribution, ACQR adapts to cover their desired high-DoF actions more than QR.

We highlight this takeaway in the task of **7DOF Goal-reaching**. We use the base model trained from the **latent preferences** context. Training demonstrations are labeled with low-dimensional inputs via a deterministic, heuristic strategy

using the change in x-direction and y-direction of the end effector.

In the calibration phase, target users provide low-dimensional input labels for a single calibration trajectory to the blue goal (see Section **??** for results on a trajectory to the red goal). For each trajectory, we collected the low-dimensional input labels for pairs of consecutive robot states, downsampled along the full trajectories. We considered five input schemes: three real human annotators (H_2, H_3, H_4), one simulated user operating under an alternative heuristic scheme (H_1), and one in-distribution simulated user (H^*). Specifically, H_1 gives input $u_{\mathcal{H}}$ equal to the change in z-direction and y-direction of the end effector, representing a slightly out-of-distribution low-dimensional input scheme. Users H_2, H_3, H_4 are three real, human users with technical expertise but novice to the task of robot teleoperation with low-dimensional control (see left hand side of Figure 5 for the query setup).

In Figure 5 (right), we see that the base model without calibration, **QR**, achieves 68% percent coverage on the in-distribution user H^*, but achieves poor coverage ($< 10\%$) on out-of-distribution input schemes (H_1-H_4). In contrast, calibration using **ACQR** increases coverage on the calibration data from all in- and out-of-distribution users, increasing coverage towards the desired level of 90%. Note that because **ACQR** gives a coverage guarantee asymptotic in time, the coverage achieved by **ACQR** nears but does not reach the desired level given the limited trajectory length.

Next, we study how our uncertainty monitoring mechanism

is influenced by OOD input schemes. In Figure 5 (middle), our uncertainty monitoring mechanism (at the threshold $\beta_{goal,human} = 0.2$, increased from 0.05 for noisy human inputs) flags high uncertainty more frequently as the user input schemes become more OOD. For example, user H_1, is relatively in-distribution: they give input via the z- and y-displacement of the end effector which is similar to user H^* from the training population. Because of the similarity in input sequence, **ACQR**'s uncertainty intervals are not as large for H_1 and thus the monitor only activates briefly along the middle of the trajectory (for 12 timesteps). On the other extreme, for our most OOD user, H_4, our monitor activates directly after the first input and continues to flag each consecutive input as uncertain. Finally, for users H_2 and H_3 whose input schemes are extremely similar in pattern but still quite OOD, our uncertainty monitor shows a similar detection pattern, indicating that our monitor's behavior is consistent.

Takeaway 4: While ensembles can sometimes capture the high-uncertainty of the learned mapping, they are highly sensitive to the problem domain and in general do not come with any assurances.

We highlight this takeaway in the setting of **GridWorld** and **7DOF Cup-grasping** settings under uncertainty induced by **low-control precision**. In the low-control precision context, our target user is the same as the demonstrator. The GridWorld domain requires demonstrators to navigate from an open region through a narrow tunnel to reach the goal state (Figure 6). Trajectories in $\mathcal{D}_{\mathrm{train}}$ begin at random locations in the open region with noisy actions representing variance in paths taken at less critical regions. Once the user is in the hallway, a narrow region where precision is critical, there is no stochasticity in their actions. For calibration, $\mathcal{D}_{\mathrm{calib}}^A$ contains 36 heldout trajectories of the user moving to the goal from different starting states. Intuitively, we find that the areas of highest uncertainty occur at the low-precision parts of the task, where there are, for the target user, many paths to approach the hallway. Once inside the tunnel, uncertainty is low (right, Figure 6). In this domain, we see **ACQR** achieve 92.7% coverage with mean interval lengths of 0.86 while the **Ensemble** struggles to achieve the target coverage (81.7%) with mean interval lengths (0.697).

