Auto-Encoding Bayesian Inverse Games

Xinjie Liu[1*][0000−0001−7826−4506], Lasse Peters[2*][0000−0001−9008−7127], Javier Alonso-Mora[2][0000−0003−0058−570X], Ufuk Topcu[1][0000−0003−0819−9985], and David Fridovich-Keil[1][0000−0002−5866−6441]

[1] The University of Texas at Austin, Austin, TX 78712, USA
{xinjie-liu,utopcu,dfk}@utexas.edu
[2] Delft University of Technology, Delft, 2600 AA, Netherlands
{l.peters,j.alonsomora}@tudelft.nl

Abstract. When multiple agents interact in a common environment, each agent's actions impact others' future decisions, and noncooperative dynamic games naturally capture this coupling. In interactive motion planning, however, agents typically do not have access to a complete model of the game, e.g., due to unknown objectives of other players. Therefore, we consider the *inverse* game problem, in which some properties of the game are unknown a priori and must be inferred from observations. Existing maximum likelihood estimation (MLE) approaches to solve inverse games provide only point estimates of unknown parameters without quantifying uncertainty, and perform poorly when many parameter values explain the observed behavior. To address these limitations, we take a Bayesian perspective and construct *posterior distributions* of game parameters. To render inference tractable, we employ a variational autoencoder (VAE) with an embedded differentiable game solver. This structured VAE can be trained from an unlabeled dataset of observed interactions, naturally handles continuous, multimodal distributions, and supports efficient sampling from the inferred posteriors without computing game solutions at runtime. Extensive evaluations in simulated driving scenarios demonstrate that the proposed approach successfully learns the prior and posterior game parameter distributions, provides more accurate objective estimates than MLE baselines, and facilitates safer and more efficient game-theoretic motion planning.

Keywords: Bayesian Inverse Games · Amortized Inference · Game-Theoretic Motion Planning · Machine Learning in Robotics · Human-Robot Interaction

1 Introduction

Autonomous robots often need to interact with other agents to operate seamlessly in real-world environments. For example, in the scenario depicted in Fig. 1, a robot encounters a human driver while navigating an intersection. In such settings, coupling effects between agents significantly complicate decision-making: if the human acts assertively and drives straight toward its goal, the robot will be forced to brake to avoid a collision. Hence, the agents compete to optimize their individual objectives. *Dynamic noncooperative game theory* [2] provides an expressive framework to model such interaction among rational, self-interested agents.

*Equal contribution

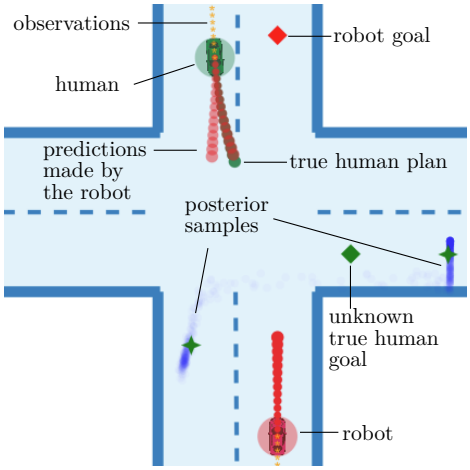


Fig. 1: A robot interacting with a human driver whose goal position is unknown. We embed a differentiable game solver in a structured variational autoencoder to infer the *distribution* of the human's objectives based on observations of their behavior.

In many scenarios, however, a robot must handle interactions under incomplete information, e.g., without knowing the goal position of the human driver in Fig. 1. To this end, a recent line of work [1, 12, 15, 18, 20, 21, 26–28, 35] solves *inverse dynamic games* to infer the unknown parameters of a game—such as other agents' objectives—from observed interactions.

A common approach to solve inverse dynamic games uses maximum likelihood estimation (MLE) to find the most likely parameters given observed behavior [1, 12, 15, 18, 20, 21, 26–28]. However, MLE solutions provide only a point estimate without any uncertainty quantification and can perform poorly in scenarios where many parameter values explain the observations [18], yielding overconfident, unsafe motion plans [25].

Bayesian formulations of inverse games mitigate these limitations of MLE methods by inferring a posterior *distribution*, i.e., belief, over the unknown parameters [17, 24]. Knowledge of this distribution allows the robot to account for uncertainty and generate safer yet efficient plans [11, 17, 25, 30].

Unfortunately, exact Bayesian inference is typically intractable in dynamic games, especially when dynamics are nonlinear. Prior work [17] alleviates this challenge by using an unscented Kalman filter (UKF) for approximate Bayesian inference. However, that approach is limited to unimodal uncertainty models and demands solving multiple games per belief update, thereby posing a computational challenge.

The main contribution of this work is a framework for tractable Bayesian inference of posterior distributions over unknown parameters in dynamic games[3]. To this end, we approximate exact Bayesian inference with a structured variational autoencoder (VAE). Unlike conventional VAEs, our method embeds a differentiable game solver in the decoder, thereby facilitating unsupervised learning of an interpretable behavior model from an unlabeled dataset of observed interactions. At runtime, the proposed approach

[3] Project website: https://xinjie-liu.github.io/projects/bayesian-inverse-games

can generate samples from the predicted posterior without solving additional games. As a result, our approach naturally captures continuous, multi-modal beliefs, and addresses the limitations of MLE inverse games without resorting to the simplifications or computational complexity of existing Bayesian methods.

Through extensive evaluations of our method in several simulated driving scenarios, we support the following key claims. The proposed framework (i) learns the underlying prior game parameter distribution from unlabeled interactions, and (ii) captures potential multi-modality of game parameters. Our approach (iii) is uncertainty-aware, predicting narrow unimodal beliefs when observations clearly reveal the intentions of other agents, and beliefs that are closer to the learned prior in case of uninformative observations. The proposed framework (iv) provides more accurate inference performance than MLE inverse games by effectively leveraging the learned prior information, especially in settings where multiple parameter values explain the observed behavior. As a result, our approach (v) enables safer downstream robot plans than MLE methods. We also qualitatively demonstrate the scalability of our inference scheme in a 4-player intersection driving scenario.

2 Related Work

This section provides an overview of the literature on dynamic game theory, focusing on both forward games (Section 2.1) and inverse games (Section 2.2).

2.1 Forward Dynamic Games

This work focuses on noncooperative games where agents have partially conflicting but not completely adversarial goals and make sequential decisions over time [2]. Since we assume that agents take actions simultaneously without leader-follower hierarchy and we consider coupling between agents' decisions through both objectives and constraints, our focus is on generalized Nash equilibrium problems (GNEPs). GNEPs are challenging coupled mathematical optimization problems. Due to the computational challenges involved in solving such problems under feedback information structure [16], most works aim to find open-loop Nash equilibria (OLNE) [2] instead, where players choose their action sequence—an open-loop strategy—at once. Open-loop generalized Nash equilibria (GNE) have been solved using iterated best response [29, 32, 32–34, 36], sequential quadratic approximations [4, 37], or by leveraging the mixed complementarity problem (MCP) structure of their first-order necessary conditions [7, 20, 25]. This work builds upon the latter approach.

