Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Ask or Act: A Calibrated-Confidence Framework for Risk-Aware Navigation Under Uncertain Human Intent

Daniel Udekwe
Florida State University
Qianwen Guo
qguo@eng.famu.fsu.edu

Florida State University

Article

Keywords: risk-aware navigation, human-robot interaction, uncertainty calibration, intent prediction, distribution shift, social navigation

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11542425/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Ask or Act: A Calibrated-Confidence Framework for Risk-Aware Navigation Under Uncertain Human Intent

Daniel Udekwe[1] and Qianwen (Vivian) Guo[1*]

[1*]Department of Civil & Environmental Engineering, FAMU-FSU College of Engineering, Tallahassee, 32310, Florida, USA.

*Corresponding author(s). E-mail(s): qguo@eng.famu.fsu.edu;
Contributing authors: dau24@fsu.edu;

Abstract

Mobile robots that share space with people must decide within seconds whether a pedestrian is about to cross their path, and confidence-based rules that ask the person only when unsure promise to combine safety with efficiency. Such rules are only as safe as their predicted probabilities, yet calibration is usually fitted and evaluated over all available samples, most of which come from moments far from any decision, and its closed-loop effect under distribution shift is rarely tested. We present Ask or Act, a framework in which a robot predicts the probability that a pedestrian will cross, calibrates it, and at a decision deadline yields, proceeds or asks. We propose decision-aware calibration, fitted only on the window in which the robot decides, and evaluate it with a TurtleBot3 in ROS 2 and Gazebo, using eight scripted pedestrian behaviors with known intent under five distribution shifts and a pre-specified analysis plan on held-out scenarios. Under degraded tracking, raw and conventionally recalibrated confidence led to collisions in 34.5% and 15.5% of crossing encounters, against 1.3% with decision-aware recalibration, which remained the safer method for all five predictors trained with different seeds (1.3–10.5% against 5.5–15.8%). Under nominal conditions it asked 16% fewer questions than always asking, within a pre-set non-inferiority margin. Offline results agreed with Gazebo in 97.5% of paired episodes, and the pipeline ran on the robot's onboard computer in about 3 ms per step. For robots that act on predicted probabilities, calibration should be fitted and evaluated where decisions are made.

Keywords: risk-aware navigation, human-robot interaction, uncertainty calibration, intent prediction, distribution shift, social navigation

1 Introduction

Mobile robots operate in human-populated spaces such as corridors, cafés, hospitals, and warehouses. They must continuously negotiate space with people whose intentions are not directly observable [1, 2]. Safe and efficient coexistence requires more than obstacle avoidance. The robot must reason about what a person will do next and act on that judgment, even when the evidence remains ambiguous [3].

This ambiguity is not mere sensor noise. A pedestrian walking toward a robot's path may cross in front of it, stop, or turn away. Until the last moment, all three possibilities can look identical. Existing risk-aware navigation methods absorb such uncertainty into the motion plan. They widen safety margins, slow down, or optimize a worst-case or distributionally robust cost over the uncertain human state [4–6]. This is a reasonable default. However, it treats every instance of high uncertainty the same way. In many

cases the uncertainty could be resolved quickly and cheaply by simply asking the person. People resolve exactly this kind of ambiguity with one another through brief communicative acts rather than by always assuming the worst.

Robots that ask for help when uncertain have been studied in several settings [7–9]. Recent work has linked calibrated, set-valued intent predictions to the decision of when a robot should seek clarification during navigation [10]. A confidence-based rule is attractive: ask only when the predictor is unsure, and otherwise act. Its safety, however, rests entirely on the predicted probabilities being trustworthy at the moment the robot decides. Two practical facts threaten that assumption. First, predictive models are often poorly calibrated, and calibration degrades under distribution shift [11, 12], for example when tracking quality changes. Second, calibration is normally fitted and evaluated over all available samples. Most of those samples come from situations far from any decision.

This paper studies what it takes for a confidence-based ask-or-act rule to be safe. We make four contributions.

1. **The Ask or Act framework:** A robot predicts the probability that a pedestrian will cross its path and calibrates that probability. At a decision deadline it yields, proceeds, or asks. The choice depends on whether the calibrated probability is confidently high, confidently low, or still uncertain.
2. **Decision-aware calibration:** Probabilities can be well calibrated on average yet poorly calibrated inside the decision window. We therefore fit the calibrator only on samples from that window.
3. **A reproducible benchmark with ground-truth intent:** The benchmark contains eight scripted pedestrian behaviors. These include matched ambiguous pairs whose motion is identical until the trajectories diverge.
4. **A pre-specified evaluation, including negative results:** Hypotheses, tests, and robustness checks were fixed before the final runs on untouched scenarios. We report every outcome.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 formalizes the problem, and Section 3.2 presents the framework. Section 3.4 describes the simulation platform, scenarios and evaluation protocol. Section 4 reports the results, Section 5 discusses them and their limitations, and Section 6 concludes.

2 Related work

2.1 Risk-aware navigation among people

Human-aware navigation has a long history [1]. Surveys identify the uncertainty of human behavior as one of its central open challenges [2]. Early approaches treated people as moving obstacles. The dynamic window approach selects velocities that remain collision-free over a short horizon [13], and reciprocal velocity obstacles extend this reasoning to agents that themselves avoid collisions [14]. Later work learned socially compliant behavior from human demonstrations [15] or through deep reinforcement learning [16, 17]. A complementary line of work makes predictive uncertainty explicit. Confidence-aware planners estimate online how well a human model explains the observed motion, and widen their predictions when it explains the motion poorly [4]; the same idea has been scaled to several robots and humans [18]. Risk-sensitive controllers optimize risk measures over multimodal trajectory forecasts [6, 19], and distributionally robust formulations also guard against errors in the predicted distribution itself [5, 20]. Conformal prediction provides prediction regions with statistical coverage guarantees for planning [21], and adaptive variants update these regions online as prediction quality changes [22]. A well-known failure mode of treating uncertainty purely as caution is the "freezing robot problem" [23]: when predictive uncertainty grows, the planner may judge every forward path unsafe and the robot stops. Modeling the cooperation between the robot and the people around it reduces this effect [24]. Across this literature, however, uncertainty is an input to motion planning that the robot absorbs by slowing down, detouring or stopping. None of these methods treats high uncertainty as a cue to resolve it by communicating with the person, which is the option studied here.

2.2 Pedestrian intent and motion prediction

Predicting where people will go is a mature field [3]. Planning-based models treat pedestrians as approximately rational agents moving toward goals, and infer their destinations and paths by inverse optimal control [25, 26]. Data-driven models learn interactions among pedestrians from trajectory datasets. They range from recurrent networks with social pooling [27] and adversarially trained generators that produce several plausible futures [28] to graph-structured generative models that output full trajectory distributions [29]. Closer to our binary formulation, several works predict a discrete intent. Crossing intention near vehicles has been studied with dedicated datasets and models [30, 31], path prediction with switching dynamics anticipates when a pedestrian will stop or keep walking [32], and anticipatory indicators of walking motion help a robot predict which way a person will turn [33]. For evaluation, classical simulators use the social-force model [34], and more recent ROS 2 tools such as HuNavSim provide richer human behaviors and social-navigation metrics [35]. These predictors are typically evaluated by displacement error or classification accuracy on held-out data from the training distribution. Whether their probabilities remain trustworthy when tracking or behavior changes, and at the moments when a robot must commit, is rarely examined; we therefore deliberately use a simple predictor and focus on that question.

2.3 Calibration of predictive uncertainty

A predictor is calibrated when events assigned probability p occur with empirical frequency close to p. Modern neural networks are often overconfident [11], although calibration depends strongly on the model architecture [36]. Post-hoc recalibration maps the predictor's outputs to calibrated probabilities using held-out data, for example by Platt scaling, temperature scaling [11, 37] or isotonic regression [38, 39]. Alternatively, ensembles and Monte Carlo dropout estimate uncertainty as part of training [40, 41]. Calibration quality is usually summarized by the expected calibration error (ECE) [42] or the Brier score [43], both of which have known limitations [44, 45]. Calibration typically degrades under dataset shift [12]. Conformal methods address shift by reweighting calibration data under covariate shift [46] or by adapting online to arbitrary shifts [47]. Two lines of work are especially close to decision-aware calibration. Multicalibration requires a predictor to be calibrated simultaneously on many subpopulations, rather than only on average [48]. Decision calibration requires predicted and true outcome distributions to be indistinguishable to a given set of downstream decision-makers [49]. Decision-aware calibration can be viewed as a simple instance of both ideas: it calibrates on the subpopulation of inputs that arise when the robot actually decides. These methods, however, have been developed and evaluated mostly on static classification benchmarks. They have not, to our knowledge, been applied to a robot acting sequentially, where the decision-relevant inputs are defined by timing within an episode, nor evaluated by their closed-loop safety consequences under distribution shift.

2.4 Robots that ask and communicate

Robots have long used questions to resolve uncertainty. In collaborative control, a teleoperated robot asks its operator questions to compensate for limits of its autonomy, and a user study found that human assistance is a limited resource that must be carefully managed [50]. Office robots ask bystanders for help with spatially situated tasks [7], robots recover from failures by generating help requests that a person can understand [8, 51], and robot learners ask questions to learn from people more efficiently [52]. More recently, calibrated uncertainty in language-model planners triggers help requests with statistical guarantees [9]. Closest to our setting, Lidard et al. [10] form sets of plausible human intents and calibrate them with statistical risk control grounded in conformal prediction [53]; the robot asks for clarification whenever more than one intent remains plausible, in tasks that include navigating a cluttered home. In machine learning, the same choice between acting and deferring appears as the reject option [54], selective classification [55], and learning to defer to a human expert [56, 57]. Communication can also flow the other way: robots signal their own intentions through legible motion [58, 59], expressive lights [60], or gaze cues, which people interpret more easily than turn signals [61]. These works either study asking in tasks where waiting carries little risk, provide guarantees that assume calibration and deployment data come from the same distribution, or communicate the robot's intention rather than query the person's. The

