Efficient Model-based Multi-agent Reinforcement Learning via Optimistic Equilibrium Computation

Pier Giuseppe Sessa [1] **Maryam Kamgarpour** [*2] **Andreas Krause** [*1]

Abstract

We consider model-based multi-agent reinforcement learning, where the environment transition model is unknown and can only be learned via expensive interactions with the environment. We propose H-MARL (Hallucinated Multi-Agent Reinforcement Learning), a novel sample-efficient algorithm that can efficiently balance *exploration*, i.e., learning about the environment, and *exploitation*, i.e., achieve good equilibrium performance in the underlying general-sum Markov game. H-MARL builds high-probability confidence intervals around the unknown transition model and sequentially updates them based on newly observed data. Using these, it constructs an *optimistic hallucinated game* for the agents for which equilibrium policies are computed at each round. We consider general statistical models (e.g., Gaussian processes, deep ensembles, etc.) and policy classes (e.g., deep neural networks), and theoretically analyze our approach by bounding the agents' *dynamic regret*. Moreover, we provide a convergence rate to the equilibria of the underlying Markov game. We demonstrate our approach experimentally on an autonomous driving simulation benchmark. H-MARL learns successful equilibrium policies after a few interactions with the environment and can significantly improve the performance compared to non-optimistic exploration methods.

1. Introduction

Multi-Agent Reinforcement Learning (MA-RL) has shown promising successes in solving complex sequential decision-making tasks faced by multiple interacting agents, such as in robotics (Levine et al., 2016) and game playing (Mnih et al., 2015). However, its applicability to several real-world problems is still limited by the required large amount of training data. MA-RL methods can be classified as *model-free*, where agents are trained directly on the obtained rewards, and *model-based* where agents are trained based on an estimated model of the environment. For comprehensive surveys, see (Busoniu et al., 2008; Gronauer & Diepold, 2021).

A central challenge for MA-RL, which is crucial for its scalability to the real world, is the problem of trading off *exploration* with *exploitation* (Busoniu et al., 2008). Exploration is the ability to learn about the environment and generalize over unseen states and actions, hence being important to avoid suboptimal policies and enable task generalization. Exploitation of the data observed so far, on the other hand, is necessary to ensure that the agents' policies achieve high performance throughout the learning.

While there has been an extensive set of techniques addressing this challenge in single-agent RL (see, e.g., Jaksch et al. (2010); Luo et al. (2019); Curi et al. (2020) and references therein), it remains fairly unexplored in the MA-RL domain. In MA-RL, this is particularly difficult since the joint action space grows *exponentially* with the number of agents and, as a result, effective single-agent RL techniques (such as ϵ-greedy) can provably fail Mahajan et al. (2019). Recently, new methods have been proposed to circumvent these challenges and encourage exploration in different game settings, as discussed in Section 1.1. However, they are typically concerned with agents' asymptotic performance (i.e., neglecting agents' performance during learning), specific game settings (cooperative or two-player zero-sum), and tabular domains (i.e., finite state and action spaces).

In this work, we propose a novel sample-efficient algorithm for MA-RL which can efficiently balance exploration and exploitation. We consider *general-sum Markov games* (also known as mixed cooperative-competitive setting in the MA-RL literature) with *continuous action and state spaces*, and quantify agents' performance with the notion of *dynamic regret*. The dynamic regret measures the cumulative distance (throughout the learning rounds) of the played games from being at equilibrium. We provide a regret bound for our method, together with sample-efficient convergence rates

[*]Equal contribution [1]ETH Zürich, Rämistrasse 101, 8092 Zürich. [2]EPFL Lausanne, Rte Cantonale, 1015 Lausanne. Correspondence to: Pier Giuseppe Sessa <sessap@ethz.ch>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

to the equilibria of the Markov game. To the best of our knowledge, our results are the first guarantees in such a setting. We further illustrate our approach on an autonomous driving simulation benchmark.