2.2 Inverse Dynamic Games

Inverse games study the problem of inferring unknown game parameters, e.g. of objective functions, from observations of agents' behavior [35]. In recent years, several approaches have extended single-agent inverse optimal control (IOC) and inverse reinforcement learning (IRL) techniques to multi-agent interactive settings. Early approaches [1, 28] minimize the residual of agents' first-order necessary conditions, given

full state-control observations, in order to infer unknown objective parameters. This approach is further extended to maximum-entropy settings in [12].

More recent work [26] proposes to maximize observation likelihood while enforcing the Karush–Kuhn–Tucker (KKT) conditions of OLNE as constraints. This approach only requires partial-state observations and can cope with noise-corrupted data. Approaches [18, 20] propose an extension of the MLE approach [26] to inverse feedback and open-loop games with inequality constraints via differentiable forward game solvers. To amortize the computation of the MLE, works [8, 20] demonstrate integration with neural network (NN) components.

In general, MLE solutions can be understood as point estimates of Bayesian posteriors, assuming a *uniform* prior [22, Ch.4]. When multiple parameter values explain the observations equally well, this simplifying assumption can result in ill-posed problems—causing MLE inverse games to recover potentially inaccurate estimates [18]. Moreover, in the context of motion planning, the use of point estimates without awareness of uncertainty can result in unsafe plans [11, 25].

To address these issues, several works take a Bayesian view on inverse games [17, 24], aiming to infer a posterior *distribution* while factoring in prior knowledge. Since exact Bayesian inference is intractable in these problems, the belief update may be approximated via a particle filter [24]. However, this approach requires solving a large number of equilibrium problems online to maintain the belief distribution, posing a significant computational burden. A sigma-point approximation [17] reduces the number of required samples but limits the estimator to unimodal uncertainty models.

To obtain multi-hypothesis predictions tractably, works in [5] and [19] integrate game-theoretic layers in NNs for motion forecasting. Both methods show that the inductive bias of games improves performance on real-world human datasets. However, both approaches are limited to the prediction of a fixed number of intents and offer no clear Bayesian interpretation of the learned model.

To overcome the limitations of MLE approaches while avoiding the intractability of exact Bayesian inference over continuous game parameter distributions, we propose to approximate the posterior via a VAE [14] that embeds a differentiable game solver [20] during training. The proposed approach can be trained from an unlabeled dataset of observed interactions, naturally handles continuous, multi-modal distributions, and does not require computation of game solutions at runtime to sample from the posterior.

3 Preliminaries: Generalized Nash Game

In this work, we consider strategic interactions of self-interested rational agents in the framework of *generalized Nash equilibrium problems (GNEPs)*. In this framework, each of N agents seeks to unilaterally minimize their respective cost while being conscious of the fact that their "opponents", too, act in their own best interest. We define a parametric N-player GNEP as N coupled, constrained optimization problems,

$$\mathcal{S}^i_\theta(\tau^{\neg i}) := \arg\min_{\tau_i} \quad J^i_\theta(\tau^i, \tau^{\neg i}) \tag{1a}$$

$$\text{s.t.} \quad g^i_\theta(\tau^i, \tau^{\neg i}) \geq 0, \tag{1b}$$

where $\theta \in \mathbb{R}^p$ denotes a parameter vector whose role will become clear below, and player $i \in [N] := \{1, \ldots, N\}$ has cost function J^i_θ and private constraints g^i_θ. Furthermore, observe that the costs and constraints of player i depend not only on their own strategy $\tau^i \in \mathbb{R}^{m_i}$, but also on the strategy of all other players, $\tau^{\neg i} \in \mathbb{R}^{\sum_{j \in [N] \setminus \{i\}} m_j}$.

Generalized Nash Equilibria. For a given parameter θ, the *solution* of a GNEP is an equilibrium strategy profile $\tau^* := (\tau^{1*}, \ldots, \tau^{N*})$ so that each agent's strategy is a best response to the others', i.e.,

$$\tau^{i*} \in \mathcal{S}^i_\theta(\tau^{\neg i*}), \forall i \in [N]. \tag{2}$$

Intuitively, at a GNE, no player can further reduce their cost by unilaterally adopting another feasible strategy.

Example: Online Game-Theoretic Motion Planning. This work focuses on applying games to online motion planning in interaction with other agents, such as the intersection scenario shown in Fig. 1. In this context, the strategy of agent i, τ_i, represents a *trajectory*—i.e., a sequence of states and inputs extended over a finite horizon—which is recomputed in a receding-horizon fashion. This paradigm results in the game-theoretic equivalent of model-predictive control (MPC): *model-predictive game-play (MPGP)*. The construction of a trajectory *game* largely follows the procedure used in single-agent trajectory *optimization*: $g^i_\theta(\cdot)$ encodes input and state constraints including those enforcing dynamic feasibility and collision avoidance; and $J^i_\theta(\cdot)$ encodes the i^{th} agent's objective such as reference tracking. Adopting a convention in which index 1 refers to the ego agent, the equilibrium solution of the game then serves two purposes simultaneously: the opponents' solution, $\tau^{\neg 1*}$, serves as a game-theoretic *prediction* of their behavior while the ego agent's solution, τ^{1*}, provides the corresponding best response.

The Role of Game Parameters $\boldsymbol{\theta}$. A *key difference* between trajectory games and single-agent trajectory optimization is the requirement to provide the costs and constraints of *all* agents. In practice, a robot may have insufficient knowledge of their opponents' intents, dynamics, or states to instantiate a complete game-theoretic model. This aspect motivates the parameterized formulation of the game above: for the remainder of this manuscript, θ will capture the unknown aspects of the game. In the context of game-theoretic motion planning, θ typically includes aspects of opponents' preferences, such as their unknown desired lane or preferred velocity. For conciseness, we denote game (1) compactly via the parametric tuple of problem data $\Gamma(\theta) := (\{J^i_\theta, g^i_\theta\}_{i \in [N]})$. Next, we discuss how to infer these parameters online from observed interactions.

4 Formalizing Bayesian Inverse Games

This section presents our main contribution: a framework for inferring unknown game parameters θ based on observations y. This problem is referred to as an *inverse* game [35].

Several prior works on inverse games [3, 18, 26] seek to find game parameters that directly maximize observation likelihood. However, this MLE formulation of inverse games (i) only provides a point estimate of the unknown game parameters θ, thereby precluding the consideration of uncertainty in downstream tasks such as motion planning; and (ii) fails to provide reasonable parameter estimates when observations are uninformative, as we shall also demonstrate in Section 5.

4.1 A Bayesian View on Inverse Games

In order to address the limitations of the MLE inverse games, we consider a *Bayesian* formulation of inverse games, and seek to construct the belief distribution

$$b(\theta) = p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}. \tag{3}$$

In contrast to MLE inverse games, this formulation provides a full posterior distribution over the unknown game parameters θ and factors in prior knowledge $p(\theta)$.

