Robots that Learn to Safely Influence via Prediction-Informed Reach-Avoid Dynamic Games

Ravi Pandya, Changliu Liu, Andrea Bajcsy

***Abstract*— Robots can influence people to accomplish their tasks more efficiently: autonomous cars can inch forward at an intersection to pass through, and tabletop manipulators can go for an object on the table first. However, a robot's ability to influence can also compromise the physical safety of nearby people if naively executed. In this work, we pose and solve a novel robust reach-avoid dynamic game which enables robots to be maximally influential, but only when a safety backup control exists. On the human side, we model the human's behavior as goal-driven but conditioned on the robot's plan, enabling us to capture influence. On the robot side, we solve the dynamic game in the joint physical and belief space, enabling the robot to reason about how its uncertainty in human behavior will evolve over time. We instantiate our method, called SLIDE (Safely Leveraging Influence in Dynamic Environments), in a high-dimensional (39-D) simulated human-robot collaborative manipulation task solved via offline game-theoretic reinforcement learning. We compare our approach to a robust baseline that treats the human as a worst-case adversary, a safety controller that does not explicitly reason about influence, and an energy-function-based safety shield. We find that SLIDE consistently enables the robot to leverage the influence it has on the human when it is safe to do so, ultimately allowing the robot to be less conservative while still ensuring a high safety rate during task execution. Project website: `https://cmu-intentlab.github.io/safe-influence/`**

I. Introduction

Whether intentional or not, influence underlies many multi-agent interactions, from nudging into someone's lane while driving to merge faster, to grabbing your favorite bottle first so that your partner has to get a different one (Fig. 1, top right). While exploiting such influence can enable agents like robots to be more efficient, it can also lead to unsafe outcomes: if you quickly reach for your favorite mug but your partner doesn't adapt fast enough or is unwilling to change, then you can cause a collision (Fig. 1, bottom left).

In this work, we seek to enable robots to *safely* influence during human-robot interactions. However, we face two challenges, one from the human modeling perspective and the other from the robot control perspective. On one hand, it is difficult to hand-design a model that captures the complexity of how people can be influenced by the robot's behavior. On the other hand, the robot actions that are maximally influential are also often those than can lead to states of irrecoverable failure where *no* safe robot action exists.

To tackle this complexity, we pose a novel robust reach-avoid dynamic game between the human and robot. First, we take inspiration from data-driven trajectory forecasting [1] and inform the human's behavior in the dynamic game via a deep conditional behavior prediction (CBP) model. With CBPs, the robot can learn implicit patterns in the responses of the human *conditioned* on other agents' future behavior. Second, we solve the reach-avoid game in the joint physical and robot belief space. This enables the robot to reach its goal while staying robust to uncertainty over the human's future behavior, instead of always trusting what the conditional model predicts for a short horizon. Finally, to solve this high-dimensional game offline, we adopt approximate reach-avoid reinforcement learning solvers [2] that have recently shown promise in scaling to high-dimensional systems [3], [4].

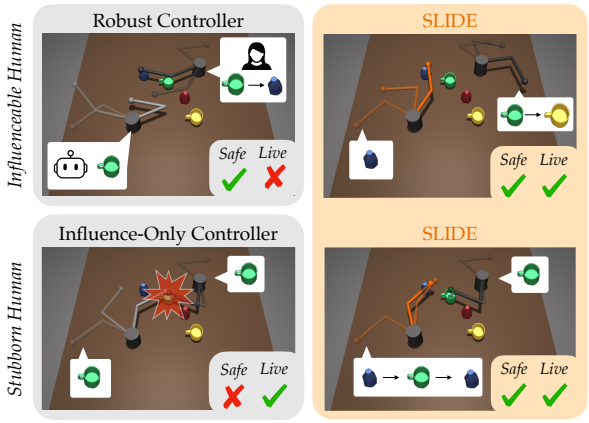

Fig. 1: Both human and robot arms want to reach their desired objects on the table, but they don't know who is going for which object. **Top Row**: The human's desired object can be influenced by the robot. Using an influence-*unaware* safety shield, the robot stays safe, but fails to reach its own object (not *live*). With our method (SLIDE), the robot influences the human's goal and safely reaches its object. **Bottom Row:** The human never changes their desired object. Naive influence-aware robot controllers are over-confident and collide (not *safe*). SLIDE recognizes that this can be unsafe and chooses a different goal for the robot, staying safe and live.

With our framework, called **SLIDE** (Safely Leveraging Influence in Dynamic Environments), we can compute robot policies that exploit influence to maximize efficiency (i.e., liveness) while staying robust to uncertainty and minimizing safety violations (right, Fig. 1). Through extensive simulations in a 39-dimensional human-robot collaborative manipulation scenario (Sec. V), we show that SLIDE is less conservative than prior safe control approaches while staying safe even in the presence of out-of-distribution human behavior.

II. Related Work

Application: Safe Collaborative Manipulation. We ground our approach in human-robot collaborative manipulation

Authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania, {rapandya, cliu6, abajcsy}@andrew.cmu.edu

tasks, which are common in industrial manufacturing [5] and will become common in home environments (e.g. cooking and cleaning tasks) as robotic assistants grow more popular [6]. Ensuring *safety* is critical in these domains [7], has been studied by prior works [8], [9], [10], [11], notably via energy-function-based safe control (e.g. Control Barrier Functions [12] and the Safe Set Algorithm [13]). We are motivated by this domain, but focus specifically on the safety challenges stemming from *influence* in collaborative manipulation.

