

In View or Out of View? A Vision-Aware Filter that Infers and Reacts to the Human’s Field of View

Mengxue Fu, Ya-Chuan Hsu, Yinlong Dai, Erdem Bıyık*, and Dylan P. Losey*

Abstract—During human-robot interaction sight plays an important role. When the human sees a robot, they can observe the robot’s actions and better predict the robot’s future behaviors. Conversely, when the human cannot see the robot, they become uncertain about the robot’s intent and plans. In this paper we focus on the human’s *field of view*. Existing human-aware planners either ignore this field of view and assume the human can observe the entire scene, or intentionally design their interactions so the robot is always visible to the human. By contrast, we enable robots to infer what the human has and has not seen during interaction, i.e., the human’s field of view. We then apply this estimate as a *vision-aware filter* on the robot’s actions. Given a nominal robot policy and a model of the interaction, we roll out the policy over a short horizon to see how the robot’s state updates. We then reweight the actions based on how those actions impact the robot’s predicted visibility: higher weights are assigned to actions that bring the robot into (or away from) the human’s line of sight. In practice, our framework provides an online estimation and filtering process that repeatedly updates its understanding of the human’s field of view and then locally perturbs robot actions based on the human’s vision. We evaluate our vision-aware filter on robots that assist humans in collaborative cooking and coordinate with them in driving environments. Across simulations and a 12-participant user study, we find that our vision-aware filter changes both performance and preference: participants completed over three times as many subtasks with in-view assistance while cooking, and had fewer collisions with cars that purposely stayed out of their view while driving.

I. INTRODUCTION

We focus on robots that must coordinate with humans whose field of view (FOV) is limited [1]. Consider the example in Figure 1. Autonomous cars are sharing the road with a human driver, and — like any human driver — the user can only see out of the windshield. If the autonomous cars merge into lanes that are visible to the human, then the human must reason about these vehicles when driving (e.g., the human might slow down to pass the other cars more carefully). But if the autonomous cars recognize what the human can and cannot see, then they can intentionally move out of the human’s FOV and enable a more seamless driving experience for the user. More broadly, FOV affects driving decisions when merging, yielding, and overtaking.

Existing works on human-robot collaboration recognize that modeling the human and predicting their behaviors are essential for seamless interactions. But they typically make these predictions while *assuming that the environment is*

fully visible to the human [2]–[4]. This is problematic from both human and robot perspectives. When the robot fails to account for the human’s line of sight, that robot may mistakenly attribute the human’s decisions to the wrong intent, or incorrectly predict the human’s future actions. Accordingly, recent works have tried to address these issues by (a) assuming the human has a known, static field of view [5], [6] and by (b) treating entry into that FOV as the way to assist [7]. Here, two practical challenges emerge. First, the human’s FOV may change dynamically during the task as they gain or lose focus, and second, we do not always want the robot to remain within the human’s vision.

In this paper, we therefore explore how robots can dynamically infer and leverage the human’s FOV during coordination. Our insight is that:

When robots infer the human’s FOV, they can modify their actions to remain within (or stay outside of) that FOV without changing the robot’s underlying policy.

Applying this insight, we develop a *vision-aware filter* that can be connected to arbitrary policies for human-robot interaction. The filter only requires (a) a predictive model of the human and robot and (b) a designer-specified parameter that tunes whether the robot steers within or outside of the human’s sight. Given the human model, our approach leverages Bayesian inference to estimate the human’s FOV based on their recent state-action pairs. We then roll out future robot behaviors under the nominal policy, and evaluate how those behaviors overlap with the human’s estimated FOV. The policy actions are then reweighted based on whether the designer wants to steer into or out of sight. This intuitively results in a robot that locally modifies its behaviors based on the human’s vision. We emphasize that — even in coordination tasks — it is not necessarily beneficial to remain inside the human’s FOV, and we explore this hypothesis by testing our method while biasing actions both towards and away from the human’s sight.

Overall, we make the following contributions:

Formulating Dynamic FOV Inference. We enable robots to infer the human’s FOV based on the system states and human behaviors. We find that this FOV inference is tightly coupled to inferring the human’s intended task, and that methods that fail to jointly consider both vision and task may misattribute the human’s actions to the wrong cause.

Developing a Vision-Aware Filter. Equipped with our estimate of the human’s FOV and intent, we next develop a model-based filter that locally adjusts the robot’s actions to steer into or out of the human’s line of sight. This filter

*Equal advising. Research performed jointly by the Collaborative Robotics Lab (Collab), Dept. of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061, and LiraLab, Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA 90089. Corresponding author’s email: mishafu@usc.edu

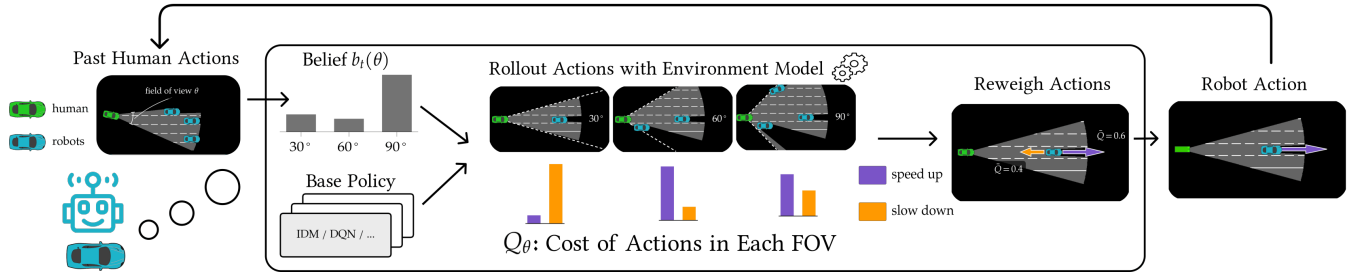


Fig. 1. Overview of our vision-aware filter. The human (green) drives with an unknown field of view θ while robots (blue) control the surrounding traffic. Based on the human’s actions the robots infer the human’s current field of view (FOV). More formally, the robots maintain a belief $b_t(\theta)$ over what the human can see. The robots then use this belief to modify the actions of a base policy. The robot’s base policy proposes candidate actions (e.g., speed up, slow down), and the robot leverages its model of the human and environment to roll out each candidate action and predict the time the robot will spend in the human’s vision under each FOV hypothesis (Q_θ). The scores are weighted by the belief ($\bar{Q}$). The robot then applies these scores to adjust the action distribution of the base policy, modifying how the robot acts to account for the human’s FOV. Designers can tune whether this steers the robot into or away from the human’s line of sight. Here our approach is used to minimize visibility, driving other robots out of the human’s FOV.