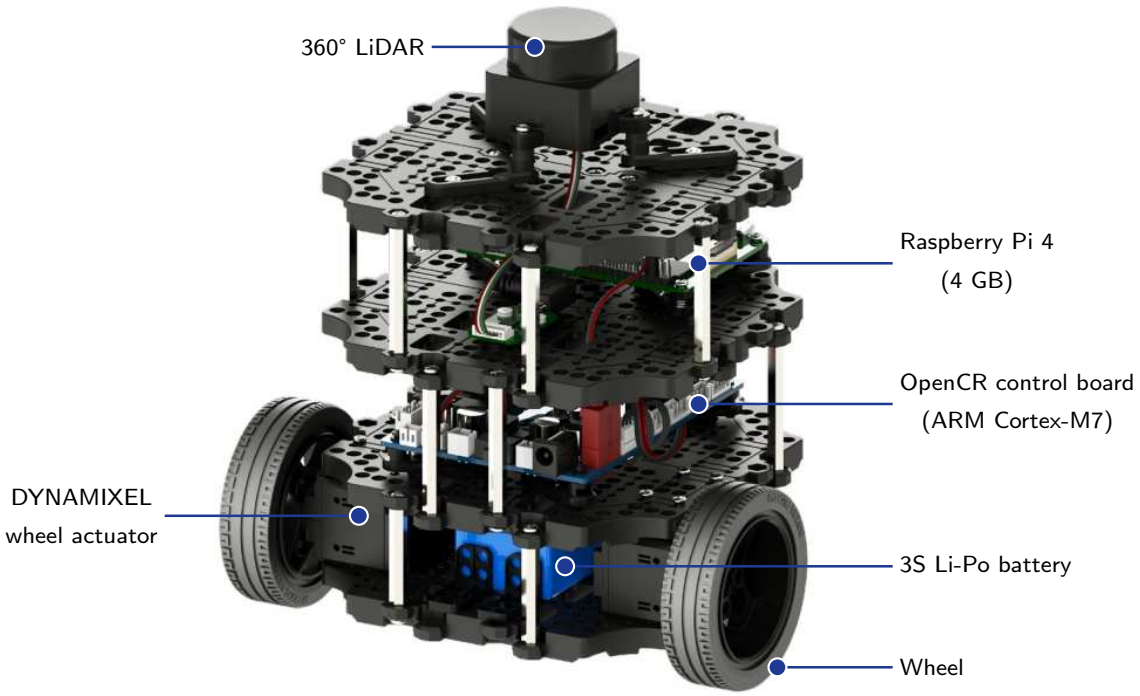

Fig. 1 Hardware of the TurtleBot3 Burger. The robot carries a 360° LiDAR, a Raspberry Pi 4 single-board computer (4 GB, quad-core Cortex-A72), an OpenCR control board (32-bit ARM Cortex-M7) that drives the two DYNAMIXEL wheel actuators, and a 3S Li-Po battery. The same robot model is used in the Gazebo experiments, and the Raspberry Pi 4 is the onboard computer used for the computation benchmark (Section 4.8). Image: ROBOTIS.

closed-loop safety of a confidence-based decision to ask, when that confidence may itself be miscalibrated at decision time, has received little attention; this is the question we address.

3 Methodology

3.1 Problem formulation

Setting: A robot drives along a known path toward a goal in an environment shared with a pedestrian. The pedestrian has a binary intent $y \in \{0, 1\}$: $y = 1$ if the pedestrian will cross the robot's path during the encounter (*crossing*), and $y = 0$ otherwise (*not crossing*). The intent is not observed.

Observations: At rate 10 Hz, the robot observes the pedestrian's position with additive Gaussian noise of standard deviation σ, which differs between operating conditions. From a short history of observations it computes a feature vector $\mathbf{f}_t$ (Section 3.2.1).

Prediction: A predictor outputs $\hat{p}_t = P(y = 1 \mid \mathbf{f}_t)$, optionally passed through a calibration map $c(\cdot)$ to give $p_t = c(\hat{p}_t)$.

Engagement and deadline: Let a_t be the distance along the robot's path from the robot to the pedestrian's position projected onto it. The pedestrian is *engaged* when $-0.5 \le a_t \le d_{\text{eng}}$ and it has not yet cleared the path; a pedestrian has *cleared* the path once it is on the opposite side from where it started and more than $d_{\text{clr}} = 0.6\,\text{m}$ from the path. A cleared pedestrian is no longer considered. The robot must commit to a decision by the *deadline*, when $a_t \le d_{\text{ask}}$, since at that point it can still stop at a safe distance.

Actions: At each control step the robot either Goes (drives at its nominal speed) or Stops. Once per pedestrian it may also Ask, which means stopping and requesting the pedestrian's intent.

Response model: An asked pedestrian answers with probability ρ, after a delay δ, and the answer is correct with probability α. The baseline is $\rho = 1$, $\delta = 1.5\,\text{s}$, $\alpha = 1$.

Objectives: The robot should avoid collisions with crossing pedestrians, avoid unnecessary stops and questions, and reach its goal without undue delay. We measure these with the collision rate on crossing

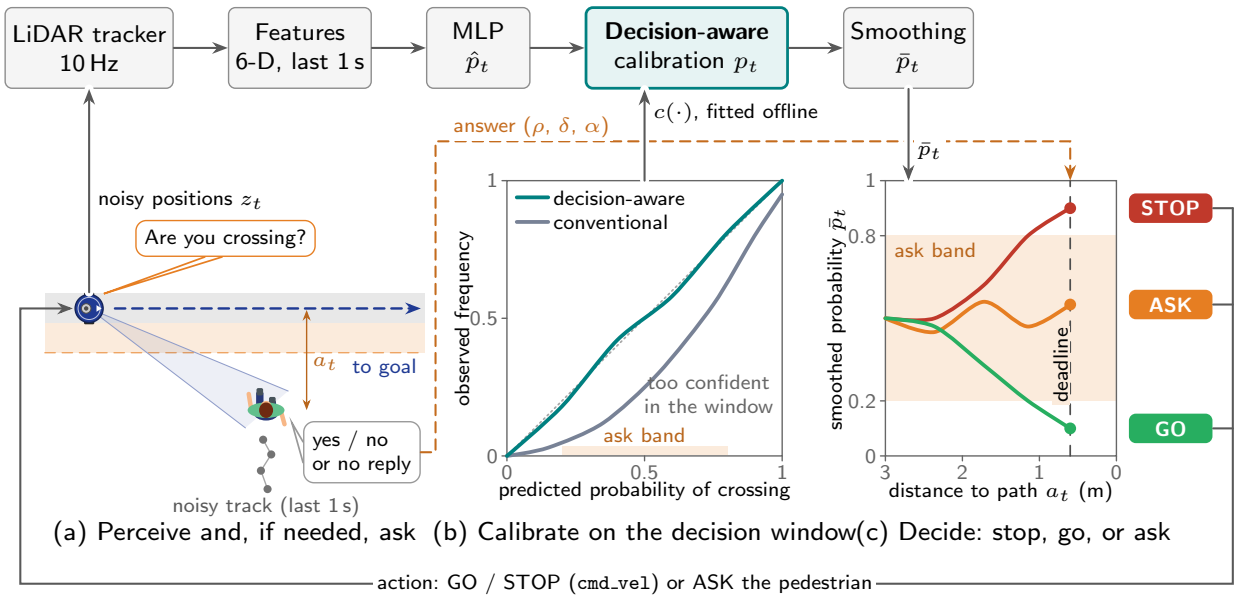

Fig. 2 The Ask or Act framework as a closed loop, run at every 10 Hz tracker update. (a) The robot tracks the pedestrian with its LiDAR, computes six features from the last second of noisy positions, and an MLP predicts the probability that the pedestrian will cross. When it asks, the pedestrian answers with probability ρ, after a delay δ, and correctly with probability α. (b) Calibration fitted on all samples (schematic reliability curves) can remain too confident in the decision window, the last 3.5 s before a crossing pedestrian reaches the path; decision-aware calibration is fitted only on that window. (c) At the deadline ($a_t = 0.6$ m), the robot stops if the smoothed probability is above the ask band $[0.2, 0.8]$, goes if it is below, and asks if it is still inside; the action is sent back to the robot, and an answer, if one arrives, replaces the prediction in the decision.

encounters, the number of questions per episode, the delay relative to driving without stopping, and the time spent stopped for pedestrians who do not cross (Section 3.4.6).

3.2 The Ask or Act framework

Figure 2 summarizes the framework. The robot tracks the pedestrian, computes features, predicts the crossing probability, calibrates it, smooths it over a short window, and applies a decision rule with an explicit deadline. Questions are answered by the pedestrian according to the response model.

3.2.1 Features

The features are computed from the last 1 s of observations (and 2 s for the stopping feature), relative to the robot's path, the line $y = 0$: (i) distance to the path; (ii) velocity toward the path; (iii) absolute velocity along the path; (iv) speed; (v) change in the velocity toward the path over the window (acceleration); and (vi) the fraction of the last 2 s spent nearly still (speed below 0.2 m/s, estimated from displacements over 0.5 s). Velocities are least-squares slopes over the window, and the current position is the mean of the last three observations, which reduces the effect of tracking noise. The same code computes the features for training and at run time.

3.2.2 Predictor

The primary predictor is a multilayer perceptron (MLP), a small feed-forward neural network, with two hidden layers of 32 units and ReLU activations, and a sigmoid output giving $\hat{p}_t$. Features are standardized to zero mean and unit variance. The MLP is trained with the Adam optimizer (learning rate 10^{-3}, L2 penalty 10^{-4}, mini-batches of 200 samples) to minimize the log loss, for at most 500 epochs, with early stopping when the loss on a held-out 10% of the training data has not improved for 10 epochs. As a simpler comparison we also evaluate logistic regression (L2-regularized, $C = 1$), which is linear in the features. We used the scikit-learn implementations [62]. Both models are trained only on nominal data (Section 3.4.4),

and each training sample is labeled with the scenario's ground-truth intent. To check that results do not depend on one training run, we retrain the MLP with five different random seeds (Section 4.5).

3.2.3 Calibration

We compare two calibration maps, each fitted on a separate calibration set, against the raw predictor. *Isotonic regression* [38] fits a non-decreasing, piecewise-constant map from raw to calibrated probability; unlike temperature scaling [11], which rescales the log-odds by a single parameter, it can correct miscalibration that differs across the probability range. Temperature scaling was examined during development but was not part of the pre-specified evaluation (Section 3.4.7), so we do not report it. The second map is decision-aware calibration, described next.

Decision-aware calibration. Standard calibration uses every sample in the calibration set. In a navigation episode, however, most samples come from moments far from any decision—a pedestrian standing still at a distance, or walking long before an encounter, and these dominate the fit. We therefore fit the isotonic map only on samples from the *decision window*: the last $W = 3.5\,\mathrm{s}$ before a crossing pedestrian would reach the robot's path, which covers the moments when the robot reaches its deadline. For pedestrians who do not cross, the window is defined by the same time reference (Section 3.4.4). The resulting map is calibrated where decisions are made, at the cost of calibration elsewhere (Section 4.1).

Each calibrator is fitted either on nominal data (*stale* under a shift) or on a small calibration set from the operating condition (*recalibrated*), which models a robot collecting a few minutes of data in a new setting.

3.2.4 Decision rule

The robot compares the smoothed probability $\bar{p}_t$ (the mean of the last five calibrated predictions, 0.5 s) with a band $[p_\ell, p_h]$ (default $[0.2, 0.8]$). For an engaged pedestrian it

- Stops (yields) if $\bar{p}_t \geq p_h$;
- Goes if $\bar{p}_t \leq p_\ell$;
- otherwise, keeps going and observing until the deadline, and then Asks once and stops until the answer arrives, after which it follows the answer.

The deadline is essential: without it, the rule asks as soon as it is uncertain, which in practice means at first engagement, before the pedestrian's behavior has revealed anything (Section 5). Two timeouts bound the waiting. If an answer does not arrive within $T_{\mathrm{ans}} = 5\,\mathrm{s}$, or if the robot has yielded to the same pedestrian for more than $T_{\mathrm{yield}} = 8\,\mathrm{s}$ without that pedestrian crossing, it falls back to a cautious rule for that pedestrian: stop only while the pedestrian is within $d_{\mathrm{c}} = 1.5\,\mathrm{m}$ of the path. Algorithm 1 gives the complete rule.