Modeling Human-Robot Influence. While many works have studied robot influence on humans via expressions or appearance [14], [15], [16], [17], we focus on physical action. One line of prior work has modeled influence *implicitly* by learning the dynamics of a latent representation of the collaborator's strategy [18], [19], [20]. Notably, these methods do not consider influence within a single interaction, only between interactions. Other work has considered modeling influence *explicitly* by predicting the future actions of other agents in driving scenarios using learned models [1], [21], [22] or stackelberg games [23], [24], [25], [26] and in collaborative manipulation scenarios [27], [28], [29]. In contrast, we focus specifically on the *safety* problems that emerge when robots take influential actions.

Embedding Human Models in Safe Robot Control. While foundational works treat the human as a disturbance [30], recent works embed predictive human models into safe controllers to reduce their conservativeness. This has been done by adapting a dynamics and uncertainty model of the human online [31], [11], [32], [33], or via limiting the forward reachable sets of a human based on predictions [34], [35], [36], [37]. We instead consider a *backward* reachability approach, similar to [38], [3] so that the robot can explicitly reason about what it can do to prevent unsafe outcomes, thereby further reducing conservativeness. However, we are focused on collaborative manipulation and leverage data-driven conditional behavior prediction to capture robot influence on people.

III. Background: Reach-Avoid Dynamic Games

Our method is rooted in reach-avoid dynamic games [30]. While there are several ways to solve these games, we leverage Hamilton-Jacobi (HJ) reachability analysis [39] which is a safe control technique compatible with general nonlinear systems, control constraints and disturbances, and is associated with a suite of numerical synthesis techniques [40], [41], [42]. Here we provide a brief overview of HJ reachability (see [43] for a review).

Human & Robot Dynamics. We model the human and robot as the two players in the dynamic game. Let the robot state be denoted by $x^t_{\mathcal{R}} \in \mathcal{S}_{\mathcal{R}}$ and the human as $x^t_{\mathcal{H}} \in \mathcal{S}_{\mathcal{H}}$. The human-robot system state is $x^t = (x^t_{\mathcal{R}}, x^t_{\mathcal{H}}) \in \mathcal{S}$ and evolves via the deterministic discrete-time dynamics $x^{t+1} = f(x^t, u^t_{\mathcal{R}}, u^t_{\mathcal{H}})$ where the robot and human control inputs are denoted by $u^t_{\mathcal{R}} \in \mathcal{U}_{\mathcal{R}}$ and $u^t_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}$ respectively.

Reach-Avoid Games via HJ Reachability. HJ reachability computes a backward reachable tube (BRT), $\mathcal{S}^* \subset \mathcal{S}$, which characterizes the set of initial states from which the robot is guaranteed to reach a desired target set while also avoiding a set of failure states, despite the best effort of an adversary. It also synthesizes a corresponding reach-avoid robot policy, $\pi^*_{\mathcal{R}}$. Let the target set $\mathcal{T} := \{x \mid l(x) \leq 0\}$ and failure set $\mathcal{F} := \{x \mid g(x) > 0\}$ be encoded via the Lipschitz-continuous margin functions $l(\cdot)$ and $g(\cdot)$. For robustness, the reach-avoid game models the robot as attempting to stay safe while reaching the goal, and the human as a virtual adversary who attempts to thwart this. Solving the game amounts to computing the value function characterized by the fixed-point Isaacs equation [44]:

$$V(x) = \max\Big\{g(x), \min\Big\{l(x), \min_{u_{\mathcal{R}} \in \mathcal{U}_{\mathcal{R}}} \max_{u_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}} V(x^+)\Big\}\Big\}, \tag{1}$$

where $x^+ = f(x, u_{\mathcal{R}}, u_{\mathcal{H}})$ is the next state. The sub-zero level set of the value function $\mathcal{S}^* = \{x \mid V(x) \leq 0\}$ encodes our desired set of states (i.e., BRT) from which there exists a robot control signal that can reach the target set without ever entering failure, despite a worst-case adversary. The corresponding optimal robot policy $\pi^*_{\mathcal{R}}$ can be obtained via:

$$\pi^*_{\mathcal{R}}(x) = \operatorname*{argmin}_{u_{\mathcal{R}} \in \mathcal{U}_{\mathcal{R}}} \max_{u_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}} V(x^+). \tag{2}$$

IV. SLIDE: Safely Leveraging Influence in Dynamic Environments

The reach-avoid formulation from Section III establishes the backbone of our approach. However, applying it directly to influencing human-robot interactions would face two challenges. First, the model of the human is far too pessimistic by treating them as a best-effort adversary (thus making $\pi^*_{\mathcal{R}}$ overly conservative). Second, it assumes that the robot can never learn about the human during interaction and thus can never decrease (or increase) its uncertainty about their future behavior. To extend this mathematical framework to safely account for human-robot influence, we introduce two key modifications which we detail below: (1) a conditional behavior prediction model of human influence, and (2) a belief-space formulation of reach-avoid games.

Modeling Influence via Conditional Behavior Prediction. Let $\mathbf{u}_{\mathcal{H}} := [u^t_{\mathcal{H}}, \ldots, u^{t+k}_{\mathcal{H}}]^\top$ be a trajectory of human actions for a horizon of length k and $\mathbf{u}_{\mathcal{R}} := [u^t_{\mathcal{R}}, \ldots, u^{t+k}_{\mathcal{R}}]^\top$ represent the same for the robot. Our approach leverages a pre-trained conditional behavior prediction (CBP) model which outputs a multimodal distribution over human action trajectories $\mathbf{u}_{\mathcal{H}}$ conditioned on the robot's future plan $\mathbf{u}_{\mathcal{R}}$: $\mathcal{P} := P(\mathbf{u}_{\mathcal{H}} \mid x^t, \mathbf{u}_{\mathcal{R}}; \theta)$ where $\theta \in \Theta$ represents the mode of the distribution. Our approach is agnostic to the particular form of the CBP model, only requiring that the model outputs M discrete modes (i.e. $|\Theta| = M$) and their associated probabilities p_θ. Note that learning a Gaussian Mixture Model (which adheres to this form) is already common among existing trajectory forecasting models [1], [45], [46]. The belief of the robot then is the M modes and their associated probabilities: $b^t_{\mathcal{R}} := \{(\theta_i, p_{\theta_i})\}_{i=1}^M$.