is invariant to the policy it is connected to, and therefore our approach generalizes across (and can be combined with) existing strategies for human-robot interaction.

Comparing In-View and Out-of-View Coordination. We conduct simulations and a user study where we compare robots that ignore and reason over the human’s FOV. Across a discrete collaborative cooking game and a continuous driving environment, we see that vision-aware robots are more effective and preferred partners. We also show that moving into the human’s FOV is not always beneficial, even when the robot is trying to help the human.

II. RELATED WORK

A. Assistive Robot Behavior and the Human’s View

Some existing work assumes that the robot’s actions are in the human’s view, and study how robots should move given that a human is watching. Within this context *legible motion* shapes the robot’s trajectory so that the human can quickly infer its intent [8], and information-revealing behaviors can convey facts the human is missing [9]–[11]. On the other hand, other assistive actions do not depend on being seen. For example, subtask allocation reduces the human’s workload to improve team efficiency [12], [13], and conservative trajectories steer the robot away from the human to maintain safety and build trust [14], [15]. However, prior works rarely consider how the human’s FOV may evolve during the task — meaning that these assistive actions fail to consider whether or not the human is currently watching.

B. Models of What the Human Perceives

Whether an action is seen depends on what the human perceives. Theory-of-mind and interactive optimization-based formulations maintain a belief over the human’s beliefs and goals [16], [17], and situation awareness models track which parts of the environment the human has registered [18]. More specifically, the human’s FOV has been considered at several levels of fidelity. Some works include the human’s heading direction or co-presence as a state feature [19] or model the FOV as a geometric cone that determines what the human can observe [7]. Occlusion-aware planning reasons about

what the human cannot see and adjusts the robot’s trajectory to compensate [20]. We note that these works generally treat the human’s FOV as a constant, and decide whether or not to enter the human’s FOV based on a task-specific heuristic.

C. Human Factors in Visible Assistance

Whether entering the human’s view is desirable depends on the person and the task. Task complexity changes how much information a person can take in [21], [22], and interruption studies show that poorly timed information is costly even when it is correct [23], [24]. Preference also varies with expertise. Novices benefit from slow and steady disclosure, while experts prefer to be left to their own plan [25]. The same is true from a trust perspective: keeping the robot within the human’s FOV for an extended period may be useful for building the user’s confidence in the system [26]. The value of visible assistance therefore varies with load, task, and person. When existing planners account for visibility at all, they treat it as something to increase, moving assistance into the human’s view whenever the task reward permits it, and the opposite direction is left unexplored. To the best of our knowledge, the literature lacks a controlled study that isolates this visibility variable by holding the task constant and evaluating in-view versus out-of-view assistance.

III. PROBLEM STATEMENT

We consider a two-agent human-robot setting. Let $s = (s_R, s_H, s_W)$ be the system state, where the components represent the robot, the human, and the environment, and s_H includes the human’s position and orientation. The robot fully observes state s at every timestep. The human has a limited FOV and only perceives $o = \phi(s, \theta)$. We parameterize the human’s FOV θ as the viewing angle, which follows the human’s current heading, so the visible region changes as the human moves and turns (e.g., the cone shown in Figure 1). The function ϕ retains the components of s inside the FOV defined by θ , and masks the rest. We assume θ is fixed over an interaction and unknown to the robot.

The human and robot interact in a shared environment. We assume that the human is performing some task with goal

g_H (e.g., completing a recipe or reaching a destination). The human takes actions a_H to complete this goal based on their history of observations, i.e., $a_H^t \sim \pi_H(\cdot \mid o^{0:t}, g_H)$. We note that the human’s behavior depends on θ via observations o . The robot has a model of π_H , and takes action $a_R \in \mathcal{A}_R$ which transitions the state according to the shared dynamics $s^{t+1} = f(s^t, a_R^t, a_H^t)$. Every robot action lands inside or outside the human’s view, and the question we study is how this FOV placement affects human-robot coordination.

IV. VISION-AWARE FILTER

Our method for inferring and accounting for the human’s FOV consists of two parts, as shown in Figure 1. First, the robot *infers* the human’s FOV θ during the interaction. Since the human acts based on what they have observed, their actions carry evidence about θ , and the robot can therefore leverage the human’s actions a_H to maintain and update a posterior over θ . Second, the robot *filters* its own actions to account for the human’s FOV. We perform this vision-aware filtering in a policy-invariant way by refining the action distribution of a given base policy to bias those actions towards (or away from) the human’s FOV. More formally, we add a cost term that the designer can tune to shape this filter. The sign of the cost determines whether the robot’s preferred coordination strategy is to steer into or out of the human’s view, which lets us treat the direction of steering as a controlled variable and evaluate both directions in Sections V and VI. We note that our method relies on having a model of system, including both the human’s policy π_H and the environment dynamics f .

