A Comparison of Learning-Aware Safety Methods

This section provides a table comparing our proposed Belief Game framework with recent learning-aware safe planning methods. To the best of our knowledge, our method is the first safety analysis framework that jointly reasons the agent's physical states and the robot's belief states in a closed-loop fashion and can scale up to high-dimensional systems with implicit learning dynamics.

Table 3: Comparison of learning-aware safe planning methods.

Feature/Method	Peters et al. [26]	Tian et al. [7]	Hu et al. [27]	Bajcsy et al. [10]	Zhang et al. [8]	Packer et al. [9]	**Ours**
Recursive safety guarantees	N	N	**Y**	**Y**	**Y**	N	**Y**
Active information gathering	N	N	**Y**	N	**Y**	**Y**	**Y**
Uncertainty-dependent safety analysis	N/A	N	N/A	**Y**	**Y**	N/A	**Y**
Belief refinement based on observations	N	**Y**	**Y**	**Y**	N/A	N/A	**Y**
Scaling to high dimension ($n_z > 10$)	**Y**	N	N	N	N	**Y**	**Y**
Allowing implicit learning dynamics	N	N	N	**Y**	N	**Y**	**Y**
Allowing continuous hypotheses space	N	N	**Y**	N	N	N/A	N
Fully online policy computation	**Y**	N	**Y**	N	**Y**	N	N

B Belief Game with Motion Transformer

B.1 Problem Setup

Consider the traffic scenario when the robot aims to traverse the intersection without violating the road bound or causing a collision with the opponent. We model both vehicles using the kinematic bicycle dynamics with longitudinal acceleration and steering angle controls. In addition, we limit their state and action space by bounding velocity $v \in [0, 10]$ m/s, acceleration $a \in [-5, 5]$ m/s^2, and steering angle $\delta \in [-0.5, 0.5]$ rad.

To infer the opponent's future actions, the robot uses a state-of-the-art trajectory prediction model, Motion Transformer (MTR) [2], which outputs a Gaussian Mixture Model of trajectories over the next 8 seconds from the 1.1 seconds of scene history. To construct the predictive control bound, we utilize a proportional controller to track the mean trajectory of each mixture component (mode θ) as the opponent's nominal policy $\pi^o_t(x_t; \theta)$. In addition, we set $d^o_t(\theta)$ by assuming it can deviate from the nominal policy up to ± 2 m/s^2 in acceleration and ± 0.1 rad in steering angle. Since MTR outputs 64 modes using a prior motion query, we aggregate overlapping trajectories using non-maximum suppression and mask out modes with $b_t(\theta) < 0.05$.

B.2 Network Architecture and Training Stratergy

The MTR model is first trained with the entire Waymo Open Motion Dataset [24] for 30 epochs and achieves claimed results in their paper. Then, we set up a simulation environment with the pre-trained MTR model in the loop to generate predictions and predictive control bounds for the opponents. We represent the state with the absolute pose of the robot w.r.t the map, the opponent's relative pose w.r.t the robot, the nominal actions for each valid prediction mode, and their probabilities. The Belief Game is trained using the Iterative Soft Adversarial Soft Actor-Critic (ISAACS) framework, where four neural networks are trained asynchronously.

- The *ego actor* is the policy of the robot. It first encodes the states and each prediction mode independently by multi-layer perceptions (MLP). Then the state feature is concatenated to each prediction feature and passed through another MLP. We conduct max-pooling across all prediction modes to generate the aggregated feature, which is processed by the final MLP and becomes the mean and standard deviation of the robot's action. Finally, we sample the action using the squashed Gaussian distribution described in Soft Actor-Critic [28].
- The *adversarial actor* is the policy of the opponent. It first encodes the states, the action of the robot, and each prediction mode independently by MLPs. Then we concatenate

the state, action, and each prediction features, which are later aggregated by another MLP. The final MLP processes the aggregated feature and outputs each prediction mode's mean, standard deviation, and probability. We sample the opponent's action using a mixture of the squashed Gaussian distribution to enforce the predictive control bound.

- The *static critic* is a simple MLP return the Q value of the robot **only** considering the road boundary and target set.
- The *interaction critic* returns the *residual* Q value of the interaction between the robot and the opponent. It first encodes the states, the action of both actors, and each prediction mode independently by MLPs. Then we concatenate the state, action, and each prediction features, and generate the aggregated feature of each mode through an MLP. The final MLP processes this feature and outputs Q values for each prediction mode.

Unlike the standard ISAACS procedure, the *ego actor* and *static critic* are first trained by ignoring the collision with the opponent. Then we train *ego actor*, *static critic*, and *adversarial actor* jointly through domain randomization by randomly sampling the initial states of both actors and the opponent's action from its predictive control bound. In this process, We take the largest Q value from *adversarial actor* among all modes, activate it through the SoftPlus function, and add it to the output from the *static critic* as the final Q value for the robot. In this way, the resulting Q value is strictly equal to or larger than that from the *static critic* as the robot will lose the game regardless of the opponent's state when violating the road constraint. Through our experiment, we found that pre-training the *ego actor* and *static critic* is necessary to stabilize learning process.

