SHARP: Shielding-Aware Robust Planning for Safe and Efficient Human-Robot Interaction

Haimin Hu[1], Kensuke Nakamura[2], and Jaime F. Fisac[1]

Abstract—Jointly achieving safety and efficiency in human-robot interaction settings is a challenging problem, as the robot's planning objectives may be at odds with the human's own intent and expectations. Recent approaches ensure safe robot operation in uncertain environments through a supervisory control scheme, sometimes called "shielding", which overrides the robot's nominal plan with a safety fallback strategy when a safety-critical event is imminent. These reactive "last-resort" strategies (typically in the form of aggressive emergency maneuvers) focus on preserving safety without efficiency considerations; when the nominal planner is unaware of possible safety overrides, shielding can be activated more frequently than necessary, leading to degraded performance. In this work, we propose a new shielding-based planning approach that allows the robot to plan efficiently by explicitly accounting for possible future shielding events. Leveraging recent work on Bayesian human motion prediction, the resulting robot policy proactively balances nominal performance with the risk of high-cost emergency maneuvers triggered by low-probability human behaviors. We formalize Shielding-Aware Robust Planning (SHARP) as a stochastic optimal control problem and propose a computationally efficient framework for finding tractable approximate solutions at runtime. Our method outperforms the shielding-agnostic motion planning baseline (equipped with the same human intent inference scheme) on simulated driving examples with human trajectories taken from the recently released Waymo Open Motion Dataset.

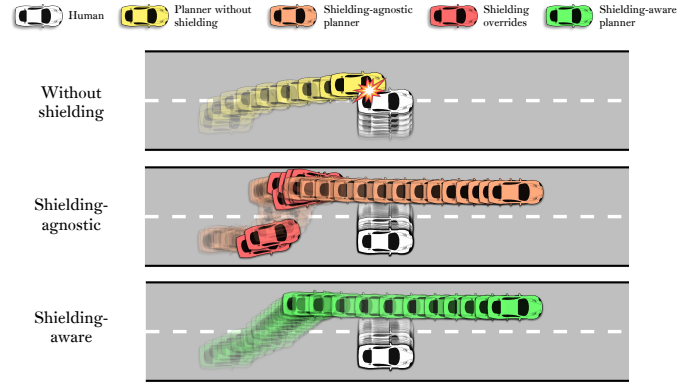*Index Terms*—Human-aware motion planning, safety in HRI, planning under uncertainty.

Fig. 1: An autonomous car seeks to overtake a vehicle driven by a distracted human (longitudinal positions are shown in relative coordinates). *Top:* A planner without formal safety guarantees incurs in a collision. *Middle:* A shielding-agnostic planner triggers safety overrides unnecessarily often, maintaining safety at the cost of performance and comfort. *Bottom:* Our proposed planner reasons about future shielding events and avoids relying on safety overrides if possible, which significantly improves the resulting performance.

I. INTRODUCTION

IN recent years, much effort has been devoted to developing robotic systems that can coexist and interact with humans. Indeed, in order to serve people in daily life, autonomous systems must competently predict and seamlessly adapt to human behaviour. Examples include autonomous driving [1], [2], indoor aerial robots [3] and robotic arms [4]. These applications are safety-critical, since inappropriate robot behaviours can pose significant danger to humans. Therefore, it is crucial to develop motion planning algorithms for human-robot interaction that not only yield high performance but also guarantee safety at all times.

In typical human-robot interaction scenarios, since the robot's safety and performance are naturally coupled with the human's movements, the robot must be able to make real-time inferences about the human's future motion during planning.

Predicting human motion while planning the robot's trajectory can be generally cast as a partially-observable stochastic game [5]. In [2], the authors modeled the interaction between the human and the robot as a dynamic game that allows for real-time trajectory planning. In [1], the authors simplified the problem to an open-loop Stackelberg game and showed that the human's objective function can be learned using inverse reinforcement learning methods [6].

Comparing to the large body of work on performance-oriented planning for human-robot interaction, ensuring safe interactions subject to uncertain human motion is a relatively less explored topic. One popular way of achieving safety for human-robot interaction tasks is by adding to the planning problem a chance constraint or cost penalty for collision avoidance, which is then accounted for via probabilistic predictions of the human's future motion (see for example [1]). In [3], the authors proposed to have the robot maintain a runtime measure of its degree of confidence in a learned human model. This allows the robot to plan probabilistically safe trajectories accounting for the observed accuracy of its own human motion predictions. Ultimately, however, under any such probabilistic approaches, safety can be compromised when the human takes low-probability actions. This is also known as the issue of the "long tail" of unlikely events [7].

In general, all-time safety in human-robot interaction can be ensured by a least-restrictive supervisory control scheme, often referred to as *shielding*. This approach involves synthesizing and implementing a reactive safety fallback policy as the "last-resort", which overrides a nominal policy *only when*

[1]Department of Electrical and Computer Engineering, Princeton University, {haiminh,jfisac}@princeton.edu
[2]Department of Mechanical and Aerospace Engineering, Princeton University, k.nakamura@princeton.edu

a safety-critical event, e.g. a collision, is imminent. Such shielding mechanisms include, for example, reachability analysis [8]–[10], control barrier functions [11], [12], Lyapunov methods [13], and model predictive control [14], [15]. Despite being effective at guaranteeing safety, applying shielding too frequently can greatly degrade the planning performance of the robot, since the safety controllers are typically designed without performance consideration such as task completion time, passenger comfort or energy consumption.

Simultaneously ensuring safety and optimizing performance for human-robot interaction tasks can be formulated as a stochastic optimal control problem (OCP), which combines propagating uncertainty (i.e. human motion), guaranteeing safety and optimizing the robot's objectives altogether in a single optimization problem. In principle, a stochastic OCP can be solved using stochastic dynamic programming [16], which is, however, only tractable for toy examples [17]. Recent work [18] proposes to approximately solve the OCP using stochastic model predictive control (SMPC) methods [19].

Statement of contributions: In this paper, we propose a novel shielding-aware planning framework that jointly achieves safety and performance for human-robot interaction. The key element of our approach is the formulation of a stochastic OCP that reasons about future shielding events via human motion prediction, while optimizing the robot's trajectory. The resulting policy improves the planning performance by preventing the robot from having to apply a costly shielding maneuver *in the future*. We reformulate the OCP by exploiting the structure in the human uncertainty model and solve it using efficient approximate dynamic programming methods. We evaluated our approach on simulated driving scenarios, with the human driver's trajectories taken from the Waymo Open Motion Dataset [20]. On average, our proposed planner improved the planning performance by at least 16% comparing to the state-of-the-art SMPC baseline across all testing scenarios.

II. PRELIMINARIES

A. Dynamical Systems

We consider a broad class of discrete-time dynamical systems for the robot and human, respectively,

$$x_{t+1}^R = f^R(x_t^R, u_t^R), \quad x_{t+1}^H = f^H(x_t^H, u_t^H), \quad (1)$$

where the input constraints are $u_t^R \in \mathcal{U}^R \subseteq \mathbb{R}^{m_R}$ and $u_t^H \in \mathcal{U}^H \subseteq \mathbb{R}^{m_H}$. We now define a joint system that captures the interactions between the human and robot subsystems,

$$x_{t+1} = f(x_t, u_t^R, u_t^H), \quad (2)$$

where $f : \mathbb{R}^{n_x} \times \mathcal{U}^R \times \mathcal{U}^H \to \mathbb{R}^{n_x}$ are the joint human-robot dynamics, whose state vector is given by $x_t = \Phi \left[x_t^R, x_t^H \right]$ and $\Phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_R + n_H}$ is a change-of-coordinates matrix.