A. Human Model

The robot first needs to build an internal human model in order to (a) infer the human’s FOV from observed human behavior and (b) predict the human’s next actions using the FOV inference. The human’s true policy is unknown to the robot. Instead, the robot has an approximate human model, represented as a conditional distribution over human actions:

$$\hat{\pi}_H(a_H^t \mid h^t, \theta) := \hat{P}(a_H^t \mid s^{0:t}, a_H^{0:t-1}, a_R^{0:t-1}, \theta) \quad (1)$$

where $\hat{P}$ approximates the true human policy π_H , $s^{0:t}$ is the state history observed by the robot, $a_H^{0:t}$ denotes the human actions observed up to timestep t , and $h^t = (s^{0:t}, a_H^{0:t-1}, a_R^{0:t-1})$ denotes the interaction history. Intuitively, the human model describes how likely the human is to choose action a_H at time t based on the interaction history and what they have observed. By connecting observed human actions to their hidden FOV θ , this model enables the robot to infer the human’s FOV as shown in Section IV-B. The human model may be learned (for example using a neural network trained on recent human behaviors). In our implementation, we approximate the human’s policy through a series of predefined decision rules.

B. Inferring Human’s Field of View

We use the approximate human model $\hat{\pi}_H$ to infer the human’s FOV from observed actions. The robot maintains a

posterior belief:

$$b_t(\theta) = P(\theta \mid h^t, a_H^t) \quad (2)$$

This belief is updated recursively according to how likely the observed human action is under each candidate θ , as predicted by the human model. Using Bayes’ theorem:

$$b_t(\theta) \propto \hat{\pi}_H(a_H^t \mid h^t, \theta) b_{t-1}(\theta) \quad (3)$$

The likelihood measures how well a candidate θ explains the human’s observed action a_H^t ; that is, it captures what the robot thinks the human can see based on how they act. Utilizing this belief, we next construct an FOV filter that can be applied to any robot base policy to steer its actions for better coordination with the human.

C. Vision-Aware Policy Refinement

We next use the posterior belief $b_t(\theta)$ to refine the action distribution proposed by a nominal robot policy.

Base Policy. Let π_R be a base robot policy that maps the observation history to a distribution over robot actions:

$$a_R^t \sim \pi_R(\cdot \mid s^{0:t}, a_H^{0:t-1}) \quad (4)$$

Note that we impose no restrictions on the form of this nominal robot policy. It may or may not use the human model $\hat{\pi}_H$, and it does not need to directly account for the inferred human’s FOV $b_t(\theta)$. We treat the base policy as a black box and do not assume it to be optimal. Its action distribution may contain errors arising from an inaccurate human model, imperfect task reasoning, or other modeling approximations. However, we do assume that any safety requirements of the domain are enforced at this level, namely, the base policy assigns zero probability to unsafe actions.

Vision-Aware Filter. We modify the outputs of the base policy using our vision-aware filter. Given the base action distribution at timestep t and the posterior belief $b_t(\theta)$, let $Q_\theta(s^t, a_R)$ denote the predicted cost of taking action a_R at s^t when the human has FOV θ . We will describe this cost Q further in the following section. For now, we aggregate the costs across θ hypotheses according to:

$$\bar{Q}_t(s^t, a_R) = \sum_{\theta \in \Theta} b_t(\theta) Q_\theta(s^t, a_R). \quad (5)$$

The FOV-aware action is then selected as

$$a_R^t \in \arg \min_{a_R \in \mathcal{A}_R} \left[-\log \pi_R(a_R \mid s^{0:t}, a_H^{0:t-1}) + \lambda \bar{Q}_t(s^t, a_R) \right] \quad (6)$$

where $\mathcal{A}_R$ is the robot’s action space. The first term favors actions assigned high probability by the base policy, while the second term incorporates the FOV-conditioned rollout cost. The coefficient λ controls the influence of the FOV filter and hence the robot’s behavior: when $\lambda = 0$, the method reduces to the base policy, whereas increasing the magnitude of λ gives greater weight to the FOV-conditioned evaluation.

FOV-Conditioned Cost. The choice of the FOV-conditioned cost Q_θ is open-ended. To preserve generality, its con-

struction should not depend on domain- or task-specific knowledge. In this work, we use a visibility-based cost. We first assume access to the transition model f that predicts the next state given the current state and both agents' actions (see Section III). Let $v(s, \theta) \in \{0, 1\}$ indicate whether the robot lies inside the visible region induced by θ in state s . For each candidate action a_R , we use f to generate an H -step rollout, with the robot executing a_R initially and following its base policy thereafter, while the human acts according to the approximated human model $\hat{\pi}_H$. The cost is the expected fraction of the rollout the robot spends in the human's view:

$$Q_\theta(s^t, a_R) = \frac{1}{H} \sum_{k=1}^H \mathbb{E}[v(s^{t+k}, \theta)]. \quad (7)$$

With this choice, $\lambda > 0$ in Equation (6) penalizes visibility and drives the robot out of the human's FOV, $\lambda < 0$ rewards visibility and drives the robot into the human's FOV, and $\lambda = 0$ recovers the base policy.

Task and FOV Trade-Off. Since the FOV filter does not use domain- or task-specific knowledge, its adjustments to the robot's actions may conflict with the base policy's task objective. While the filter can potentially reweight actions to favor lower visibility costs and sacrifice task performance, we impose three constraints that protect the base policy's performance.

First, the filter never selects an action to which the base policy assigns zero probability. In practice, the filter reweights the base policy's top- k actions. Second, each candidate must remain within a bounded margin of the base policy's own choice. More specifically, when the base policy plans a sequence of actions toward its current objective, the length of that sequence serves as a cost d , and a candidate is discarded if its own cost exceeds $(1 + c) \cdot d$, where $c \geq 0$ is a hyperparameter. For base policies that emit a single action rather than a distribution, candidates are bounded offsets on that action, penalized in proportion to their deviation. Finally, if no candidate satisfies these constraints, or if the posterior $b_t(\theta)$ has not yet concentrated, the filter returns the base policy's action unchanged. The filter thus intervenes only within the base policy's own tolerance, in the spirit of safety filters for safety-critical control [27].