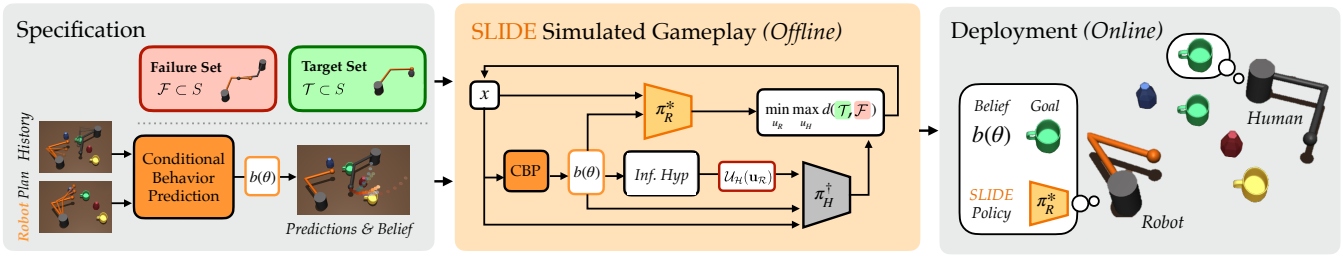

Fig. 2: SLIDE Framework. (left) Before solving the reach-avoid game, we specify the target set (goal locations), failure set (collisions), and a conditional behavior prediction (CBP) model that can predict the human's future trajectory conditioned on the robot's future plan. (center) During simulated gameplay, the SLIDE policy, $\pi^*_{\mathcal{R}}(x_e)$, is trained against a simulated human adversary, $\pi^\dagger_{\mathcal{H}}(x_e)$ whose control bounds are informed by the CBP model. (right) Online, the robot uses its robust SLIDE policy to safely influence against *any* human.

Our core idea is that the CBP model enables us to inform the actions we expect the human to take within the reach-avoid game (i.e., $\mathcal{U}_{\mathcal{H}}$ in Eqn. 1) while also capturing the influence the robot has over this action set. Specifically, for each predicted behavior mode $\theta \in \Theta$, we construct the set of $\delta-$likely trajectories under the predicted distribution $\mathbb{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}}; \theta) := \{\mathbf{u}_{\mathcal{H}} \mid P(\mathbf{u}_{\mathcal{H}} \mid x^t, \mathbf{u}_{\mathcal{R}}; \theta) \geq \delta\}$. We construct our *influence-informed control bounds* for the human by taking the time-wise minimum and maximum actions:

$$\mathcal{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}}; \theta) := [\min_{t \in \{t, \dots t+k\}} \mathbb{U}_{\mathcal{H}}, \max_{t \in \{t, \dots t+k\}} \mathbb{U}_{\mathcal{H}}]. \quad (3)$$

Similar to [3], we have an *inference hypothesis*: at each time t, the robot's belief $b^t_{\mathcal{R}}$ will assign at least probability ϵ to the human's next action $u^t_{\mathcal{H}}$. Mathematically, we assume the human's action will belong to the inferred control bound during reach-avoid analysis:

$$\mathcal{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}}) := \bigcup_{\theta_i \in \{\theta \mid b^t_{\mathcal{R}}(\theta) \geq \epsilon\}} \mathcal{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}}; \theta_i). \quad (4)$$

Here, $\epsilon \geq 0$ is a hyperparameter that captures the reliability of the trajectory forecasting model. Now that we have a influence-aware human model we can inject into our reach-avoid game, we have to contend with the challenge that the robot's belief—and therefore Eqn. 4—changes over time.

Belief-Space Reach-Avoid Games. To enable the robot to account for uncertainty in the human's future behavior, we modify the reach-avoid problem from Eqn. (1) by *extending* the state to include both the physical state (x) and the robot's *belief state* ($b_{\mathcal{R}}(\theta)$). Here, the robot's belief state is precisely the Gaussian mixture model over possible behavior modes θ output by the CBP. Note that at each timestep the robot re-generates its predictions of the human behavior given new observations. This induces the belief-space dynamics:

$$b^{t+1}_{\mathcal{R}} = f_L(b^t_{\mathcal{R}}, x^t, u_{\mathcal{R}}, u_{\mathcal{H}}). \quad (5)$$

Note that in this work, f_L is implicitly modeled by subsequent calls to the forecasting model during reach-avoid reinforcement learning.

Finally, let the extended state be $x^t_e = (x^t, b^t_{\mathcal{R}})$ and the extended physical-belief dynamics to be $x^{t+1}_e = F(x^t_e, u_{\mathcal{R}}, u_{\mathcal{H}})$. Putting our model from Eqn. 4, we obtain a modified fixed-point Isaacs equation:

$$V(x_e) = \max\left\{g(x_e), \min\left\{l(x_e), \min_{u_{\mathcal{R}}} \max_{u_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}})} V(x^+_e)\right\}\right\}. \quad (6)$$

where $x^+_e = F(x_e, u_{\mathcal{R}}, u_{\mathcal{H}})$ is the next joint physical and belief state. The differences from the standard Isaacs equation in Eqn. 1 is highlighted in orange: this is the inclusion of the extended state and the adversary's control bound being a function of the ego's nominal long-term plan.