3.3 Design properties

By construction: (i) the robot asks each pedestrian at most once, and only at or after the deadline; (ii) waiting for an answer lasts at most T_{ans}; (iii) yielding on the basis of an answer or a confident prediction lasts at most T_{yield} before the cautious fallback applies; and (iv) the fallback stops only while the pedestrian is near the path. Property (iii) prevents the robot from waiting indefinitely for a pedestrian who never crosses, for example after a wrong answer. In all final experiments, no episode timed out.

3.4 Experimental setup

3.4.1 Simulation platform

Experiments use ROS 2 Humble [63] and Gazebo 11 (Classic) [64] on Ubuntu 22.04, with a TurtleBot3 Burger in the café world distributed with HuNavSim [35]. The TurtleBot3 Burger (Figure 1) is a small differential-drive robot whose onboard Raspberry Pi 4 runs ROS 2, while an OpenCR board controls its two wheel actuators; its 360° LiDAR is simulated in Gazebo. The robot follows a straight 5.8 m path from

Algorithm 1 Ask or Act decision rule for one engaged pedestrian (evaluated at 20 Hz)

Require: smoothed probability $\bar{p}_t$, distance ahead a_t, distance to path e_t

if yield timeout expired for this pedestrian **then**
  return STOP **if** $e_t < d_c$ **else** GO
else if an answer has been received **then**
  return STOP **if** answer = crossing **else** GO
else if $\bar{p}_t \geq p_h$ **then**
  return STOP ▷ confident: crossing
else if $\bar{p}_t \leq p_\ell$ **then**
  return GO ▷ confident: not crossing
else if not yet asked **and** $a_t > d_{\text{ask}}$ **then**
  return GO ▷ uncertain: keep observing
else if not yet asked **then**
  ASK; **return** STOP
else if waiting less than T_{ans} **then**
  return STOP
else
  return STOP **if** $e_t < d_c$ **else** GO ▷ no answer
end if

Table 1 Scenario families. Parameters are drawn uniformly per scenario; g is the distance from the robot's path at which the pedestrian stops or turns. All routes are shifted sideways by up to ±0.2 m.

Family	Intent	Behavior	Type	Parameters
Clear crossing	crossing	walks straight across	–	speed 0.8–1.2 m/s
Clear non-crossing	not crossing	walks away, parallel to the path	–	speed 0.8–1.2 m/s
Hesitant	both	approaches, pauses, then crosses or turns back	ambiguous	g 0.8–1.2 m, pause 1.5–2.5 s
Late turn	both	approaches, then crosses or veers away at the last moment	ambiguous	g 0.4–0.7 m
Slow crossing	crossing	crosses slowly and steadily	readable	speed 0.35–0.55 m/s
Waits far back	not crossing	approaches, stops well back and waits	readable	g 1.6–2.2 m

$(2.9, 0.1)$ to $(-2.9, 0.0)$ at 0.18 m/s with a proportional heading controller. A fixed path isolates the ask-or-act decision: differences between methods come only from when they stop, not from path planning. All timing uses simulation time. Table 2 lists all parameters used in the final experiments.

Scripted pedestrians. We initially used HuNavSim's social-force agents, but found that their trajectories varied between repeated runs of the same scenario and that paused agents occasionally failed to resume, which made ground-truth intent unreliable. We therefore implemented scripted pedestrians: each follows a plan of walking segments and pauses with a trapezoidal speed profile (acceleration 0.8 m/s^2, keeping 60% of its speed through non-stopping waypoints), as a pure function of time since release. Each pedestrian is a cylinder of radius 0.3 m that the robot's LiDAR can see but that does not physically push the robot; collisions are recorded, not simulated. Pedestrians do not react to the robot, which tests the robot on the harder case in which avoiding a conflict is entirely its responsibility.

3.4.2 Scenario families

Table 1 and Figure 3 describe the eight scenario families. The four *ambiguous* ones come in matched pairs: for the same random seed, the crossing and the not-crossing versions share every parameter and follow exactly the same motion until they diverge. Two *readable* families were added so that confidence can save questions: a slow, steady crosser whose intent is visible before the deadline, and a pedestrian who stops well back from the path. Routes run along an open corridor of the café, checked against the occupancy map.

Encounter timing. Each scenario draws a target distance $d_{\text{enc}} \sim U(0, 0.4)$ m between the robot and the crossing point at the moment a crossing pedestrian reaches the path. The pedestrian's release time and the robot's start time are then set so that a robot that never stops is exactly this far from the crossing point at that moment, producing a genuine conflict for every crossing scenario. Ambiguous pairs share their timing, which is computed from the crossing version, so timing never reveals intent. For waiting pedestrians, the robot is 1.2–1.8 m from the pedestrian when it stops.

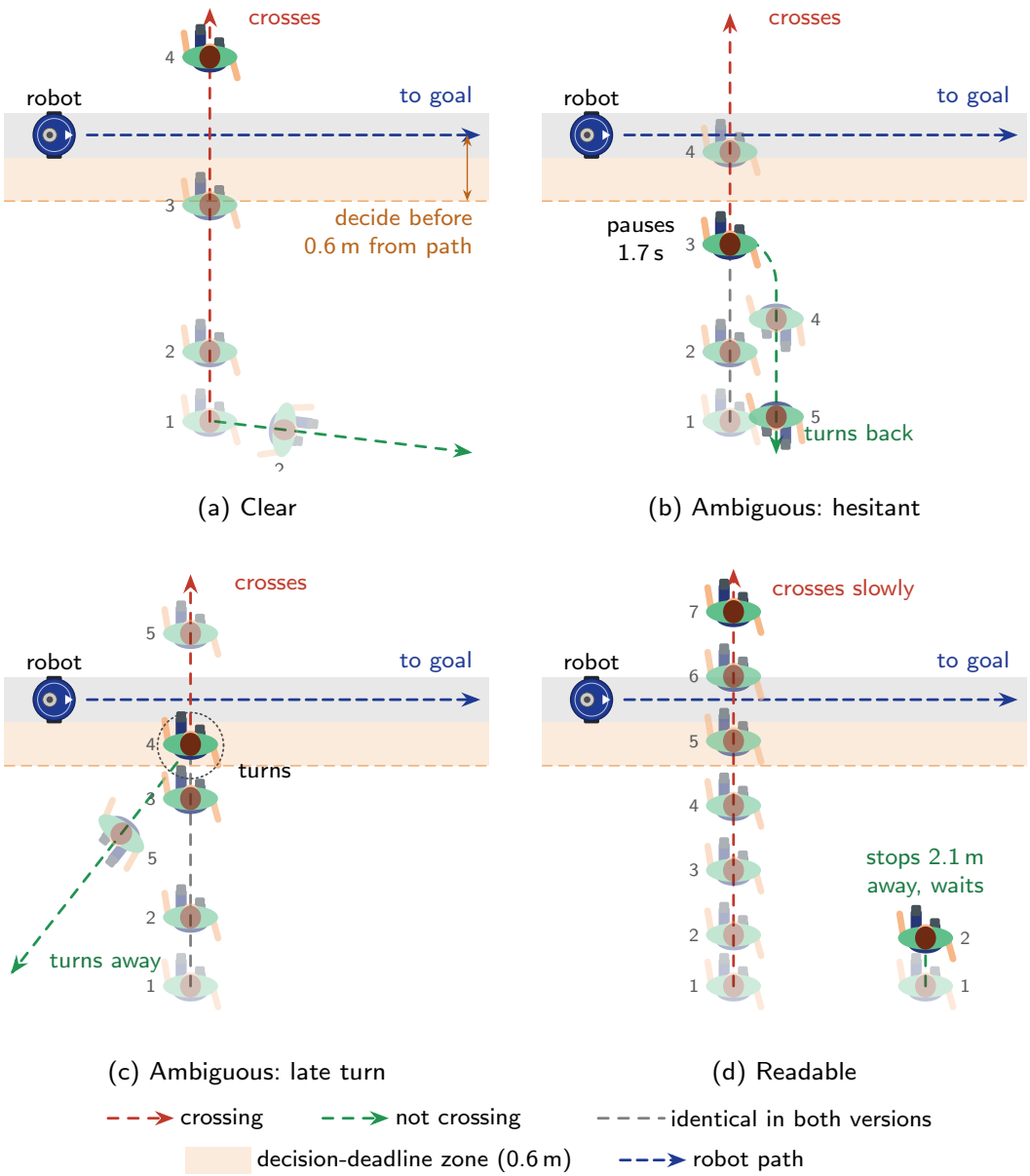

Fig. 3 The scenario families, seen from above (seed 7000). The robot drives along the gray lane toward the goal, and must decide before the pedestrian enters the orange zone, 0.6 m from its path. Pedestrians are shown every 1.2 s (numbered, fading into the past), so closely spaced icons mean slow motion. (a) Clear: the pedestrian crosses or walks away from the first step. (b) Hesitant and (c) late turn are ambiguous: the crossing and not-crossing versions are identical (gray) until the pedestrian ends a 1.7 s pause at 0.9 m from the path, or turns 0.4 m from it, so intent cannot be read from early motion. (d) Readable: the pedestrian crosses slowly or stops far from the path and waits, so intent is visible before the deadline.

3.4.3 Distribution shifts

Table 3 lists the operating conditions. The predictor is always trained on nominal data; the other conditions shift how pedestrians behave, how well the robot tracks them, or both.

3.4.4 Data

Predictor data are generated offline from the same scenario generator by computing each pedestrian's trajectory and simulating noisy tracking at 10 Hz, including 2 s of standing still before release. Samples are taken from release until the moment a crossing pedestrian would reach the path (the same moment for both members of an ambiguous pair), since after that the intent is no longer uncertain; for pedestrians

Table 2 Parameters used in all final experiments.

Parameter	Value
Robot speed, path	0.18 m/s; $(2.9, 0.1) \to (-2.9, 0.0)$
Engagement / deadline distance	1.0 m / 0.6 m
Ask band $[p_\ell, p_h]$	$[0.2, 0.8]$
Smoothing	mean of 5 predictions (0.5 s)
Answer / yield timeout	5 s / 8 s
Caution / clearance distance	$d_c = 1.5$ m / $d_{clr} = 0.6$ m
Collision distance	0.4 m
Tracking rate / noise	10 Hz; $\sigma = 0.03$ m (nominal), 0.10 m (noisy)
Response (baseline)	$\rho = 1$, $\delta = 1.5$ s, $\alpha = 1$
Pedestrian acceleration	0.8 m/s^2
Encounter distance d_{enc}	$U(0, 0.4)$ m
Predictor	MLP, 2 hidden layers of 32 ReLU units; Adam, learning rate 10^{-3}, L2 10^{-4}, batch 200, $\leq$ 500 epochs, early stopping (10% validation, patience 10); logistic regression (L2, $C = 1$)
Decision window W	3.5 s
ECE bins	15
Controller / simulator step	20 Hz
Software	ROS 2 Humble, Gazebo 11.10.2, Ubuntu 22.04, Python 3.10, numpy 1.21.5, pandas 1.3.5, scikit-learn 0.23.2 [62], scipy 1.8.0
Onboard computer	Raspberry Pi 4 Model B, 4 GB, Ubuntu 22.04 (aarch64), same Python and library versions

Table 3 Operating conditions. Only the listed quantities differ from nominal.