Observation Model $p(y \mid \theta)$. In autonomous online operation of a robot, y represents the (partial) observations of other players' recent trajectories—e.g., from a fixed lag buffer as shown in orange in Fig. 1. Like prior works on inverse games [17, 18, 26], we assume that, given the unobserved true trajectory of the human, y is Gaussian-distributed; i.e., $p(y \mid \tau) = \mathcal{N}(y \mid \mu_y(\tau), \Sigma_y(\tau))$. Assuming that the underlying trajectory is the solution of a game with known structure Γ but unknown parameters θ, we express the observation model as $p(y \mid \theta) = p(y \mid \mathcal{T}_\Gamma(\theta))$, where $\mathcal{T}_\Gamma$ denotes a game solver that returns a solution τ^* of the game $\Gamma(\theta)$.

Challenges of Bayesian Inverse Games. While the Bayesian formulation of inverse games in Eq. (3) is conceptually straightforward, it poses several challenges: (i) The prior $p(\theta)$ is typically unavailable and instead must be learned from data. (ii) The computation of the normalizing constant, $p(y) = \int p(y \mid \theta)p(\theta)\mathrm{d}\theta$, is intractable in practice due to the marginalization of θ. (iii) Both the prior $p(\theta)$ and posterior $p(\theta \mid y)$ are in general non-Gaussian or even *multi-modal* and are therefore difficult to represent explicitly in terms of their probability density function (PDF). Prior work [17] partially mitigates these challenges by using a UKF for approximate Bayesian inference, but that approach is limited to unimodal uncertainty models and requires solving multiple games for a single belief update, thereby posing a computational challenge.

Fortunately, as we shall demonstrate in Section 5, many practical applications of inverse games do not require an explicit evaluation of the belief PDF, $b(\theta)$. Instead, a *generative model* of the belief—i.e., one that allows drawing samples $\theta \sim b(\theta)$—often suffices. Throughout this section, we demonstrate how to learn such a generative model from an unlabeled dataset $\mathcal{D} = \{y_k \mid y_k \sim p(y), \forall k \in [K]\}$ of observed interactions.

4.2 Augmentation to Yield a Generative Model

To obtain a generative model of the belief $b(\theta)$, we augment our Bayesian model as summarized in the top half of Fig. 2. The goal of this augmentation is to obtain a model structure that lends itself to approximate inference in the framework of variational autoencoders (VAEs) [14]. As in a conventional VAE, we introduce an auxiliary random variable z with known isotropic Gaussian prior $p(z) = \mathcal{N}(z \mid \mu = 0, \Sigma = I)$ to model the data distribution as the marginal $p_\phi(y) = \int p_\phi(y \mid z)p(z)\mathrm{d}z$. We thus convert the task of learning the data distribution into learning the parameters of the conditional distribution $p_\phi(y \mid z)$, commonly referred to as the *decoder* in terminology of VAEs. The key difference of our approach to a conventional VAE is the special *structure* of this decoder. Specifically, while a conventional VAE employs an (unstructured) NN, say d_ϕ, to

map z to the parameters of the observation distribution, i.e. $p_\phi(y \mid z) = p(y \mid d_\phi(z))$, our decoder composes this NN with a game solver $\mathcal{T}_\Gamma$ to model the data distribution as $p(y \mid (\mathcal{T}_\Gamma \circ d_\phi)(z))$. From the perspective of a conventional VAE, the game solver thus takes the role of a special "layer" in the decoder, mapping game parameters predicted by d_ϕ to the observation distribution. In more rigorous probabilistic terminology, our proposed generative model can be interpreted as factoring the conditional distribution as $p_\phi(y \mid z) = \int p(y \mid \theta) p_\phi(\theta \mid z) \mathrm{d}\theta$, with $p_\phi(\theta \mid z) = \delta(\theta - d_\phi(z))$. Here, δ denotes the Dirac delta function[4] and the observation model $p(y \mid \theta)$ includes the game solver $\mathcal{T}_\Gamma$ as discussed in Section 4.1.

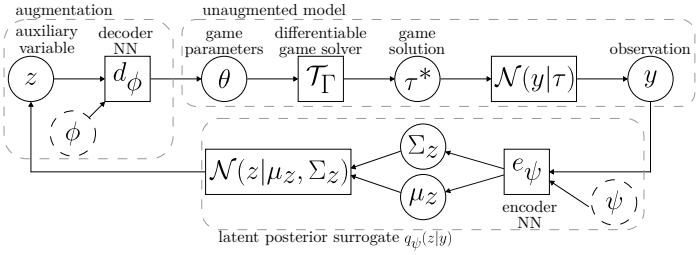


Fig. 2: Overview of a structured VAE for generative Bayesian inverse games. Top (left to right): decoder pipeline. Bottom (right to left): variational inference via an encoder.

4.3 Auto-Encoding Bayesian Inverse Games

The modeling choices made in Section 4.2 imply that the game parameters θ and observations y are conditionally independent given the latent variable z. Therefore, knowledge of the distribution of z fully specifies the downstream distribution of θ. Specifically, observe that we can express the prior over the game parameters as $p_\phi(\theta) = \int p_\phi(\theta \mid z) p(z) \mathrm{d}z$ and the corresponding posterior as $p_\phi(\theta \mid y) = \int p_\phi(\theta \mid z) p_\phi(z \mid y) \mathrm{d}z$. Intuitively, this means that by propagating samples from the latent prior $p(z)$ through d_ϕ, we implicitly generate samples from the prior $p_\phi(\theta)$. Similarly, by propagating samples from the latent posterior $p_\phi(z \mid y)$ through d_ϕ, we implicitly generate samples from the posterior $p_\phi(\theta \mid y)$. We thus have converted the problem of (generative) Bayesian inverse games to estimation of ϕ and inference of $p_\phi(z \mid y)$.

As in conventional VAEs, we can perform both these tasks through the lens of (amortized) variational inference (VI). That is, we seek to approximate the latent posterior $p_\phi(z \mid y)$ with a Gaussian $q_\psi(z \mid y) = \mathcal{N}(z \mid e_\psi(y))$. Here, e_ψ takes the role of an *encoder* that maps observation y to the mean $\mu_z(y)$ and covariance matrix $\Sigma_z(y)$ of the Gaussian posterior approximation; cf. bottom of Fig. 2. The design of this encoder model directly follows that of a conventional VAE [14] and can be realized via a NN.

[4] The modeling of $p_\phi(\theta \mid z)$ as a Dirac delta function—i.e. a deterministic relationship between z and θ—is consistent with the common modeling choice in non-hierarchical VAEs [14] which employ a *deterministic* decoder network. We found good performance with this setup in our experiments; cf., Section 5. We discuss extensions to other modeling choices in Section 6.

As is common in amortized VI, we seek to fit the parameters of both the encoder and the structured decoder via gradient-based optimization. A key challenge in taking this approach, however, is the fact that this training procedure requires back-propagation of gradients through the entire decoder; including the game solver $\mathcal{T}_\Gamma$. It is this gradient information from the game-theoretic "layer" in the decoding pipeline that induces an interpretable structure on the output of the decoder-NN d_ϕ, forcing it to predict the hidden game parameters θ. We recover the gradient of the game solver using the implicit differentiation approach proposed in [20] to differentiate through the PATH solver [6].