1.1. Related Work

A set of different techniques have been proposed to encourage exploration in *cooperative* MA-RL, i.e, where groups of agents share common team-level objectives. Among those, Mahajan et al. (2019) enforce agents' exploration using a latent variable controlled by a hierarchical policy, Wang et al. (2020) use influence-based techniques, Liu et al. (2021a) proposes a normalized entropy-based method, while Zheng et al. (2021) achieve exploration by promoting agents' curiosity. Moreover, Mahajan et al. (2021) and Van Der Vaart et al. (2021) develop model-based algorithms which utilize low-rank tensor decompositions of rewards and the transition function. All of these techniques, however, are not applicable to our setting of general-sum Markov games where each agent is concerned with its own reward function. Moreover, while demonstrating good experimental performance, these approaches lack theoretical guarantees or consider only asymptotic convergence.

A theoretically grounded approach to guide exploration, which has been extensively studied in *single-agent* RL, is the celebrated *optimism in the face of uncertainty* (OFU) principle. In a nutshell, it consists of choosing actions that maximize an optimistic estimate of the agent's value function. OFU can efficiently balance exploration with exploitation and has been applied in several single-agent RL domains (e.g., Jaksch et al., 2010; Luo et al., 2019; Curi et al., 2020), yielding sample-efficient regret guarantees. Inspired by this line of work, our approach utilizes the OFU principle in our multi-agent domain. In particular, we utilize the model-based techniques of Curi et al. (2020) to compute optimistic agents' *equilibria* (as opposed to single-agent optimal policies), and generalize the obtained guarantees to our much more complex MA-RL setting.

Applications of the OFU principle in MA-RL are fairly unexplored, with a few recent exceptions. The line of works by Bai & Jin (2020), Xie et al. (2020), Bai et al. (2020), Jin et al. (2021), Loftin et al. (2021) and Chen et al. (2022) propose optimistic approaches which, however, consider only the special case of *two-player zero-sum* Markov games. Paszto r et al. (2021), instead, study the problem of optimistic mean-field control. In our N players general-sum setting, the first guarantees are by Liu et al. (2021b) who propose a centralized optimistic value iteration scheme by adding bonus terms to the estimated Q-functions. Their algorithm is applicable only to the *tabular* case (i.e., finite number of state S and actions A^i for each agent i) and enjoys a sample-complexity guarantee of $\tilde{\mathcal{O}}(H^4S^2\Pi_{i=1}^N A^i/\epsilon^2)$ to reach ϵ equilibria, where H is the game horizon. In the same setting, Mao & Başar (2022) shows that this can be improved to $\tilde{\mathcal{O}}(H^6S\max_i A^i/\epsilon^2)$ by a fully decentralized scheme. Compared to these works, we consider Markov games with *continuous* states and actions, a setting where the aforementioned methods do not apply. Similar to Liu et al. (2021b), our approach follows under the 'centralized training with decentralized execution' paradigm (Lowe et al., 2017). However, differently from Liu et al. (2021b), we construct optimistic value functions using general statistical models for the environment transition (such as Gaussian processes or deep ensembles), which allow exploiting the correlations in the transition function over continuous domains. Our guarantees capture the degrees of freedom in the environment transition function and the generalization ability of the used statistical model.

1.2. Our contributions

We propose H-MARL (Hallucinated MA-RL), a novel sample-efficient algorithm for model-based MA-RL which can efficiently guide exploration by hallucinating optimistic value functions for the agents. Regressing on past observed data, H-MARL builds confidence estimates around the true transition function using general statistical models. This allows exploiting correlations in the environment transition and generalizing for unseen game outcomes. Using these estimates, at each round H-MARL constructs upper confidence bounds on the agents' value functions and utilizes an equilibrium-finding subroutine to compute the respective agents' equilibria. We theoretically analyze our approach by bounding the agents' dynamic regret and providing a sample-complexity guarantee to reach equilibria of the underlying general-sum Markov game. To the best of our knowledge, ours are the first guarantees of this sort for continuous state and action spaces. We demonstrate our approach on an autonomous driving MA-RL benchmark, where H-MARL can quickly learn successful equilibrium policies and leads to superior performance compared to non-optimistic exploration methods.

