Learning Game-Theoretic Models of Multiagent Trajectories Using Implicit Layers

Philipp Geiger, Christoph-Nikolas Straehle
Bosch Center for Artificial Intelligence, Renningen, Germany
philipp.w.geiger@de.bosch.com, christoph-nikolas.straehle@de.bosch.com

Abstract

For prediction of interacting agents' trajectories, we propose an end-to-end trainable architecture that hybridizes neural nets with game-theoretic reasoning, has interpretable intermediate representations, and transfers to downstream decision making. It uses a net that reveals preferences from the agents' past joint trajectory, and a differentiable implicit layer that maps these preferences to local Nash equilibria, forming the modes of the predicted future trajectory. Additionally, it learns an equilibrium refinement concept. For tractability, we introduce a new class of continuous potential games and an equilibrium-separating partition of the action space. We provide theoretical results for explicit gradients and soundness. In experiments, we evaluate our approach on two real-world data sets, where we predict highway drivers' merging trajectories, and on a simple decision-making transfer task.

1 Introduction

Prediction of interacting agents' trajectories has recently been advanced by flexible, tractable, multi-modal data-driven approaches. But it remains a challenge to use them for safety-critical domains with additional verification and decision-making transfer requirements, like automated driving or mobile robots in interaction with humans. Towards addressing this challenge, the following seem sensible *intermediate goals*: (1) incorporation of well-understood *principles*, prior knowledge and reasoning of the multiagent domain, allowing to generalize well and to transfer to robust downstream decision making; (2) *interpretability* of models' latent variables, allowing for verification beyond just testing the final output; (3) theoretical *analysis* of soundness.

In this paper, we take a step towards addressing multiagent trajectory prediction including these intermediate goals, while trying to keep as much as possible of the practical strength of data-driven approaches. For this, we *hybridize neural learning with game-theoretic reasoning* – because game theory provides well-established explanations of agents' behavior based on the principle of instrumental rationality, i.e., viewing agents as utility maximizers. Roughly speaking, we *"fit a game to the observed trajectory data"*.

Along this hybrid direction one major obstacle – and a general reason why game theory often remains in abstract settings – lies in classic game-theoretic solution concepts like the Nash equilibrium (NE) notoriously suffering from computational *intractability*. As one way to overcome this, we build on *local* NE (Ratliff, Burden, and Sastry 2013, 2016). We combine this with a specific class of games – *(continuous) potential games* (Monderer and Shapley 1996) – for which local NE usually coincide with local optima of a single objective function, simplifying search. Another challenge lies in combining game theory with neural nets in a way that makes the overall model still efficiently *trainable* by gradient-based methods. To address this, we build on *implicit layers* (Amos and Kolter 2017; Bai, Kolter, and Koltun 2019). Implicit layers specify the functional relation between a layer's input and its output not in closed form, but only implicitly, usually via an equation. Nonetheless they allow to get exact gradients by "differentiating through" the equation, based on the *implicit function theorem*.

Main contributions and outline. We propose a modular architecture that outputs a multi-modal prediction of interacting agents' joint trajectory (where modes are interpretable as local Nash equilibria), from their past trajectory as input (Sec. 3). The architecture is depicted in Fig. 1, alongside the motivating example of highway drivers' merging trajectories. It builds on the following components:

- a tractable, differentiable *game solver implicit layer* (Sec. 3.1) with explicit gradient formula, mapping game parameters to local Nash equilibria (Thm. 1). It is based on a *new class of continuous-action trajectory games* that allow to encode prior knowledge on agents' preferences (Def. 4). We prove that they are potential games (Lem. 1). And it builds on an *equilibrium-separating concave partition of the action space* that we introduce to ensure tractability (Def. 5).
- Furthermore, the architecture contains a *neural net that reveals the agents' preferences* from their past, and a *net that learns an equilibrium refinement concept* (Sec. 3.2).

This architecture forms a model class where certain latent representations have clear game-theoretic interpretations and certain layers encode game-theoretic principles that help induction (also towards strategically-robust decision-making). At the same time, it has neural net-based capacity for learning, and is end-to-end trainable with analytic gradient formula. Furthermore:

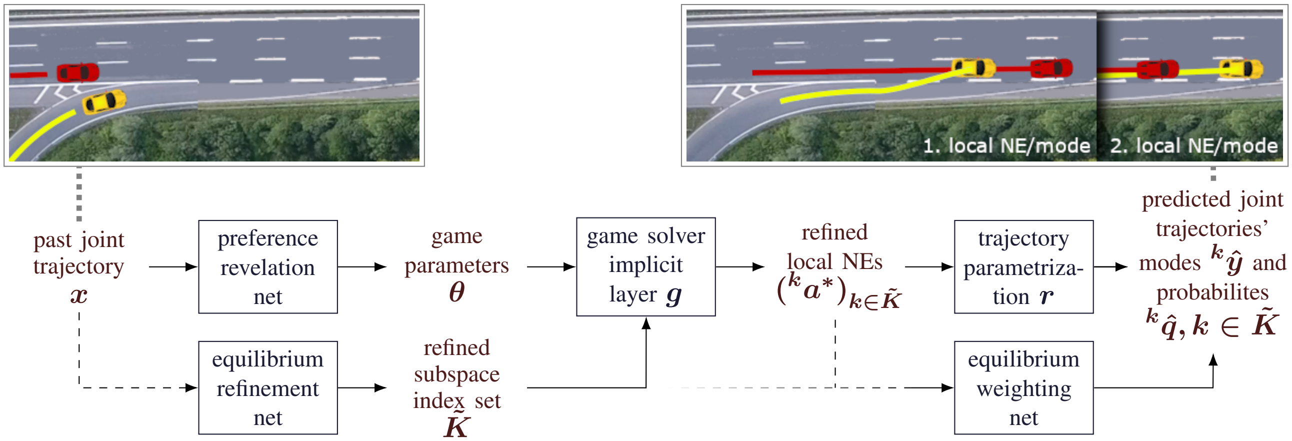

Figure 1: *Bottom:* Our full architecture (Sec. 3.2). *Top:* Example: highway merging scenario, where reliable models of (human) driver interaction are key for safe automated driving. *Top left:* input x*: initial trajectories* of drivers. *Top right:* prediction of *future trajectory* y: depicted are *two modes* ${}^1\hat{y}, {}^2\hat{y}$ corresponding to *two local Nash eq.* ${}^1a^*, {}^2a^*$: red going first vs. yellow first.