Remark on Inference Time. At inference time, to generate samples from the estimated posterior $q_{\psi,\phi}(\theta \mid y) = \int p_\phi(\theta \mid z) q_\psi(z \mid y) \mathrm{d}z$, we do not need to evaluate the game solver $\mathcal{T}_\Gamma$: posterior sampling involves only the evaluation of NNs d_ϕ and e_ψ; cf. $y \to \theta$ in Fig. 2. That is, given an observation y, we extensively sample from the latent Gaussian distribution $q_\psi(z \mid y)$ and map the latent samples z through the decoder network d_ϕ to recover a posterior distribution $q_{\psi,\phi}(\theta \mid y)$. This inference pipeline can operate repetitively at high frequencies, enabling the generation of updated posteriors in real time for downstream receding-horizon motion planning.

4.4 Training the Structured Variational Autoencoder

Below, we outline the process for optimizing model parameters in our specific setting. For a general discussion of VAEs and VI, we refer to [23].

Fitting a Prior to $\mathcal{D}$. First, we discuss the identification of the prior parameters ϕ. We fit these parameters so that the data distribution induced by ϕ, i.e., $p_\phi(y) = \int p_\phi(y \mid z) p(z) \mathrm{d}z$, closely matches the unknown true data distribution $p(y)$. For this purpose, we measure closeness between distributions via the Kullback–Leibler (KL) divergence

$$D_{\mathrm{KL}}\left(p \parallel q\right) := \mathbb{E}_{x \sim p(x)}\left[\log p(x) - \log q(x)\right]. \tag{4}$$

The key properties of this divergence metric, $D_{\mathrm{KL}}\left(p \parallel q\right) \geq 0$ and $D_{\mathrm{KL}}\left(p \parallel q\right) = 0 \iff p = q$, allow us to cast the estimation of ϕ as an optimization problem:

$$\begin{aligned} \phi^* &\in \arg\min_\phi \quad D_{\mathrm{KL}}\left(p(y) \parallel p_\phi(y)\right) && \text{(5a)} \\ &= \arg\min_\phi \quad -\mathbb{E}_{y \sim p(y)}\left[\log \mathbb{E}_{z \sim p(z)}\left[p_\phi(y \mid z)\right]\right]. && \text{(5b)} \end{aligned}$$

With the prior recovered, we turn to the approximation of the posterior $p_\phi(\theta \mid y)$.

Variational Belief Inference. We can find the closest surrogate $q_\psi(z \mid y)$ of $p_\phi(z \mid y)$ by minimizing the expected KL divergence between the two distributions over the data distribution $y \sim p(y)$ using the framework of VI:

$$\begin{aligned} \psi^* &\in \arg\min_\psi \mathbb{E}_{y \sim p(y)}\left[D_{\mathrm{KL}}\left(q_\psi(z \mid y) \parallel p_\phi(z \mid y)\right)\right] && \text{(6a)} \\ &= \arg\min_\psi \mathbb{E}_{\substack{y \sim p(y) \\ z \sim q_\psi(z|y)}}\left[\ell(\psi, \phi, y, z)\right], && \text{(6b)} \end{aligned}$$

$$\text{where} \quad \ell(\psi, \phi, y, z) := \log \underbrace{q_\psi(z \mid y)}_{\mathcal{N}(z|e_\psi(y))} - \log \underbrace{p_\phi(y \mid z)}_{\mathcal{N}(y|(\mathcal{T}_\Gamma \circ d_\phi)(z))} - \log \underbrace{p(z)}_{\mathcal{N}(z)}.$$

We have therefore outlined, in theory, the methodology to determine all parameters of the pipeline from the unlabeled dataset $\mathcal{D}$.

Considerations for Practical Realization. To solve Eqs. (5) and (6) in practice, we must overcome two main challenges: first, the loss landscapes are highly nonlinear due to the nonlinear transformations e_ψ, d_ϕ and $\mathcal{T}_\Gamma$; and second, the expected values cannot be computed in closed form. These challenges can be addressed by taking a *stochastic* first-order optimization approach, such as stochastic gradient descent (SGD), thereby limiting the search to a *local* optimum and approximating the gradients of the expected values via Monte Carlo sampling. To facilitate SGD, we seek to construct unbiased gradient estimators of the objectives of Eq. (5b) and Eq. (6b) from gradients computed at individual samples. We can construct an unbiased gradient estimator when the following conditions hold: (**C1**) any expectations appear on the outside, and (**C2**) the sampling distribution of the expectations are independent of the variable of differentiation; cf. [23, Chapter 10.2]. The objective of the optimization problem in Eq. (6b) takes the form

$$\mathcal{L}(q, \phi) := \mathbb{E}_{\substack{y\sim p(y) \\ z\sim q(z|y)}} \left[\ell(\psi, \phi, y, z)\right], \tag{7}$$

which clearly satisfies **C1**. Similarly, it is easy to verify that the objective of Eq. (5b) can be identified as $\mathcal{L}(p_\phi(z \mid y), \cdot)$. Unfortunately, this latter insight is not immediately actionable since $p_\phi(z \mid y)$ is not readily available. However, if instead we use our surrogate model from Eq. (6)—which, by construction closely matches $p_\phi(z \mid y)$—we can cast the estimation of the prior and posterior model as a joint optimization of $\mathcal{L}$:

$$\tilde{\psi}^*, \tilde{\phi}^* \in \arg\min_{\psi,\phi} \mathcal{L}(q_\psi, \phi). \tag{8}$$

When e_ψ and d_ϕ are sufficiently expressive to allow $D_{\mathrm{KL}}\left(q_\psi(z \mid y) \parallel p_\phi(z \mid y)\right) = 0$, then this reformulation is exact; i.e., $\tilde{\phi}^*$ and $\tilde{\psi}^*$ are also minimizers of the original problems Eq. (5) and Eq. (6), respectively. In practice, a perfect match of distributions is not typically achieved and $\tilde{\phi}^*$ and $\tilde{\psi}^*$ are biased. Nonetheless, the scalability enabled by this reformulation has been demonstrated to enable generative modeling of complex distributions, including those of real-world images [9]. Finally, to also satisfy **C2** for the objective of Eq. (8), we apply the well-established "reparameterization trick" to cast the inner expectation over a sampling distribution independent of ψ. That is, we write

$$\mathcal{L}(q_\psi, \phi) = \mathbb{E}_{\substack{y\sim p(y) \\ \epsilon\sim\mathcal{N}}} \left[\ell(\psi, \phi, y, r_{q_\psi}(\epsilon, y))\right], \tag{9}$$

where the inner expectation is taken over ϵ that has multi-variate standard normal distribution, and $r_{q_\psi}(\cdot, y)$ defines a bijection from ϵ to z so that $z \sim q_\psi(\cdot \mid y)$. Since we model the latent posterior as Gaussian—i.e., $q_\psi(z \mid y) = \mathcal{N}(z \mid e_\psi(y))$—the reparameterization map is $r_{q_\psi}(\epsilon, y) := \mu_z(y) + L_z(y)\epsilon$, where $L_z L_z^\top = \Sigma_z$ is a Cholesky decomposition of the latent covariance. Observe that Eq. (9) only involves Gaussian PDFs, and all terms can be easily evaluated in closed form.