Remark 1: The theoretical analysis in this paper extends to multi-human interaction by letting x_t^H, u_t^H in (2) represent the *joint* state and actions of multiple humans. Computational scalability is limited in practice by the exponential complexity common to combinatorial problems of this kind.

Running example: We consider a highway driving scenario, as depicted in Fig. 1, involving an autonomous vehicle (R) and a human-driven vehicle (H), each modeled by simplified dynamics taken from [2]. The states are the relative longitudinal position p_x^r, relative velocity v_r and lateral positions p_y^i, $i \in \{R, H\}$. The controls are the desired lateral velocity v_{lat}^i and acceleration a^i. The robot's task is to safely overtake the human.

B. Safe Human-Robot Interactive Planning via Shielding

In this paper, we focus on safety-critical human-robot interaction applications in which the state trajectory of the human-robot joint system must not enter a failure set $\mathcal{F} \subseteq \mathbb{R}^{n_x}$. This includes, for example, the robot colliding with the human. To ensure that $x_t \notin \mathcal{F}$ for all $t \geq 0$ despite the *worst-case* human actions, we make use of a supervisory safe control strategy, often referred to as "shielding", which is defined as a tuple (Ω, π^s). Here, set $\Omega \subseteq \mathbb{R}^{n_x}$ is a *safe set* that satisfies $\Omega \cap \mathcal{F} = \emptyset$, and $\pi^s : \mathbb{R}^{n_x} \to \mathcal{U}^R$ is a safe control policy that keeps the state inside Ω even under the worst-case human action. This is formalized in the following definition.

Definition 1 (Robust controlled-invariant set): Given dynamics (2) with a bounded uncertain input $u_t^H \in \mathcal{U}^H$, a set $\Omega \subseteq \mathbb{R}^{n_x}$ is a robust controlled-invariant set if there exists a control policy $\pi^s : \mathbb{R}^{n_x} \to \mathcal{U}^R$ that keeps x_t from leaving Ω:

$$x_0 \in \Omega \Rightarrow x_t \in \Omega, \ \forall t > 0, \ \forall u_t^H \in \mathcal{U}^H, \ u_t^R = \pi^s(x_t). \quad (3)$$

The definition suggests that the safe control $\pi^s(x_t)$ is needed only when the state is *about to leave* the safe set. Let the *shielding set* $\mathcal{S}^R \subset \Omega \times \mathcal{U}^R$ contain all state-action pairs that *might* result in the next state being outside of the safe set:

$$\mathcal{S}^R = \{(x, u^R) \in \Omega \times \mathcal{U}^R \mid \exists \tilde{u}^H \in \mathcal{U}^H : f\left(x, u^R, \tilde{u}^H\right) \notin \Omega\}. \quad (4)$$

We can then define a "least-restrictive" supervisory safety filter for arbitrary candidate control actions $\tilde{u}_t^R$:

$$u_t^R = \pi^{\mathbf{0}}(x_t; \tilde{u}_t^R) := \begin{cases} \tilde{u}_t^R, & \text{if } (x_t, \tilde{u}_t^R) \notin \mathcal{S}^R \\ \pi^s(x_t), & \text{if } (x_t, \tilde{u}_t^R) \in \mathcal{S}^R \end{cases} \quad (5)$$

The *shielding mechanism* (5) allows the robot to apply *any* nominal controller $\pi_t : \mathbb{R}^{n_x} \to \mathcal{U}^R$ as long as $\left(x_t, \pi_t(x_t)\right)$ is not in the shielding set $\mathcal{S}^R$; otherwise, it overrides $\pi_t(x_t)$ with the safety policy $\pi^s(x_t)$. The result below follows.

Proposition 1 (Shielding): If a set Ω is robust controlled-invariant under $\pi^s(\cdot)$, then it is robust controlled-invariant under $\pi^{\mathbf{0}}\left(\cdot; \pi_t(\cdot)\right)$, for any nominal control policy $\pi_t(\cdot)$.

Equation (5) and Proposition 1 describe a variety of shielding mechanisms, from Hamilton-Jacobi and Lyapunov analysis [8], [13] to predictive policy rollouts [9], [14], [15]. In this paper, we focus on efficient shielding-aware planning, only assuming that we have access to *some* shielding mechanism $\pi^{\mathbf{0}}$. Our framework is therefore quite general and can work in conjunction with many existing shielding methods.

Running example: A typical failure set for system (2) is $\mathcal{F} := \{x \in \mathbb{R}^4 \mid \left(|p_x^r| < 5.5 \text{ m} \wedge |p_y^R - p_y^H| < 2.0 \text{ m}\right) \vee |p_y^R| > 3.7 \text{ m}\}$, including any loss of separation between the two vehicles as well as R exceeding the road edges. Note

that $\mathcal{F}$ is a static set in the joint state space, even as H and R move. We use Hamilton-Jacobi (HJ) reachability [8] to compute the safe set Ω and control policy π^s for shielding.

C. Predicting Human Motion

The robot's main task is to achieve desirable performance through minimizing a cost function $\ell^R(x_t, u_t^R)$ over time. Note that both the cost function and the safe controller $\pi^s(x_t)$ depend on human's state x_t^H. Therefore, in order to plan efficiently, the robot must be able to predict the human's actions, since they can not only affect the robot's cost directly, but also indirectly by triggering (costly) shielding events. Here, we use the "noisily-rational" Boltzmann model originated from cognitive science [21] to predict human's future motion. Concretely, the probability of H taking a specific action $u_t^H \in \mathcal{U}^H$ is given by,

$$P\left(u_t^H \mid x_t, \beta_t, \theta_t\right) = \frac{e^{-\beta_t Q_{\theta_t}^H\left(x_t, u_t^H\right)}}{\sum_{\tilde{u}_t^H \in \tilde{\mathcal{U}}^H} e^{-\beta_t Q_{\theta_t}^H\left(x_t, \tilde{u}_t^H\right)}}, \quad (6)$$

where $Q_{\theta_t}^H(x_t, u_t^H)$ is the human's state-action value function, characterized by a set of time-varying parameters $\theta_t \in \mathbb{R}^{n_\theta}$ indicating human's possible intents. The inverse temperature $\beta_t > 0$, sometimes called "rationality coefficient" or "model confidence", quantifies the tendency of the human's actions to concentrate around the modeled optimum. This model assumes that the human is exponentially likelier to pick actions with better state-action values.

Remark 2: Our framework is agnostic to the concrete methods for determining the human's possible intents θ, which is usually specified by the system designer based on domain knowledge or learned from prior data. Goal-driven models of human motion are well-established in the literature. See for example [1], [6].