Condition	Change from nominal
Nominal	speed 0.8–1.2 m/s, pause 1.5–2.5 s, $\sigma = 0.03$ m
Slow	speed 0.5–0.8 m/s
Fast	speed 1.2–1.6 m/s
Long pause	pause 3–5 s
Noisy	tracking noise $\sigma = 0.10$ m
Combined	slow, long pause and noisy together

Table 4 Data splits. Each seed yields one scenario per family (eight per seed).

Split	Seeds	Size
Training	1000–1049	400 scenarios, 23,344 samples (nominal only)
Calibration	2000–2009	80 scenarios per condition (4,600–5,600 samples)
Development test	3000–3019	160 scenarios per condition
Final predictor test	7000–7099	800 scenarios per condition; 26,000–28,000 decision-window samples
Final closed loop	7000–7099	800 episodes per method and condition (400 crossing)
Gazebo validation	7000–7004	80 scenarios $\times$ 4 methods

who wait far back, until the robot would have passed them, and for clear non-crossers, for the first 5 s. Table 4 lists the splits. Seed ranges never overlap, and each condition has its own block of seeds. Scenarios used during method development were never used for the final evaluation.

3.4.5 Methods compared

We compare four baselines and four versions of the decision rule: *Always act* (never reacts); *Always cautious* (at the deadline, stops while the pedestrian is within 1.5 m of the path); *Always ask* (asks every engaged pedestrian once at the deadline and follows the answer); *Oracle* (knows the true intent and yields

only to crossing pedestrians); and the threshold rule of Section 3.2.4 with the *raw* predictor, *isotonic (stale)*, *isotonic (recalibrated)*, and *decision-aware recalibration*. Always ask and Always cautious use the same deadline and fallback as the threshold rule.

3.4.6 Metrics

A *collision* is recorded when the robot and pedestrian centers come within 0.4 m (pedestrian radius 0.3 m plus the robot's approximately 0.1 m). The primary metrics are the collision rate on crossing encounters and the number of questions per episode. Secondary metrics are the delay relative to driving without stopping, the time stopped with pedestrians who do not cross, and the timeout rate. For the predictor we report ECE with 15 equal-width bins, the Brier score, and AUC, over all test samples and within the decision window.

3.4.7 Evaluation protocol

All hypotheses, tests, parameters and robustness checks were written down and fixed before the final runs, and the code was tagged. The pre-specified hypotheses were:

- **H1**: under the noisy shift, raw (H1a) and stale-calibrated (H1b) confidence give more collisions on crossing encounters than decision-aware recalibration;
- **H2**: under the noisy shift, decision-aware recalibration gives fewer collisions than conventional (all-sample) recalibration;
- **H3**: under nominal conditions, decision-aware recalibration asks fewer questions than Always ask, with a collision rate no more than 2 pp higher (non-inferiority);
- **H4**: decision-aware recalibration has lower decision-window ECE than conventional recalibration under every shifted condition.

All comparisons between methods are paired, since every method sees the same scenarios. H1 and H2 use McNemar's exact test [65] on paired collision outcomes; H3 uses the one-sided Wilcoxon signed-rank test [66] on questions and a one-sided 95% bootstrap bound [67] on the paired collision difference; H4 uses scenario-level bootstrap intervals. The Holm correction [68] is applied across H1a, H1b, H2 and H3. Proportions are reported with 95% Wilson intervals [69]. Robustness checks: an ask-band sweep, predictors trained with four further random seeds and logistic regression, decision deadlines of 0.5 and 0.7 m, and a full grid over the response model. The onboard computation benchmark (Section 4.8) was added after the pre-specified evaluation and is reported as an additional feasibility measurement.

3.4.8 Offline simulator and Gazebo validation

Large-scale runs use an offline simulator that replicates a Gazebo episode: the robot moves along its path at constant speed or stops, the pedestrian follows its trajectory, tracking noise and answers are drawn with the same random seeds, and the same decision code is used. The offline simulator makes the final evaluation (over 290,000 episodes, including the response grid) feasible; its validity is checked against full Gazebo simulation on a pre-specified subset, with a criterion set in advance of at least 90% agreement in collision outcome.

4 Results

4.1 Predictor calibration under distribution shift

Figure 4 shows reliability diagrams for nominal and noisy tracking. Changes in pedestrian behavior (slow, fast, long pause) have only a modest effect on calibration, whereas degraded tracking has a large one: under noise, the raw predictor becomes overconfident. Over all samples, the raw MLP's ECE is 0.040 under nominal conditions and 0.047–0.066 under the behavioral shifts, but 0.076 under noise and 0.101 under the combined shift (Table 5). Recalibrating on a small calibration set from the new condition restores calibration over all samples, to an ECE of 0.010–0.018 in every condition. Noise also costs discrimination, which no calibration can recover: the decision-window AUC falls from 0.896 (nominal) to 0.826 (noisy) and 0.847 (combined).

Table 5 Calibration of the MLP on the final predictor test set: ECE over all samples and within the decision window, for the raw predictor, conventional isotonic recalibration and decision-aware recalibration (both recalibrated on each condition), and decision-window AUC, which calibration does not change.

	Raw		Isotonic (recalibrated)		Decision-aware (recalibrated)		
Condition	All	Window	All	Window	All	Window	AUC (window)
Nominal	0.040	0.053	0.011	0.036	0.047	0.015	0.896
Slow	0.066	0.063	0.010	0.034	0.031	0.009	0.907
Fast	0.047	0.069	0.015	0.041	0.042	0.017	0.903
Long pause	0.051	0.070	0.015	0.052	0.055	0.019	0.903
Noisy	0.076	0.093	0.013	0.056	0.070	0.017	0.826
Combined	0.101	0.079	0.018	0.036	0.049	0.014	0.847

Table 6 Decision-window ECE (H4) on the final predictor test set, for isotonic recalibration fitted on all calibration samples versus only on the decision window. Both columns are evaluated on decision-window test samples; the interval is a 95% scenario-bootstrap interval of the improvement.

Condition	Fitted on all	Fitted on window	Improvement	95% CI
Nominal	0.036	0.015	0.020	[−0.002, 0.029]
Slow	0.034	0.009	0.025	[0.001, 0.026]
Fast	0.041	0.017	0.024	[0.000, 0.032]
Long pause	0.052	0.019	0.033	[0.007, 0.043]
Noisy	0.056	0.017	0.039	[0.009, 0.051]
Combined	0.036	0.014	0.022	[−0.010, 0.033]

Figure 5 shows the central observation of this paper. Conventional recalibration achieves the lowest ECE over all test samples (0.010–0.018), but within the decision window its ECE is two to four times higher (0.034–0.056); under noise, for example, it is 0.013 over all samples and 0.056 in the window. Decision-aware calibration shows the opposite pattern: it is worse over all samples (0.031–0.070) but best in the decision window (0.009–0.019; Tables 5 and 6). A standard evaluation over all samples would therefore choose the calibrator that performs worse where decisions are made.

H4 predicted lower decision-window ECE under *every* shifted condition. Its decision-window ECE was lower in all five shifted conditions, but the bootstrap interval of the improvement excluded zero in only four of them (slow, fast, long pause, noisy), not under the combined shift. As pre-specified, H4 is therefore not supported; descriptively, decision-aware calibration reduced decision-window ECE in all conditions, with a reliable improvement in four of the five shifted ones.

4.2 Closed-loop safety and questions

Table 7 gives the closed-loop results for the two primary conditions, and Table 8 the hypothesis tests. Always act collides in every crossing encounter, confirming that every crossing scenario is a genuine conflict. Always ask and the Oracle never collide. Always cautious collides in 16–18% of crossings, because a pedestrian who is still far from the path at the deadline can reach it quickly afterward.

Under the noisy shift, the raw and stale-calibrated threshold rules become overconfident, skip questions they should ask (about 0.5 questions per episode, against 0.86 for decision-aware recalibration), and collide in 34.5% and 31.8% of crossing encounters. Decision-aware recalibration reduces this to 1.3%. The paired comparison is entirely one-sided: in 133 and 122 scenarios, respectively, only the raw or stale rule collided, and in none did only decision-aware recalibration collide (H1a, H1b). Conventional recalibration helps but leaves 15.5% of crossings ending in collision (57 scenarios versus 0; H2).

Under nominal conditions, decision-aware recalibration asks 0.73 questions per episode against 0.88 for Always ask (16% fewer), with a collision rate 0.8 pp higher (3 of 400 crossing encounters). The one-sided 95% upper bound on the difference, 1.5 pp, is inside the 2 pp margin, so decision-aware recalibration is non-inferior to Always ask (H3). It also stops for less time with pedestrians who do not cross (0.81 against 1.13 s). All four tests remain significant after the Holm correction.