Stochastic Optimization. As in SGD-based training of conventional VAEs, at iteration k, we sample $y_k \sim \mathcal{D}$, and encode y_k into the parameters of the latent distribution via e_{ψ_k}. From the latent distribution we then sample z_k, and evaluate the gradient

estimators $\nabla_\psi \ell(\psi, \phi_k, y_k, z_k)|_{\psi=\psi_k}$ and $\nabla_\phi \ell(\psi_k, \phi, y_k, z_k)|_{\phi=\phi_k}$ of $\mathcal{L}$ at the current parameter iterates, ψ_k and ϕ_k. Due to our model's structure, the evaluation of these gradient estimators thereby also involves the differentiation of the game solver $\mathcal{T}_\Gamma$.

5 Experiments

To assess the proposed approach, we evaluate its online inference capabilities and its efficacy in downstream motion planning tasks in several simulated driving scenarios. Experiment videos can be found in our supplementary video: https://xinjie-liu.github.io/projects/bayesian-inverse-games.

5.1 Planning with Online Inference

First, we evaluate the proposed framework for downstream motion planning tasks in the two-player intersection scenario depicted in Fig. 3. This experiment is designed to validate the following hypotheses:

- **H1 (Inference Accuracy).** Our method provides more accurate inference than MLE by leveraging the learned prior information, especially in settings where multiple human objectives explain the observations.
- **H2 (Multi-modality).** Our approach predicts posterior distributions that capture the multi-modality of agents' objectives and behavior.
- **H3 (Planning Safety).** Bayesian inverse game solutions enable safer robot plans compared to MLE methods.

Experiment Setup In the test scenario shown in Fig. 3, the red robot navigates an intersection while interacting with the green human, whose goal position is initially unknown. In each simulated interaction, the human's goal is sampled from a Gaussian mixture with equal probability for two mixture components: one for turning left, and one for going straight. To simulate human behavior that responds strategically to the robot's decisions, we generate the human's actions by solving trajectory games in a receding-horizon fashion with access to the sampled ground-truth game parameters.

We model the agents' dynamics as kinematic bicycles. The state at time step t includes position, longitudinal velocity, and orientation, i.e., $x_t^i = (p_{x,t}^i, p_{y,t}^i, v_t^i, \xi_t^i)$, and the control comprises acceleration and steering angle, i.e., $u_t^i = (a_t^i, \eta_t^i)$. We assign player index 1 to the robot and index 2 to the human. Over a planning horizon of $T = 15$ time steps, each player i seeks to minimize a cost function that encodes incentives for reaching the goal, reducing control effort, and avoiding collision:

$$J_\theta^i = \sum_{t=1}^{T-1} \|p_{t+1}^i - p_{\text{goal}}^i\|_2^2 + 0.1\|u_t^i\|_2^2 + 400 \max(0, d_{\min} - \|p_{t+1}^i - p_{t+1}^{\neg i}\|_2)^3, \quad (10)$$

where $p_t^i = (p_{x,t}^i, p_{y,t}^i)$ denotes agent i's position at time step t, $p_{\text{goal}}^i = (p_{\text{goal},x}^i, p_{\text{goal},y}^i)$ denotes their goal position, and $d_{\min}$ denotes a preferred minimum distance to other agents. The unknown parameter θ inferred by the robot contains the two-dimensional goal position p_{goal}^2 of the human.

We train the structured VAE from Section 4.3 on a dataset of observations from 560 closed-loop interaction episodes obtained by solving dynamic games with the opponent's ground truth goals sampled from the Gaussian mixture described above. The 560 closed-loop interactions are sliced into 34600 15-step observations y. Partial-state observations consist of the agents' positions and orientations. We employ fully-connected feedforward NNs with two 128-dimensional and 80-dimensional hidden layers as the encoder model e_ψ and decoder model d_ϕ, respectively. The latent variable z is 16-dimensional. We train the VAE with Adam [13] for 100 epochs, which took 14 hours.

Baselines We evaluate the following methods to control the robot interacting with a human of unknown intent, where both the robot and the human trajectories are computed game-theoretically at every time step in a receding-horizon fashion:

Ground truth (GT): This planner generates robot plans by solving games with access to ground truth opponent objectives. We include this oracle baseline to provide a reference upper-bound on planner performance.

Bayesian inverse game (ours) + planning in expectation (B-PinE) solves multi-hypothesis games [25] in which the robot minimizes the *expected* cost $\mathbb{E}_{\theta\sim q_{\psi,\phi}(\theta|y)}\left[J^1_\theta\right]$ under the posterior distributions predicted by our structured VAE based on new observations at each time step.

Bayesian inverse game (ours) + maximum a posteriori (MAP) planning (B-MAP) constructs MAP estimates $\hat{\theta}_{\mathrm{MAP}} \in \arg\max_\theta q_{\psi,\phi}(\theta \mid y)$ from the posteriors and solves the game $\Gamma(\hat{\theta}_{\mathrm{MAP}})$.

Randomly initialized MLE planning (R-MLE): This baseline [20] solves the game $\Gamma(\hat{\theta}_{\mathrm{MLE}})$, where $\hat{\theta}_{\mathrm{MLE}} \in \arg\max_\theta p(y \mid \theta)$. Hence, this baseline utilizes the *same game structure* as our method but only makes a point estimate of the game parameters. The MLE problems are solved online via gradient descent based on new observations at each time step as in [20]. The initial guess for optimization is sampled uniformly from a rectangular region covering all potential ground truth goal positions.

Bayesian prior initialized MLE planning (BP-MLE): Instead of uniformly sampling heuristic initial guesses as in R-MLE, this baseline solves for the MLE with initial guesses from the Bayesian *prior* learned by our approach.

Static Bayesian prior planning (St-BP) samples $\hat{\theta}$ from the learned Bayesian *prior* and uses the sample as a *fixed* human objective estimate to solve $\Gamma(\hat{\theta})$. This baseline is designed as an ablation study of the effect of online objective inference.

For those planners that utilize our VAE for inference, we take 1000 samples at each time step to approximate the distribution of human objectives, which takes around $7\,\mathrm{ms}$. For B-PinE, we cluster the posterior samples into two groups to be compatible with the multi-hypothesis game solver from [25]. We note that the observation distributions may differ between training and testing because, during runtime, observations are generated based on inferred objectives and may originate from different solvers, such as B-PinE.

Qualitative Behavior Figure 3 shows the qualitative behavior of B-PinE and R-MLE.

B-PinE: The top row of Fig. 3 shows that our approach initially generates a bimodal belief, capturing the distribution of potential opponent goals. In the face of this uncertainty, the planner initially computes a more conservative trajectory to remain safe.