***Offline:* Reach-Avoid Reinforcement Learning Solution.** Our problem in Eqn. (6) quickly becomes computationally intractable with traditional level-set methods [40], [47] due to the dimensionality of the extended state. However, we build on the recent ISAACS reach-avoid reinforcement-learning based solver [2] to solve a time-discounted version of the safety value function (6). We use a soft actor-critic formulation to train the value function critic $V(\cdot)$ and two actor networks, $\pi^*_{\mathcal{R}}(\cdot)$ and $\pi^\dagger_{\mathcal{H}}(\cdot)$, representing the optimal robot and simulated human adversary control policies. To enforce our inference hypothesis in Eqn. (4), we project the actions of the adversary agent's policy to be within the inferred control bound $\mathcal{U}_{\mathcal{H}}(\mathbf{u}_{\mathcal{R}})$. At each step during training, the robot's future plan $\mathbf{u}_{\mathcal{R}}$ is computed by querying a nominal policy to drive the robot towards the closest goal.

***Online*: Safe and Influencing Control.** Online, we deploy the trained reach-avoid policy $\pi^*_{\mathcal{R}}(x_e)$ directly via solving Eqn. (2). We input the current state x and query the trajectory prediction model for the belief state $b_{\mathcal{R}}$ at each timestep.

V. Experimental Setup

Task: Tabletop Object Reaching. We deploy our framework in a scenario where a human and robot arm need to reach their desired objects on the table, but they do not know who is going for what object. They must choose objects without colliding into each other. We instantiate four objects on the table, two mugs and two bottles (shown in Fig. 1) denoted by $g_i \in \mathcal{G}$ in Cartesian space (i.e. end-effector goals) and via discrete semantic class, $c : \mathcal{G} \to \mathbb{W}$.

Human-Robot System Dynamics. We model the human and robot as 2-link planar manipulators (i.e. no gravity) and each agent $i \in \{\mathcal{H}, \mathcal{R}\}$'s dynamics are:

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i = B_i u_i, \quad (7)$$

where the state $q_i \in \mathcal{Q}_i$ consists of the joint angles, M_i is the inertia matrix, C_i captures Coriolis forces, and B_i represents how the control input affects the system. The agents' actions $u_i \in \mathcal{U}_i$ are the joint torques bounded by a box constraint: $\mathcal{U}_i := \{u \mid u_i^{min} \leq u \leq u_i^{max}\}$. We model the robot as having larger control authority than the human. The configuration space states q_i are transformed by an observation map $\mathcal{O}_i : \mathcal{Q}_i \to \mathbb{R}^{n_i}$ where n_i is the dimensionality of the observation for agent i. The physical state input to the model is the concatenation of observations of both agents: $\mathcal{S} := \mathcal{O}_\mathcal{R}(\mathcal{Q}_\mathcal{R}) \times \mathcal{O}_\mathcal{H}(\mathcal{Q}_\mathcal{H})$.

Multi-Arm Interaction Data Generation. To create our conditional behavior prediction model, we need a dataset of multi-agent interactions to learn from. Unlike in autonomous driving, where many large, diverse, multi-agent datasets already exist [48], [49], [50], these types of large datasets are underrepresented in collaborative manipulation. We thus choose to create our own synthetic dataset of multi-agent interactions so that we can control how the other agent is influenced and can cleanly analyze the effect of our influence-aware safe control policy. Our simulated human always tries to choose an object with a different type than the robot. We simulate this by having the human keep their own belief over the goal that robot is currently choosing, $b^t_\mathcal{H}(g) \propto P(u^t_\mathcal{R} \mid q^t_\mathcal{R}, g) b^{t-1}_\mathcal{H}(g)$, which gets updated online via Bayes' Rule. The human changes their goal to be the least likely robot goal based on $b^t_\mathcal{H}$ if the robot's most likely goal both has probability > 0.3 and has the same semantic class as the human's goal. The dataset is generated by randomly sampling initial goals for the two agents and rolling out for fixed horizon of 15 seconds.

Methods. We compare two human prediction models: marginal and conditional (CBP). We also compare our method, **SLIDE**, with four other robot policies: **NoSafety** (computed torque control), the Safe Set Algorithm (**SSA-Basic**) [13], Robust reach-avoid (**Robust-RA**) [2], and Marginal reach-avoid (**Marginal-RA**) [3]. We use a naive version of SSA to understand the effect of adding simple safe control. The robust policy solves the same reachability problem but without including a prediction model, and the marginal policy has the same structure as SLIDE but its prediction model is not conditioned on the robot's future plan.

Metrics. For trajectory prediction models, we measure the average displacement error (ADE), final displacement error (FDE), and size of the inferred control bound $|\mathcal{U}_\mathcal{H}|$. For closed-loop simulations of the agents, we measure: 1) collision rate (safety) 2) task completion rate (liveness) and 3) completion time (average trajectory length).

Training: Prediction Models for Manipulation. The prediction models are both 3-layer MLPs[1] with hidden sizes of 256 neurons and output the parameters of a Gaussian Mixture Model (GMM) to predict the human's actions for the next 1 second. Both models' input consists of a 1-second history of both agents' end effector positions, the goal positions Θ, and the goals' semantic classes $c(\theta_i)$ as a flattened vector. The CBP model additionally takes in a 2-second future plan of the robot's end effector position. The models are trained with the sum of two loss function terms (similar to [1]): 1) negative log-likelihood and 2) MSE of most-likely GMM mode.

[1]We found this simple architecture to be sufficient for our task, but our method is agnostic to the complexity of the model.