V. SIMULATION EXPERIMENTS

With our vision-aware filter introduced, we next want to test whether reweighting an FOV-agnostic base policy by visibility alone can improve coordination between agents. We start with simulated environments where we can directly control the human model and test a variety of base robot policies. We consider two types of collaborative domains (overcooked and driving), and explore whether steering towards or away from the human's FOV affects coordination differently in these different domains.

A. Experimental Domains

We evaluate our vision-aware filter in two domains that require different types of coordination, permit diverse, dis-

tinguishable FOV configurations, and span discrete and continuous state and action spaces.

- **Overcooked** (Figure 2) is a collaborative kitchen environment [7] built on top of Overcooked-AI by [2]. A human and robot must collaborate and deliver as many dishes as possible within a limited number of timesteps. The human sees only within a cone centered on its heading, and walls further occlude its line of sight. The four layouts confine the agents to separate sections of the kitchen, enforcing coordination by requiring agents to pass ingredients to each other to complete orders.
- **Highway** (Figure 1) is a continuous, safety-critical driving environment [28] where the human operates an ego vehicle through dense traffic controlled by robot policies. We customize the environment to use an 8-lane road with 80 vehicles at $1.5\times$ the density of the environment's defaults on the "highway-v0" scene, making interactions with surrounding traffic more frequent. The human's goal is to travel a fixed distance as quickly as possible while avoiding collision with surrounding vehicles. The robots do not share the human's task; they influence the human only by shaping how the other vehicles share the road.

B. Simulated Humans

In both domains the simulated human has a limited FOV. This human acts greedily to make immediate progress towards the task's objective following the human model in Section IV-A. We next introduce the simulated human's observation and action spaces in each domain.

In Overcooked the simulated human's observation space includes ingredient locations currently within its FOV and those retained in a memory buffer from the past 40 timesteps. Visible locations are updated to their observed values, while remembered locations expire 40 timesteps after they were last observed. Memory is essential for a plausible partner, since a human who instantly forgets everything outside its FOV would be ineffective, while one who never forgets would be unrealistic. Following prior work [29], we define a human's high-level action space as a set of subtasks $\tau \in \mathbb{T}$, allowing the human to plan at the subtask level. Each subtask has preconditions under which it is available. For example, "cook food" requires that uncooked food is present and the grill is empty. At each step, the human selects the highest-priority subtask available according to memory, and plans to execute it with A*, treating locations has it never observed as unwalkable. If it last saw the robot carrying an item it needs, it goes to look for where that item was placed.

In Highway the human's observation space consists of information about surrounding vehicles within their FOV and a sensing distance of 40 m. The human's action space consists of speed adjustments between 1-35 m/s, as well as moving to adjacent lanes. The simulated human changes lanes when they believes a faster lane is free, unless the merging action leads to a conflict with a visible vehicle.

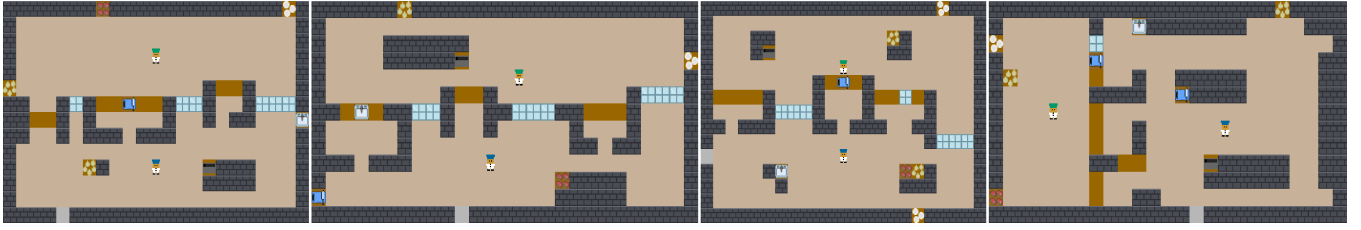


Fig. 2. Overcooked environment. Here we show the four kitchen layouts used in our simulation experiments. We visualize the start state for each option, which includes items such as a back bar, banquet pass, chefs table, and divider. Brick tiles are walls, which block the human’s line of sight; pane tiles are glass, which is impassable but see-through. Each kitchen contains meat, onion, and plate dispensers, a grill, a chopping board, a sink, and a serving window. The green-hat agent is the robot and the blue-hat agent is the human.

C. Base Policies

For each domain we implement a distinct set of base robot policies. Viewed together, we test five policies ranging from rule-based to learned control, and from no partner reasoning to full Bayesian inference. None of these nominal policies reason over the human’s FOV. We test whether our filter is able to synergize with these different base policies and improve team coordination. In **Overcooked**, we consider:

- *Greedy* [2], which selects subtasks by task progress and ignores the presence of a partner;
- *Solo*, which adjusts the robot’s behavior based on the human’s current position to minimize interference;
- *Bayes* [29], which collaborates and infers the human’s current subtask while assuming that the human observes the full system state.

Each policy outputs its subtask ranking to the filter as a discrete action distribution. In **Highway**, we consider:

- *IDM*, where each background vehicle follows the Intelligent Driver Model [30] for car-following with MOBIL lane changes [31];
- *DQN*, a pretrained deep Q-network policy that selects among discrete acceleration and lane-change actions.

D. Filter Design

In both domains the vision-aware filter follows Section IV. The filter is given candidate actions from the base policy, forward simulates each candidate action under the human and environment models, and then scores that action by its expected visibility under the estimated human FOV. Finally, the filter chooses the candidate (or locally modifies the candidate) with the lowest cost according to Equation (6).

In Overcooked, candidates are the base policy’s three highest-ranked subtasks; rollouts that are used to compute the FOV-conditioned cost extend $H = 12$ timesteps, roughly the duration of one subtask. In Highway, the filter runs on every neighboring vehicle within 45m of the human, and adjusts only each vehicle’s acceleration. Lane-change decisions are always left to the base policy. Candidates are bounded offsets on the nominal acceleration. To score a candidate, the filter rolls the environment forward for 3s under that acceleration, with neighboring vehicles at constant velocity.