2. Problem Setup

We consider a Multi-Agent Reinforcement Learning (MA-RL) problem, formulated as a stochastic (Markov) game (Shapley, 1953) among N agents over a finite episode of H steps. At each time step h, the environment's state is $s_h \in \mathcal{S} \subseteq \mathbb{R}^p$ and each agent i selects action $a_h^i \in \mathcal{A}^i \subseteq \mathbb{R}^q$. Then, each agent obtains reward $r^i(s_h, a_h^1, \dots, a_h^N)$ according to her reward function $r^i : \mathcal{S} \times \Pi_{i=1}^N \mathcal{A}^i \to \mathbb{R}$, and the environment transitions to state $s_{h+1} \sim P(\cdot|s_h, a_h^1, \dots, a_h^N)$ where P is the transition probability function. Each agent plays according to a policy $\pi^i : \mathcal{S} \to \mathcal{A}^i$ which maps states to actions (our results extend also to the partially observable case, where agents have access only to a subset of the state),

with the goal of maximizing her value function:

$$V^i(\pi^i, \pi^{-i}) = \mathbb{E}\left[\sum_{h=0}^{H-1} r^i(s_h, \pi^1(s_h), \dots, \pi^N(s_h))\right],$$

where we have used the notation π^{-i} to indicate the policies of all agents except agent i. For simplicity, we assume that the initial state s_0 and agent policies are deterministic, but our results can be naturally extended to the stochastic case. We let Π^i be the policy space of agent i (e.g., neural network policies, Foerster et al. (2016), Gaussian policies, Duan et al. (2016), etc.), and let $\boldsymbol{\Pi} := \Pi^1 \times \cdots \times \Pi^N$ be the joint policy space. Similarly, we define $\boldsymbol{\mathcal{A}} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ and let $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and $\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ be joint policy and action profiles, respectively. We also define $[N] := \{1, \dots, N\}$.

We make no assumption (e.g., cooperative or zero-sum) on the game rewards structure, and consider the most general class of general-sum Markov games, also known as mixed cooperative-competitive setting in MA-RL literature. In such a setting, natural solution concepts are game *equilibria*, i.e, outcomes from which rational agents' do not have incentives to deviate. We consider the most general class of equilibria, denoted as Coarse-Correlated Equilibria (CCE) (Moulin & Vial, 1978) defined as follows.

Def. 1 (CCE)**.** *A Coarse Correlated Equilibrium (CCE) is a distribution $\boldsymbol{\mathcal{P}}_\star$ over $\boldsymbol{\Pi}$ such that, for each agent i and any policy $\pi^i \in \Pi^i$,*

$$\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_\star}[V^i(\boldsymbol{\pi})] \geq \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}_\star^{-i}}[V^i(\pi^i, \pi^{-i})].$$

CCEs generalize other equilibrium notions such as Nash equilibria (Nash, 1950), and have received significant interest from the learning community because they can be computed by decentralized algorithms in polynomial time (Cesa-Bianchi & Lugosi, 2006; Marris et al., 2021; Mao & Başar, 2022). In practice, one can usually compute CCEs only up to some approximation factor. We denote such outcomes as ϵ-CCE, where the condition above is satisfied only up to a $\epsilon > 0$ accuracy.

Model-based MA-RL In this work, we take a model-based approach to compute game equilibria. Namely, we assume agents' reward functions are known[1] (often, they are suitably designed depending on the agents' goals), and the environment's state transition follows the dynamics:

$$s_{h+1} = f(s_h, a_h^1, \dots, a_h^N) + w_h, \tag{1}$$

where $f : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathcal{S}$ is the environment transition function, and w_h is zero-mean sub-Gaussian noise i.i.d. over time.

[1]The proposed approach can also be extended to unknown reward functions, but we assume them known for ease of exposition.