Training: Safety Value Function. The actor and safety critic networks are 4-layer MLPs with a hidden size of 256 and the actor outputs a Guassian distribution for the action of the robot. The input to the networks consists of the 24D observed state (plus 15D extended belief state for marginal and SLIDE methods). The belief state consists of the mode means and mixture weights of the trajectory predictor's GMM output. The policies are trained using one NVIDIA GeForce RTX 4090 GPU and the RL environment is run on 16 parallel threads on an AMD Ryzen Threadripper 7960X CPU. The wall-clock time for training the robust policy is approximately 5 hours, the marginal policy is approximately 28 hours, and SLIDE is 61 hours. This includes time for pretraining the ego and adversary agents, following the training scheme laid out by [2].

VI. Experimental Results

A. Influence-Aware vs. Influence-Unaware Safety

We first study the closed-loop performance of **SLIDE**'s controller, which safely exploits influence, compared to alternative safe controllers described in Sec. V.

	Collision rate	Completion rate	Completion Time (s)
NoSafety	28.5%	71.5%	3.5 ± 1.8
SSA-Basic	19.1%	52.3%	8.9 ± 4.7
Robust-RA	1.4%	97.0%	2.6 ± 2.1
Marginal-RA	1.5%	98.0%	2.5 ± 1.3
SLIDE (ours)	1.9%	98.1%	1.9 ± 0.8

TABLE I: Closed-loop Results. Failure rate, completion rate and completion time for all methods over 1,000 randomized trials. The task is incomplete if the robot does not reach any goal within 15 seconds. **SLIDE** is able to reach its goal most often and in the shortest time without becoming significantly less safe.

Results: Quantitative. Table I shows all methods' performance. As expected, **Robust-RA** has the smallest collision rate and all safe controllers have a lower collision rate than **NoSafety**. However, what **Robust-RA** and **Marginal-RA** gain in safety, they lose in completion rate and time. In contrast, **SLIDE** has the highest completion (liveness) rate while having a comparable collision rate to the robust and marginal baselines. Moreover, **SLIDE** allows the robot to reach goals significantly faster: e.g., 24% faster than **Marginal-RA**. We further plot the histogram of completion times in Fig. 4. We see that **SLIDE** almost never times out, while other methods do. We note that **SSA-Basic** collides and times out the most out of the safe controllers. The reason for this is because the safety index for **SSA-Basic** was hand-tuned (as proposed in [13]), so it does not always respect

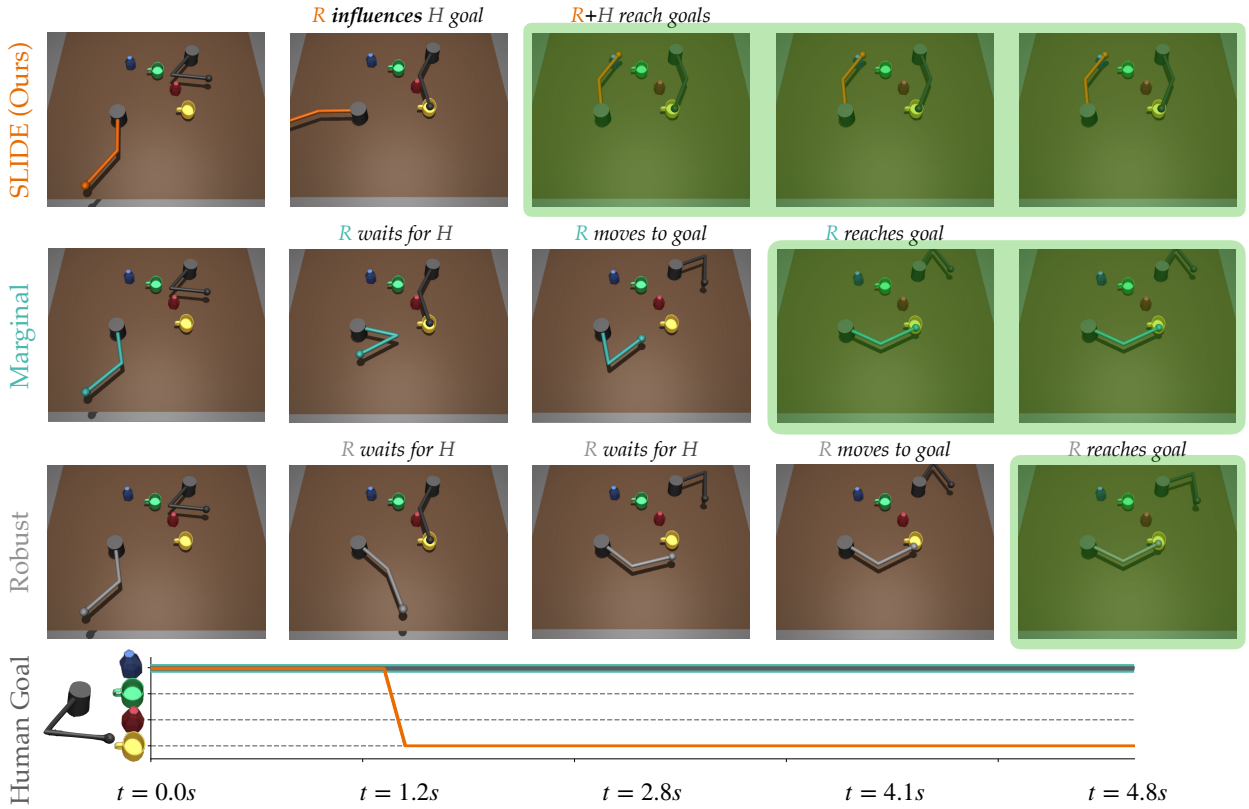

Fig. 3: Closed-Loop Simulations. SLIDE, Marginal-RA and Robust-RA policies starting from the same initial condition. SLIDE confidently understands that the human will be influenced to move out of its way as it chooses the blue bottle and reaches the fastest (the human changes its mind from the blue bottle to the yellow mug at $t = 1.2s$). Marginal-RA waits until the human is out of its way and chooses the yellow mug. Robust-RA stays cautious even as the human is moving towards a different goal and finishes last.