Running example: The human's state-action value function is expressed as the convex combination of two basis functions, $Q_{\theta_t}^H(\cdot) = \theta_t Q_1^H(\cdot) + (1 - \theta_t)Q_2^H(\cdot), \theta_t \in [0, 1]$, where $Q_1^H(\cdot)$ and $Q_2^H(\cdot)$ are quadratic functions capturing H tracking two possible intents: driving in the left and right lane, respectively, at the cruising speed 30 m/s.

D. Inferring Human Model Parameters

In general, parameters $(\beta_t, \theta_t) \in \Xi \subseteq \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_\theta}$ at each time instance t are unknown to the robot and therefore can only be estimated from past observations. To address this, we define the information vector $\mathcal{I}_t := \left[x_t, u_{t-1}^H, \mathcal{I}_{t-1}\right]$ as the collection of all *causally observable* information at time $t \geq 0$, with $\mathcal{I}_0 = [x_0]$. We then define the *belief state* $b_t := P\left(\beta_t, \theta_t \mid \mathcal{I}_t\right) \in \Delta$ as the probability distribution of parameters (β_t, θ_t) conditioned on $\mathcal{I}_t$, and $b_0 := P\left(\beta_0, \theta_0\right)$ is a given prior distribution. When the robot receives a new observation $u_t^H \in \mathcal{I}_{t+1}$, the current belief state $b_t \in \Delta$ is updated using the recursive Bayesian estimation,

$$b_{t+1}^- := P(\beta_t, \theta_t \mid \mathcal{I}_{t+1})$$
$$= \frac{P(u_t^H \mid x_t, \beta_t, \theta_t)b_t(\beta_t, \theta_t)}{\sum_{(\tilde{\beta}, \tilde{\theta}) \in \tilde{\Xi}} P(u_t^H \mid x_t, \tilde{\beta}, \tilde{\theta})b_t(\tilde{\beta}, \tilde{\theta})} \quad (7)$$

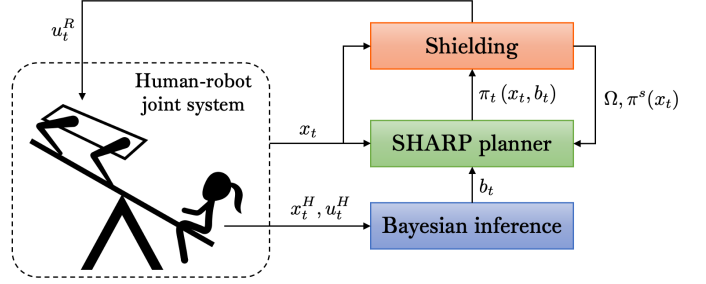Fig. 2: Overview of the proposed SHARP framework.

$$b_{t+1} = P(\beta_{t+1}, \theta_{t+1} \mid \mathcal{I}_{t+1})$$
$$= \sum_{(\tilde{\beta}, \tilde{\theta}) \in \tilde{\Xi}} P(\beta_{t+1}, \theta_{t+1} \mid \tilde{\beta}, \tilde{\theta})P(\tilde{\beta}, \tilde{\theta} \mid \mathcal{I}_{t+1}) \quad (8)$$

where $P(\beta', \theta' \mid \beta, \theta)$ is a transition model and a set $\tilde{\Xi}$ discretized from Ξ is used as the support. We can then rewrite (7) and (8) compactly as a dynamical system,

$$b_{t+1} = g(b_t, x_t, u_t^H). \quad (9)$$

III. SHARP: SHIELDING-AWARE ROBUST PLANNING

In this paper, our goal is to plan an efficient trajectory for the robot while ensuring safety at all times. A naïve approach would be using a shielding-agnostic nominal planner in (5), whose main focus is on performance but is unaware of the possibility of being overridden by the shielding mechanism. This approach can, however, yield a trajectory far from optimal in the presence of noisily rational human agents. The main reason is that shielding-agnostic planners tend to *unwittingly* activate shielding, resulting in frequent discrepancies between the efficiently planned trajectory, which will not be allowed to take place, and the costly executed trajectory, which was unaccounted for in planning. Conversely, a planner with shielding awareness reasons about potential future shielding events based on human motion predictions and preempts unnecessary overrides, thereby improving closed-loop performance.

Based on this central insight, we propose a new planning formulation that accounts for possible future shielding events, which we call Shielding-Aware Robust Planning (SHARP). The core of SHARP is a stochastic optimal control problem formulated as follows:

$$\min_{\pi_{[0:N-1]}} \mathbb{E}_{\substack{\beta_{[0:N-1]}, \theta_{[0:N-1]}, \\ u_{[0:N-1]}^H}} \sum_{k=0}^{N-1} \ell^R(x_k, u_k^R) + \ell_F^R(x_N) \quad (10a)$$

$$\text{s.t.} \quad x_0 = x_t, \ b_0 = b_t, \quad (10b)$$
$$\forall k = 0, \dots, N-1:$$
$$x_{k+1} = f\left(x_k, u_k^R, u_k^H\right) \quad (10c)$$
$$b_{k+1} = g\left(b_k, x_k, u_k^H\right) \quad (10d)$$
$$u_k^R = \pi^{\bullet}\left(x_t; \pi_k(x_k, b_k)\right) \quad (10e)$$

where $\ell^R : \mathbb{R}^{n_x} \times \mathcal{U}^R \to \mathbb{R}_{\geq 0}$ and $\ell_F^R : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ are designer-specified stage and terminal cost function, and $\pi_k : \mathbb{R}^{n_x} \times \Delta \to \mathcal{U}^R$ is a *causal* feedback policy that leverages the (yet-to-be-acquired) knowledge of x_k and b_k.

In theory, problem (10) can be solved using stochastic dynamic programming [16]. An optimal value function

$V_k(x_k, b_k)$ and control policy $\pi_k^*(x_k, b_k)$ can be obtained backwards in time using the Bellman recursion,

$$
\begin{aligned}
V_k(x_k, b_k) = \min_{\pi_k(x_k, b_k)} \; &\ell^R(x_k, u_k^R) \\
+ \mathop{\mathbb{E}}_{(\beta_k, \theta_k) \sim b_k, u_k^H} &\left[V_{k+1}(x_{k+1}, b_{k+1}) \mid \mathcal{I}_k \right]
\end{aligned} \quad (11)
$$
$$
\text{s.t.} \quad (10c) - (10e)
$$

with terminal condition $V_N(x_N, b_N) = \ell_F^R(x_N)$. Due to causal feedback, the controller obtained by solving (11) takes into account information that will become available in the future. As a result, the robot is able to predict upcoming shielding events using not only the *current* belief state b_t, but also a series of potential *future* belief states propagated via (10d), thus gaining an opportunity to plan a more efficient trajectory while staying safe *without* relying on the (usually) costly shielding maneuvers. Unfortunately, (11) is computationally intractable in all but the simplest cases. Even with spatial discretization, the belief states b_k generally live in a high dimensional space, which makes solving (11) infeasible in practice due to the "curse of dimensionality" [16].