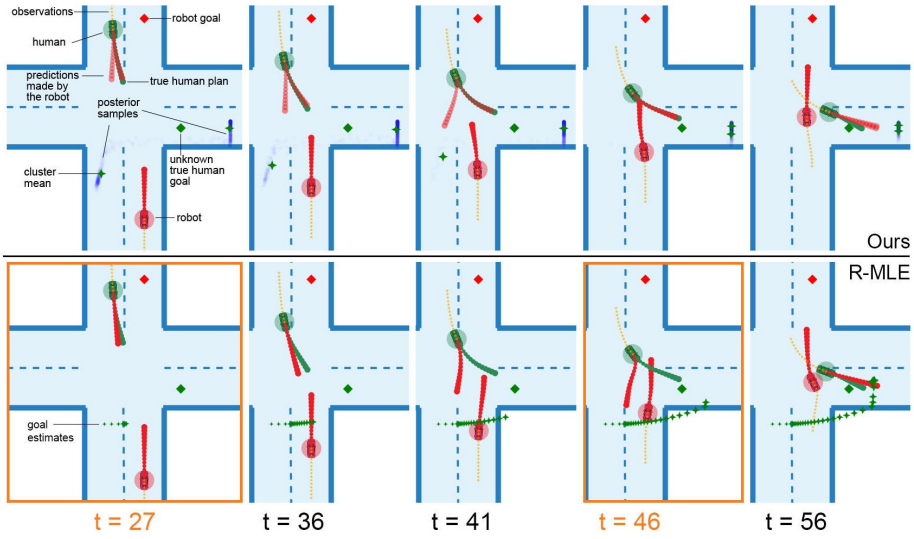


Fig. 3: Qualitative behavior of B-PinE (ours) vs. R-MLE. In the bottom row, the size of the green stars increases with time.

As the human approaches the intersection and reveals its intent, the left-turning mode gains probability mass until the computed belief eventually collapses to a unimodal distribution. Throughout the interaction, the predictions generated by the multi-hypothesis game accurately cover the true human plan, allowing safe and efficient interaction.

R-MLE: As shown in the bottom row of Fig. 3, the R-MLE baseline initially estimates that the human will go straight. The true human goal only becomes clear later in the interaction and the over-confident plans derived from these poor point estimates eventually lead to a collision.

Observability Issues of MLE: To illustrate the underlying issues that caused the poor performance of the R-MLE baseline discussed above, Fig. 4 shows the negative observation log-likelihood for two time steps of the interaction. As can be seen, a large region in the game parameter space explains the observed behavior well, and the baseline incorrectly concludes that the opponent's goal is always in front of their current position; cf. bottom of Fig. 3. In contrast, by leveraging the learned prior distribution, our approach predicts objective samples that capture potential opponent goals well even when the observation is uninformative; cf. top of Fig. 3.

*In summary, these results validate hypotheses **H1** and **H2**.*

Quantitative Analysis To quantify the performance of all six planners, we run a Monte Carlo study of 1500 trials. In each trial, we randomly sample the robot's initial position along the lane from a uniform distribution such that the resulting ground truth interaction covers a spectrum of behaviors, ranging from the robot entering the intersection first to the human entering the intersection first. We use the minimum distance between players in each trial to measure safety and the robot's cost as a metric for efficiency.

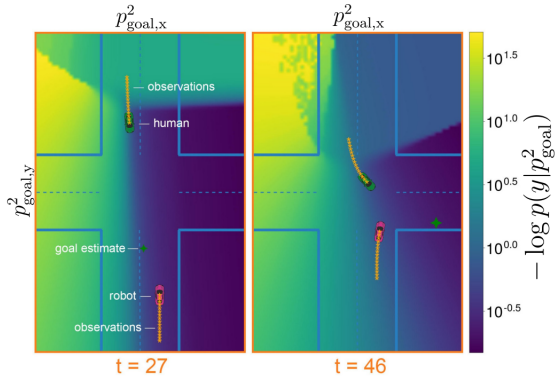


Fig. 4: Negative observation log-likelihood $-\log p(y \mid p^2_{\text{goal}})$ for varying human goal positions p^2_{goal} at two time steps of the R-MLE trial in Fig. 3.

We group the trials into three settings based on the ground-truth behavior of the agents: **(S1)** The human turns left and passes the intersection first. In this setting, a robot recognizing the human's intent should yield, whereas blindly optimizing the goal-reaching objective will likely lead to unsafe interaction. **(S2)** The human turns left, but the robot reaches the intersection first. In this setting, we expect the effect of inaccurate goal estimates to be less pronounced than in **S1** since the robot passes the human on the right irrespective of their true intent. **(S3)** The human drives straight. Safety is trivially achieved in this setting.

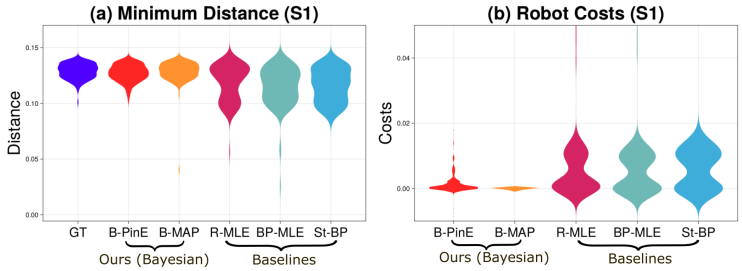


Fig. 5: Quantitative results of **S1**: (a) Minimum distance between agents in each trial. (b) Robot costs (with GT costs subtracted).

*Results, **S1**:* Figure 5(a) shows that methods using our framework achieve better planning safety than other baselines, closely matching the ground truth in this metric. Using the minimum distance between agents among all ground truth trials as a collision threshold, the collision rates of the methods are **0.0% (B-PinE)**, 0.78% (B-MAP), 17.05% (R-MLE), 16.28% (BP-MLE), and 17.83% (St-BP), respectively. Moreover, the improved safety of the planners using Bayesian inference does not come at the cost of reduced planning efficiency. As shown in Fig. 5(b), B-PinE and B-MAP consistently achieve low interaction costs.

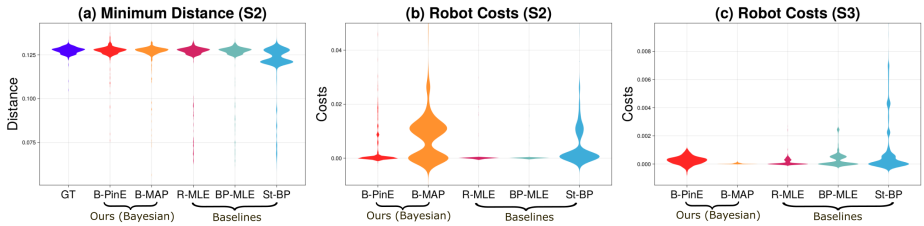


Fig. 6: Quantitative results of **S2** and **S3**: (a) Minimum distance between agents in each trial of **S2**. (b-c) Robot costs of **S2-3** (with GT costs subtracted).

*Results, **S2**:* Figure 6(a) shows that all the approaches except for the St-BP baseline approximately achieve ground truth safety in this less challenging setting. While the gap between approaches is less pronounced, we still find improved planning safety of our methods over the baselines. Taking the same collision distance threshold as in **S1**, the collision rates of the methods are **0.86% (B-PinE)**, 2.24% (B-MAP), 7.59% (R-MLE), 6.03% (BP-MLE), and 7.59% (St-BP), respectively. B-MAP gives less efficient performance in this setting. Compared with B-PinE, B-MAP only uses point estimates of the unknown goals and commits to over-confident and more aggressive behavior. We observe that this aggressiveness results in coordination issues when the two agents enter the intersection nearly simultaneously, causing them to accelerate simultaneously and then brake together. Conversely, the uncertainty-aware B-PinE variant of our method does not experience coordination failures, highlighting the advantage of incorporating the inferred *distribution* during planning. Lastly, St-BP exhibits the poorest performance in **S2**, stressing the importance of online objective inference.