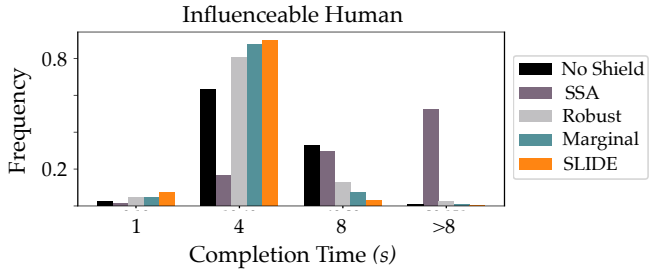

Fig. 4: Closed-loop Completion Times. Histogram of completion times for all methods interacting with the influenceable human model. SLIDE has the highest frequency of short trajectories, while SSA-Basic and Robust-RA have the highest incidence of timing out.

the system's control limits. If we used some additional safety index synthesis techniques [51] to properly account for the control bounds, the performance would likely approach that of Robust-RA, though without the goal-reaching policy built-in.

Results: Qualitative. We visualize SLIDE and Robust-RA's closed-loop trajectories in Fig. 3. Robust-RA initially keeps the robot arm far away from all goals, waiting for the human to move out of the way before reaching for a mug. In contrast, SLIDE immediately recognizes that it can influence the human's target object (bottom, Fig. 3). It switches to reach the bottle, ensuring the human picks a cup, stays out of its way, and enables the robot to complete the task faster.

B. Ablation: When Does Modeling Influence Matter?

Next, we study when it matters that we use influence-aware human models for safe control. We ablate if the robot uses a *conditional* or *marginal* prediction model (i.e. no conditioning on robot's future plan). We compare both the performance of the predictors and their effect on the learned reach-avoid policies.

	All Data (14,000)	Interactive Data (2,457)	Non-Interactive Data (11,543)
Marginal	0.002 (0.12)	0.007 (0.40)	0.001 (0.07)
CBP	0.001 (**0.09**)	0.007 (**0.30**)	0.0007 (**0.04**)

TABLE II: Open-Loop Prediction Error. Average (ADE) and Final Displacement Error (FDE) of marginal and CBP predictors.

	All Data (14,000)	Interactive Data (2,457)	Non-Interactive Data (11,543)
Marginal	[12.3, 12.7]	[13.8, 13.6]	[11.9, 12.5]
CBP	[7.8, 8.3]	[12.7, 12.9]	[6.7, 7.3]

TABLE III: Inferred $\mathcal{U}_\mathcal{H}$ Size. Average inferred control bound size from Marginal-RA ($|\mathcal{U}_\mathcal{H}|$) and SLIDE ($|\mathcal{U}_\mathcal{H}(\mathbf{u}_\mathcal{R})|$). Entries shown per ctrl. dimension and the max. dyn. feasible range is 20.

Approach. In these experiments, the human agent acts according to the data distribution in Sec. V. We evaluate

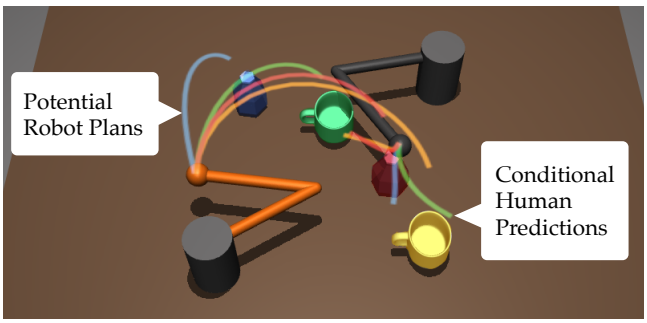

Fig. 5: Conditional Behavior Predictions. Most-likely mode of SLIDE CBP model given different future robot plans. Each robot plan has a corresponding human prediction in the same color. The prediction is highly dependent on the robot's plan and captures the idea that the human will change goals to a different semantic class.

the predictors on a dataset $\mathcal{D}$ of 100 held-out trajectories, each 15 seconds long, resulting in 14,000 data points[2].

Open-Loop Results: Quantitative & Qualitative. Table II shows ADE and FDE results. Averaged across the held-out dataset, ADE looks very similar for both marginal and CBP, with CBP performing better on FDE (left, Table II). However, only a *subset* of all trajectories in this held-out dataset exhibit highly interactive scenarios. Thus, we further decompose dataset $\mathcal{D} = \mathcal{D}_I \cup \mathcal{D}_{\neg I}$ into data points where influence is happening $\mathcal{D}_I$ (2,457 data points) and those where it is not $\mathcal{D}_{\neg I}$ (11,543 data points). This is done by tracking the timesteps where the human's goal changes: if it does, then this and directly adjacent timesteps are added to $\mathcal{D}_I$. For data in $\mathcal{D}_I$, the CBP model has a significantly lower FDE compared to the marginal trajectory predictor (center column, Table II), indicating that the CBP matters to predict long-term behavior in highly interactive scenarios. Fig. 5 shows the CBP model that SLIDE uses as a function of the robot's plan. Each robot plan results in distinct human predictions. This capability (not present in a marginal prediction model) gives the SLIDE policy the freedom to learn which goals to move towards to best influence the human to move out of its way. We further study the implications of each model on the inferred control bounds, $\mathcal{U}_{\mathcal{H}}$, that we use during reach-avoid computations (Sec. IV). This will directly tell us how the prediction model will affect the policy training loop, since this determines the set of allowable actions for the adversary agent. In Table III we show the size of the inferred control bounds. The CBP action bounds enlarge at interactive states, but do not expand as much as the marginal predictor's bounds. On the full dataset, the CBP model results in a smaller control bound on average. This implies that SLIDE's downstream policy (which uses the CBP) will be able to exploit the human's influence and thus choose less conservative actions.