Although **Ensemble** mispredicts the actions more frequently than **ACQR** in **Gridworld** setting, we see different results in **7DOF Cup-Grasping** environment (see Table I). Here, $\mathcal{D}_{\mathrm{train}}$ contains 7 kinesthetic demonstrations by a single user where all demonstrations pick up the cup by the handle. The trajectories are very precise near the handle, but less precise in the path towards the cup where there is more free space. We calibrate on the same target user, $\mathcal{D}_{\mathrm{calib}}^A$, who demonstrates 3 unseen test trajectories of picking up the cup by the handle. Empirically, **Ensemble** exhibits higher coverage than ACQR: **ACQR** achieves 93.8% coverage with mean interval lengths of 0.044, while **Ensemble** achieves 100% coverage with a larger mean interval length of 0.745. While empirically our ACQR algorithm achieves coverage
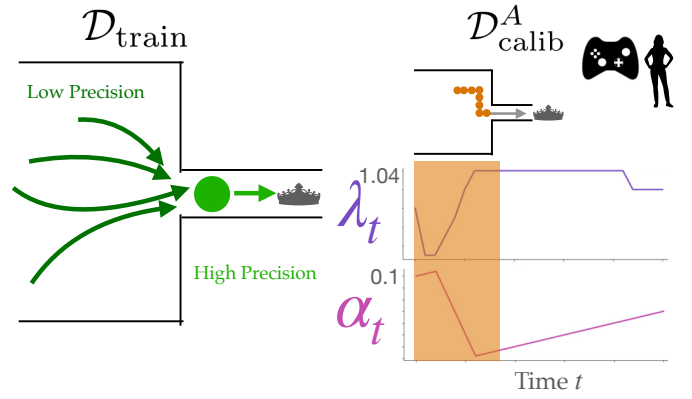

Fig. 6: **GridWorld with low control precision**. The areas of highest uncertainty ($\beta = 1.0$) occur as the heldout demonstration approaches the tunnel. Inside the tunnel, uncertainty is low. α_t decreases throughout the approach to the tunnel, and increases once inside the tunnel.

TABLE I: Low-Control Precision: Coverage and Interval Size for $\mathcal{D}_{\mathrm{calib}}^A$

Algorithm	Coverage			Interval Length		
	Grid $\mathcal{D}_{\mathrm{calib}}^A$	Goal $\mathcal{D}_{\mathrm{calib}}^A$	Cup $\mathcal{D}_{\mathrm{calib}}^A$	Grid $\mathcal{D}_{\mathrm{calib}}^A$	Goal $\mathcal{D}_{\mathrm{calib}}^A$	Cup $\mathcal{D}_{\mathrm{calib}}^A$
ACQR	0.927	0.890	0.938	0.860	0.009	0.044
Ensemble	0.817	1.0	1.0	0.697	0.745	0.742
QR	0.910	0.994	0.402	0.689	0.012	0.035

of $1 - \alpha = 0.9$ in both environments, the Ensemble does not provide any statistical guarantees and coverage can vary significantly between environments. We hypothesize that the latter point stems from the training action distribution: in the Gridworld, the training action distribution is multi-modal around the values $\{-1, 0, 1\}$, inducing high variance; in the 7DOF robot domain, the robot's joint velocities are bounded to $[-1, 1]$ per-joint and the training action distribution has small variance centered around the mean, making mis-prediction less likely.

Takeaway 5: Our proposed detection mechanism based on the uncertainty interval size is an effective way to separate high and low uncertainty states and inputs.