*Results, **S3**:* In Fig. 6(c), B-MAP achieves the highest efficiency, while B-PinE produces several trials with increased costs due to more conservative planning. Again, the St-BP baseline exhibits the poorest performance.

Summary: These quantitative results show that our Bayesian inverse game approach improves motion planning safety over the MLE baselines, validating hypothesis **H3**. Between B-PinE and B-MAP, we observe improved safety in **S1-2** and efficiency in **S2**.

5.2 Additional Evaluation of Online Inference

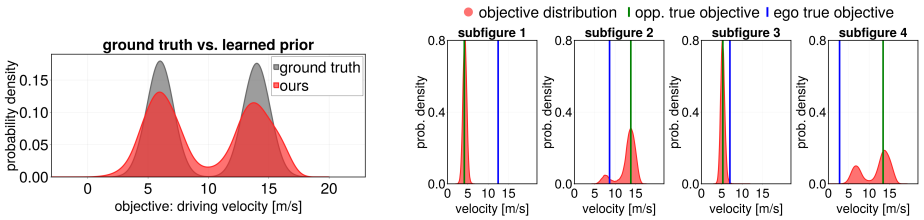
To isolate inference effects from planning, we consider two additional online inference settings. In these settings, agents operate with fixed but unknown ground-truth game parameters. Our method acts as an external observer, initially aware only of the robots' intent, and is tasked to infer the humans' intent online.

Two-Player Highway Game This experiment is designed to validate the following hypotheses:

- **H4 (Bayesian Prior).** Our method learns the underlying multi-modal prior objective distribution.
- **H5 (Uncertainty Awareness).** Our method computes a narrow, unimodal belief when the unknown objective is clearly observable and computes a wide, multi-modal belief that is closer to the prior in case of uninformative observations.

We consider a two-player highway driving scenario, in which an ego robot drives in front of a human opponent whose desired driving speed is unknown. The rear vehicle is responsible for decelerating and avoiding collisions, and the front vehicle always drives at their desired speed. To simplify the visualization of beliefs and thereby illustrate **H4-5**, we reduce the setting to a single spatial dimension and model one-dimensional uncertainty: we limit agents to drive in a single lane and model the agents' dynamics as double integrators with longitudinal position and velocity as states and acceleration as controls. Furthermore, we employ a VAE with the same hidden layers as in Section 5.1 but only *one-dimensional latent space*. The VAE takes a 15-step observation of the two players' velocities as input for inference.

We collect a dataset of 20000 observations to train a VAE. In each trial, the robot's reference velocity is sampled from a uniform distribution from $0\,\mathrm{m\,s^{-1}}$ to the maximum velocity of $20\,\mathrm{m\,s^{-1}}$; the human's desired velocity is sampled from a bi-modal Gaussian mixture distribution shown in grey in Fig. 7(a) with two unit-variance mixture components at means of 30% and 70% of the maximum velocity.



(a) Prior distributions.

(b) Posterior distributions.

Fig. 7: (a) Learned and ground truth priors for the human's objective. (b) Inferred objective posterior distributions.

Figure 7(a) shows that the learned prior objective distribution captures the underlying training set distribution closely and thereby validates our hypothesis **H4**. Figure 7(b) shows beliefs inferred by our approach for selected trials. The proposed Bayesian inverse game approach recovers a narrow distribution with low uncertainty when the opponent's desired speed is clearly observable, e.g., when the front vehicle drives faster so that the rear vehicle can drive at their desired speed (Fig. 7, subfigures 1, 3). The rear driver's objective becomes unobservable when they wish to drive fast but are blocked by the car in front. In this case, our approach infers a wide, multi-modal distribution with high uncertainty that is closer to the prior distribution (Fig. 7, subfigures 2, 4). These results validate hypothesis **H5**.

Multi-Player Demonstration To demonstrate the scalability of our method, we tested our inference approach in a four-agent intersection scenario. As shown in Fig. 8, a red robot interacts with three green humans, and our structured VAE infers the humans' goal positions. Initially, our method predicts multi-modal distributions with high uncertainty about the humans' objectives. As more evidence is gathered, uncertainty decreases,

and the belief accurately reflects the true objectives of the other agents. The objective distribution inference operates in real-time at approximately 150 Hz without requiring additional game solves.

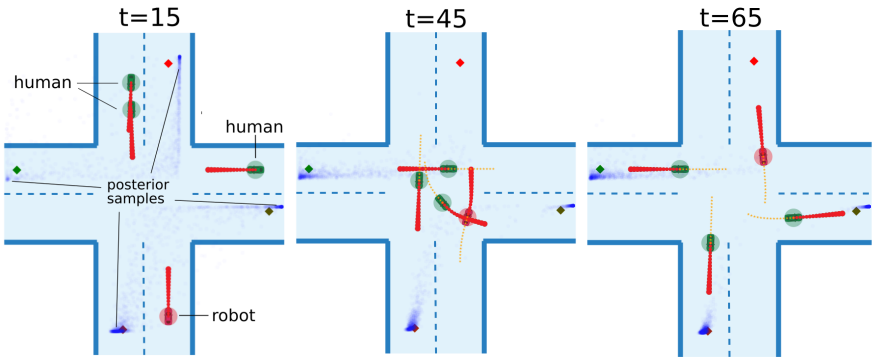


Fig. 8: Belief evolution of our structured VAE in a 4-player intersection scenario.

6 Conclusion & Future Work

We presented a tractable Bayesian inference approach for dynamic games using a structured VAE with an embedded differentiable game solver. Our method learns prior game parameter distributions from unlabeled data and infers multi-modal posteriors, outperforming MLE approaches in simulated driving scenarios.

Future work could explore this pipeline with other planning frameworks, incorporating active information gathering and intent signaling. Furthermore, in this work we employed a decoder architecture that assumes a deterministic relationship between the latent z and game parameters θ—a modeling choice that is consistent with conventional VAEs [14]. Future work could extend our structured amortized inference pipeline to models that assume a *stochastic* decoding process as in hierarchical VAEs [31] or recent diffusion probabilistic models [10]. Finally, our pipeline's flexibility allows for investigating different equilibrium concepts, such as entropic cost equilibria [21].