Closed-Loop Results: Quantitative & Qualitative. Finally, we study the closed-loop safety and liveness performance of of the SLIDE policy and the Marginal-RA policy. In Table I, we see that both methods have a similar task completion rate (SLIDE is 98.1% while Marginal-RA is 98.0%). However, the main difference comes in their completion *time*: SLIDE completes the task in $1.9s$ compared to Marginal-RA at $2.5s$. This implies that our SLIDE policy exploits the influence it has on the human to maximize task completion efficiency. Qualitatively, we see that SLIDE chooses the top (blue) bottle (top row, Fig. 3) to influence the human into picking to the bottom (yellow) mug, making the two agents stay out of each others' way. In contrast, the Marginal-RA policy chooses the goal closest to itself (middle row, Fig. 3), but then must wait for the human to move out of the way before reaching the yellow mug.

[2]We discount predictions in the last 1 second of each trajectory since this is length of the prediction horizon.

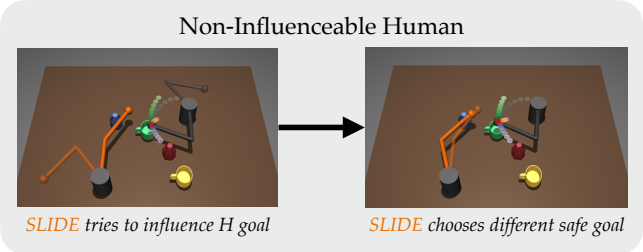

Fig. 6: Shows an interaction between the SLIDE controller and a stubborn (non-influenceable) human model. The modes of the GMM output by the CBP prediction model are visualized as dots from the human's end effector.

	Collision rate	Completion rate	Completion Time (s)
Influenceable Human	1.9%	98.1%	1.9 ± 0.8
Stubborn Human (OOD)	1.8%	98.1%	2.0 ± 0.9
Adversarial Human (OOD)	3.8%	96.2%	1.9 ± 0.7

TABLE IV: Closed-Loop OOD Human Results. SLIDE interacts with in-distribution and out-of-distribution (OOD) humans over 1,000 randomized trials. Stubborn human never changes their goal. Adversarial human always chooses the goal it thinks the robot is moving towards. SLIDE is relatively robust to the stubborn human, but starts to degrade against the adversarial human.

C. How Robust is SLIDE to Out-of-Distribution Humans?

Finally, we study how robust the SLIDE policy is when interacting with human behavior that is out-of-distribution (OOD) from the conditional behavior prediction model. We measure the same closed-loop metrics described in Sec. V.

Approach. We ablate the deployment-time human model (right, Fig. 2) to be OOD relative to the prediction model training data. Specifically, we change the human to be **non-influenceable** (also referred to as *stubborn*) and **adversarial** (where the human always chooses the goal it thinks the robot is likely moving towards[3]).

Results: Quantitative & Qualitative. Results are shown in Table IV for 1,000 random initial conditions. We see that when interacting with a stubborn human, SLIDE generally performs similarly to the in-distribution human model. We hypothesize this is precisely because of the *inference hypothesis* from Sec. IV: instead of blindly trusting the output, we

[3]Note that this is different from the optimal disturbance policy $\pi^{\dagger}_{\mathcal{H}}$ trained in simulated gameplay.

allow the simulated opponent agent to take any $\delta-$likely action from $\epsilon-$likely modes. This means as long as the modes of the prediction model have some coverage of the agent's true behavior, the learned policy will still be robust to it. We can see an example of this in Fig. 6. At first, **SLIDE** tries to influence the human to switch to the red bottle goal. After a few timesteps, however, **SLIDE** backs off and chooses a different (safe) goal (the blue bottle) to complete the task. When interacting with the **adversarial human** that always chooses the goal it thinks the robot is moving towards, we can see that **SLIDE** starts to collide more often (bottom row, Table IV). This makes sense since the human sometimes takes actions that are in direct opposition to the predictor's training data, which our inference hypothesis is unable to account for. However, it is a promising sign that there is not a catastrophic degradation in the collision rate. Future work should investigate, for example, the use of a *hybrid* approach which switches between the **Robust-RA** controller and **SLIDE** depending on the prediction quality observed during deployment.

VII. Conclusion

In this work, we enable robots to safely influence humans. We pose and solve a new reach-avoid dynamic game (called SLIDE) that (1) accounts for influence via the use of a data-driven conditional behavior prediction model and (2) accounts for the robot's ability to learn online via a belief-state. In simulations, we find that SLIDE can accomplish tasks significantly faster than prior safe control approaches, and remains relatively robust in the face of out-of-distribution human interactions. Future work should investigate calibrating the prediction model [52] as well as performing post-verification on the approximate reach-avoid value function [53], [54] for stronger safety assurances.

Acknowledgment

This work is supported by NSF Award #2246447. RP is also funded by the NSF GRFP.

References

[1] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3473–3479.

[2] K.-C. Hsu, D. P. Nguyen, and J. F. Fisac, "Isaacs: Iterative soft adversarial actor-critic for safety," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 90–103.

[3] H. Hu, Z. Zhang, K. Nakamura, A. Bajcsy, and J. F. Fisac, "Deception game: Closing the safety-learning loop in interactive robot autonomy," in *7th Annual Conference on Robot Learning*, 2023.

[4] D. P. Nguyen, K.-C. Hsu, J. F. Fisac, J. Tan, and W. Yu, "Gameplay filters: Robust zero-shot safety through adversarial imagination," in *8th Annual Conference on Robot Learning*, 2024.