• In Sec. 4, we give two concrete example scenarios that provably satisfy our approach's conditions (Prop. 1, etc.).

• In the experiments reported in Sec. 5, we apply our architecture to prediction of real-world highway on-ramp merging driver interaction trajectories, on one established and one new data set we publish alongside this paper. We also apply it to a simple decision-making transfer task.

Keep in mind that proofs are available in Sec. A of the extended version including appendix (Geiger and Straehle 2021) of this paper. In what follows, we first discuss related work and introduce setting and background (Sec. 2).

Closest related work. Regarding general *multiagent model learning* from observational behavioral data *with game-theoretic components*: closest related is work by Ling, Fang, and Kolter (2018, 2019), who use game solvers as differentiable implicit layers, learning these layers' input (i.e., agents' preferences) from covariates. They focus on discrete actions while we address continuous trajectory prediction. And they use different solution concepts, and do not consider equilibrium refinement. There is further work more broadly related in this direction (Kita 1999; Kita, Tanimoto, and Fukuyama 2002; Liu et al. 2007; Kang and Rakha 2017; Tian et al. 2018; Li et al. 2018; Fox et al. 2018; Camara et al. 2018; Ma et al. 2017; Sun, Zhan, and Tomizuka 2018), sometimes also studying driver interaction, but they have no or little data-driven aspects (in particular no implicit layers) and/or use different approximations to rationality than our local NE, such as level-k reasoning, and often are less general than us, often focusing on discrete actions. More broadly related is multiagent inverse reinforcement learning (Wang and Klabjan 2018; Reddy et al. 2012; Zhang et al. 2019; Etesami and Straehle 2020), usually discrete-action.

For *multiagent trajectory prediction*, there generally is a growing number of papers on the *machine learning* side, often building on deep learning principles and allowing multimodality – but without game-theoretic components. Without any claim to completeness, there is work using long-short term memories (LSTMs) (Alahi et al. 2016; Deo and Trivedi 2018; Salzmann et al. 2020), generative adversarial networks (GANs) (Gupta et al. 2018), and attention-based encoders (Tang and Salakhutdinov 2019). Kuderer et al. (2012) uses a partition ("topological variants") of the trajectory space related to ours. There is also work related to the principle of "social force" (Helbing and Molnar 1995; Robicquet et al. 2016; Blaiotta 2019), and related rule-based driver modeling approaches (Treiber, Hennecke, and Helbing 2000; Kesting, Treiber, and Helbing 2007).

Regarding *additional game-theoretic elements*: W.r.t. the class of trajectory potential games we introduce (Def. 4), the closest related work we are aware of is (Zazo et al. 2016) who consider a related class, but they do not allow agents' utilities to have differing additive terms w.r.t. their own actions. Worth mentioning is further related work based on games (different ones than ours though), but towards pure *control* (not prediction) tasks (Peters et al. 2020; Zhang et al. 2018; Spica et al. 2018; Fisac et al. 2019). Peters et al. (2020) use a latent variable for the equilibrium selection, similar to our equilibrium weighting. For further related work see Sec. E in (Geiger and Straehle 2021).

2 General Setting, Goals and Background

We consider *scenes*, each consisting of:

• a set $I := \{1, \dots, n\}$ of agents.

• Each agent $i \in I$ at each time $t \in [0, T]$ has an *individual state* $y^i_t \in \mathbb{R}^{d_Y}$. They yield an *individual trajectory* $y^i = (y^i_t)_{t\in[0,T]}$ (think of 0 as the present time point and T as the horizon up to which we want to predict).

• And $y := ((y^1_t, \dots, y^n_t))_{t\in[0,T]} \in Y$ denotes the agents' *joint (future) trajectory*.

• We assume that the *past joint trajectory* $x \in X$ of the agents until time point 0 is available as side information.

Now, besides the other goals mentioned in Sec. 1, we formulate the *main (passive) predictive problem* as follows:

- *goal:* in a new scene, predict the future joint trajectory y by a list of pairs $(\hat{y}, \hat{q})$, corresponding to y's modes, where each $\hat{y}$ is a point prediction of y, and $\hat{q}$ the associated probability (for more details and metrics etc., see Sec. 3.2, 5);

- *given:* (1) the past trajectory x of that new scene, as well as (2) a training set consisting of previously sampled scenes, i.e., pairs $(x', y'), (x'', y''), \ldots$ of past and future trajectory (discrete-time subsampled of course). (We assume all scenes sampled from the same underlying distribution.)

We assume that agent i's (future) trajectory y^i is parameterized by a finite-dimensional vector $a^i \in A^i \subseteq \mathbb{R}^{d_A}$, which we refer to as i's *action*, with A^i the *action space* of i. So, in particular, there is a *(joint) trajectory parameterization* $r : A \to Y$, with $A := A^1 \times \ldots \times A^n$ the *joint action space*. Keep in mind that $a = (a^1, \ldots, a^n)$, a^{-i} means $(a^1, \ldots, a^{i-1}, a^{i+1}, a^n)$ and (a^i, a^{-i}) reads a.

We use games (Shoham and Leyton-Brown 2008; Osborne and Rubinstein 1994) to model our setting. A game specifies the set of agents (also called "players"), their possible actions and their utility functions. The following formal definition is slightly tailored to our setting: utilities are integrals over the trajectories parameterized by the actions.