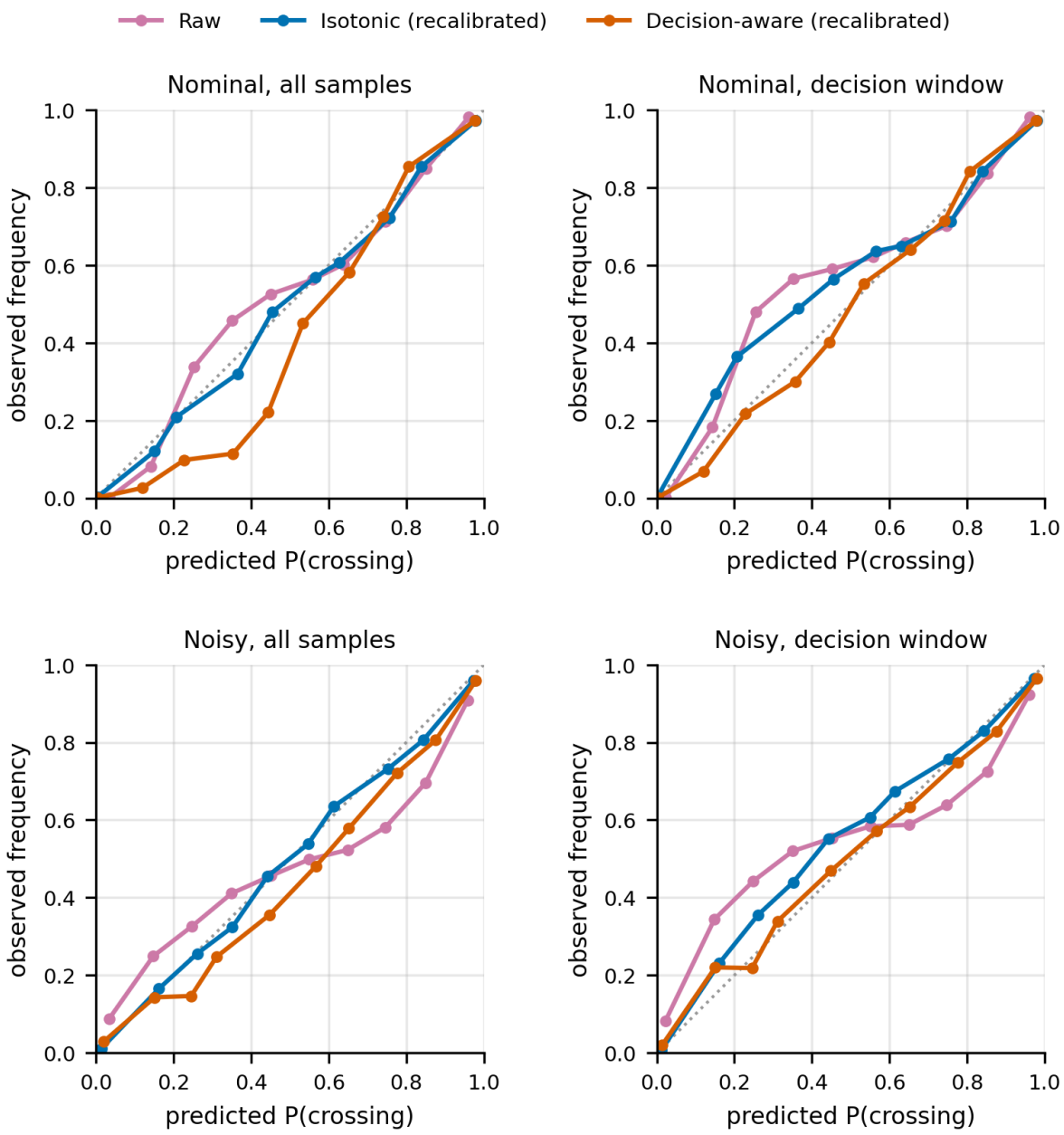

Fig. 4 Reliability diagrams on the final predictor test set, over all samples and within the decision window. The dotted diagonal indicates perfect calibration.

4.3 Results across all conditions

Figure 6 and Table 9 show collisions on crossing encounters in all six conditions. Decision-aware recalibration has the fewest collisions among the threshold rules in five of six conditions, with rates between 0.3% and 1.8% outside the combined shift. Under the combined shift, however, it collides in 14.5% of crossings, more than conventional recalibration (8.5%). We return to this in Section 5. Always cautious is unreliable: its collision rate ranges from 2.5% to 28.3% depending on the condition.

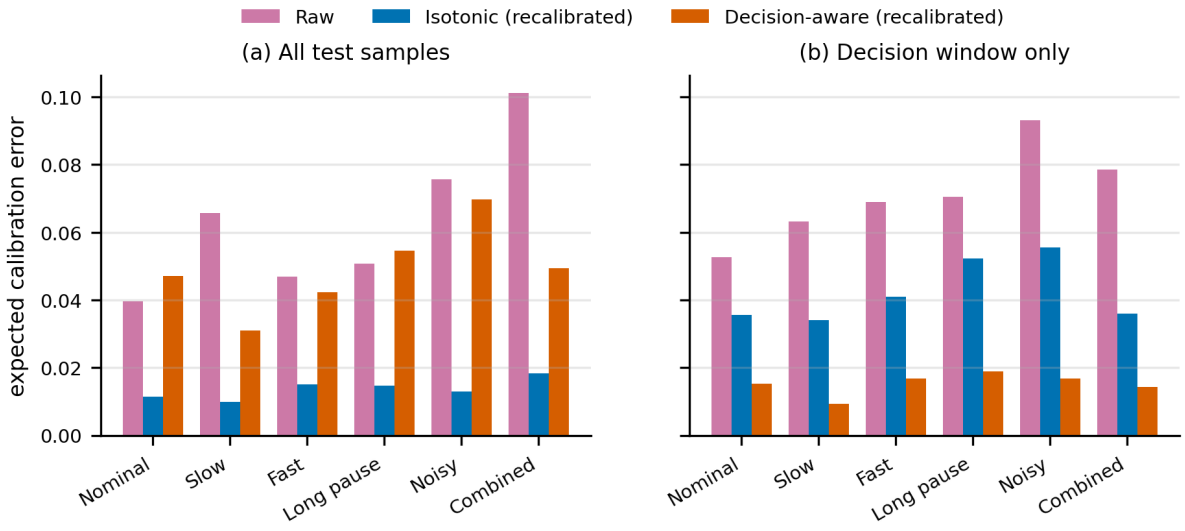

Fig. 5 Aggregate calibration hides miscalibration where decisions are made. Expected calibration error on the final predictor test set (a) over all samples and (b) within the decision window, for the raw predictor, conventional isotonic recalibration and decision-aware recalibration.

Table 7 Closed-loop results on the final scenarios (800 episodes per method and condition, of which 400 are crossing encounters). Collisions are on crossing encounters, with 95% Wilson intervals (one crossing encounter is 0.25 pp; values ending in 0.25 or 0.75 are rounded up); delay is relative to driving without stopping; "stop" is the time stopped with pedestrians who do not cross.

	Nominal				Noisy			
Method	Collisions (%)	Questions	Delay (s)	Stop (s)	Collisions (%)	Questions	Delay (s)	Stop (s)
Always act	100.0 [99.0, 100]	0.00	0.02	0.00	100.0 [99.0, 100]	0.00	0.02	0.00
Always cautious	18.0 [14.5, 22.1]	0.00	1.86	1.03	16.5 [13.2, 20.5]	0.00	1.69	0.92
Always ask	0.0 [0.0, 1.0]	0.88	2.16	1.13	0.0 [0.0, 1.0]	0.89	2.21	1.15
Oracle	0.0 [0.0, 1.0]	0.00	2.60	0.00	0.0 [0.0, 1.0]	0.00	2.39	0.00
Raw	4.5 [2.9, 7.0]	0.69	1.98	0.80	34.5 [30.0, 39.3]	0.52	1.58	0.83
Isotonic (stale)	4.5 [2.9, 7.0]	0.70	1.96	0.75	31.8 [27.4, 36.5]	0.54	1.60	0.79
Isotonic (recalibrated)	4.5 [2.9, 7.0]	0.70	1.96	0.75	15.5 [12.3, 19.4]	0.69	1.85	0.86
Decision-aware (recalibrated)	**0.8 [0.3, 2.2]**	0.73	2.04	0.81	**1.3 [0.5, 2.9]**	0.86	2.22	1.15

Table 8 Pre-specified hypothesis tests. "Discordant" counts the scenarios in which only one of the two methods collided. p-values are Holm-corrected.

	Comparison (condition)	Result	p (Holm)	Supported
H1a	Raw vs. decision-aware (noisy)	collisions 34.5% vs. 1.3%; discordant 133 vs. 0	< 0.001	yes
H1b	Isotonic stale vs. decision-aware (noisy)	collisions 31.8% vs. 1.3%; discordant 122 vs. 0	< 0.001	yes
H2	Isotonic recalibrated vs. decision-aware (noisy)	collisions 15.5% vs. 1.3%; discordant 57 vs. 0	< 0.001	yes
H3	Decision-aware vs. Always ask (nominal)	questions 0.73 vs. 0.88; collisions +0.8 pp (upper 95%: 1.5 pp)	< 0.001	yes
H4	Decision-window ECE (5 shifted conditions)	lower in all five; interval excludes 0 in four (not under the combined shift)	–	no

Table 9 Collisions on crossing encounters (%) in all conditions. Always act collides in 100% and Always ask and the Oracle in 0% of crossing encounters in every condition.

Method	Nominal	Slow	Fast	Long pause	Noisy	Combined
Always cautious	18.0	2.8	28.3	14.3	16.5	2.5
Raw	4.5	2.8	4.0	5.5	34.5	23.3
Isotonic (stale)	4.5	2.5	3.0	5.0	31.8	20.5
Isotonic (recalibrated)	4.5	2.8	4.0	1.8	15.5	8.5
Decision-aware (recalibrated)	0.8	1.8	0.5	0.3	1.3	14.5

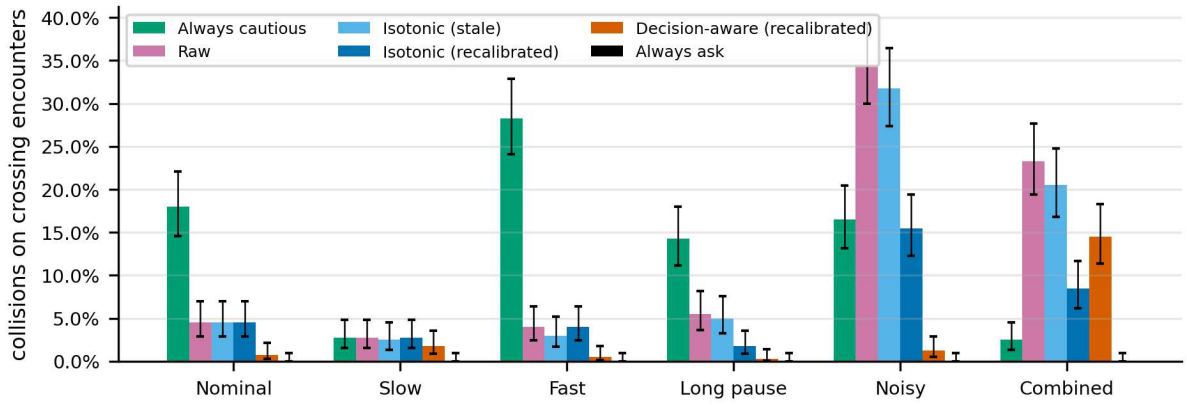

Fig. 6 Collisions on crossing encounters in all six conditions, with 95% Wilson intervals (400 crossing encounters per bar). Always act (100% in every condition) and the Oracle (0%) are omitted.

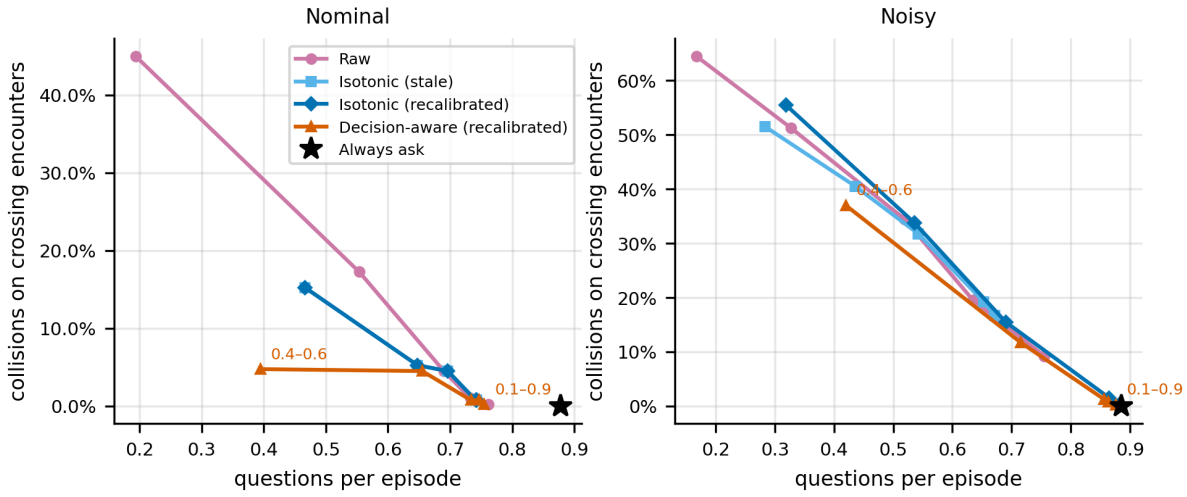

Fig. 7 Trade-off between questions and collisions as the ask band varies (labels give the narrowest and widest bands for decision-aware recalibration); the star marks Always ask.