E. Evaluation

In Overcooked we evaluate all combinations of the four layouts (see Figure 2), four FOVs $\theta \in \{30^\circ, 60^\circ, 90^\circ, 360^\circ\}$,

and three base policies, with 20 seeds per combination. Each episode lasts up to 400 timesteps and ends once three dishes have been delivered. In Highway we evaluate all combinations of the two base policies and three FOVs, $\theta \in \{30^\circ, 60^\circ, 90^\circ\}$, with 20 seeds per combination. Each session lasts 60 seconds of simulated driving; a collision restarts the traffic while elapsed time and distance carry over. We report mean dishes delivered (task success) in Overcooked and collisions (safety), distance covered (traffic efficiency), and the human’s completed lane changes (maneuvering induced by the traffic) in Highway. For both domains we also report the filtered policy’s total computation per timestep and its overhead relative to the base policy, which together determine whether this vision-aware filter can be applied in real-time and support the user study in Section VI.

F. Results

We compare three versions of our filter: no adjustment to the base policy ($\lambda = 0$), steering the robot away from the human’s inferred FOV ($\lambda > 0$), and steering the robot into the human’s FOV ($\lambda < 0$). In our results we refer to these conditions as **Base**, **Out**, and **In**, respectively.

Overcooked. Table I reports mean dishes delivered per condition. The *in-view filter* ($\lambda < 0$) improves over every base policy at every FOV, increasing mean dishes per episode (averaged over layouts and FOVs) from 1.16 to 1.40 for greedy, 1.14 to 1.39 for solo, and 0.69 to 1.06 for Bayes. Note that even the omnidirectional $\theta = 360^\circ$ setting falls short of full observability, since the layouts’ walls still occlude the view, leaving room for the filter to improve.

The gain is largest for Bayes, whose assumption that the human observes the full state is most wrong at narrow FOVs, and which the filter helps to correct. The benefit is steepest precisely where sight is narrowest: at $\theta = 30^\circ$ the in-view filter increases deliveries by over 60% relative to each base policy, and the margin narrows as the FOV widens. In general, the less the human can see, the more valuable it is for the robot to work where the human is looking.

Highway. Table II reports collisions and traveled distance per session. Results show that staying *out of the human’s view* ($\lambda > 0$) is the better approach in this domain, helping under both base policies at all but the narrowest FOV. Under DQN traffic, the out-of-view filter reduces collisions at every FOV, from 4.03 to 2.62 per session on average. Under IDM

TABLE I

OVERCOOKED SIMULATION: MEAN DISHES DELIVERED PER 400-TIMESTEP EPISODE (MAXIMUM 3 DELIVERABLE ORDERS PER EPISODE). RESULTS ARE AVERAGED OVER FOUR LAYOUTS, THREE BASE POLICIES, AND 20 SEEDS. EACH ROW CORRESPONDS TO A DIFFERENT HUMAN FOV θ . COLUMNS CORRESPOND TO THE VISION-AWARE FILTER: UNFILTERED (BASE POLICY), OUT-OF-VIEW FILTER (OUT, $\lambda > 0$), IN-VIEW FILTER (IN, $\lambda < 0$); BEST IS IN BOLD.

θ	Greedy			Solo			Bayes		
	Base	Out	In	Base	Out	In	Base	Out	In
30°	0.43	0.08	0.70	0.40	0.08	0.65	0.20	0.12	0.50
60°	1.27	1.25	1.48	1.20	1.25	1.45	0.80	1.05	1.10
90°	1.25	1.30	1.50	1.42	1.28	1.50	0.73	1.02	1.12
360°	1.70	1.70	1.93	1.55	1.70	1.95	1.02	1.45	1.50

TABLE II

HIGHWAY SIMULATION: MEAN COLLISIONS | DISTANCE (M) PER 60 s SESSION OVER 20 PAIRED SEEDS. EACH ROW CORRESPONDS TO A DIFFERENT HUMAN FIELD OF VIEW θ . COLUMNS CORRESPOND TO THE DIFFERENT TRAFFIC POLICIES AND VISION-AWARE FILTERS: UNFILTERED (BASE), OUT-OF-VIEW FILTER (OUT), IN-VIEW FILTER (IN); BEST IN BOLD (FEWER COLLISIONS; TIES BROKEN BY MORE DISTANCE).

θ	IDM						DQN					
	Base	Out	In	Base	Out	In	Base	Out	In	Base	Out	In
30°	3.50	1778	4.20	1900	3.90	1711	5.15	1437	2.85	1461	4.10	1284
60°	0.75	1686	0.40	1907	1.50	1667	3.55	1268	2.50	1232	3.05	1267
90°	0.50	1702	0.35	1902	1.00	1699	3.40	1237	2.50	1248	3.45	1372

traffic, it reduces collisions at $\theta = 60^\circ$ and 90° while adding roughly 180 m of distance covered per session. We attribute this to how the filter reshapes traffic, namely, vehicles ahead of the human accelerate out of the FOV, traffic spreads out, and the human brakes less.

The human’s maneuvering shows the same effect. Under IDM traffic, the in-view filter pulls vehicles into the FOV and raises the human’s lane changes from 19.9 to 27.8 per session, alongside more collisions. Under the DQN traffic, the out-of-view filter reduces the number of lane changes from 13.3 to 10.0 as vehicles clear out of the human’s path. Neither direction helps under IDM at $\theta = 30^\circ$. We attribute this to the FOV itself rather than to FOV inference: with such a narrow view, the human drives more recklessly, colliding mostly with vehicles it never gets to see, so rearranging which vehicles are visible cannot fully overcome the natural FOV limitation.