[5] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran, "A survey of robot learning strategies for human-robot collaboration in industrial settings," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102231, 2022.

[6] A. Abou Allaban, M. Wang, and T. Padır, "A systematic review of robotics research in support of in-home care for older adults," *Information*, vol. 11, no. 2, p. 75, 2020.

[7] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[8] C. Liu and M. Tomizuka, "Algorithmic safety measures for intelligent industrial co-robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3095–3102.

[9] A. Singletary, S. Kolathaya, and A. D. Ames, "Safety-critical kinematic control of robotic systems," *IEEE Control Systems Letters*, vol. 6, pp. 139–144, 2021.

[10] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2565–2570.

[11] R. Liu, R. Chen, and C. Liu, "Safe interactive industrial robots using jerk-based safe set algorithm," in *Proceedings of the International Symposium on Flexible Automation 2022 International Symposium on Flexible Automation*. The Institute of Systems, Control and Information Engineers, 2022, pp. 196–203.

[12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[13] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *Dynamic Systems and Control Conference*, vol. 46209. American Society of Mechanical Engineers, 2014, p. V003T42A003.

[14] S. Saunderson and G. Nejat, "How robots influence humans: A survey of nonverbal communication in social human–robot interaction," *International Journal of Social Robotics*, vol. 11, no. 4, pp. 575–608, 2019.

[15] M. Siegel, C. Breazeal, and M. I. Norton, "Persuasive robotics: The influence of robot gender on human behavior," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2563–2568.

[16] I. Rae, L. Takayama, and B. Mutlu, "The influence of height in robot-mediated communication," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2013, pp. 1–8.

[17] H. Admoni and B. Scassellati, "Social eye gaze in human-robot interaction: a review," *Journal of Human-Robot Interaction*, vol. 6, no. 1, pp. 25–63, 2017.

[18] A. Xie, D. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," in *Conference on robot learning*. PMLR, 2021, pp. 575–588.

[19] W. Z. Wang, A. Shih, A. Xie, and D. Sadigh, "Influencing towards stable multi-agent interactions," in *Conference on robot learning*. PMLR, 2022, pp. 1132–1143.

[20] S. Parekh, S. Habibian, and D. P. Losey, "Rili: Robustly influencing latent intent," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 01–08.

[21] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[22] Z. Huang, H. Liu, J. Wu, and C. Lv, "Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7244–7258, 2023.

[23] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and systems*, vol. 2. Ann Arbor, MI, USA, 2016, pp. 1–9.

[24] S. Sagheb, Y.-J. Mun, N. Ahmadian, B. A. Christie, A. Bajcsy, K. Driggs-Campbell, and D. P. Losey, "Towards robots that influence humans over long-term interaction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7490–7496.

[25] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9590–9596.

[26] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[27] A. Bestick, R. Bajcsy, and A. D. Dragan, "Implicitly assisting humans to choose good grasps in robot to human handovers," in *2016 International Symposium on Experimental Robotics*. Springer, 2017, pp. 341–354.

[28] K. Kedia, A. Bhardwaj, P. Dan, and S. Choudhury, "Interact: Transformer models for human intent prediction conditioned on robot

actions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 621–628.

[29] R. Pandya, Z. Wang, Y. Nakahira, and C. Liu, "Towards proactive safe human-robot collaborations via data-efficient conditional behavior prediction," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 956–12 963.

[30] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proceedings of the 18th international conference on hybrid systems: computation and control*, 2015, pp. 11–20.

[31] C. Liu and M. Tomizuka, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 465–470.

[32] R. Pandya and C. Liu, "Safe and efficient exploration of human models during human-robot interaction," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6708–6715.

[33] R. Pandya, T. Wei, and C. Liu, "Multimodal safe control for human-robot interaction," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 2672–2678.

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

[35] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, "A scalable framework for real-time multi-robot, multi-human collision avoidance," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 936–943.

[36] A. Bajcsy, S. Bansal, E. Ratner, C. J. Tomlin, and A. D. Dragan, "A robust control framework for human motion prediction," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 24–31, 2020.

[37] K. Nakamura and S. Bansal, "Online update of safety assurances using confidence-based predictions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 765–12 771.

[38] R. Tian, L. Sun, A. Bajcsy, M. Tomizuka, and A. D. Dragan, "Safety assurances for human-robot interaction via confidence-aware game-theoretic human models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 229–11 235.

[39] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[40] I. M. Mitchell and J. A. Templeton, "A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *International workshop on hybrid systems: computation and control*. Springer, 2005, pp. 480–494.

[41] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8550–8556.

[42] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1817–1824.

[43] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[44] R. Isaacs, "Differential games i: Introduction," *Technical Report*, 1954.

[45] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

[46] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[47] M. Bui, G. Giovanis, M. Chen, and A. Shriraman, "Optimizeddp: An efficient, user-friendly library for optimal control and dynamic programming," *arXiv preprint arXiv:2204.05520*, 2022.

[48] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[49] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[50] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[51] W. Zhao, T. He, T. Wei, S. Liu, and C. Liu, "Safety index synthesis via sum-of-squares programming," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 732–737.

[52] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.

[53] A. Lin and S. Bansal, "Verification of neural reachable tubes via scenario optimization and conformal prediction," in *6th Annual Learning for Dynamics & Control Conference*. PMLR, 2024, pp. 719–731.

[54] Y. Yang, H. Hu, T. Wei, S. E. Li, and C. Liu, "Scalable synthesis of formally verified neural value function for hamilton-jacobi reachability analysis," *arXiv preprint arXiv:2407.20532*, 2024.