Definition 1 (Game)**.** *A (trajectory) game consists of: the set I of agents, and for each agent $i \in I$: the action space $A^i \subseteq \mathbb{R}^{d_A}$, and a utility function $u^i\colon A \to \mathbb{R}$. We assume u^i, $i \in I$, to be of the form*

$$u^i(a) = \int_0^T u_t^i(y_t) d\mu(t),$$

where $a \in A$, $y = r(a)$; u_t^i, $t \in [0, T]$, are the stage-wise utility functions*, and μ is a measure on $[0, T]$.*[1]

This game formalizes the agents' "decision-making *problem*". Game theory also provides the "Nash equilibrium" as a concept of how rational agents will/should act to *"solve"* the game. Here we use a "local" version – for tractability:

Definition 2 (Local Nash equilibrium (NE) (Ratliff, Burden, and Sastry 2016, 2013))**.** *Given a game, a joint action $a \in A$ is a* (pure) local Nash equilibrium (local NE) *if there are open sets $S^i \subset A^i$ such that $a^i \in S^i$ and for each i,*

$$u^i(a^i, a^{-i}) \geq u^i(a^{i\prime}, a^{-i}),$$

for any $a^{i\prime} \in S^i$.[2] *If $S^i = A^i$ for all i, then a is called a* (pure, global) NE.

The following type of game can reduce finding local NE to finding local optima of a single objective ("potential function"), allowing for tractable gradient ascent-based search.

Definition 3 (Potential game (Monderer and Shapley 1996))**.** *A game is called an* (exact continuous) potential game*, if there is a so-called* potential function ψ *such that, for all agents i, all actions $a^i, a^{i'}$ and remaining actions a^{-i},*

$$u^i(a^{i'}, a^{-i}) - u^i(a^i, a^{-i}) = \psi(a^{i'}, a^{-i}) - \psi(a^i, a^{-i}).$$

[1]This general integral-based formulation contains discrete-time Scenario 1 as special case with μ's mass on discrete time points.

[2]I.e., no agent can improve its utility by unilaterally and locally deviating from its action in a local NE – a "consistency" condition.

Let us also give some neural net-related background:

Remark 1 (Implicit layers (Amos and Kolter 2017; Amos et al. 2018; Bai, Kolter, and Koltun 2019; El Ghaoui et al. 2019))**.** *Classically, one specifies a neural net layer by specifying the functional relation between its input v and output w* explicitly*, in* closed form*, $w = f(v)$, for some function f (e.g., a softmax). The idea of* implicit layers *is to specify the relation* implicitly*, usually via an equation $h(v, w) = 0$ (coming from, e.g., a stationarity condition of an optimization or dynamics modeling problem). To ensure that this specification is indeed useful in prediction and training, there are two important requirements: (1) the equation has to determine a unique, tractable function f that maps v to w, and (2) f has to be differentiable, ideally with explicitly given analytic gradients.*

3 General Approach With Analysis

We now describe our general approach. It consists of (1) a game-theoretic model and differentiable reasoning about how the agents behave (Sec. 3.1), and (2) a neural net architecture that incorporates this game-theoretic model/reasoning as an implicit layer and combines it with learnable modules, with tractable training and decision-making transfer abilities (Sec. 3.2).

3.1 Common-Coupled Games, Equilibrium-Separation and Induced Implicit Layer

For the rest of the paper, let $(\Gamma_\theta)_{\theta \in \Theta}$, $\Theta \subseteq \mathbb{R}^{d_\Theta}$, be a *parametric family of trajectory games* (Def. 1). First let us introduce the following type of trajectory game to strike a balance between adequate modeling and tractability:

Definition 4 (Common-coupled game)**.** *We call Γ_θ a* common-coupled(-term trajectory) game*, if the stage-wise utility functions (Def. 1) have the following form, for all agents $i \in I$, $t \in [0, T]$:*

$$u_t^{i,\theta}(y_t) = u_t^{com,\theta}(y_t) + u_t^{own,i,\theta}(y_t^i) + u_t^{oth,i,\theta}(y_t^{-i}), \quad (1)$$

where $y = r(a)$ (action parameterizes trajectory, Sec. 2), $u_t^{com,\theta}$ is a term that depends on all agents' trajectories and is common between agents, $u_t^{own,i,\theta}$ and $u_t^{oth,i,\theta}$ are terms that only depend on agent i's trajectory, or all other agents' trajectories, respectively, and may differ between agents.

Common-coupled games adequately approximate many multiagent trajectory settings where agents trade off (1) social norms and/or common interests (say, traffic rules or the common interest to avoid crashes), captured by the common utility term $u_t^{\text{com},\theta}$, against (2) individual inclinations related to their own state, captured by the terms $u_t^{\text{own},i,\theta}$. It is non-cooperative, i.e., utilities differ, but more on the cooperative than adversarial end of games. For tractability we can state:

Lemma 1. *If Γ_θ is a common-coupled game, then it is a potential game with the following potential function, where, as usual, $y = r(a)$:*

$$\psi(\theta, a) = \int_0^T u_t^{com,\theta}(y_t) + \sum_{i \in I} u_t^{own,i,\theta}(y_t^i) d\mu(t).$$

Note that this implies existence of NE, given continuity of the utilities and compactness (Monderer and Shapley 1996).

We now show how the mappings from parameters θ to local NE of the game Γ_θ can be soundly defined, tractable and differentiable, so that we can use this game-theoretic reasoning[3] as one implicit layer (Rem. 1) in our architecture. For this, a helpful step is to (tractably) partition the action space into subspaces with exactly one equilibrium each – if the game permits this. For the rest of the paper, let $(\tilde{A}_k)_{k\in K}$ be a finite *collection of subspaces of* A, i.e., $\tilde{A}_k \subseteq A$.

Definition 5 (Equilibrium-separating action subspaces)**.** *For a common-coupled game Γ_θ, we call the action subspace collection $(\tilde{A}_k)_{k\in K}$* equilibrium-separating (partition) *if, for all $k \in K$ and $\theta \in \Theta$, the game's potential function $\psi(\theta, \cdot)$ is strictly concave on $\tilde{A}_k$.*[4]

As a simplified example, a first partition towards equilibrium-separation in the highway merging scenario of Fig. 1 would be into two subspaces: (1) those that result in *joint trajectories where the red car goes first* and (2) those where *yellow goes first*. More details follow in Scenario 1.