Computation Time. The filter fits both domains’ real-time budgets. In Overcooked the filter computes its action in 91 ms on average, against under 1 ms for the underlying Bayes policy alone. Although the filter adds substantial overhead relative to the base policy, the total remains well within the real-time budget of the game, which is around 3 Hz. This added time is caused by rolling out candidate actions and considering their future effects on visibility: the time delay of the filter is correlated with the simulation time and depth

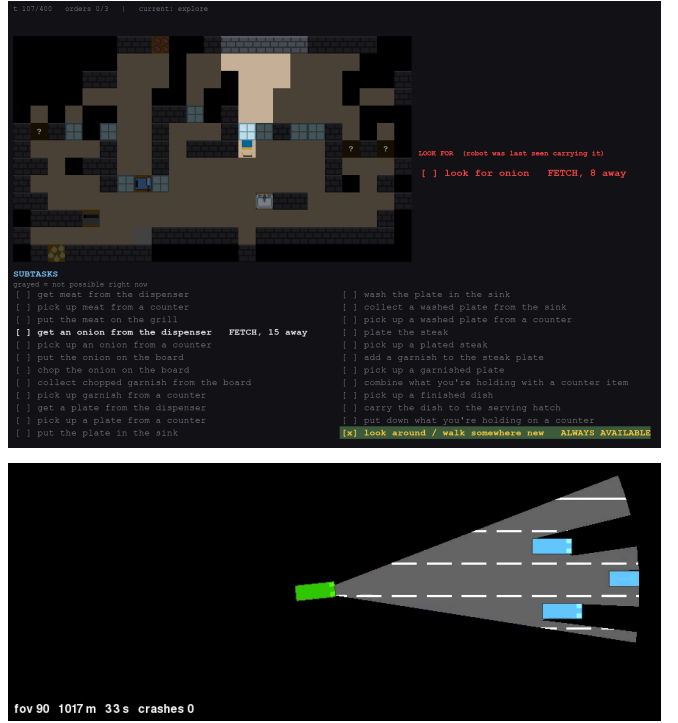


Fig. 3. The user study interfaces. Top: Overcooked, where the participant sees only the region inside their FOV and selects among currently available high-level subtasks from the menu. Bottom: Highway, where the participant steers the ego vehicle and sees only vehicles inside their FOV.

H. In Highway, computing all background vehicles’ actions takes 21 ms with the filter compared to 17.5 ms with just the base policy under the IDM model, and 10–14 ms for the DQN model. The total is still under the real-time budget for Highway, which is around 15 Hz. We therefore run the same filters without any changes in the user study.

VI. USER STUDY

Our simulations enabled us to test the vision-aware filter in controlled conditions. Next, we explore how this filter helps robots collaborate with *actual humans*. Our user study setup follows the design of our simulations, again using the **Overcooked** and **Highway** domains where a robot and human must collaborate. We apply visual masks to these environments to regulate what the human can and cannot perceive (see Figure 3). The study tests whether the effects observed in simulation hold with real users, and additionally examines how users perceive the differences between the base policy and the in-view and out-of-view filters.

Participants. We conducted the study with 12 participants (3 female, 9 male; ages 20–63, mean = 27.9). Nine of our participants had experience playing the commercial game Overcooked. One participant played driving simulators regularly, and seven participants had used driving simulators occasionally. Four participants had prior research experience with the Overcooked-AI environment [2], and three with the Highway environment [28]. All participants provided informed written consent following the IRB protocol at the University of Southern California.

Experimental Setup. Each participant interacted with three types of robot agents: the base policy alone (no filter, $\lambda = 0$), the base policy with the out-of-view filter ($\lambda > 0$), and the base policy with the in-view filter ($\lambda < 0$). The order of robot policies was counterbalanced across participants, and conditions were unnamed: participants knew only that they would interact with three different robot policies. The FOV was assigned between users ($\theta = 30^\circ$ for 6 participants, 60° for 6) and held fixed for that user in both domains. These two θ values were chosen based on preliminary experiments; we are most interested in how the filter operates in low-visibility settings. Each participant completed the overcooked and highway domains in one session, beginning each domain with a familiarization round under a near-unrestricted view that was excluded from analysis.

In Overcooked the participant played one 400-timestep episode with each robot policy using *Bayes* as the base policy. As shown in Figure 3 (top), the participant saw the kitchen only inside their FOV and acted at the subtask level. A menu listed the high-level subtasks, and a row was clickable only when that subtask was currently available given what the participant could observe. Clicking a row made their chef carry that task out automatically, and the SPACE bar canceled the current subtask. In Highway, the participant drove the ego vehicle with the arrow keys through IDM traffic for 60s (Figure 3, bottom); a collision reset the surrounding traffic while elapsed time and distance carried over. After each episode, participants rated the robot on 7-point Likert scales. After completing each domain, participants ranked the three robot policies and optionally described any differences they noticed.

Objective Results. We measured the human’s subtask completions and dishes delivered in Overcooked, and collisions and distance covered in Highway. Our results are summarized in Table III. In Overcooked — with real humans in the loop — the *in-view* filter’s advantage from simulation not only holds but widens. Participants completed 11.75 subtasks on average with the in-view robot, over three times the 3.17 they completed with the base policy ($p < .001$, Wilcoxon signed-rank test). Six of the seven dishes delivered across the entire study came from the in-view condition. This affirms the simulation results, showing that acting within the human’s FOV keeps their knowledge of the kitchen current: in the study, the subtask of looking for an item could only be selected once the participant observed the robot carrying that item, so a robot that passed through where the participant was looking directly expanded what they could act on, making the overall task more effective.