Finally, we seek to verify the hypothesis that instances of high uncertainty in the user's demonstration at test time can be meaningfully identified through our proposed detection mechanism. To test this, for each environment, we choose an uncertainty threshold, β (tuned per domain) and test if there is a statistically significant difference in prediction error $\|\hat{a}_t - a_t\|_2$ at states with high uncertainty (uncertainty above β) and states with low uncertainty (uncertainty below or equal to β). While ultimately this is a design parameter, we aim to understand if even simple choices of β demonstrate statistically meaningful separation between the classification of certain and

TABLE II: Prediction Error in Contexts with Uncertainty Induced by Low-Precision

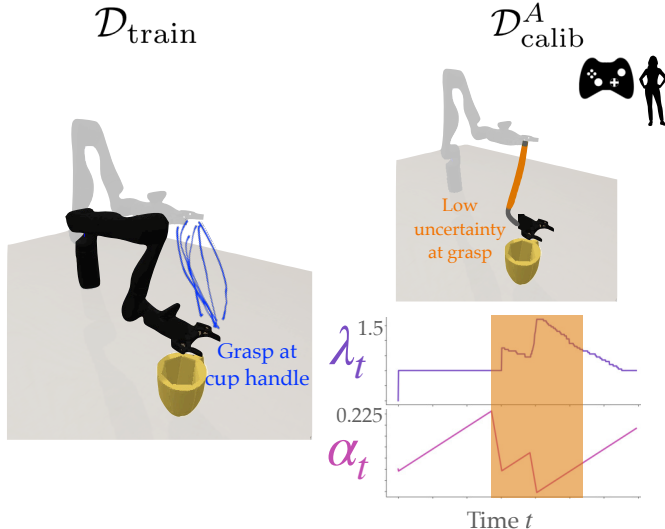| Environment	Target User	β	$\hat{a}_{\mathcal{H}}$ **Error at Uncertain Inputs**: mean, (std)	$\hat{a}_{\mathcal{H}}$ **Error at Certain Inputs**	**T-test**
GridWorld	$\mathcal{D}^A_{calib}$	1.0	0.680, (0.415)	0.040, (0.073)	$t(966) = 36.62 \; p < 0.001$
7DOF Goal	$\mathcal{D}^A_{calib}$	0.05	0.014, (0.005)	0.006, (0.001)	$t(3109) = 55.47 \; p < 0.001$
7DOF Cup	$\mathcal{D}^A_{calib}$	0.12	0.018, (0.008)	0.010 (0.002)	$t(2041) = 36.57 \; p < 0.001$

Fig. 7: **7DOF Cup-grasping with low control precision.** Demonstrations from a single user position the gripper to grasp a cup at the handle (left). ACQR flags high uncertainty in the path towards the cup, but has the lowest uncertainty at the grasp location (right), where high precision is required.

uncertain states. If this is true, then this indicates the potential for using our approach to proactively flag potential robot low-to-high dimensional mapping errors before they occur and ask for further assistance from the human.

We detail the results in the **7DOF Goal-reaching** domain with **low-control precision** uncertainty. Specifically, we perform a T-test of unequal variances [30] between the distribution of prediction errors for inputs flagged as anomalous by our mechanism and the distribution of prediction errors for inputs flagged as nominal on $\mathcal{D}^A_{\text{calib}}$ (see Table II). We find that our mechanism separates low and high uncertainty inputs in a statistically significant manner ($p < 0.001$ in this environment). We evaluate distributional separation using a T-test to confirm the prediction errors from low and high uncertainty inputs come from different distributions. Furthermore, the mean prediction error is higher in uncertain states than certain states. We find this result particularly promising, because this indicates that **ACQR** can be used as a principled detection mechanism for mitigating the robot's learned mapping errors and potentially providing a way of proactively asking the user for assistance.

VII. Discussion & Conclusion

Discussion. One limitation of our approach is that the model cannot differentiate between these different types of semantic uncertainty (e.g., preferences vs. control precision). Here, we are excited to investigate how additional context (e.g., images [27] or language [29]) can reduce the robot's uncertainty intervals. Future work should also investigate seeking appropriate assistance from human users, through other modalities, like language [25, 45, 28].