4.4 The ask-band trade-off

Figure 7 shows collisions against questions as the ask band varies from $[0.1, 0.9]$ to $[0.4, 0.6]$.

Under noisy tracking the curves of all methods nearly coincide, and collisions fall roughly linearly with questions. When tracking is poor, the predictor carries little information at the deadline, so the robot must effectively choose between asking and risking a collision. At equal question rates, decision-aware recalibration is only slightly better; its advantage at the fixed default band comes mainly from asking more (0.86 against 0.69 questions per episode). Conventional recalibration with a wider band reaches a similar operating point. The practical difference is that a single fixed band behaves appropriately in both conditions with decision-aware recalibration, whereas conventional recalibration would need its band retuned when tracking degrades.

4.5 Robustness

Predictor seeds and model class. Table 10 repeats the comparison with four further MLPs trained with different random seeds, and with logistic regression. For every predictor, in both conditions, decision-aware recalibration had the fewest collisions among the calibration methods. The magnitudes, however,

Table 10 Collisions on crossing encounters (%): range over five MLP predictors trained with different random seeds; logistic regression in parentheses.

Method	Nominal	Noisy
Raw	0.8–4.5 (43.0)	18.5–52.5 (19.0)
Isotonic (stale)	0.5–4.5 (51.2)	15.0–56.5 (21.5)
Isotonic (recalibrated)	0.5–4.5 (51.2)	5.5–15.8 (11.0)
Decision-aware (recalibrated)	0.0–0.8 (32.3)	1.3–10.5 (2.8)

vary considerably between MLPs: under noise, collision rates ranged from 18.5% to 52.5% for the raw predictor, from 5.5% to 15.8% for conventional recalibration and from 1.3% to 10.5% for decision-aware recalibration. The ordering holds within every predictor, but the ranges overlap across predictors: the best conventionally recalibrated MLP (5.5%) collided less often than the worst decision-aware one (10.5%). Under noise, the main predictor is the most favorable case for decision-aware recalibration; under nominal conditions it is the least favorable (0.8%, against a range of 0.0–0.8%). The ranges are therefore the fairer summary. Logistic regression shows a different limitation: under nominal conditions it collided in 32–51% of crossings with every calibration method. A linear model cannot separate a pedestrian walking steadily toward the path from one about to turn away, and calibration cannot create discrimination that the model does not have. It collided less under noise (2.8–21.5%), Part of the explanation is that noise pushes its probabilities into the ask band more often, so that it asks rather than acts on a confident but wrong prediction: it asked more questions under noise with every calibration method (0.39–0.56 questions per episode under nominal conditions against 0.51–0.84 under noise), most strongly after recalibration (0.39 to 0.78 for conventional and 0.56 to 0.84 for decision-aware recalibration). For the raw predictor, however, questions increased only slightly (0.47 to 0.53) while collisions fell from 43.0% to 19.0%, so more frequent asking does not fully account for the difference.

Decision deadline. With deadlines of 0.5 and 0.7 m instead of 0.6 m, collision rates were identical for every method in both conditions; only questions and delay changed slightly (for decision-aware recalibration under nominal conditions, 0.73 and 0.75 questions per episode, and 1.78 and 2.32 s delay).

4.6 Unreliable answers

We varied the probability of an answer (1.0, 0.8, 0.5), the answer delay (0.5, 1.5, 3 s), and the answer accuracy (1.0, 0.9) in a full grid. We had expected that a calibrated threshold rule, which asks less, would degrade less than Always ask as answers become unreliable. This was not the case.

With truthful answers, collision rates were identical across all nine combinations of response probability and delay, for every method. Because the robot stops while it waits, missing or slow answers cost time but not safety. Wrong answers do cost safety, and their effect depends strongly on the delay (Figure 8). With 90% accurate answers and every question answered, Always ask collided in 9.0%, 3.0% and 0% of crossing encounters at delays of 0.5, 1.5 and 3 s: a quick wrong "I'm not crossing" sends the robot on while the pedestrian is still approaching, whereas a slow wrong answer arrives after the pedestrian has already crossed. For the same reason, fewer answers made the robot *safer* when answers could be wrong (at 0.5 s delay: 9.0%, 7.8% and 4.0% for response probabilities of 1.0, 0.8 and 0.5), because an unanswered question triggers the cautious fallback. Decision-aware recalibration tracked Always ask within about 1.3 pp in every setting. Its advantage was efficiency: under nominal conditions its delay was lower than that of Always ask, by about 0.1 s at a 1.5 s answer delay and 0.3 s at 3 s.

4.7 Gazebo validation

We ran 80 final-seed scenarios (five per family, nominal and noisy) with four methods in full Gazebo simulation and compared each episode with its offline counterpart (Table 11, Figure 9). Collision outcomes agreed in 97.5% of the 320 episodes, above the pre-specified 90% criterion; minimum distances correlated with $r = 0.984$ (mean absolute difference 5.6 cm), and question counts matched in 95.6% of episodes. Always ask agreed perfectly, and decision-aware recalibration in 79 of 80 episodes. For Always act, Gazebo encounters were occasionally less close than offline (median difference 2 cm, mean 9 cm–12 cm), consistent

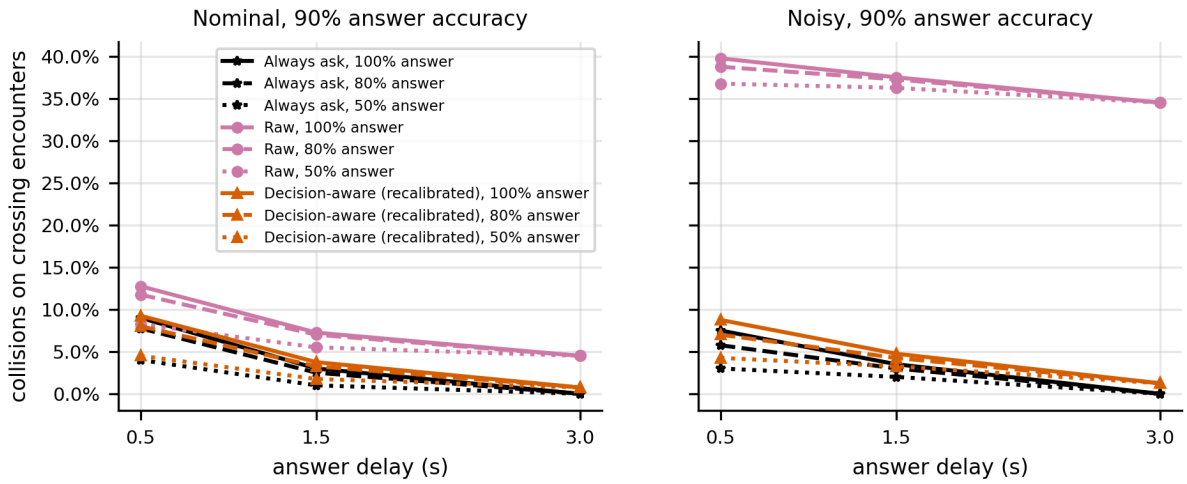

Fig. 8 Collisions on crossing encounters with 90% accurate answers, against answer delay, for three response probabilities. Wrong answers are dangerous mainly when they arrive quickly.

Table 11 Gazebo validation: agreement with the offline simulator on the same scenarios (40 episodes per method and condition).

Method	Condition	Agreement (%)	r	Same questions (%)
Always act	nominal	100.0	0.996	100.0
	noisy	97.5	0.987	100.0
Always ask	nominal	100.0	1.000	100.0
	noisy	100.0	0.999	100.0
Raw	nominal	95.0	0.918	92.5
	noisy	90.0	0.988	77.5
Decision-aware	nominal	100.0	1.000	100.0
	noisy	97.5	0.998	95.0
All		97.5	0.984	95.6

with small start-up timing differences that matter only when both robot and pedestrian are moving; this makes the offline simulator slightly conservative. The raw predictor agreed least (95% and 90%), with disagreements in both directions and matching aggregate collision rates. A plausible explanation, which we have not tested, is that its overconfident probabilities often lie near a threshold, so small differences in tracking noise between the two simulators change its decisions.

4.8 Onboard computation

To check that the method can run on the robot itself, we ran the decision pipeline used in all experiments—feature extraction, the MLP, decision-aware calibration, smoothing and the decision rule—on the robot's onboard computer, a Raspberry Pi 4 Model B (4 GB, four cores; Figure 1), and on a desktop computer for comparison, with identical software versions. Inputs were noisy pedestrian observations from 80 scenarios per condition (nominal and noisy), 141 observations per scenario at 10 Hz (11,280 in total). The decision rule was called twice per observation, at 20 Hz, giving 22,560 calls per condition. Prediction steps were timed only once enough observation history existed to compute the features, and after excluding the first 200 steps as warm-up, giving 10,280 timed prediction steps per condition. We also ran the decision rule at its real 20 Hz rate for 60 s (1,200 decisions), on the Raspberry Pi both alone and while the robot's standard ROS 2 drivers were running (motors, sensors and the LiDAR). This measures the decision pipeline only; a LiDAR people tracker, which a physical deployment would also need, is not included.

Table 12 gives the results. On the robot, a complete prediction step took a median of 2.9 ms (3.1 ms with the robot's drivers running), about 6% of the 50 ms control period, and the longest step in the real-time runs took 9.1 ms. No deadline was missed, and the pipeline used about 1% of one processor core.

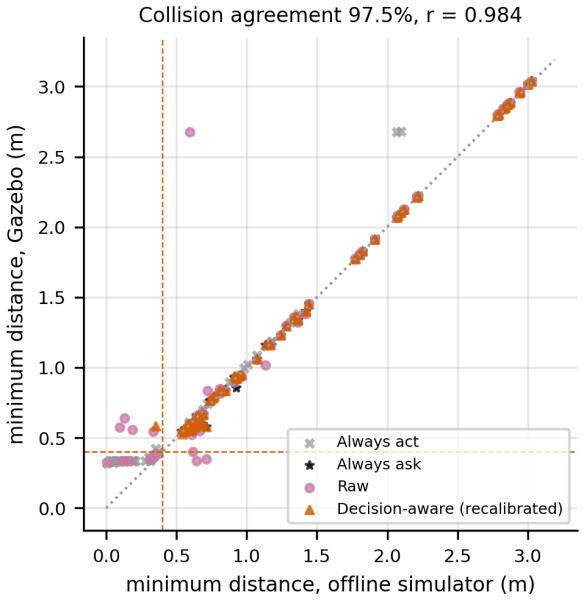

Fig. 9 Minimum robot–pedestrian distance in Gazebo against the offline simulator for the same 320 episodes. Dashed lines mark the 0.4 m collision distance.