Next, we focus on developing a tractable and efficient computation framework for solving OCP (10) approximately. Our road map is to reformulate (10) in two ways, each tackled with a different approximate dynamic programming method. Our main focus is on reformulating (10) as a scenario-tree-based stochastic model predictive control (ST-SMPC) problem, which is a real-time trajectory optimization method originally developed in [19]. This approach estimates the expectation in (10a) and propagates the belief states in (10d) based on a small number of likely uncertainty realizations, thereby preserving a simplified but representative truncation of the original problem's structure. However, we first present a simpler relaxation of (11) based on the QMDP assumption [22], which allows computing a tabular solution offline. The solution is a value function that approximately captures the cost-to-go over the full horizon N, and can be used as a guiding terminal cost function in ST-SMPC to implicitly extend the planning horizon. The overall SHARP framework is illustrated in Fig. 2.

A. Problem Simplification with the QMDP Assumption

In this section, we discuss how to solve a relaxation of (11) with an offline tabular dynamic programming scheme. We start by discretizing the joint state space and robot's action space into $\tilde{\mathbb{X}}, \tilde{\mathcal{U}}^R$, and letting $z_t := [x_t, \beta_t, \theta_t]$. Now, under perfect observability of z_t, we would have a fully certain belief $b_t \equiv \mathbb{1}_{(\beta_t, \theta_t)}$ and (11) would reduce to a full-information problem that can be numerically solved with the Bellman recursion:

$$
\begin{aligned}
\tilde{V}_k(z_k) = \min_{\pi_k(z_k)} \; &\ell^R(x_k, u_k^R) \\
+ \sum_{(\tilde{\beta}, \tilde{\theta}) \in \tilde{\Xi}} &P(\tilde{\beta}, \tilde{\theta} \mid \beta_k, \theta_k) \mathop{\mathbb{E}}_{u_k^H} \left[\tilde{V}_{k+1}(\tilde{z}_{k+1}) \right]
\end{aligned} \quad (12)
$$
$$
\text{s.t.} \quad (10c), (10e),
$$

where $\tilde{z}_{k+1} := [x_{k+1}, \tilde{\beta}, \tilde{\theta}]$. This simplified Bellman recursion follows the QMDP assumption [22], which optimistically

assumes that the uncertainties in the current belief states (β, θ) disappear in one time-step. Here, in lieu of evolving the belief states with the measurement update (7), uncertainties in (β, θ) are now propagated only by the transition model $P(\beta', \theta' \mid \beta, \theta)$ in (8). As a result, the Bellman recursion (12) can be computed efficiently, at the cost of losing the ability to account for future uncertainties. Given a state x_t, a belief state b_t and a lookup table of $\tilde{V}_0(\cdot)$ obtained by (12), we can obtain a value function,

$$
\begin{aligned}
V_F(x_t, b_t) := \min_{\pi_k(x_t, b_t)} \; &\ell^R(x_t, u_t^R) \\
+ \mathop{\mathbb{E}}_{(\beta_t, \theta_t) \sim b_t} \mathop{\mathbb{E}}_{u_k^H} \mathop{\mathbb{E}}_{(\tilde{\beta}, \tilde{\theta})} &\left[\tilde{V}_0(\tilde{z}_{t+1}) \right],
\end{aligned} \quad (13)
$$

which is an optimistic estimate of the true cost-to-go $V_t(x_t, b_t)$ of (11). In Section III-B4, we will use this approximate value function as a guiding terminal cost in ST-SMPC. As a byproduct of (13), we can obtain a causal feedback control policy, which we refer to as SHARP-QMDP. In the next section, we will use this policy to construct a scenario tree for ST-SMPC. Nonetheless, it can also be used directly as the nominal planner in (5) for online planning. Although this policy no longer propagates belief states, it is still effective at predicting shielding events and gains an information advantage over a shielding-unaware policy due to causal feedback and the shielding constraint (10e).

B. ST-SMPC with the Sparse LQG Tree

The performance of SHARP-QMDP can be limited by its inability to propagate the belief states with measurements on human uncertainties. In this section, we focus on developing a shielding-aware planner that propagates the belief states and leverages them to better predict future shielding events. Motivated by recent advances in approximate dynamic programming for uncertain systems [18], [23], we propose to propagate the belief states in (10d) using samples of u_k^H. This leads to a scenario tree that allows us to reformulate (10) as a computationally tractable ST-SMPC problem. With discretized human action and parameter spaces $\tilde{\mathcal{U}}^H, \tilde{\Xi}$, the (intractable) Bellman recursion (11) can be evaluated for any given state x_0 and belief state b_0:

$$
\begin{aligned}
V_0(x_0, b_0) = \min_{u_0 \in \mathcal{U}^R} \ell^R(x_0, u_0^R) + \sum_{\beta, \theta} b_0(\beta, \theta) \cdot \\
\sum_{\tilde{u}_0^H \in \tilde{\mathcal{U}}^H} P(\tilde{u}^H \mid x_0^H, \beta, \theta) V_1(\tilde{x}_1, \tilde{b}_1),
\end{aligned} \quad (14)
$$

with value functions at subsequent times obtained recursively in an analogous manner. The next state $\tilde{x}_1$ and belief state $\tilde{b}_1$ are obtained by computing $\tilde{x}_1 = f\left(\mathrm{pre}(\tilde{x}_1), \tilde{u}_0^R, \tilde{u}_0^H\right)$ and $\tilde{b}_1 = g(\mathrm{pre}(\tilde{b}_1), \mathrm{pre}(\tilde{x}_1^H), \tilde{u}_0^H)$. Here, $\mathrm{pre}(\tilde{x}_1) := x_0$ is the predecessor state of $\tilde{x}_1$, similarly for beliefs. Given a sequence of human uncertainty realizations $(\tilde{\beta}_{[0:N-1]}, \tilde{\theta}_{[0:N-1]}, \tilde{u}_{[0:N-1]}^H)$, we refer to the corresponding state and belief state trajectory $(\tilde{x}_{[0:N]}, \tilde{b}_{[0:N]})$ as a *scenario*. Note that by expanding (11) using (14), the total number of scenarios is $(|\tilde{\Xi}||\tilde{\mathcal{U}}^H|)^N$. As a result, the optimization problem can quickly become intractable due to an exponentially growing number of decision variables. Therefore, we use ST-SMPC to solve the problem over a subset of representative human uncertainty realizations.
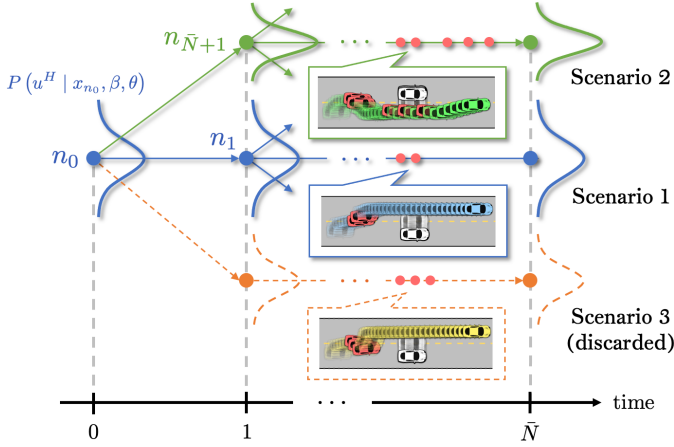
Fig. 3: Illustration of a sparse LQG scenario tree. Red (colored) dots denote (non-)shielding nodes. The bell curve at node $\tilde{n}$ represents the Gaussian distribution $P(u^H \mid x_{\tilde{n}}, \beta, \theta)$. Two scenarios (1 and 2) are branched out from the root node n_0, while Scenario 3 is discarded due to its similarity with Scenario 1.