Transition function $f(\cdot)$ is a-priori *unknown* and can only be learned online via sequential interactions with the environment. Hence, the learning protocol goes as follows. At each round t, agents choose policies $\boldsymbol{\pi}_t = \{\pi_t^1, \dots, \pi_t^N\}$ (possibly via randomization), play an episode of the Markov game, observe corresponding state transitions data $\mathcal{D}_t = \{s_h^t, \mathbf{a}_h^t, h = 0, \dots, H-1\}$ and use these to improve their policies for the next round.

Interacting with the environment can be very costly, hence we seek to minimize the number of interaction rounds. At the same time, however, we want agents to achieve good performance across the played Markov games. To do so, we measure the performance of agent i after T rounds by its dynamic regret:

Def. 2 (Dynamic regret)**.**

$$R^i(T) := \sum_{t=1}^{T} \max_{\pi \in \Pi^i} \mathbb{E}_{\pi_t^{-i}}\left[V^i(\pi, \pi_t^{-i})\right] - \mathbb{E}_{\boldsymbol{\pi}_t}\left[V^i(\boldsymbol{\pi}_t)\right].$$

The dynamic regret measures the cumulative difference between the best achievable expected value of the game at each round t, and the actual value obtained. Thus, it is an indicator of how far the played games were from being at equilibrium (note that if $\boldsymbol{\pi}_t$ is sampled from a CCE at each round t, $R^i(T) = 0$ for all i by Def.1). The notion of dynamic regret is widely adopted in multi-agent learning (see, e.g., Zhang et al., 2020b) and represents a stronger performance benchmark than the *static* regret (Cesa-Bianchi & Lugosi, 2006), which compares the obtained rewards with the ones of the best fixed policy in hindsight. Intuitively, we can compete with such a stronger benchmark because we allow centralized training of MA-RL agents and learn to play a time-invariant Markov game repeated over rounds.

In the next section, we propose an algorithm that can efficiently balance exploration (learning about the environment) and exploitation (achieving low regret). Provided "sufficient model generalizability" (as formalized later on), it ensures that the agents' regret grows sublinearly with the interaction rounds T, i.e., $\lim_{T\to\infty} \frac{1}{T} R^i(T) \to 0$.

3. The H-MARL Algorithm

We propose H-MARL, a novel sample-efficient algorithm for the model-based MA-RL setting defined above. The algorithm falls under the widely adopted 'centralized training with decentralized execution' (CTDE) paradigm (Lowe et al., 2017), i.e, agent policies are trained centrally but, once deployed, they require only agents' local information. The proposed method utilizes two main building blocks: 1) learning the transition model from observed data, and 2) hallucinating optimistic agents' value functions. We describe them in detail next.

3.1. Learning the transition model from data

At the end of each round t, we can use observed transition data (along with possibly available offline data) $\{\mathcal{D}_\tau\}_{\tau=1}^t$ to estimate the transition function f via its posterior *mean* and *confidence* functions

$$\mu_t(s, \mathbf{a}) \in \mathbb{R}^p, \quad \Sigma_t(s, \mathbf{a}) \in \mathbb{R}^{p \times p},$$

respectively. For this purpose, a variety of statistical models can be used, depending on the problem. Statistical models allow exploiting correlations in the observed data and generalize for non-visited states and joint actions. A concrete example are Gaussian Processes (GPs, cf., Williams & Rasmussen, 2006). GPs are powerful non-parametric models widely used in Bayesian optimization and have recently found new application areas, e.g., in online learning (Sessa et al., 2019; 2020) and model-based RL (Curi et al., 2020; 2021). According to the GP model, μ_t and Σ_t are obtained via kernel-ridge regression on the observed data. Popular kernel choices include linear, squared exponential, and Màtern kernels (Srinivas et al., 2010). For higher dimensional state and action spaces, a more scalable alternative is represented by deep Neural Network (NN) ensembles (Lee et al., 2015; Lakshminarayanan et al., 2016). In such a case, an ensemble of NNs can be trained on the data $\{\mathcal{D}_\tau\}_{\tau=1}^t$ (e.g., with bootstrapping techniques or by random shuffling the whole dataset), and posterior mean μ_t and confidence Σ_t are computed by aggregating the ensemble predictions.