Table 12 Onboard computation of the decision pipeline (nominal condition). A prediction step is feature extraction, MLP and calibration; a decision call is one evaluation of the decision rule at 20 Hz, which reuses the prediction between 10 Hz observations. The real-time run executed 1,200 decisions at 20 Hz with a 50 ms deadline.

	Prediction step (ms)			Decision call (ms)	Real-time run at 20 Hz		
Computer	Median	99th pct.	Max	Median	Max (ms)	Misses	CPU (% of a core)
Desktop (Intel i7-14700K)	0.28	0.67	2.51	0.006	6.1	0	0.5
Robot (Raspberry Pi 4)	2.91	3.07	5.03	0.083	8.9	0	0.8
Robot (Raspberry Pi 4), drivers on	3.06	3.69	5.05	0.083	9.1	0	1.3

The Raspberry Pi was about ten times slower than the desktop. Feature extraction accounted for about 60% of the time (1.75 ms), more than the MLP (0.80 ms) or calibration (0.34 ms). Results under noisy tracking were similar (median 2.9 ms, 99th percentile 4.6 ms on the robot). The model file is 74 kB, loads in 0.14 s, and the process used 91 MB of memory.

5 Discussion

Calibrate where decisions are made. The central lesson is that a confidence-based decision rule is only as safe as its probabilities *at the moments it acts on them*. Aggregate calibration metrics are dominated by easy moments far from any decision, and in our experiments they would have selected conventional recalibration—which left 15.5% of crossings under noise ending in collision—over decision-aware recalibration, which left 1.3%. We recommend that calibration for decision-making robots be both fitted and evaluated on the decision window.

Honest uncertainty makes the robot defer. Under nominal tracking, decision-aware recalibration saves questions (16% fewer than always asking) without a meaningful loss of safety. Under degraded tracking it asks almost as often as always asking and stops longer for non-crossers than under nominal tracking (1.15 s against 0.81 s, the same as Always ask under noise). We regard this as the correct behavior: when perception degrades, calibrated uncertainty tells the robot that its own judgment has become unreliable, and it defers to asking without needing to know that conditions have changed. The band analysis (Section 4.4) qualifies this: under noise, the benefit comes from appropriate deferral at a fixed band rather than from a better trade-off at equal question rates.

The deadline matters. In an early version of the rule, the robot asked as soon as it was uncertain. Because the predictor is honestly uncertain when it first sees a pedestrian who has not yet revealed anything, this made the threshold rule identical to always asking. Asking only at the deadline, while observing until then, lets evidence accumulate and is what allows confidence to save questions.

Where it fails. Under the combined shift, decision-aware recalibration collided more often than conventional recalibration (14.5% against 8.5%). A plausible, post-hoc explanation is that the decision window is defined as a fixed time before the pedestrian reaches the path: with slow pedestrians and long pauses, the robot's actual decisions happen earlier than that window, so the calibrator is fitted on the wrong moments. Defining the window by the robot's own state—for example, samples in which the pedestrian is engaged—is a natural remedy that we have not evaluated. Results also depend strongly on the predictor: magnitudes varied several-fold between MLPs trained with different seeds, and a linear model failed regardless of calibration. Calibration is necessary but not sufficient; the predictor must discriminate well in the decision window.

The risk of asking lies in fast wrong answers. Missing and slow answers cost only time, because the robot waits safely. The danger is a quick, confident, wrong "go ahead". This suggests that an asking robot should treat an answer that conflicts with its own calibrated prediction with caution, for example by requiring agreement before proceeding when its prediction is strongly in favor of crossing.

Limitations. Our pedestrians are scripted and do not react to the robot, being asked, or each other; real people would adapt, which may make asking more effective (people pause when asked) or less (people ignore repeated questions). The response model is simple, and its parameters are assumptions rather than measurements. Each episode involves a single pedestrian and a fixed robot path at low speed. The intent is binary. All closed-loop results are in simulation, with one environment; the offline simulator was validated against Gazebo but not against the physical world. On the physical robot we measured only computation, not perception: the onboard benchmark excludes a LiDAR people tracker, whose cost and tracking noise a deployment would add. The decision-window length was chosen once and not tuned, and the explanation of the combined-shift result is post hoc.

Future work. We see three directions for future work.

(1) Decision windows that follow the robot's state, with guarantees under shift. The failure under the combined shift suggests that the decision window should be defined by the robot's own state rather than by a fixed time before the pedestrian reaches the path. For example, the window could include all samples in which a pedestrian is engaged, or those within a fixed distance of the deadline. Such a window would follow the robot's actual decisions when pedestrians walk slowly or pause for long periods. Beyond the choice of window, the recalibrated condition in our study assumes that the robot can collect a small labeled set in each new setting. In practice the robot would need to detect the shift and adapt without labels. Calibration under covariate shift through unsupervised domain adaptation [70] and online calibration that holds even against adversarial sequences [71] address this need. Adaptive conformal inference [47] and multivalid conformal prediction [72] go further and provide coverage that holds over time and within subgroups, which matches the decision-window idea in spirit. A related step is to replace the fixed ask band with a band chosen by conformal risk control [73]. That would bound the expected collision rate on crossing encounters directly rather than through calibration error. Combining such guarantees with set-valued intent prediction [10] and with sample-efficient safety assurances for planners [74] would connect the empirical findings here to formal safety statements.

(2) Interactive pedestrians, crowds and real-world validation. Our pedestrians are scripted, do not react to the robot, and appear one at a time along a fixed path. Real people adapt to a robot's motion, and a robot's actions in turn shape what people do [75]. Interaction-aware planners that reason over multimodal, action-conditioned forecasts [29, 76] would let us test whether asking remains useful once the robot can also influence the pedestrian by slowing or swerving. Reactive agents such as those in HuNavSim [35] are a natural first step, provided their ground-truth intent can be recorded reliably. Scenes with several pedestrians and full path planning would test whether a single question remains the right unit of communication, or whether the robot should decide whom to ask. Intent should also extend beyond a binary label to destinations and timing. Evaluation should then move from scripted scenarios to recorded human trajectories, such as the ETH and UCY datasets [77, 78] and real-world robot navigation data [79]. Benchmarking frameworks for social navigation [80], together with recent guidelines for evaluating social navigation algorithms [81], would make comparisons with other methods fair. Finally,

physical experiments on the TurtleBot3, with an onboard LiDAR people tracker, are needed to show that the method survives the gap between simulation and reality. Domain randomization of tracking noise and pedestrian behavior during training [82] may reduce that gap before such experiments are run.

(3) Measured human responses and fusing answers with predictions. The response model is an assumption, and our results show that its accuracy and timing matter most: a fast, wrong "go ahead" is the main risk of asking. Work on how pedestrians and automated vehicles communicate [83] and on external human–machine interfaces [84] shows that the way a request is expressed strongly affects how people interpret it and how quickly they respond. A user study should therefore measure response probability, delay and accuracy directly, and should compare spoken, visual and motion-based ways of asking. Immersive virtual reality offers a safe and controlled setting for such studies of pedestrian decisions [85]. It also allows the scripted ambiguous scenarios of this benchmark to be reproduced with real participants. Repeated questions may also change human trust and reliance over time [86, 87], so a robot that asks too often may receive less careful answers. On the algorithmic side, the robot should treat an answer as noisy evidence rather than ground truth. One option is to fuse the answer with its calibrated prediction using models of annotator reliability [88]. Another is to require agreement before proceeding when its prediction strongly favors crossing. Studies of human–AI teams show that combined decisions outperform either party alone only when each side's reliability is represented explicitly [89]. Learning when to defer [56, 57] could then be extended from a single decision to the sequential setting studied here.

6 Conclusion

We presented Ask or Act, a framework in which a robot decides whether to proceed, yield or ask a pedestrian based on a calibrated probability that the pedestrian will cross its path, with a decision deadline that lets evidence accumulate before the robot commits. Under a pre-specified evaluation on held-out scenarios, miscalibrated confidence under degraded tracking led the robot to skip questions it should have asked, resulting in collisions in roughly a third of crossing encounters; calibrating on the decision window reduced this to 1.3%, while conventional recalibration left 15.5%. Across predictors trained with different seeds, decision-aware recalibration was always the safer of the two, but the size of the gain varied (1.3–10.5% against 5.5–15.8%). Under nominal conditions the framework asked 16% fewer questions than always asking, within a pre-set safety margin, and its decision pipeline ran in real time on the robot's onboard computer using about 1% of one processor core. The broader lesson is that for robots that act on predicted probabilities, calibration should be fitted and evaluated where decisions are made.

Declarations

- **Funding:** The authors gratefully acknowledge funding support from the United States Department of Agriculture (USDA), Grant No. 14066635.

- **Competing interests:** The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- **Ethics approval and consent to participate:** Not applicable. The study used only simulated pedestrians and involved no human participants.
- **Consent for publication:** Not applicable.
- **Data availability:** The datasets generated and analyzed during the current study are available in the GitHub repository at https://github.com/daniel-udekwe/turtlebot3-ask_or_act.
- **Materials availability:** Not applicable.
- **Author contribution: Daniel Udekwe:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Qianwen Guo:** Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing.

References

[1] Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: A survey. Robotics and Autonomous Systems **61**(12), 1726–1743 (2013)

[2] Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., Oh, J.: Core challenges of social robot navigation: A survey. ACM Transactions on Human-Robot Interaction **12**(3), 1–39 (2023)

[3] Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: A survey. The International Journal of Robotics Research **39**(8), 895–935 (2020)

[4] Fisac, J.F., Bajcsy, A., Herbert, S.L., Fridovich-Keil, D., Wang, S., Tomlin, C.J., Dragan, A.D.: Probabilistically safe robot planning with confidence-based human predictions. arXiv preprint arXiv:1806.00109 (2018)

[5] Ryu, K., Mehr, N.: Integrating predictive motion uncertainties with distributionally robust risk-aware control for safe robot navigation in crowds. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 2410–2417 (2024). IEEE

[6] Majumdar, A., Pavone, M.: How should a robot assess risk? towards an axiomatic theory of risk in robotics. In: Robotics Research: The 18th International Symposium ISRR, pp. 75–84 (2019). Springer

[7] Rosenthal, S., Veloso, M., Dey, A.K.: Is someone in this office available to help me? proactively seeking help from spatially-situated humans. Journal of Intelligent & Robotic Systems **66**(1), 205–221 (2012)

[8] Tellex, S., Knepper, R., Li, A., Rus, D., Roy, N.: Asking for help using inverse semantics. Robotics: Science and Systems Proceedings (2014)

[9] Ren, A.Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., Xu, P., Takayama, L., Xia, F., Varley, J., et al.: Robots that ask for help: Uncertainty alignment for large language model planners. arXiv preprint arXiv:2307.01928 (2023)