Keep in mind that the equation $\nabla_a\psi(\theta, a)=0$ is a necessary condition for a to be a local NE of Γ_θ (for interior points), since local optima of the potential function correspond to local NE. This equation induces an implicit layer:

Assumption 1. *Let Γ_θ be a common-coupled game. Let $(\tilde{A}_k)_{k\in K}$ be equilibrium-separating subspaces for it, and let all $\tilde{A}_k, k \in K$ be compact, given by the intersection of linear inequality constraints. On each subspace $\Theta \times \tilde{A}_k, k \in K$, let Γ_θ's potential function ψ be continuous.*

Theorem 1 (Games-induced differentiable implicit layer)**.** *Let Assumption 1 hold true.*[5] *Then, for each $k \in K$, there is a continuous mapping $g_k : \Theta \to \tilde{A}_k$, such that for any $\theta \in \Theta$, if $g_k(\theta)$ lies in the interior of $\tilde{A}_k$, then*

- *$g_k(\theta)$ is a local NE of Γ_θ,*
- *$g_k(\theta)$ is given by the unique argmax*[6] *of $\psi(\theta, \cdot)$ on $\tilde{A}_k$, with ψ the game's potential function (Lem. 1),*
- *g_k is continuously differentiable in θ with gradient*

$$J_\theta g_k(\theta) = -\left(H_a\psi(\theta, a)\right)^{-1} J_\theta\nabla_a\psi(\theta, a),$$

whenever ψ is twice continuously differentiable on an open set containing (θ, a), for $a = g_k(\theta)$, where ∇, J and H denote gradient, Jacobian and Hessian, respectively.

The specifics of how the g_k of Thm. 1 form an implicit layer will be discussed in Sec. 3.2.

Remark on boundaries. There remain several questions: e.g., whether the action space partition introduces "artificial" local NE at the boundaries of the subspaces; and also regarding what happens to the gradient if $g_k(\theta)$ lies at the boundary of A or $\tilde{A}_k$. Here we state a preliminary answer[7] to the latter:

Lemma 2. *Assume Assumption 1 and that ψ is twice continuously differentiable on a neighborhood of $\Theta\times\tilde{A}_k, k\in K$. If $a=g_k(\theta)$ lies on exactly one constraining affine hyperplane of $\tilde{A}_k$, defined by orthogonal vector v, with multiplier λ and optimum $\lambda^*>0$ of $\psi(\theta, a)$'s Lagrangian (details see proof), then $J_\theta g_k(\theta)$ is the upper left $nd_A\times nd_A$-submatrix of*

$$-\begin{pmatrix} H_a\psi(\theta, a) & v \\ \lambda^* v^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_\theta\nabla_a\psi(\theta, a) \\ 0 \end{pmatrix}.$$

Remark on identifiability. Another natural question is whether the game's parameters are *identifiable* from observations, and, especially, whether the g_k are invertible. While difficult to answer in general, we investigate this for one scenario in Sec. C in (Geiger and Straehle 2021).

3.2 Full Architecture With Further Modules, Tractable Training and Decision Making

Now for the overall problem of mapping past joint trajectories x to predictions of their future continuations y, we propose the architecture depicted in Fig. 1 alongside a training procedure. We call it *trajectory game learner (TGL).* (Its forward pass is explicitly sketched in Alg. 1 in Sec. B in (Geiger and Straehle 2021).) It contains the following modules (here we leave some of the modules fairly abstract because details depend on size of the data set etc.; for one concrete instances see the experimental setup in Sec. 5), which are well-defined under Assumption 1:

- ***Preference revelation net***: It maps the past joint trajectory $x \in X$ to the inferred game parameters $\theta \in \Theta$ (encoding agents preferences).[8] For example, this can be an LSTM.
- ***Equilibrium refinement net*:** This net maps the past joint trajectory $x \in X$ to a subset $\tilde{K} \subset K$(we encode $\tilde{K}$ e.g. via a multi-hot encoding), with $|\tilde{K}| = \tilde{k}$, for $\tilde{k}$ arbitrary but fixed. This subset $\tilde{K}$ selects a *subcollection* $(\tilde{A}_k)_{k\in\tilde{K}}$ of the full equilibrium-separating action space partition $(\tilde{A}_k)_{k\in K}$

[3] Here, we mean reasoning in the sense of drawing the "local NE conclusions" from the game Γ_θ, due to the principle of rationality. More generally, game-theoretic reasoning also comprises equilibrium refinement/selection (Sec. 3.2).

[4] We loosely speak of a *partition* of A, but we do not require to cover the full A, and we allow overlaps, so it is not a partition in the rigorous set-theoretic sense. NB: The subspaces also have the interpretation as *macroscopic/high-level joint action* of the agents: for instance, which car goes first in the merging scenario in Fig. 1.

[5] Note: (1) (Parts of) this theorem translate to general potential games, not just common-coupled games. (2) For tractability/analysis reasons we consider the simple deterministic game form of Def. 1 instead of, say, a Markov game – which we leave to future work. (3) Our framework may still be applicable if assumptions like concavity, which is quite strong, are relaxed. However, deriving guarantees may become arbitrarily hard.

[6] Due to the concavity assumption, we can use established tractable, guaranteeably sound algorithms to calculate this argmax.

[7] Note that similar results have already been established, in the sense of constrained optimizers as implicit layers (Amos and Kolter 2017), but we give the precise preconditions for our setting. See also Sec. E in (Geiger and Straehle 2021). Moreover, note that under the conditions of this lemma, i.e., when $g_k(\theta)$ lies at the boundary, then the above gradient $J_\theta g_k(\theta)$ in fact often becomes zero, which can be a problem for parameter fitting.

[8] In a sense, this net is the inverse of the game solver implicit layer on x, but can be more flexible.

(introduced in Sec. 3.1, Def. 5). This directly determines a *subcollection of local NE of the game* Γ_θ, denoted by $({}^k a^*)_{k \in \tilde{K}}$ – those local NE that lie in one of the subspaces $\tilde{A}_k, k \in \tilde{K}$.[9] The purpose is to narrow down the set of all local NE to a "refined" set of local NE that form the "most likely" candidates to be selected by the agents. [10] The reason why we not directly output the refined local NE (instead of the subspaces) is to simplify training (details follow). As a simple example, take a feed forward net with softmax as final layer to get a probability distribution over K, and then take the $\tilde{k}$ most probable $k \in K$ to obtain the set $\tilde{K}$.