B.3 Additional Results

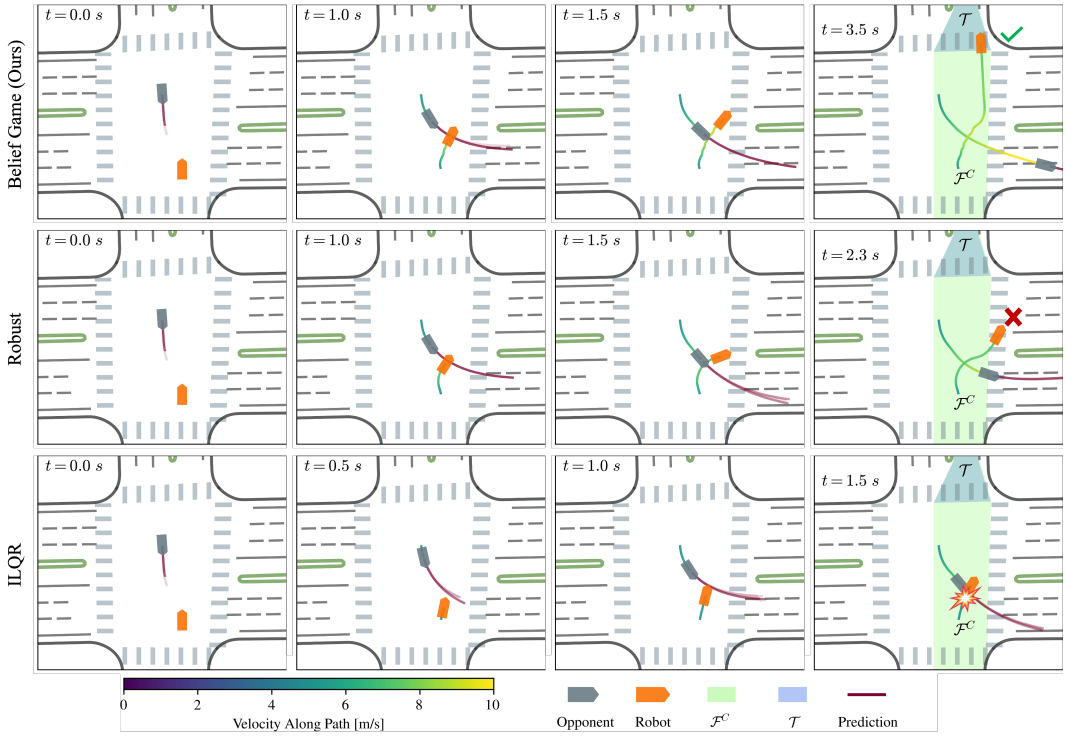

Figure 7: The opponent made an unprotected left turn when the oncoming robot entered the intersection. Robots using Belief Game (top) safely reached the target $\mathcal{T}$ by taking a proactive action even when the opponent was predicted to yield. The Robust Policy (middle) overreacted to the opponent's action, violated the road boundary constraints, and entered its failure set $\mathcal{F}$. The ILQR policy (bottom) was overly-optimistic about the prediction and caused a collision

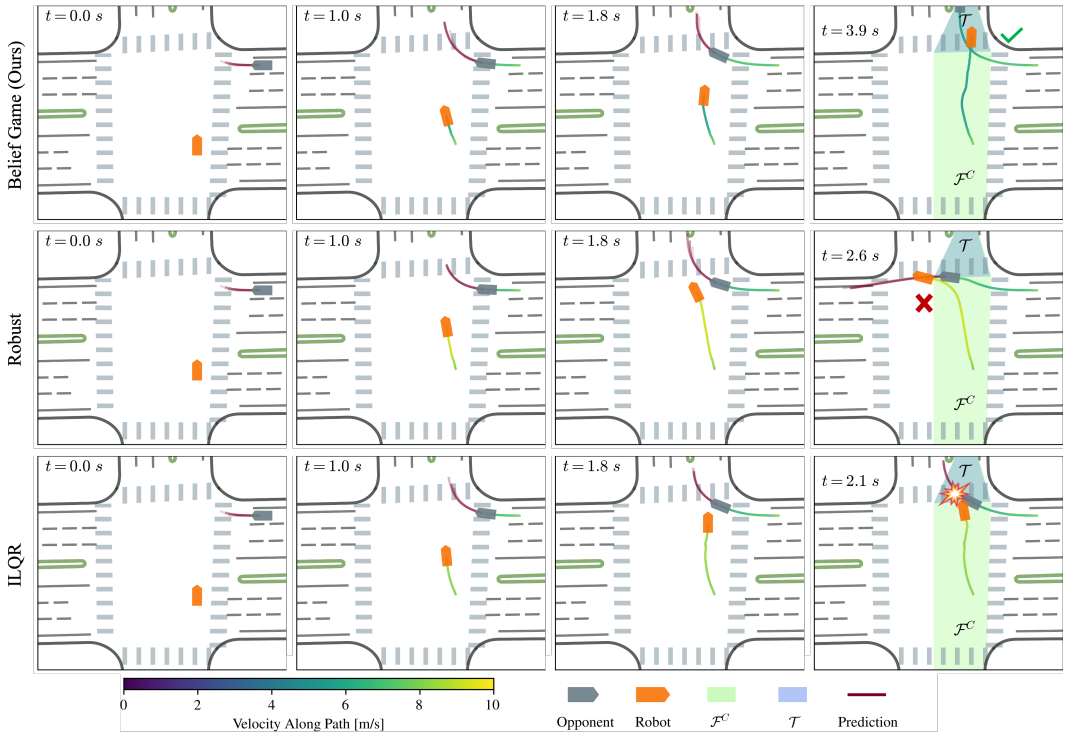

Figure 8: The robot interacted with the merging opponent. Robots using Belief Game (top) safely reached the target $\mathcal{T}$ by yielding to the opponent. The Robust Policy (middle) overreacted to the opponent's action, violated the road boundary constraints, and entered its failure set $\mathcal{F}$. The ILQR policy (bottom) was overly-optimistic about the prediction and causes a collision