Conclusion. In this work, we present Conformalized Teleoperation, an approach for quantifying uncertainty in learned assistive teleoperation controllers that map from low-DoF human inputs to high-DoF robot actions. We find that the dataset with which the controller was trained on significantly impacts the robot's ability to infer the user's intended high-DoF action, and off-the-shelf controllers can be over-confident even when interacting with significantly out-of-distribution end-users. However, we find that our proposed adaptive conformal quantile regression approach can meaningfully detect these state and inputs that cause uncertainty, providing a promising path forward towards the detection and mitigation of such assistive teleoperation failures.

References

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

[2] Reuben M Aronson and Henny Admoni. Eye gaze for assistive manipulation. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 552–554, 2020.

[3] RG Boboc, H Moga, and D Talaba. A review of current applications in teleoperation of mobile robots. *Bulletin of the Transilvania University of Brasov. Engineering Sciences. Series I*, 5(2):9, 2012.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

[5] Leo Breiman. Random forests. *Machine learning*, 45: 5–32, 2001.

[6] Alexander Broad, Todd Murphey, and Brenna Argall. Learning models for shared control of human-machine systems with unknown dynamics. *arXiv preprint arXiv:1808.08268*, 2018.

[7] Alexander Broad, Ian Abraham, Todd Murphey, and Brenna Argall. Data-driven koopman operators for model-based shared control of human–machine systems. *The International Journal of Robotics Research*, 39(9):1178–1195, 2020.

[8] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6):509–521, 2010.

[9] Yuxiao Chen, Ugo Rosolia, Chuchu Fan, Aaron Ames, and Richard Murray. Reactive motion planning with probabilistic safety guarantees. In *Conference on Robot Learning*, pages 1958–1970. PMLR, 2021.

[10] Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[11] Jianye Ching, James L Beck, and Keith A Porter. Bayesian state and parameter estimation of uncertain dynamical systems. *Probabilistic engineering mechanics*, 21(1):81–96, 2006.

[12] Thomas G Dietterich and Jesse Hostetler. Conformal prediction intervals for Markov decision process trajectories. *arXiv preprint arXiv:2206.04860*, 2022.

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

[14] Anca D Dragan and Siddhartha S Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[15] Håkan Eftring and Kerstin Boschian. Technical results from manus user trials. In *Proc. ICORR*, volume 99, pages 136–141. Citeseer, 1999.

[16] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

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

[18] Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pages 45–87, 2020.

[19] Deepak E Gopinath and Brenna D Argall. Active intent disambiguation for shared control robots. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(6):1497–1506, 2020.

[20] Takumi Hara, Takashi Sato, Tetsuya Ogata, and Hiromitsu Awano. Uncertainty-aware haptic shared control with humanoid robots for flexible object manipulation. *IEEE Robotics and Automation Letters*, 2023.

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[22] Sandy H Huang, Kush Bhatia, Pieter Abbeel, and Anca D Dragan. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3929–3936. IEEE, 2018.

[23] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S Srinivasa, and J Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, 37(7):717–742, 2018.

[24] Hong Jun Jeon, Dylan P Losey, and Dorsa Sadigh. Shared autonomy with learned latent actions. *arXiv preprint arXiv:2005.03210*, 2020.

[25] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

[26] Ananth Jonnavittula, Shaunak A Mehta, and Dylan P Losey. Learning to share autonomy from repeated human-robot interaction. *arXiv preprint arXiv:2205.09795*, 2022.

[27] Siddharth Karamcheti, Albert J Zhai, Dylan P Losey, and Dorsa Sadigh. Learning visually guided latent actions for assistive teleoperation. In *Learning for Dynamics and Control*, pages 1230–1241. PMLR, 2021.

[28] Siddharth Karamcheti, Raj Palleti, Yuchen Cui, Percy Liang, and Dorsa Sadigh. Shared autonomy for robotic manipulation with language corrections. In *ACL Workshop on Learning with Natural Language Supervision*, 2022.

[29] Siddharth Karamcheti, Megha Srivastava, Percy Liang, and Dorsa Sadigh. Lila: Language-informed latent actions. In *Conference on Robot Learning*, pages 1379–1390. PMLR, 2022.