- ***Game solver implicit layer***[11] $g := (g_k)_{k \in \tilde{K}}$**:** It maps the revealed game parameters $\theta \in \Theta$ together with the refined $\tilde{K}$ to the refined subcollection $({}^k a^*)_{k \in \tilde{K}}$ of local NE[12] (described in the equilibrium refinement net above). This is done by performing, for each $k \in \tilde{K}$, the concave optimization over the subspace $\tilde{A}_k$:

$$ {}^k a^* = g_k(\theta) = \arg\max_{a \in \tilde{A}_k} \psi(\theta, a), $$

based on Thm. 1. See also Line 3 to 4 in Alg. 1 in Sec. B in (Geiger and Straehle 2021).

- ***Equilibrium weighting net*****:** It outputs probabilities $({}^k \hat{q})_{k \in \tilde{K}}$ over the refined equilibria, and thus *probabilities of the modes of our prediction* (introduced in Sec. 2). We think of them as the probabilities of the mixture components in a mixture model, but leave the precise metrics open. As input, in principle the variables $\theta, ({}^k a^*)_{k \in \tilde{K}}$ are allowed, plus possibly the agents' utilities attained in the respective equilibrium. And one can think of various function classes, for instance a feed forward net with softmax final layer. Its purpose is to (probabilistically) learn agents' "equilibrium selection" mechanism considered in game theory.[13]

- **Trajectory parameterization** r**:** This is the predetermined parameterization from Sec. 2: it maps each local NE's joint action ${}^k a^*$ to the corresponding joint trajectory ${}^k \hat{y}$ that results from it, corresponding to *mode* k *of the prediction*, where $k \in \tilde{K}$ are the indices of the refined equilibria.

Training and tractability. Training of the architecture in principle happens as usual by fitting it to past scenes in the training set, sketched in Alg. 2 in Sec. B in (Geiger and Straehle 2021). The implicit layer's gradient for backpropagation is given in Thm. 1. By default, we take the mean absolute error (MAE) averaged over the prediction horizon $[0, T]$ (see also Sec. 5). Note that the full architecture – all modules plugged together – is not differentiable, because the equilibrium refinement net's output is discrete. However, it is easy to see that (1) the equilibrium refinement net and (2) the rest of the architecture can be trained *separately* and are both differentiable themselves: in training, for each sample (x, y), we directly know which subspace y lies in, so we first only train the equilibrium refinement net with this subspace's index k as target, and then train the full architecture with the equilibrium refinement net's weights fixed.[14] [15]

Observe that in training there is an outer (weight fitting) and an inner (game solver, i.e., potential function maximizer, during forward pass) optimization loop, so their speed is crucial. For the game solver, we recommend quasi-Newton methods like L-BFGS, because this is possible due to the subspace-wise concavity of the potential function (Assumption 1). For the outer loop, we recommend recent stochastic quasi-Newton methods (Wang et al. 2017; Li and Liu 2018).

Transferability to decision making. Once the game Γ_θ's parameters θ are learned (for arbitrary numbers of agents) as described above, it does not just help for *prediction* – i.e., a model of how an *observed* set of strategic agents *will* behave – but also for *prescription*. This means (among other things) that it tells how a newly introduced agent *should* decide to *maximize its utility*, while aware of how the other agents respond to it based on their utilities in Γ_θ (think of a self-driving car entering a scene with other – human – drivers).[16] Note: the knowledge of Γ_θ cannot resolve the remaining equilibrium selection problem (but the equilibrium weighting net may help). For an example see Sec. 5.2.

4 Concrete Example Scenarios With Analysis

We give two examples of settings alongside games and action space partitions that provably fulfill the conditions for our general approach (Sec. 3) to apply. First we consider a scenario that captures various non-trivial driver interactions like overtaking or merging at on-ramps. Essentially, it consists of a straight road section with multiple (same-directional) lanes, where some lanes can end within the section. Fig. 1 and 2 (left) are examples. This setting will be used in the experiments (Sec. 5).

Scenario 1 (Multi-lane driver interaction)**.** ***Setting:*** *The set of possible individual states, denote it by* Y_0*, is of the*

[9] To be exact, in rare cases it can happen that some of these local NE are "artificial" as discussed in Sec. 3.1.

[10] In game theory, "equilibrium refinement concepts" mean hand-crafted concepts that narrow down the set of equilibria of a game (for various reasons, such as achieving "stable" solutions) (Osborne and Rubinstein 1994). For us, the "locality relaxation" makes the problem of "too many" equilibria particularly severe, since the number of *local* NE can be even bigger than global NE; it can grow exponentially in the number of agents in our scenarios.

[11] NB: Here, the implicit layer does not have parameters. Generally, implicit layers with parameters can be handled similarly.

[12] At first sight, local NE are a poorer approximation to rationality than global NE, and are mainly motivated by tractability. However, we found that in various scenarios, like the highway merging, local NE do seem to correspond to something meaningful, like the intuitive modes of the distribution of joint trajectories. NB: Generally, we do not consider humans as fully (instrumentally) rational, but we see (instrumental) rationality as a useful approximation.

[13] "Equilibrium selection" (Harsanyi, Selten et al. 1988) refers to the problem of which *single* equilibrium agents will end up choosing if there are multiple – possibly even after a refinement.

[14] Therefore we loosely refer to the full architecture as "end-to-end *trainable*", not "end-to-end *differentiable*".