Remark 3: Recall that in (6) we use a human's state-action value function $Q_{\theta_t}^H(x_t, u_t^H)$ that depends on the robot's state x_t^R. This introduces coupling between uncertainties and decision variables in (14), which significantly increases the complexity of the optimization. In order to plan in real time, we consider a class of human state-action value function parametrized as $Q_{\theta_t}^H(x_t^H, u_t^H)$, which (conservatively) assumes that the human does not react to the robot. As a result, the human's action model (6) equals $P(u_t^H \mid x_t^H, \beta_t, \theta_t)$. Nonetheless, we show in Section IV that our method is still effective with a "responsive" human, whose unmodeled responses cause a reduction in the inferred inverse temperature β_t, similar to [3]. Our work may be extended to explicitly account for human reactions leveraging recent advances in dual SMPC with state-dependent uncertainty [23].

1) Constructing a sparse scenario tree: Our proposed scenario tree construction procedure is summarized in Alg. 1 and depicted in Fig. 3. We start by introducing some useful definitions. We denote a *node* in the tree as n, whose state and belief state are denoted as x_n and b_n. The set of all nodes is defined as $\mathcal{N}$. We define the transition probability from a parent node $\text{pre}(n)$ to its child node n as $\bar{P}_n := \sum_{(\beta,\theta)\sim b_n} b_n(\beta,\theta) \cdot P(u^H \mid \text{pre}(x_n^H), \beta, \theta)$. Subsequently, the *path transition probability* of node n, i.e. the transition probability from the root node n_0 to node n can be computed recursively as $P_n := \bar{P}_n \cdot \bar{P}_{\text{pre}(n)} \cdots \bar{P}_{n_0}$.

In order to efficiently leverage belief state propagation for predicting future shielding events, our scenario tree construction procedure differs from the conventional ones [18], [19], [23] in three key aspects. First, at scenario branching time (Alg. 1, Line 21), we only need to draw samples for human's actions u^H but not for the parameters (β, θ). Importantly, those u^H samples are only used for updating the belief states (Alg. 1, Line 18). In the next section, we show that by exploiting the problem structure, the robot's action obtained by solving the SMPC will adapt to the belief states instead of the samples. Second, after each scenario branching, instead of propagating the (belief) states for only one time step, we perform a forward simulation up to a truncated horizon of

---

Algorithm 1 Constructing a sparse LQG scenario tree

Input: Current state $x_t \in \Omega$ and belief state b_t, maximum number of nodes $M > 0$, truncated horizon $\bar{N} \leq N$, surrogate policy $\pi_{\text{QMDP}}(x, b)$
Output: A scenario tree defined by node sets $\mathcal{N}_t, \mathcal{N}_t^s$

    // Initialization:
1: $x_{n_0} \leftarrow x_t, b_{n_0} \leftarrow b_t, t_{n_0} \leftarrow 0, P_{n_0} \leftarrow 1$
2: $\mathcal{N}_t \leftarrow \{n_0\}, \mathcal{N}_t^s \leftarrow \emptyset, m \leftarrow 1, n_{\text{br}} \leftarrow n_0$
3: **while** $m \leq M$ **do**
      // Forward Simulation for One Scenario:
4:     $\tilde{n} \leftarrow n_{\text{br}}$
5:     **for all** $k \leftarrow t_{n_{\text{br}}}, t_{n_{\text{br}}} + 1, \ldots, \bar{N} - 1$ **do**
      // Robot Control:
6:       $u_{\tilde{n}}^R \leftarrow \pi_{\text{QMDP}}(x_{\tilde{n}}, b_{\tilde{n}})$
7:       **if** $(x_{\tilde{n}}, u_{\tilde{n}}^R) \in \mathcal{S}^R$ **then**   *// Shielding required*
8:         $u_{\tilde{n}}^R \leftarrow \pi^s(x_{\tilde{n}})$
9:         $\mathcal{N}_t^s \leftarrow \mathcal{N}_t^s \cup \{\tilde{n}\}$
10:      **end if**
      // Human Control:
11:      **if** $k = t_{n_{\text{br}}}$ **and** $|\mathcal{N}_t| > 1$ **then**   *// Branching*
12:        $u_{\tilde{n}}^H \leftarrow u_{\text{br}}^H$
13:      **else**   *// Non-branching*
14:        $u_{\tilde{n}}^H \leftarrow \arg\max \sum_{\beta,\theta} b_{\tilde{n}}(\beta,\theta) P(u^H | x_{\tilde{n}}^H, \beta, \theta)$
15:      **end if**
16:      Compute path transition probability: $P_{n_m} \leftarrow P_{\tilde{n}} \cdot \sum_{\beta,\theta} b_{\tilde{n}}(\beta,\theta) \cdot P(u_{\tilde{n}}^H \mid x_{\tilde{n}}^H, \beta, \theta)$
17:      Update state: $x_{n_m} \leftarrow f(x_{\tilde{n}}, u_{\tilde{n}}^R, u_{\tilde{n}}^H)$
18:      Update belief state: $b_{n_m} \leftarrow g(b_{\tilde{n}}, x_{\tilde{n}}, u_{\tilde{n}}^H)$
19:      $\mathcal{N}_t \leftarrow \mathcal{N}_t \cup \{n_m\}, \tilde{n} \leftarrow n_m, m \leftarrow m + 1$
20:     **end for**
21:     $(n_{\text{br}}, u_{\text{br}}^H) \leftarrow \text{GETBRANCHNODE}(\mathcal{N}_t)$
22: **end while**
23: $\mathcal{N}_t \leftarrow \text{NORMALIZEPATHTRANSPROB}(\mathcal{N}_t)$

$\bar{N} \leq N$ (Alg. 1, Line 3-20). This generally leads to a *sparse* scenario tree with an increased depth, allowing us to capture more shielding events in the future. Finally, when branching out new nodes (Alg. 1, Line 21), instead of selecting nodes with higher realization probabilities [19], we are interested in those that lead to *distinct trajectories*, which are essentially shaped by different shielding events. Concretely, when picking a new branch node n_{br}, we prioritize one with a smaller time step t_{br}, which is likelier to result in a distinct trajectory from the existing ones in the tree. At node n_{br}, we sample several human's action $\tilde{u}^H \in \mathcal{U}^H$, each of which produces a scenario $(\tilde{x}_{[0:\bar{N}]}, \tilde{b}_{[0:\bar{N}]})$ via forward simulation. We then pick $u_{\text{br}}^H = \tilde{u}^H$ that leads to the most different scenario from all existing ones in the tree. The difference between two scenarios is measured in terms of the difference in the metric $\xi^\top H \xi$, where ξ is a vector stacking all components of $\tilde{x}_{[0:\bar{N}]}, \tilde{b}_{[0:\bar{N}]}$ and H is a positive semidefinite matrix.