[30] Tae Kyun Kim. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540–546, 2015.

[31] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[32] Igor Kononenko. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989.

[33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.

[35] Jongmin M Lee, Temesgen Gebrekristos, Dalia De Santis, Mahdieh Nejati-Javaremi, Deepak Gopinath, Biraj Parikh, Ferdinando A Mussa-Ivaldi, and Brenna D Ar-

gall. Learning to control complex robots using high-dimensional interfaces: Preliminary insights. *arXiv preprint arXiv:2110.04663*, 2021.

[36] Mengxi Li, Dylan P Losey, Jeannette Bohg, and Dorsa Sadigh. Learning user-preferred mappings for intuitive robot control. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10960–10967. IEEE, 2020.

[37] Albert Lin and Somil Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. *L4DC (submitted)*, 2023.

[38] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

[39] Dylan P Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. Controlling assistive robots with learned latent actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–384. IEEE, 2020.

[40] Dylan P Losey, Hong Jun Jeon, Mengxi Li, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, Jeannette Bohg, and Dorsa Sadigh. Learning latent actions to control assistive robots. *Autonomous robots*, 46(1):115–147, 2022.

[41] Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 149–169. Springer, 2022.

[42] Joao Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa. Ensemble approaches for regression: A survey. *Acm computing surveys (csur)*, 45(1):1–40, 2012.

[43] David P Miller. Assistive robotics: an overview. *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing*, pages 126–136, 2006.

[44] Anish Muthali, Haotian Shen, Sampada Deglurkar, Michael H Lim, Rebecca Roelofs, Aleksandra Faust, and Claire Tomlin. Multi-agent reachability calibration with conformal prediction. *arXiv preprint arXiv:2304.00432*, 2023.

[45] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[46] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

[47] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33: 3581–3591, 2020.

[48] JC Rosier, JA Van Woerden, LW Van der Kolk, BJF Driessen, HH Kwee, JJ Duimel, JJ Smits, AA Tuinhof de Moed, G Honderd, and PM Bruyn. Rehabilitation robotics: The manus concept. In *Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments*, pages 893–898. IEEE, 1991.

[49] Afshar Shamsi, Hamzeh Asgharnezhad, Moloud Abdar, AmirReza Tajally, Abbas Khosravi, Saeid Nahavandi, and Henry Leung. Improving mc-dropout uncertainty estimates with calibration error-based optimization. *arXiv preprint arXiv:2110.03260*, 2021.

[50] Ingo Steinwart and Andreas Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 2011.

[51] Muhammad Faaiz Taufiq, Jean-Francois Ton, Rob Cornish, Yee Whye Teh, and Arnaud Doucet. Conformal off-policy prediction in contextual bandits. *Advances in Neural Information Processing Systems*, 35:31512–31524, 2022.

[52] Hylke A Tijsma, Freek Liefhebber, and Just L Herder. A framework of interface improvements for designing new user interfaces for the manus robot arm. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 235–240. IEEE, 2005.

[53] Katherine Tsui, Holly Yanco, David Kontak, and Linda Beliveau. Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 105–112, 2008.

[54] Erkang You and Kris Hauser. Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input. In *Robotics: science and systems*, volume 7, page 354. MIT Press USA, 2012.

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

[56] Bowen Zhang and Harold Soh. Large language models as zero-shot human models for human-robot interaction. *arXiv preprint arXiv:2303.03548*, 2023.

[57] Yu Zhou and Kris Hauser. Cpi: Conservativeness, permissiveness and intervention metrics for shared control evaluation. *IEEE Robotics and Automation Letters*, 7(3): 6367–6374, 2022.

[58] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE, 2017.

[59] Matthew Zurek, Andreea Bobu, Daniel S Brown, and Anca D Dragan. Situational confidence assistance for lifelong shared autonomy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2783–2789. IEEE, 2021.