[15] On a related note, we learn the common term's parameter θ (see (1)) as shared between all scenes, while the other parameters

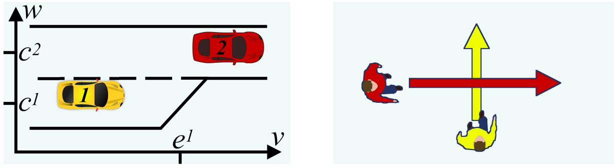

Figure 2: *Left:* Simple illustration of Scenario 1's variables. *Right:* Illustration of simplistic pedestrian encounter scenario (Sec. C in (Geiger and Straehle 2021)).

form $[b, c] \times [d, e]$ – positions on a road section. There are m parallel lanes (some of which may end), parallel to the x-axis. Agent i's action $a^i \in A^i$ is given by the sequence of planar (i.e., 2-D) positions denoted $(v^i_t, w^i_t) \in Y_0, t = 0, \ldots, T$, but not allowing backward moves (and possibly other constraints). Define the states $y^i_t := ((v^i_t, w^i_t), (v^i_{t-1}, w^i_{t-1}), (v^i_{t-2}, w^i_{t-2}))$.[17] *And let y^i be the linear interpolation.* ***Game:*** *Let, for $t = 0, \ldots, T$, the stage utilities of agent i in the game Γ_θ be the following sum of terms for distance between agents, distance to center of lane, desired velocities, acceleration penalty, and end of lane overshooting penalty, respectively:*[18]

$$u^{i,\theta}_t(y^i_t) = -\theta^{dist} \sum \frac{1}{|v^{j'}_t - v^j_t| + \zeta} - \theta^{cen,i}_t (w^i_t - c^i_t)^2 \quad (2a)$$

$$- \theta^{vel,i}_t (\delta v^i_t - \theta^{v,i})^2 - \theta^{velw,i} (\delta w^i_t)^2 - \theta^{acc,i} (\delta^2 v^i_t)^2 \quad (2b)$$

$$- \theta^{end,i} \max(0, v_t - e^i_t), \quad (2c)$$

where the sum ranges over all (j, j') such that driver j is right before j' on the same lane; $\zeta > 0$ is a constant, c^i_t is the respective center of the lane, δv^i_t means velocity along lane, δw^i_t means lateral velocity, $\delta^2 v^i_t$ means acceleration (vector), e^i_t is the end of i's lane, if it ends, otherwise $-\infty$; furthermore, μ is the counting measure on $\{0, \ldots, T\}$ (i.e., discrete). and $\theta = (\theta^{dist}, \theta^{cen,i}_{[0:T]}, \theta^{vel,i}_{[0:T]}, \theta^{v,i}, \theta^{velw,i}, \theta^{acc,i}, \theta^{end,i})_{i \in I}$.[19] ***Action subspaces:*** *Consider the following equivalence relation on the trajectory space Y: two joint trajectories $y, y' \in Y$ are equivalent if at each time point t, (1) each agent i is on the same lane in y as in y', and (2) within each lane, the order of the agents (along the driving direction) is the same in y as in y'. Now let the subspace collection $(\tilde{A}_k)_{k \in K}$ be obtained by taking the (closures of the) resulting equivalence classes.*[20]

Proposition 1 (Scenario 1's suitability)**.** *Scenario 1 satisfies Assumption 1. So, in particular, Thm. 1's implications on the induced implicit layer hold true.*

Our general approach (Sec. 3) in principle is also applicable to various other multiagent trajectory settings, such as pedestrian interaction, relevant for mobile robots. We analyze a simplistic such scenario in Sec. C in (Geiger and Straehle 2021), see Fig. 2 (right) for a foretaste.

5 Experiments

We evaluate our approach on (1) an observational prediction task[21] on two real-world data sets (Sec. 5.1), as well as (2) a simple decision-making transfer task (Sec. 5.2).

5.1 Prediction Task on Highway Merging Scenarios in Two Real-World Data Set

We consider a highway merging interaction scenario with two cars similar as sketched in Fig. 1. This is considered a challenging scenario for autonomous driving.

Implementation details for our method for these merging scenario. We use the following generic implementation of our general approach (Sec. 3), with concrete setting, game and action subspaces from Scenario 1 (with $n = 2$), referring to it as *TGL (trajectory game learner)*: We use validation-based early stopping. We combine equilibrium refinement and weighting net into one module, consisting of two nets that predict the weights $({}^k\hat{q})_{k \in \tilde{K}}$ on the combination of (1) merging order (before/after) probabilities via a cross-entropy loss (2 hidden layers: 1×16, 1×4 neurons; dropout 0.6), and (2) Gaussian distribution over merging time point (discretized and truncated, thus the support inducing a refinement; 2 hidden layers: 1×64, 1×32 neurons; dropout 0.6), given x. For the preference revelation net we use a feed forward net (two hidden layers: 1×16, 1×24 neurons).[22] As training loss we use mean absolute error (MAE; see also evaluation details below).

Besides this generic instantiation, we also consider a version of it, termed *TGL-D*: Instead of predicting the desired velocity $\theta^{v,i}$ itself, the preference revelation net predicts the difference to the past velocity, and then squashes this into a sensible range using a sigmoid. (This can be seen as encoding a bit of additional prior knowledge which may not always be easy to specify and depend on the situation.) For further details, see Sec. D in (Geiger and Straehle 2021).

Code is available at: `https://github.com/boschresearch/trajectory_games_learning`

are predicted from the individual's sample past trajectory.

[16] This is the general double nature of game theory – predictive and prescriptive (Shoham and Leyton-Brown 2008).

[17] We do this state augmentation so that utilities can also depend on velocity/acceleration (not just position) while still rigorously fitting into Def. 1. When calculating prediction errors for $\hat{y}$, only the position component is considered.

[18] Note that the *invariance over time* of the utility terms, as we assume it here, is a key element of how rationality principles can give *informative priors*.

[19] We allow some of the weights to vary with t to add some flexibility. In the experiments (Sec. 5), we use "terminal" costs only; more specifically $\theta^{vel,i}_t = 0$ for $0 \le t \le T - 6$ and $\theta^{cen,i}_t = 0$ for $0 \le t \le T - 1$, which we found works best.

[20] In the two-driver on-ramp scenario of Fig. 1 and experiments (Sec. 5.1), these subspaces roughly amount to splitting the action space A w.r.t. (1) time point of merge and (2) which driver goes first. Note that (1) are additional splits beyond the intuitive ones in (2) (see Fig. 1), but they help for concavity and for the analysis.

[21] This directly evaluates the method's abilities for the observational/passive prediction task, but it is also a proxy metric/task for decision making.

[22] For varying initial trajectory lengths, an LSTM might be more suitable.