We make no assumptions on the used statistical model, except that it is (conservatively) *calibrated*:

Assumption 1. *We assume the statistical model is calibrated, i.e., there exists $\beta_t \in \mathbb{R}_+$ such that with probability at least $1-\delta$, $|f(s, \mathbf{a}) - \mu_t(s, \mathbf{a})| \leq \beta_t \sigma_t(s, \mathbf{a})$ holds coordinate-wise, $\forall s, \forall \mathbf{a}, \forall t$, where $\sigma_t(\cdot) = diag(\Sigma_t(\cdot))$.*

A calibrated model allows us to bound, with a given confidence, how the trajectories predicted by the learned model plausibly differ from those corresponding to the true environment. We note that GP models readily satisfy Assumption 1, provided that $f(\cdot)$ has a bounded norm in the reproducing kernel Hilbert space associated to the used kernel function (cf., Srinivas et al., 2010; Chowdhury & Gopalan, 2017). Moreover, in case of ensemble NNs models, different calibration techniques have been proposed in the literature, e.g., by Kuleshov et al. (2018) and Zhang et al. (2020a).

3.2. Hallucinating agents' value functions

Using calibrated posterior mean and confidence $\mu_t(\cdot), \Sigma_t(\cdot)$, the proposed algorithm consists of building hallucinated *optimistic* value functions for the agents. For each agent i,

Algorithm 1 The H-MARL algorithm

Require: Agents' policy spaces $\Pi^1, \dots, \Pi^N$.
1: **for** $t = 1, \dots, T$ **do**
2: $\quad \boldsymbol{\mathcal{P}}_t \leftarrow$ Find-CCE$\big(\text{UCB}_{t-1}^1(\cdot), \dots, \text{UCB}_{t-1}^N(\cdot)\big)$, with $\text{UCB}_{t-1}^i(\cdot)$ defined in Eq. (2).
3: $\quad$ Episode rollout using policies

$$\boldsymbol{\pi}_t = (\pi_t^1, \dots, \pi_t^N) \sim \boldsymbol{\mathcal{P}}_t$$

4: $\quad$ Update transition model $\mu_t(\cdot,\cdot)$, $\Sigma_t(\cdot,\cdot)$, using observed H transitions.

these are obtained as:

$$\text{UCB}_t^i(\boldsymbol{\pi}) = \max_{\eta(\cdot) \in [-1,1]^p} \mathbb{E}\left[\sum_{h=0}^{H-1} r^i(s_h, \mathbf{a}_h)\right] \tag{2}$$
$$\text{s.t.} \quad \mathbf{a}_h = \boldsymbol{\pi}(s_h)$$
$$s_h = \mu_t(s_{h-1}, \mathbf{a}_{h-1}) + \beta_t \cdot \Sigma_t(s_{h-1}, \mathbf{a}_{h-1})\eta(s_{h-1}, \mathbf{a}_{h-1}) + w_h \,.$$

The function $\text{UCB}_t^i(\cdot)$ maps a joint policy profile $\boldsymbol{\pi}$ to an optimistic estimate of the value function for agent i. The optimism is injected through the auxiliary function $\eta(\cdot)$ which, for each state and agents' joint action, selects the state transition that leads to the largest expected cumulative reward, yet being plausible (by Assumption 1 and since $\eta(s, \mathbf{a}) \in [-1, 1]^p, \forall (s, \mathbf{a})$). More precisely, under Assumption 1 it holds $\text{UCB}_t^i(\boldsymbol{\pi}) \geq V^i(\boldsymbol{\pi})$, for all joint policies $\boldsymbol{\pi}$ and all rounds $t \geq 1$, as stated in Lemma 1 in Appendix A. Hence, the functions $\text{UCB}_t^i(\cdot)$ are provable upper confidence bounds for the agents' value functions. Their computational bottleneck is represented by the outer maximization over general functions $\eta(\cdot)$. To alleviate this, in Section 3.4 we provide a practical approximation of $\text{UCB}_t^i(\cdot)$ via sampling, which we also use in our experiments. Finally, we note that Eq. (2) can be viewed as the multi-agent generalization of the hallucinated value function proposed by Curi et al. (2020) for single-agent RL. These functions are at the core of the proposed approach. Indeed, our proposed H-MARL algorithm utilizes an equilibrium finding subroutine to compute, at each round t, a CCE of the hallucinated Markov game defined by the value functions $\{\text{UCB}_{t-1}^i(\cdot), i = 1, \dots, N\}$. Let $\boldsymbol{\mathcal{P}}_t$ be the computed CCE equilibrium. Then, agents play the Markov game using equilibrium policies $\boldsymbol{\pi}_t \sim \boldsymbol{\mathcal{P}}_t$, and the transition models μ_t, Σ_t are updated based on the newly observed data. We summarize our overall approach in Algorithm 1.