In Highway, as in simulation, participants performed best when the robots stayed *out* of their FOV. Participants drove further (1855 versus 1779 m on average) with fewer collisions (1.17 versus 1.25 per session) as compared to the base policy. When the robot controlled the vehicles to stay out of the human’s line of sight, these vehicles often sped up (moving away from the human), allowing the human to drive unimpeded. By contrast, the in-view filter drew vehicles into

TABLE III
USER STUDY OBJECTIVE RESULTS ($N = 12$): MEAN PER PARTICIPANT WITH EACH ROBOT POLICY, EXCEPT DISHES DELIVERED, WHICH ARE TOTALED ACROSS ALL PARTICIPANTS. COLUMNS CORRESPOND TO THE BASE POLICY (BASE), OUT-OF-VIEW FILTER (OUT, $\lambda > 0$), AND IN-VIEW FILTER (IN, $\lambda < 0$); BEST IS IN BOLD.

Domain	Metric	Base	Out	In
Overcooked	Subtasks completed	3.17	5.67	11.75
	Dishes delivered (total)	0	1	6
Highway	Collisions per session	1.25	1.17	1.42
	Distance (m)	1779	1855	1733

the participant’s limited sight, and then the participants had to navigate around these blocking vehicles. Hence, filtering the robot’s actions to remain within the human’s vision resulted in the most collisions and the least distance covered (1.42 collisions, 1733 m). Together, the two domains show that a robot which infers the human’s FOV can steer its actions either into or out of view, potentially improving the team’s coordination. But it not as simple as always trying to maximize the robot’s visibility — how the robot should react to the human’s FOV depends on the current domain.

Subjective Results. After each interaction participants rated their agreement with six statements on 7-point Likert scales, adapted from established fluency and trust questionnaires in human-robot collaboration [32], [33]: “the robot was helpful” (*helpfulness*), “I understood what the robot was doing and why” (*understanding*), “I could predict what the robot would do next” (*predictability*), “the robot behaved the way a human would” (*human-likeness*), “I trusted the robot to do the right thing” (*trust*).

Figure 4 reports the user rating for these scales, and Figure 5 reports the participants’ forced rankings for each filter type. In Overcooked the participants’ preferences matched their performance: the in-view robot was rated above the base policy on all six constructs ($p < .05$ for all), and was ranked best by 10 of the 12 participants. The other two participants ranked the out-of-view filter the best, potentially representing users who prefer completing tasks alone with as little interaction as possible with the robot. Participants not only benefited from visible assistance but also recognized it. One participant wrote that the in-view robot “*passed the food to the other side of the counter [...] as if it was intentionally showing the player it is carrying the food.*”

In Highway the out-of-view robot policy was rated as significantly more helpful than the base policy ($p = .016$), with trust and understanding trending in the same direction ($p = .094$ for both). A total of 7 of 12 participants selected the out-of-view policy as their preference. By contrast, the in-view robot was rated lowest on *Not in the way*: directing traffic into a limited view was not received as assistance but rather as interference. Participants also provided descriptions, including one describing the out-of-view robots as “*actively mak[ing] way for the ego*” while the in-view robot “*somehow seemed actively in [ego vehicle’s] way.*” Across both domains

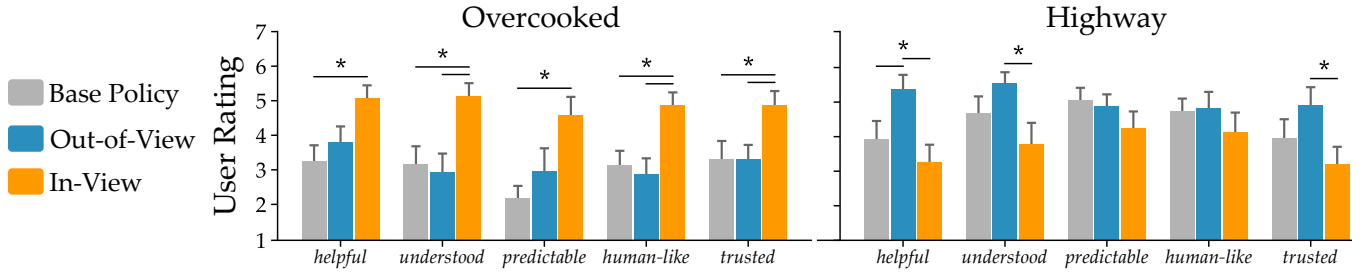


Fig. 4. User study subjective ratings. Participants responded to a 7-point Likert scale survey after working with each filter type. Bars show standard error, and an $\star$ denotes statistical significance ($p < .05$). We find that people preferred robots that used our vision-aware filter (the colored bars); the correct way to leverage that filter (in-view or out-of-view) depended on the domain.

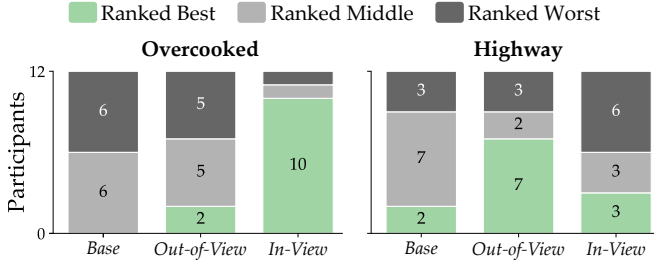


Fig. 5. Forced choice comparison. Our 12 participants ranked the filter type from best to worst in both experimental domains. In overcooked participants preferred *in-view*, while in highway participants preferred *out-of-view*.

participants were not told which types of filters they were working with, yet their rankings and descriptions recovered the distinction the filter implements.

VII. CONCLUSION

In this paper we introduced a vision-aware filter that accounts for what the human can see during human-robot interaction. Our filter had two main components: (a) dynamically updating its estimate of the human’s current FOV, and then (b) modifying the actions of a base policy to steer the robot towards — or away from — the human’s estimated FOV. We achieved this action modification in a policy-invariant manner by reweighting the action distribution according to the predicted visibility of each candidate action. Although our approach assumes access to a dynamics model and an estimated human model, the filter *does not require* any task knowledge. Instead, we provide a parameter that designers can tune to adjust whether our filter increases or decreases the robot’s visibility. During controlled simulations and a user study we found that inferring and accounting for the human’s FOV can significantly impact human-robot collaboration. The right way to steer the robot, however, appears to depend on the task and domain. In our collaborative cooking environment, maximizing the robot’s visibility resulted in more efficient teamwork. Conversely, in our shared driving environment, keeping other vehicles out of the human’s line of sight provided a clearer road and enabled faster motions.