Data set	Metric	TGL (ours)	TGL-D (ours)	CS-LSTM	MFP
highD (Krajewski et al. 2018)	MAE	3.6	2.9	5.0	5.2
	RMSE	4.9	3.7	6.8	7.1
HEE (our new data set; Sec. 5.1)	MAE	3.7	3.2	3.6	3.7
	RMSE	4.7	4.1	4.3	4.8

Table 1: *Prediction task:* Our method(s) vs. state-of-the-art (CS-LSTM (Deo and Trivedi 2018), MFP (Tang and Salakhutdinov 2019)) for a prediction task on merging scenarios in two real-world highway data sets, averaged over a 7s prediction horizon.

Figure 3: *Decision-making transfer task:* Solution trajectorie(s) that the (partially learned) game implies for the self-driving car's decision-making task (each circle/square corresponds to one time step). *Left:* First local NE: the self-driving car (red) does a full emergency break and the other (blue) merges before it. *Right:* Second local NE: the other merges after it, both slow down.

Baselines. As baselines we use the state-of-the-art data-driven methods "convolutional social pooling" – specifically: *CS-LSTM* (Deo and Trivedi 2018) – and "Multiple Futures Prediction" (MFP) (Tang and Salakhutdinov 2019). **Evaluation.** We use four-fold cross validation (splitting the data into 4 × 75% train and 25% validation). As metrics, we use rooted mean squared error (RMSE) and MAE (in meters) between predicted future trajectory $\hat{y}$ and truth y, averaged over a 7s horizon, with prediction step size of 0.2s, applying this to the most likely mode given by our method.

Data sets (one new one) and filtering: *1st data set:* We use the "highD" data set (Krajewski et al. 2018), which consists of car trajectories recorded by drones over several highway sections. It is increasingly used for benchmarking (Rudenko et al. 2019; Zhang et al. 2020). From this data set, we use the recordings done over a section with an on-ramp.

2nd data set: We publish a ***new data set*** with this paper, termed ***HEE (Highway Eagle Eye)***. It consists of ∼12000 individual car trajectories (∼4h), recorded by drones over a highway section (length ∼600m) with an entry lane. The link to the data set and further details are in Sec. D.2 in (Geiger and Straehle 2021). Keep in mind that this data set can be useful for studies like ours, but some aspects of it may be noisy, so it is only meant for such experimental purposes.

Selection of merging scenes in both data sets: We filter for all joint trajectories of two cars where one is merging from the on-ramp, one is on the rightmost highway lane, and all other cars are far enough to not interact with these two. This leaves 25 trajectories of highD and 23 of our new data set.

Results. The results are in Table 1 (with more details in Sec. D in (Geiger and Straehle 2021)). Our generic method TGL outperforms CS-LSTM and MFP on highD. And our slightly more hand-crafted method TGL-D outperforms them on both data set. Keep in mind that the data sets are small. So, while the results do indicate the practicality of our method in this small-sample regime, their significance is comparably limited.

5.2 Simple Decision-Making Transfer Task in Simulation

As discussed in Sec. 3.2 the game Γ_θ – once θ is given, e.g., by our learned preference revelation net – naturally transfers to decision-making tasks in situations with multiple strategic agents (something which predictive methods like the above CS-LSTM usually cannot do). To test and illustrate its ability for this, we consider a simple scenario: Take the above two-car highway on-ramp situation (Sec. 5.1, Scenario 1), but assume that the car on the highway lane is a self-driving car. Assume it has a technical failure roughly at the height of the on-ramp's end, and it should do an emergency break (i.e., desired velocity $\theta^{\mathrm{v},i}$ in (2) is set to 0) while at the same time ensuring that the other car coming from the on-ramp will not crash into it. Which trajectory should it choose? **Result.** Fed with this situation, our game solver suggests two possible solutions – two local NE, see Fig. 3: (1) the self-driving car completely stops and the on-ramp car will merge in front of it, accepting to touch the on-ramp's end; (2) the self-driving car moves slowly, but at a non-zero speed, with the other car right behind it (keeping a rational distance). While a toy scenario, we feel that these are sensible solutions.

6 Conclusion

For modeling of realistic continuous multiagent trajectories, in this work we proposed an end-to-end trainable model class that hybridizes neural nets with game-theoretic reasoning. We accompanied it with theoretical guarantees as well as an empirical demonstration of its practicality, on real-world highway data. We consider this as one step towards machine learning methods for this task that are more interpretable, verifiable and transferable to decision making. This is particularly relevant for safety-critical domains that involve interaction with humans. A major challenge is to make game-theoretic concepts tractable for such settings, and we were only partially able to address this. Specifically, potential for future work lies in relaxing subspace-wise concavity, common-coupled games and related assumptions we made.

Acknowledgments

We thank Jalal Etesami, Markus Spies and Mathias Buerger for insightful discussions and anonymous reviewers for their hints.

References

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 961–971.

Amos, B.; Jimenez, I.; Sacks, J.; Boots, B.; and Kolter, J. Z. 2018. Differentiable MPC for end-to-end planning and control. In *Advances in Neural Information Processing Systems*, 8289–8300.

Amos, B.; and Kolter, J. Z. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 136–145. JMLR. org.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2019. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 688–699.

Blaiotta, C. 2019. Learning generative socially aware models of pedestrian motion. *IEEE Robotics and Automation Letters* 4(4): 3433–3440.

Camara, F.; Romano, R.; Markkula, G.; Madigan, R.; Merat, N.; and Fox, C. 2018. Empirical game theory of pedestrian interaction for autonomous vehicles. In *Proceedings of Measuring Behavior 2018*. Manchester Metropolitan University.

Deo, N.; and Trivedi, M. M. 2018. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1468–1476.

El Ghaoui, L.; Gu, F.; Travacca, B.; and Askari, A. 2019. Implicit deep learning. *arXiv preprint arXiv:1908.06315* .

Etesami, J.; and Straehle, C.-N. 2020. Non-cooperative Multi-agent Systems with Exploring Agents. *arXiv preprint arXiv:2005.12360* .

Fisac, J. F.; Bronstein, E.; Stefansson, E.; Sadigh, D.; Sastry, S. S.; and Dragan, A. D. 2019. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International Conference on Robotics and Automation (ICRA)*, 9590–9596. IEEE.

Fox, C.; Camara, F.; Markkula, G.; Romano, R.; Madigan, R.; Merat, N.; et al. 2018. When should the chicken cross the road?: Game theory for autonomous vehicle-human interactions .