2) Optimizing over LQG scenarios: In ST-SMPC, given a scenario tree, one shall optimize simultaneously for each scenario a robot's action sequence, which reacts to the human uncertainty in that scenario. One key difference of our approach from ST-SMPC literature [18], [19], [23] is that the optimized robot's action $u_{\tilde{n}}^R$ at node $\tilde{n}$ does not react to the *samples*, i.e. the human's action $u_{\tilde{n}}^H$, but to the entire

distributions $P(u^H \mid x_{\tilde{n}}^H, \beta, \theta)$ and $b_{\tilde{n}}$. Specifically, given a scenario $(x_{\tilde{n}_{[0:\bar{N}]}}, b_{\tilde{n}_{[0:\bar{N}]}})$ associated with node sequence $\tilde{n}_{[0:\bar{N}]}$, the corresponding scenario optimization problem is,

$$\min_{\bar{u}_{\tilde{n}_{[0:\bar{N}-1]}}^R} \sum_{k=0}^{\bar{N}-1} \mathop{\mathbb{E}}_{\substack{(\beta,\theta)\sim b_{\tilde{n}_k}, \\ u_{\tilde{n}_k}^H \sim P(u^H|x_{\tilde{n}_k}^H,\beta,\theta)}} \ell^R(\bar{x}_{\tilde{n}_k}, \bar{u}_{\tilde{n}_k}^R) \tag{15}$$

subject to constraints (10b), (10c) and (10e), where we use $(\bar{\cdot})$ to denote decision variables. If this scenario shares nodes with other scenarios (e.g. node n_0 in Fig. 3), then the robot's action at those shared nodes should be constrained to be the same, which enforces *causality* [19].

One key observation of (15) is that if $Q_\theta^H(x^H, u^H)$ is approximated as a quadratic function of u^H, then the human's action uncertainty $P(u^H \mid x^H, \beta, \theta)$ becomes a Gaussian distribution with mean $\hat{u}^H(\beta, \theta) := \arg\max P(u^H \mid x^H, \beta, \theta)$. Furthermore, we linearize the joint dynamics (2) around scenario trajectories $x_{\tilde{n}_{[0:\bar{N}]}}$, $u_{\tilde{n}_{[0:\bar{N}]}}^R$ and $u_{\tilde{n}_{[0:\bar{N}]}}^H$ to obtain a linear dynamical system,

$$\delta x^+ = A_{\tilde{n}_k} \delta x + B_{\tilde{n}_k}^R \delta u^R + B_{\tilde{n}_k}^H \delta u_{\tilde{n}_k}^H, \tag{16}$$

where $\delta x = x - x_{\tilde{n}_k}$, $\delta u^R = u^R - u_{\tilde{n}_k}^R$, $\delta u_{\tilde{n}_k}^H = \hat{u}_{\tilde{n}_k}^H - u_{\tilde{n}_k}^H$ and $A_{\tilde{n}_k}$ is the Jacobian $D_{x_{\tilde{n}_k}} f(\cdot)$, likewise for $B_{\tilde{n}_k}^R$ and $B_{\tilde{n}_k}^H$. If we further drop the shielding constraint (10e) for a moment (we will return to this in the next section) and consider a quadratic cost ℓ^R, then (15) becomes a Linear-Quadratic-Gaussian (LQG) problem, whose optimal solution is known to be certainty-equivalent [24]. The resulting robot's control sequence $u_{\tilde{n}_{[0:\bar{N}-1]}}^R$ will be robust to distributions $P(u^H \mid x_{\tilde{n}_k}^H, \beta, \theta)$ and $b_{\tilde{n}_k}(\beta, \theta)$.

3) Convexifying the shielding constraint: The final piece we need to deal with is the shielding constraint (10e), which is in general non-convex. In this paper, we propose to convexify it using the discrete-time exponential control barrier function (CBF) developed in [25]. The main idea is to linearize the system and approximate the safe set as a halfspace at any state $x \in \mathcal{S}_{(\cdot)}^R$, in which case an affine CBF can be constructed analytically [25]. Concretely, given a shielding node $\tilde{n} \in \mathcal{N}^s$, we first obtain a linearized system at $(x_{\tilde{n}}, u_{\tilde{n}})$ according to (16). We then approximate the safe set Ω locally at $x_{\tilde{n}}$ as a halfspace defined by

$$\bar{\Omega}_{\tilde{n}} := \{x \mid \mathbf{n}_{\tilde{n}}^\top (x - x_{\tilde{n}}) \geq 0\} = \{\delta x \mid \mathbf{n}_{\tilde{n}}^\top \delta x \geq 0\}, \tag{17}$$

where $\mathbf{n}_{\tilde{n}} := f(x_{\tilde{n}}, \pi^s(x_{\tilde{n}}), u_{\tilde{n}}^H) - x_{\tilde{n}}$ approximates the normal vector of the tangent space of $\mathcal{S}_{u_{\tilde{n}}^R}^R$ at $x_{\tilde{n}}$, as illustrated in Fig. 4.

Proposition 2: [25, Prop. 4] Given a safe set $\bar{\Omega}_{\tilde{n}}$ define by (17), the affine function $h_{\tilde{n}}(\delta x) := \mathbf{n}_{\tilde{n}}^\top \delta x$ is a discrete-time exponential CBF for system (16) linearized at $(x_{\tilde{n}}, u_{\tilde{n}})$ if there exists $\gamma \in (0, 1]$ and $u^R \in \mathcal{U}^R$ such that $\forall \delta x \in \bar{\Omega}_{\tilde{n}}$, $h_{\tilde{n}}(A_{\tilde{n}} \delta x + B_{\tilde{n}}^R \delta u^R + B_{\tilde{n}}^H \delta u^H) + (\gamma - 1)h_{\tilde{n}}(\delta x) \geq 0$ holds.

Using the CBF defined in Proposition 2, we can now approximate the shielding constraint (10e) as,

$$\mathbf{n}_{\tilde{n}}^\top \left[(A_{\tilde{n}} + (\gamma - 1)I) \delta x + B_{\tilde{n}}^R \delta u^R + B_{\tilde{n}}^H \delta u^H \right] \geq 0, \tag{18}$$

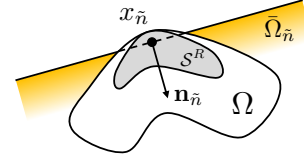which is linear (and hence convex) in δx and δu^R.

Fig. 4: Illustration of a CBF-based convex shielding constraint.

Remark 4: As pointed out in [25], constraint (18) is not necessarily feasible for bounded control input. Therefore, we incorporate it as a soft constraint in the ST-SMPC problem.