Acknowledgments. We thank Yunhao Yang, Ruihan Zhao, Haimin Hu, and Hyeji Kim for their helpful discussions. This work was supported in part by the National Science Foundation (NSF) under Grants 2211432, 271643-874F, 2211548 and 2336840, and in part by the European Union under the ERC Grant INTERACT 101041863. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the granting authorities; the granting authorities cannot be held responsible for them.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Awasthi, C., Lamperski, A.: Inverse differential games with mixed inequality constraints. In: Proc. of the IEEE American Control Conference (ACC) (2020)

[2] Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. Society for Industrial and Applied Mathematics (SIAM), 2. edn. (1999)

[3] Clarke, S., Dragotto, G., Fisac, J.F., Stellato, B.: Learning rationality in potential games. In: Proceedings of the Conference on Decision Making and Control (CDC). pp. 4261–4266. IEEE (2023)

[4] Cleac'h, L., Schwager, M., Manchester, Z.: ALGAMES: A fast augmented lagrangian solver for constrained dynamic games. Autonomous Robots **46**(1), 201–215 (2022)

[5] Diehl, C., Klosek, T., Krueger, M., Murzyn, N., Osterburg, T., Bertram, T.: Energy-based potential games for joint motion forecasting and control. In: Proc. of the Conf. on Robot Learning (CoRL) (2023)

[6] Dirkse, S.P., Ferris, M.C.: The PATH solver: A nommonotone stabilization scheme for mixed complementarity problems. Optimization methods and software **5**(2), 123–156 (1995)

[7] Facchinei, F., Pang, J.S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer Verlag (2003)

[8] Geiger, P., Straehle, C.N.: Learning game-theoretic models of multiagent trajectories using implicit layers. In: Proc. of the Conference on Advancements of Artificial Intelligence (AAAI). vol. 35 (2021)

[9] Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A.A., Visin, F., Vazquez, D., Courville, A.: Pixelvae: A latent variable model for natural images. In: Proc. of the Int. Conf. on Learning Representations (ICLR) (2016)

[10] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Proc. of the Conference on Neural Information Processing Systems (NeurIPS) **33**, 6840–6851 (2020)

[11] Hu, H., Fisac, J.F.: Active uncertainty reduction for human-robot interaction: An implicit dual control approach. In: Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR). pp. 385–401. Springer (2022)

[12] Inga, J., Bischoff, E., Köpf, F., Hohmann, S.: Inverse dynamic games based on maximum entropy inverse reinforcement learning. arXiv preprint arXiv:1911.07503 (2019)

[13] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proc. of the Int. Conf. on Learning Representations (ICLR) (2015)

[14] Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: Proc. of the Int. Conf. on Learning Representations (ICLR) (2014)

[15] Köpf, F., Inga, J., Rothfuß, S., Flad, M., Hohmann, S.: Inverse reinforcement learning for identification in linear-quadratic dynamic games. IFAC-PapersOnLine **50**(1), 14902–14908 (2017)

[16] Laine, F., Fridovich-Keil, D., Chiu, C.Y., Tomlin, C.: The computation of approximate generalized feedback nash equilibria. SIAM Journal on Optimization **33**(1), 294–318 (2023)

[17] Le Cleac'h, S., Schwager, M., Manchester, Z.: LUCIDGames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning. IEEE Robotics and Automation Letters (RA-L) **6**(3), 5485–5492 (2021)

[18] Li, J., Chiu, C.Y., Peters, L., Sojoudi, S., Tomlin, C., Fridovich-Keil, D.: Cost inference for feedback dynamic games from noisy partial state observations and incomplete trajectories. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). pp. 1062–1070 (2023)

[19] Lidard, J., So, O., Zhang, Y., DeCastro, J., Cui, X., Huang, X., Kuo, Y.L., Leonard, J., Balachandran, A., Leonard, N., et al.: Nashformer: Leveraging local nash equilibria for semantically diverse trajectory prediction. arXiv preprint arXiv:2305.17600 (2023)

[20] Liu, X., Peters, L., Alonso-Mora, J.: Learning to play trajectory games against opponents with unknown objectives. IEEE Robotics and Automation Letters (RA-L) **8**(7), 4139–4146 (2023). https://doi.org/10.1109/LRA.2023.3280809

[21] Mehr, N., Wang, M., Bhatt, M., Schwager, M.: Maximum-entropy multi-agent dynamic games: Forward and inverse solutions. IEEE Trans. on Robotics (TRO) **39**(3), 1801–1815 (2023). https://doi.org/10.1109/TRO.2022.3232300

[22] Murphy, K.P.: Probabilistic Machine Learning: An introduction. MIT Press (2022), probml.ai

[23] Murphy, K.P.: Probabilistic machine learning: Advanced topics. MIT press (2023)

[24] Peters, L.: Accommodating intention uncertainty in general-sum games for human-robot interaction (Master's thesis, Hamburg University of Technology, 2020)

[25] Peters, L., Bajcsy, A., Chiu, C.Y., Fridovich-Keil, D., Laine, F., Ferranti, L., Alonso-Mora, J.: Contingency games for multi-agent interaction. IEEE Robotics and Automation Letters (RA-L) (2024)

[26] Peters, L., Rubies-Royo, V., Tomlin, C.J., Ferranti, L., Alonso-Mora, J., Stachniss, C., Fridovich-Keil, D.: Online and offline learning of player objectives from partial observations in dynamic games. In: Intl. Journal of Robotics Research (IJRR) (2023), https://journals.sagepub.com/doi/pdf/10.1177/02783649231182453

[27] Qiu, T., Fridovich-Keil, D.: Identifying occluded agents in dynamic games with noise-corrupted observations. arXiv preprint arXiv:2303.09744 (2023)

[28] Rothfuß, S., Inga, J., Köpf, F., Flad, M., Hohmann, S.: Inverse optimal control for identification in non-cooperative differential games. IFAC-PapersOnLine **50**(1), 14909–14915 (2017). https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.2538

[29] Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., Rus, D.: Social behavior for autonomous vehicles. Proceedings of the National Academy of Sciences **116**(50), 24972–24978 (2019)

[30] Schwarting, W., Pierson, A., Karaman, S., Rus, D.: Stochastic dynamic games in belief space. IEEE Trans. on Robotics (TRO) **37**(6), 2157–2172 (2021)
[31] Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. Proc. of the Conference on Neural Information Processing Systems (NeurIPS) **29** (2016)
[32] Spica, R., Cristofalo, E., Wang, Z., Montijano, E., Schwager, M.: A real-time game theoretic planner for autonomous two-player drone racing. IEEE Trans. on Robotics (TRO) **36**(5), 1389–1403 (2020)
[33] Wang, M., Wang, Z., Talbot, J., Gerdes, J.C., Schwager, M.: Game-theoretic planning for self-driving cars in multivehicle competitive scenarios. IEEE Trans. on Robotics (TRO) **37**(4), 1313–1325 (2021). https://doi.org/10.1109/TRO.2020.3047521
[34] Wang, Z., Spica, R., Schwager, M.: Game theoretic motion planning for multi-robot racing. In: Distributed Autonomous Robotic Systems. pp. 225–238. Springer (2019)
[35] Waugh, K., Ziebart, B.D., Bagnell, J.A.: Computational rationalization: the inverse equilibrium problem. In: Proc. of the Int. Conf. on Machine Learning (ICML). pp. 1169–1176 (2011)
[36] Williams, G., Goldfain, B., Drews, P., Rehg, J.M., Theodorou, E.A.: Best response model predictive control for agile interactions between autonomous ground vehicles. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA) (2018)
[37] Zhu, E.L., Borrelli, F.: A sequential quadratic programming approach to the solution of open-loop generalized nash equilibria. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 3211–3217. IEEE (2023)