Geiger, P.; and Straehle, C.-N. 2021. Learning game-theoretic models of multiagent trajectories using implicit layers. *ArXiv e-prints* URL `https://arxiv.org/abs/2008.07303`. ArXiv:2008.07303 [cs.GT]. This is the extended version of the present paper, including appendix.

Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; and Alahi, A. 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2255–2264.

Harsanyi, J. C.; Selten, R.; et al. 1988. *A general theory of equilibrium selection in games*, volume 1. The MIT Press.

Helbing, D.; and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical review E* 51(5): 4282.

Kang, K.; and Rakha, H. A. 2017. Game theoretical approach to model decision making for merging maneuvers at freeway on-ramps. *Transportation Research Record* 2623(1): 19–28.

Kesting, A.; Treiber, M.; and Helbing, D. 2007. General lane-changing model MOBIL for car-following models. *Transportation Research Record* 1999(1): 86–94.

Kita, H. 1999. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. *Transportation Research Part A: Policy and Practice* 33(3-4): 305–312.

Kita, H.; Tanimoto, K.; and Fukuyama, K. 2002. A game theoretic analysis of merging-giveway interaction: a joint estimation model. *Transportation and Traffic Theory in the 21st Century* 503–518.

Krajewski, R.; Bock, J.; Kloeker, L.; and Eckstein, L. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*.

Kuderer, M.; Kretzschmar, H.; Sprunk, C.; and Burgard, W. 2012. Feature-based prediction of trajectories for socially compliant navigation. In *Robotics: science and systems*.

Li, N.; Kolmanovsky, I.; Girard, A.; and Yildiz, Y. 2018. Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control. In *2018 Annual American Control Conference (ACC)*, 3215–3220. IEEE.

Li, Y.; and Liu, H. 2018. Implementation of Stochastic Quasi-Newton's Method in PyTorch. *arXiv preprint arXiv:1805.02338* .

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2018. What game are we playing? End-to-end learning in normal and extensive form games. In *IJCAI-ECAI-18: The 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence*.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2019. Large scale learning of agent rationality in two-player zero-sum games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6104–6111.

Liu, H. X.; Xin, W.; Adam, Z.; and Ban, J. 2007. A game theoretical approach for modelling merging and yielding behaviour at freeway on-ramp sections. *Transportation and traffic theory* 3: 197–211.

Ma, W.-C.; Huang, D.-A.; Lee, N.; and Kitani, K. M. 2017. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 774–782.

Monderer, D.; and Shapley, L. S. 1996. Potential games. *Games and economic behavior* 14(1): 124–143.

Osborne, M. J.; and Rubinstein, A. 1994. *A course in game theory*. MIT press.

Peters, L.; Fridovich-Keil, D.; Tomlin, C. J.; and Sunberg, Z. N. 2020. Inference-Based Strategy Alignment for General-Sum Differential Games. *arXiv preprint arXiv:2002.04354* .

Ratliff, L. J.; Burden, S. A.; and Sastry, S. S. 2013. Characterization and computation of local Nash equilibria in continuous games. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 917–924. IEEE.

Ratliff, L. J.; Burden, S. A.; and Sastry, S. S. 2016. On the characterization of local Nash equilibria in continuous games. *IEEE Transactions on Automatic Control* 61(8): 2301–2307.

Reddy, T. S.; Gopikrishna, V.; Zaruba, G.; and Huber, M. 2012. Inverse reinforcement learning for decentralized noncooperative multiagent systems. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1930–1935. IEEE.

Robicquet, A.; Sadeghian, A.; Alahi, A.; and Savarese, S. 2016. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, 549–565. Springer.

Rudenko, A.; Palmieri, L.; Herman, M.; Kitani, K. M.; Gavrila, D. M.; and Arras, K. O. 2019. Human motion trajectory prediction: A survey. *arXiv preprint arXiv:1905.06113* .

Salzmann, T.; Ivanovic, B.; Chakravarty, P.; and Pavone, M. 2020. Trajectron++: Multi-Agent Generative Trajectory Forecasting With Heterogeneous Data for Control. *arXiv preprint arXiv:2001.03093* .

Shoham, Y.; and Leyton-Brown, K. 2008. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Spica, R.; Falanga, D.; Cristofalo, E.; Montijano, E.; Scaramuzza, D.; and Schwager, M. 2018. A real-time game theoretic planner for autonomous two-player drone racing. *arXiv preprint arXiv:1801.02302* .

Sun, L.; Zhan, W.; and Tomizuka, M. 2018. Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2111–2117. IEEE.

Tang, C.; and Salakhutdinov, R. R. 2019. Multiple futures prediction. In *Advances in Neural Information Processing Systems*, 15398–15408.

Tian, R.; Li, S.; Li, N.; Kolmanovsky, I.; Girard, A.; and Yildiz, Y. 2018. Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts. In *2018 IEEE Conference on Decision and Control (CDC)*, 321–326. IEEE.

Treiber, M.; Hennecke, A.; and Helbing, D. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical review E* 62(2): 1805.

Wang, X.; and Klabjan, D. 2018. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations. *arXiv preprint arXiv:1801.02124* .

Wang, X.; Ma, S.; Goldfarb, D.; and Liu, W. 2017. Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization* 27(2): 927–956.

Zazo, S.; Macua, S. V.; Sánchez-Fernández, M.; and Zazo, J. 2016. Dynamic potential games with constraints: Fundamentals and applications in communications. *IEEE Transactions on Signal Processing* 64(14): 3806–3821.

Zhang, C.; Zhu, J.; Wang, W.; and Xi, J. 2020. Spatiotemporal Learning of Multivehicle Interaction Patterns in Lane-Change Scenarios. *arXiv preprint arXiv:2003.00759* .

Zhang, Q.; Filev, D.; Tseng, H.; Szwabowski, S.; and Langari, R. 2018. Addressing Mandatory Lane Change Problem with Game Theoretic Model Predictive Control and Fuzzy Markov Chain. In *2018 Annual American Control Conference (ACC)*, 4764–4771. IEEE.

Zhang, X.; Zhang, K.; Miehling, E.; and Basar, T. 2019. Non-Cooperative Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 9482–9493.