4) Overall ST-SMPC Problem for SHARP: Given a sparse LQG scenario tree defined by node sets $\mathcal{N}_t$ and $\mathcal{N}_t^s$, we can approximate (11) as an ST-SMPC problem,

$$\begin{aligned}
\min_{\Pi_t} \quad & \sum_{\tilde{n}\in\mathcal{N}_t\setminus\mathcal{L}_t} \sum_{\beta,\theta} b_{\tilde{n}}(\beta,\theta) P_{\tilde{n}} \ell^R(\bar{x}_{\tilde{n}}^{\beta,\theta}, \pi_{\tilde{n}}) \\
& + \sum_{\tilde{n}\in\mathcal{L}_t} \sum_{\beta,\theta} b_{\tilde{n}}(\beta,\theta) P_{\tilde{n}} V_F(\bar{x}_{\tilde{n}}^{\beta,\theta}, b_{\tilde{n}}) \\
\text{s.t.} \quad & \forall \tilde{n} \in \mathcal{N}_t \setminus \mathcal{L}_t : \pi_{\tilde{n}} \in \mathcal{U}^R, \\
& \forall \tilde{n} \in \mathcal{N}_t \setminus \{n_0\} : (16), \\
& \forall \tilde{n} \in \mathcal{N}_t^s : (18),
\end{aligned} \tag{19}$$

where $\mathcal{L}_t$ is the set of all leaf nodes $\tilde{n}$ with $t_{\tilde{n}} = \bar{N}$, $\Pi_t := \{\pi_{\tilde{n}}(\bar{x}_{\tilde{n}}^{\beta,\theta}, b_{\tilde{n}}) : \tilde{n} \in \mathcal{N}_t \setminus \mathcal{L}_t\}$ is the collection of robot's control inputs associated with all non-leaf nodes, and $V_F(\cdot, \cdot)$ is the QMDP value function defined in (13). The path transition probabilities are normalized such that they sum up to 1 at each time step (Alg. 1, Line 23). Problem (19) is a quadratic program and thus can be solved efficiently. The optimal solution Π_t^* to (19) is implemented in a receding horizon fashion, i.e. $\pi_{\text{SMPC}}(x_t, b_t) = \pi_{n_0}^*$. We refer to this policy as SHARP-SMPC.

IV. RESULTS

In this section, we evaluate SHARP on simulated driving scenarios, where we use the human driver's trajectories both from the Waymo Open Motion Dataset [20] and simulated using a car-following model in [26]. For simulation purposes, vehicle dynamics are described by a kinematic bicycle model [2] and discretized with a time step of $\Delta t = 0.2$ s; for planning, we use the linearized model from the Running Example. All simulations are performed using MATLAB and YALMIP [27] on a laptop with an Intel Core i7-7820HQ CPU. The code and dataset are available at https://github.com/SafeRoboticsLab/SHARP

Ablation. We consider an ablation method that uses the state-of-the-art stochastic MPC scheme [18], which is based on the ST-SMPC technique originally developed in [19], but additionally propagates the belief states that allows for human motion prediction via (6) and (9). The MPC only has control constraints. Therefore, the scenario information is only used by the objective function and the resulting policy is *safety-unaware* (though nonetheless *safe* thanks to shielding).

Baseline. Our baseline method adds to the ablation soft collision-avoidance constraints of $x_{\tilde{n}} \notin \mathcal{F}$, $\forall \tilde{n} \in \mathcal{N}$. We used a simple grid search to determine approximately optimal weights for the soft constraints. Note that the baseline policy is *safety-aware* but *shielding-agnostic*.
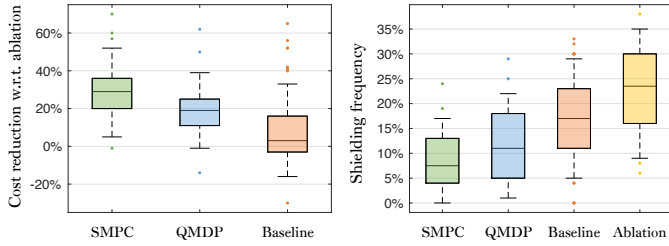
Fig. 5: Cost reduction and shielding frequency of Scenario 1 with 50 human trajectories from the Waymo Open Motion Dataset [20]. The central mark, bottom and top edges of the box indicate the median, 25th and 75th percentiles, respectively. The max whisker length is the interquartile range. Outliers are shown as points.

Simulation Setup. The ablation, baseline and SHARP planners are equipped with the same HJ-reachability-based shielding policy [8]. They also use the same human intent inference scheme (6) and (9) to obtain a prediction of human's future trajectories. All ST-SMPC problems use a bound $M = 70$ on the number of nodes in the tree, and are solved with MOSEK [28] (average solving time 60 ms).

Metrics. We first define the closed-loop cost as $J_{\text{cl}}^R := \sum_{t=0}^{T_{\text{sim}}} \ell^R(x_t, u_t^R)$, where T_{sim} is the simulation horizon, and $x_{[0:T_{\text{sim}}]}, u_{[0:T_{\text{sim}}]}$ are the *executed* state and input trajectories (with replanning). To measure the performance of the planners, we consider the following two metrics:

- Cost reduction rate: Defined as the percentage reduction of the closed-loop cost achieved by a certain planner with respect to the one achieved by the ablation.
- Shielding frequency: A number defined as $T_{\text{🛡}}/T_{\text{sim}} \times 100\%$, where $T_{\text{🛡}}$ is the number of time steps when shielding is used.

A. Scenario 1: Highway Overtaking

We first show simulation results for Scenario 1, which is the running example. We simulate the scenario for 50 times, each with a different human's trajectory taken from the Waymo Open Motion Dataset [20]. The performance metrics are presented in Fig. 5. We observe that SHARP planners outperform the baseline in both metrics, due to their ability to take advantage of human inference to predict the costly shielding events. On the other hand, even though the baseline also leverages human inference for collision avoidance, the heuristic proximity penalty can negatively interfere with the robot's actual performance criterion, and is ultimately less effective at preventing unnecessary shielding events.

Snapshots of one simulation trial are shown in Fig. 6. We observe that SHARP-SMPC accurately predicts the human's future movement to the right lane, controls the robot to stay in the left lane, following the human without incurring shielding (top left), and safely overtakes the human when a window of opportunity opens (top right). SHARP-QMDP, despite rendering a low shielding frequency as well thanks to the shielding-awareness, cannot as effectively reason about and react to the human's uncertain trajectory due to the overly optimistic QMDP assumption, resulting in a more conservative trajectory. The baseline triggers more shielding events and produces a less efficient trajectory than the SHARP planners due to lack of shielding-awareness.
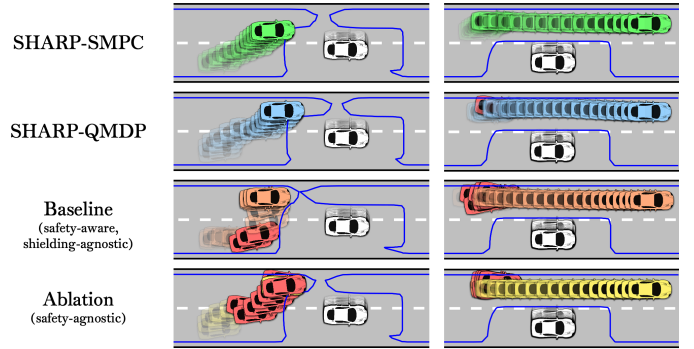

Fig. 6: Simulation snapshots of Scenario 1. Longitudinal positions are shown in relative coordinates with $p_x^H = 0$. The left column displays trajectories for $t = [0, 4.8]$ s and the right one displays the remainder of the trajectories. (p_x^R, p_y^R)-slices of the safe set Ω, taken at the terminal state in each trajectory, are indicated in blue. A red vehicle snapshot indicates a shielding override.