[10] Lidard, J., Pham, H., Bachman, A., Boateng, B., Majumdar, A.: Risk-calibrated human-robot interaction via set-valued intent prediction. arXiv preprint arXiv:2403.15959 (2024)

[11] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning, pp. 1321–1330 (2017). PMLR

[12] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. Advances in neural information processing systems **32** (2019)

[13] Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE robotics & automation magazine **4**(1), 23–33 (1997)

[14] Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics Research: the 14th International Symposium ISRR, pp. 3–19 (2011). Springer

[15] Kretzschmar, H., Spies, M., Sprunk, C., Burgard, W.: Socially compliant mobile robot navigation via inverse reinforcement learning. The International Journal of Robotics Research **35**(11), 1289–1307 (2016)

[16] Everett, M., Chen, Y.F., How, J.P.: Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3052–3059 (2018). IEEE

[17] Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: Crowd-aware robot navigation

with attention-based deep reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6015–6022 (2019). IEEE

[18] Bajcsy, A., Herbert, S.L., Fridovich-Keil, D., Fisac, J.F., Deglurkar, S., Dragan, A.D., Tomlin, C.J.: A scalable framework for real-time multi-robot, multi-human collision avoidance. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 936–943 (2019). IEEE

[19] Nishimura, H., Ivanovic, B., Gaidon, A., Pavone, M., Schwager, M.: Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11205–11212 (2020). IEEE

[20] Hakobyan, A., Yang, I.: Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. IEEE Transactions on Robotics **38**(2), 939–957 (2021)

[21] Lindemann, L., Cleaveland, M., Shim, G., Pappas, G.J.: Safe planning in dynamic environments using conformal prediction. IEEE Robotics and Automation Letters **8**(8), 5116–5123 (2023)

[22] Dixit, A., Lindemann, L., Wei, S.X., Cleaveland, M., Pappas, G.J., Burdick, J.W.: Adaptive conformal prediction for motion planning among dynamic agents. In: Learning for Dynamics and Control Conference, pp. 300–314 (2023). PMLR

[23] Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 797–803 (2010). IEEE

[24] Trautman, P., Ma, J., Murray, R.M., Krause, A.: Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. The International Journal of Robotics Research **34**(3), 335–356 (2015)

[25] Ziebart, B.D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J.A., Hebert, M., Dey, A.K., Srinivasa, S.: Planning-based prediction for pedestrians. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3931–3936 (2009). IEEE

[26] Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision, pp. 201–214 (2012). Springer

[27] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971 (2016)

[28] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018). IEEE

[29] Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: European Conference on Computer Vision, pp. 683–700 (2020). Springer

[30] Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 206–213 (2017). IEEE

[31] Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.K.: Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6262–6271 (2019)

[32] Kooij, J.F., Flohr, F., Pool, E.A., Gavrila, D.M.: Context-based path prediction for targets with switching dynamics. International Journal of Computer Vision **127**(3), 239–262 (2019)

[33] Unhelkar, V.V., Pérez-D'Arpino, C., Stirling, L., Shah, J.A.: Human-robot co-navigation using anticipatory indicators of human walking motion. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6183–6190 (2015). IEEE

[34] Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)

[35] Pérez-Higueras, N., Otero, R., Caballero, F., Merino, L.: Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation. IEEE robotics and automation letters **8**(11), 7130–7137 (2023)

[36] Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., Lucic, M.: Revisiting the calibration of modern neural networks. Advances in neural information processing systems **34**, 15682–15694 (2021)

[37] Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)

[38] Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 694–699 (2002)

[39] Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 625–632 (2005)

[40] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems **30** (2017)

[41] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059 (2016). PMLR

[42] Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)

[43] Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly weather review **78**(1), 1–3 (1950)

[44] Nixon, J., Dusenberry, M., Jerfel, G., Nguyen, T., Liu, J., Zhang, L., Tran, D.: Measuring calibration in deep learning. arXiv preprint arXiv:1904.01685 (2019)

[45] Kumar, A., Liang, P.S., Ma, T.: Verified uncertainty calibration. Advances in neural information processing systems **32** (2019)

[46] Tibshirani, R.J., Foygel Barber, R., Candes, E., Ramdas, A.: Conformal prediction under covariate shift. Advances in neural information processing systems **32** (2019)

[47] Gibbs, I., Candes, E.: Adaptive conformal inference under distribution shift. Advances in Neural Information Processing Systems **34**, 1660–1672 (2021)

[48] Hébert-Johnson, U., Kim, M., Reingold, O., Rothblum, G.: Multicalibration: Calibration for the (computationally-identifiable) masses. In: International Conference on Machine Learning, pp. 1939–1948 (2018). PMLR

[49] Zhao, S., Kim, M., Sahoo, R., Ma, T., Ermon, S.: Calibrating predictions to decisions: A novel

approach to multi-class calibration. Advances in Neural Information Processing Systems **34**, 22313–22324 (2021)

[50] Fong, T., Thorpe, C., Baur, C.: Robot, asker of questions. Robotics and Autonomous systems **42**(3-4), 235–243 (2003)

[51] Knepper, R.A., Tellex, S., Li, A., Roy, N., Rus, D.: Recovering from failure by asking for help. Autonomous Robots **39**(3), 347–362 (2015)

[52] Cakmak, M., Thomaz, A.L.: Designing robot learners that ask good questions. In: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 17–24 (2012)

[53] Angelopoulos, A.N., Bates, S.: Conformal prediction: A gentle introduction. Foundations and Trends in Machine Learning **16**(4), 494–591 (2023)

[54] Chow, C.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory **16**(1), 41–46 (1970) https://doi.org/10.1109/TIT.1970.1054406

[55] Geifman, Y., El-Yaniv, R.: Selective classification for deep neural networks. Advances in neural information processing systems **30** (2017)

[56] Madras, D., Pitassi, T., Zemel, R.: Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer (2018). https://arxiv.org/abs/1711.06664

[57] Mozannar, H., Sontag, D.: Consistent Estimators for Learning to Defer to an Expert (2021). https://arxiv.org/abs/2006.01862

[58] Dragan, A.D., Lee, K.C., Srinivasa, S.S.: Legibility and predictability of robot motion. In: 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 301–308 (2013). IEEE

[59] Mavrogiannis, C.I., Thomason, W.B., Knepper, R.A.: Social momentum: A framework for legible navigation in dynamic multi-agent environments. In: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction. HRI '18, pp. 361–369. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3171221.3171255 . https://doi.org/10.1145/3171221.3171255

[60] Baraka, K., Veloso, M.M.: Mobile service robot state revealing through expressive lights: Formalism, design, and evaluation. International Journal of Social Robotics **10**(1), 65–92 (2018)

[61] Hart, J., Mirsky, R., Xiao, X., Tejeda, S., Mahajan, B., Goo, J., Baldauf, K., Owen, S., Stone, P.: Using human-inspired signals to disambiguate navigational intentions. In: Social Robotics: 12th International Conference, ICSR 2020, Golden, CO, USA, November 14–18, 2020, Proceedings, pp. 320–331. Springer, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-62056-1_27

[62] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)

[63] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: Design, architecture, and uses in the wild. Science robotics **7**(66), 6074 (2022)

[64] Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154 (2004). Ieee

[65] McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika **12**(2), 153–157 (1947)

[66] Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin **1**(6), 80–83 (1945)

[67] Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. Chapman and Hall/CRC (1994)

[68] Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian journal of statistics, 65–70 (1979)

[69] Wilson, E.B.: Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association **22**(158), 209–212 (1927)

[70] Park, S., Bastani, O., Weimer, J., Lee, I.: Calibrated prediction with covariate shift via unsupervised domain adaptation. In: International Conference on Artificial Intelligence and Statistics, pp. 3219–3229 (2020). PMLR

[71] Kuleshov, V., Ermon, S.: Estimating uncertainty online against an adversary. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)

[72] Bastani, O., Gupta, V., Jung, C., Noarov, G., Ramalingam, R., Roth, A.: Practical adversarial multivalid conformal prediction. Advances in neural information processing systems **35**, 29362–29373 (2022)

[73] Angelopoulos, A., Bates, S., Fisch, A., Lei, L., Schuster, T.: Conformal risk control. In: International Conference on Learning Representations, vol. 2024, pp. 55198–55218 (2024)

[74] Luo, R., Zhao, S., Kuck, J., Ivanovic, B., Savarese, S., Schmerling, E., Pavone, M.: Sample-efficient safety assurances using conformal prediction. The International Journal of Robotics Research **43**(9), 1409–1424 (2024)

[75] Sadigh, D., Sastry, S., Seshia, S.A., Dragan, A.D.: Planning for autonomous cars that leverage effects on human actions. In: Robotics: Science and Systems, vol. 2, pp. 1–9 (2016). Ann Arbor, MI, USA

[76] Schmerling, E., Leung, K., Vollprecht, W., Pavone, M.: Multimodal probabilistic model-based planning for human-robot interaction. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3399–3406 (2018). IEEE

[77] Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 261–268 (2009). IEEE

[78] Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum, vol. 26, pp. 655–664 (2007). Wiley Online Library

[79] Karnan, H., Nair, A., Xiao, X., Warnell, G., Pirk, S., Toshev, A., Hart, J., Biswas, J., Stone, P.: Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. IEEE Robotics and Automation Letters **7**(4), 11807–11814 (2022)

[80] Biswas, A., Wang, A., Silvera, G., Steinfeld, A., Admoni, H.: Socnavbench: A grounded simulation testing framework for evaluating social navigation. ACM Transactions on Human-Robot Interaction (THRI) **11**(3), 1–24 (2022)

[81] Francis, A., Pérez-d'Arpino, C., Li, C., Xia, F., Alahi, A., Alami, R., Bera, A., Biswas, A., Biswas, J., Chandra, R., *et al.*: Principles and guidelines for evaluating social robot navigation algorithms. ACM Transactions on Human-Robot Interaction **14**(2), 1–65 (2025)

[82] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 23–30 (2017). IEEE

[83] Rasouli, A., Tsotsos, J.K.: Autonomous vehicles that interact with pedestrians: A survey of theory and practice. IEEE transactions on intelligent transportation systems **21**(3), 900–918 (2019)

[84] Dey, D., Habibovic, A., Löcken, A., Wintersberger, P., Pfleging, B., Riener, A., Martens, M., Terken, J.: Taming the ehmi jungle: A classification taxonomy to guide, compare, and assess the design principles of automated vehicles' external human-machine interfaces. Transportation Research Interdisciplinary Perspectives **7**, 100174 (2020)

[85] Deb, S., Carruth, D.W., Sween, R., Strawderman, L., Garrison, T.M.: Efficacy of virtual reality in pedestrian safety research. Applied ergonomics **65**, 449–460 (2017)

[86] Lee, J.D., See, K.A.: Trust in automation: Designing for appropriate reliance. Human factors **46**(1), 50–80 (2004)

[87] Desai, M., Kaniarasu, P., Medvedev, M., Steinfeld, A., Yanco, H.: Impact of robot failures and feedback on real-time trust. In: 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 251–258 (2013). IEEE

[88] Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of machine learning research **11**(4) (2010)

[89] Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M.T., Weld, D.: Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–16 (2021)