Limitations. While our current filter assumes the field of view is constant within an interaction, real attention shifts with focus and task, so extending the posterior to track a moving, task-dependent view is a natural next step. More

broadly, the benefits of a purely visibility-based signal suggest that visibility could be folded into end-to-end learning, training robot policies that can decide for themselves whether and when to be seen. Our results strongly indicate that when robots make this choice well, they become better partners when coordinating with humans.

REFERENCES

- [1] I. P. Howard and B. J. Rogers, *Binocular vision and stereopsis*. Oxford University Press, 1995.
- [2] M. Carroll *et al.*, “On the utility of learning about humans for human-AI coordination,” in *NeurIPS*, 2019.
- [3] A. Xie *et al.*, “Learning latent representations to influence multi-agent interaction,” in *Conference on Robot Learning*, 2021.
- [4] S. Parekh and D. P. Losey, “Learning latent representations to co-adapt to humans,” *Autonomous Robots*, 2023.
- [5] C. L. Baker *et al.*, “Rational quantitative attribution of beliefs, desires and percepts in human mentalizing,” *Nature Human Behaviour*, 2017.
- [6] J. Kolb and K. M. Feigh, “Inferring belief states in partially-observable human-robot teams,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.
- [7] Y.-C. Hsu, M. Defranco, R. Patel, and S. Nikolaidis, “Integrating field of view in human-aware collaborative planning,” in *IEEE International Conference on Robotics and Automation*, 2025.
- [8] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, “Legibility and predictability of robot motion,” in *ACM/IEEE International Conference on Human-Robot Interaction*, 2013.
- [9] M. Shvo, T. Q. Klassen, and S. A. McIlraith, “Resolving misconceptions about the plans of agents via theory of mind,” in *International Conference on Automated Planning and Scheduling*, 2022.
- [10] A. Hiranaka, Y.-C. Hsu, S. Nikolaidis, E. Bıyık, and D. Seita, “Fix the mind, not the move: Interpretable AI assistance via knowledge-gap localization,” in *International Conference on Machine Learning*, 2026.
- [11] S. Habibian, A. Alvarez Valdivia, L. H. Blumenschein, and D. P. Losey, “A survey of communicating robot learning during human-robot interaction,” *The International Journal of Robotics Research*, 2025.
- [12] G. Hoffman and C. Breazeal, “Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team,” in *ACM/IEEE International Conference on Human-Robot Interaction*, 2007.
- [13] S. Habibian and D. P. Losey, “Encouraging human interaction with robot teams: Legible and fair subtask allocations,” *IEEE Robotics and Automation Letters*, 2022.
- [14] K. Dautenhahn *et al.*, “How may I serve you? A robot companion approaching a seated person in a helping context,” in *ACM/IEEE International Conference on Human-Robot Interaction*, 2006.
- [15] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, “Trust-aware decision making for human-robot collaboration: Model learning and planning,” *ACM Transactions on Human-Robot Interaction*, 2020.
- [16] J. Koster-Hale and R. Saxe, “Theory of mind: A neural prediction problem,” *Neuron*, 2013.
- [17] P. J. Gmytrasiewicz and P. Doshi, “A framework for sequential planning in multi-agent settings,” *Journal of Artificial Intelligence Research*, 2005.

- [18] H. Senaratne *et al.*, “A framework for dynamic situational awareness in human-robot teams: An interview study,” *ACM Transactions on Human-Robot Interaction*, 2025.
- [19] G. Sartor and R. Alami, “An epistemic human-aware task planner integrating agents expectations,” in *ICAPS Workshop on Human-Aware and Explainable Planning*, 2026.
- [20] Z. Zhang and J. F. Fisac, “Safe occlusion-aware autonomous driving via game-theoretic active perception,” in *Robotics: Science and Systems*, 2021.
- [21] J. Sweller, “Cognitive load during problem solving: Effects on learning,” *Cognitive Science*, 1988.
- [22] C. D. Wickens, “Multiple resources and mental workload,” *Human Factors*, 2008.
- [23] F. Cini, T. Banfi, G. Ciuti, L. Craighero, and M. Controzzi, “The relevance of signal timing in human-robot collaborative manipulation,” *Science Robotics*, 2021.
- [24] Y.-C. Hsu, J. DeCastro, A. Silva, and G. Rosman, “Timing the message: Language-based notifications for time-critical assistive settings,” *arXiv preprint arXiv:2509.07438*, 2025.
- [25] P. Carreno-Medrano, S. L. Smith, and D. Kulić, “Joint estimation of expertise and reward preferences from human demonstrations,” *IEEE Transactions on Robotics*, 2023.
- [26] G. Campagna and M. Rehm, “A systematic review of trust assessments in human-robot interaction,” *ACM Transactions on Human-Robot Interaction*, 2025.
- [27] K.-C. Hsu, H. Hu, and J. F. Fisac, “The safety filter: A unified view of safety-critical control in autonomous systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, 2024.
- [28] E. Leurent, “An environment for autonomous driving decision-making,” <https://github.com/Farama-Foundation/HighwayEnv>, 2018.
- [29] S. A. Wu, R. E. Wang, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, “Too many cooks: Bayesian inference for coordinating multi-agent collaboration,” *Topics in Cognitive Science*, 2021.
- [30] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Physical Review E*, 2000.
- [31] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model MOBIL for car-following models,” *Transportation Research Record*, 2007.
- [32] G. Hoffman, “Evaluating fluency in human-robot collaboration,” *IEEE Transactions on Human-Machine Systems*, 2019.
- [33] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, “Foundations for an empirically determined scale of trust in automated systems,” *International Journal of Cognitive Ergonomics*, 2000.