B. Scenario 2: Traffic Intersection

Next, we consider a traffic intersection scenario where the human may choose to stop, go straight or make a right turn. The performance metrics obtained from 50 simulation trials with human's trajectories taken from the Waymo Open Motion Dataset are shown in Fig. 8. Snapshots of two simulation trials with the human going straight and turning right are depicted in Fig. 7.

C. Responsive Human

Finally, we revisit Scenario 1 with a responsive human (see Remark 3). We simulate the behaviour of the human with the car following model from [26, Chapter 4], which is also used in microscopic traffic simulators such as SUMO [29]. The parameter values we used are human's preferred acceleration $a = 3$ m/s^2, reaction time $\tau = 1$ s, and random velocity perturbation $\eta = 0.1$ m/s. The human also performs random lane changing maneuvers. The performance metrics obtained from 50 simulation trials are shown in Fig. 9. We observe that even in the face of unmodeled human behavior, SHARP planners still outperform the baseline.

V. DISCUSSION

Summary. We have introduced Shielding-Aware Robust Planning (SHARP), a decision-making framework for safe and efficient interaction. The SHARP policy improves robustness by accounting for possible future shielding events, proactively balancing nominal performance with costly emergency maneuvers triggered by unlikely human behaviors. **Limitations and future work.** Performance of SHARP policies can be limited by neglecting human reactions to the robot's future decisions. The scenario tree approach provides a promising avenue for extended formulations that tractably account for human responsiveness. Similarly, scalability improvements are needed in order to compute real-time SHARP policies for multi-human multi-robot interaction. Finally, the current framework assumes that the robot can accurately observe the state and past human actions, which is often unrealistic. Combining the efficient risk mitigation of SHARP with recent advances in safe perception-aware planning [9] is likely to yield more general and powerful frameworks.
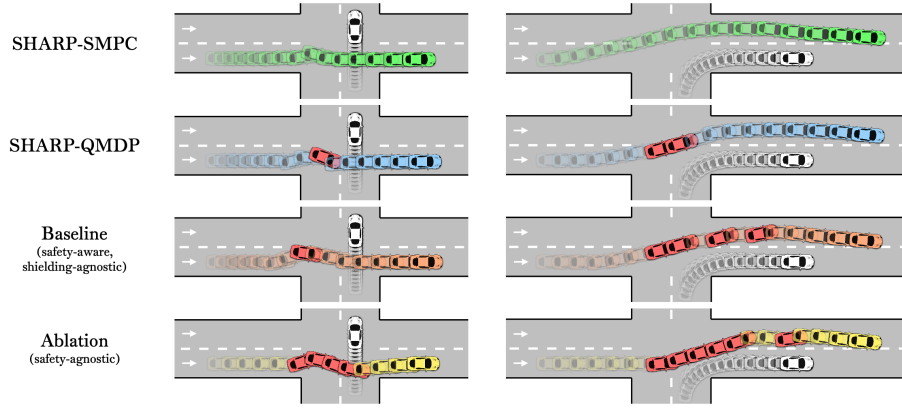
Fig. 7: Simulation snapshots of Scenario 2. A red vehicle snapshot indicates a shielding override.
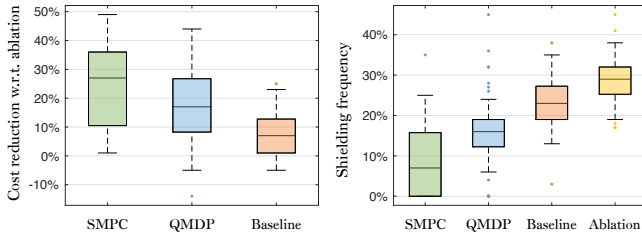

Fig. 8: Cost reduction and shielding frequency of Scenario 2 with 50 human trajectories from the Waymo Open Motion Dataset [20].
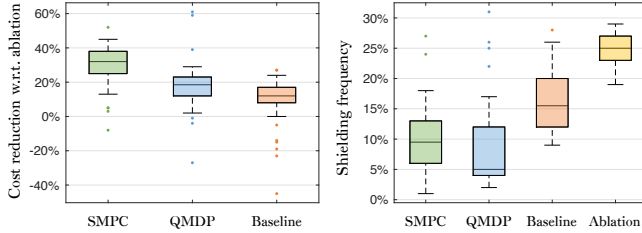

Fig. 9: Cost reduction and shielding frequency of Scenario 1 obtained from 50 trials, where the human is simulated using [26].

REFERENCES

[1] D. Sadigh, N. Landolfi, S. S. Sastry, *et al.*, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, 2018.

[2] J. F. Fisac *et al.*, "Hierarchical game-theoretic planning for autonomous vehicles," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[3] ——, "Probabilistically safe robot planning with confidence-based human predictions," *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018.

[4] H. B. Amor, G. Neumann, S. Kamthe, *et al.*, "Interaction primitives for human-robot cooperation tasks," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[5] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," *AAAI*, 2004.

[6] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," *AAAI*, Chicago, IL, USA, vol. 8, 2008.

[7] P. Koopman, "The heavy tail safety ceiling," *Automated and Connected Vehicle Systems Testing Symposium*, vol. 1145, 2018.

[8] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," *IEEE Conference on Decision and Control (CDC)*, 2017.

[9] Z. Zhang and J. F. Fisac, "Safe occlusion-aware autonomous driving via game-theoretic active perception," *Proceedings of Robotics: Science and Systems*, Held Virtually, 2021.

[10] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac, "Safety and liveness guarantees through reach-avoid reinforcement learning," *Proceedings of Robotics: Science and Systems*, Held Virtually, 2021.

[11] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, 2016.

[12] A. Robey *et al.*, "Learning control barrier functions from expert demonstrations," *IEEE Conference on Decision and Control (CDC)*, 2020.

[13] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," *Advances in Neural Information Processing Systems*, 2018.

[14] S. Li and O. Bastani, "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, 2021.

[16] D. P. Bertsekas, "Dynamic programming and optimal control", 2. Athena scientific Belmont, MA, 1995, vol. 1.

[17] E. D. Klenske and P. Hennig, "Dual control for approximate bayesian reinforcement learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, 2016.

[18] E. Arcari, L. Hewing, and M. N. Zeilinger, "An approximate dynamic programming approach for dual stochastic model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, 2020.

[19] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, 2011.

[20] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] R. D. Luce, "Individual Choice Behavior", ser. Individual Choice Behavior. Oxford, England: John Wiley, 1959.

[22] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," *Machine Learning Proceedings*, Elsevier, 1995.

[23] A. D. Bonzanini, J. A. Paulson, and A. Mesbah, "Safe learning-based model predictive control under state-and input-dependent uncertainty using scenario trees," *IEEE Conference on Decision and Control (CDC)*, 2020.

[24] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, 1971.

[25] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," *Proceedings of Robotics: Science and Systems*, Cambridge, MA, USA, vol. 13, 2017.

[26] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," 1998.

[27] J. Lofberg, "Yalmip : A toolbox for modeling and optimization in matlab," *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[28] M. ApS, "Mosek optimization toolbox for MATLAB", *User's Guide and Reference Manual*, 2019.

[29] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.