We leave the equilibrium computation step (Line 2 in Algorithm 1) very general, as this can be achieved by various MA-RL methods for general-sum Markov games. Importantly, because equilibrium is computed with respect to the hallucinated value functions, this step *does not require interacting*

with the true environment and hence sample-efficiency is not crucial here. Accuracy of the returned equilibrium can be traded off with its computational complexity and different algorithms are more suitable than others, depending on the game. A list of practical methods has demonstrated good empirical performance in computing equilibrium policies, e.g., using independent learners, actor-critic formulations (Iqbal & Sha, 2019), policy gradients (Lowe et al., 2017), or exploiting mean-field approximations (Yang et al., 2018). Provably convergent approaches also exist, e.g., using optimistic (Liu et al., 2021b) or decentralized (Mao & Başar, 2022) Q-learning. For ease of exposition we assume an exact CCE is computed at each round, but we will also discuss the case where only ϵ-CCEs are obtained.

3.3. Theoretical guarantees

We now present theoretical guarantees for H-MARL. More specifically, we obtain a dynamic regret bound for the agents after T rounds. Additionally, we provide an offline sample-complexity bound on the number of rounds T to reach an ϵ-CCE of the underlying Markov game.

Our guarantees depend on the following quantity, which characterizes the complexity of the transition model:

$$\mathcal{I}_T := \max_{\substack{\mathcal{D}_1,\dots,\mathcal{D}_T \\ \mathcal{D}_i \subset \mathcal{S}\times\boldsymbol{\mathcal{A}} \\ |\mathcal{D}_i|=H}} \sum_{t=1}^{T} \sum_{(s,\mathbf{a})\in\mathcal{D}_t} \|\sigma_{t-1}(s,\mathbf{a})\|_2^2 . \quad (3)$$

It quantifies the maximum predictive uncertainty about the model, where the worst case is taken over all possible observed transitions up to round T. Intuitively, easier transition models should be learned with less uncertainty, thus leading to a smaller $\mathcal{I}_T$. Curi et al. (2020) define the quantity $\mathcal{I}_T$ for the single-agent case and show that, although it is generally impossible to compute, it can be bounded for the special case of GP models. The same considerations apply to our setting: it holds $\mathcal{I}_T \leq pH\gamma_{HT}$, where γ_{HT} is the *maximum information gain* about f from HT noisy observations, a typical quantity in Bayesian optimization (Srinivas et al., 2010; Chowdhury & Gopalan, 2017). Known bounds for γ_{HT} exist depending on the kernel, e.g., $\gamma_{HT} \leq \mathcal{O}(\log(HT)^{d+1})$ and $\gamma_{HT} \leq \mathcal{O}(d\log(HT))$, respectively for squared-exponential and linear kernels, where $d = p + Nq$ is the domain dimension (Srinivas et al., 2010).

Moreover, the obtained guarantees rely on the following Lipschitz assumptions.

Assumption 2. *We assume the transition function f, reward functions r^i, policies $\pi \in \Pi^i$, and the posterior standard deviation function σ_t are Lipschitz continuous w.r.t. $\|\cdot\|_2$ with constants L_f, L_r, L_π, and L_σ, respectively for all $i \in [N]$ and $t \geq 0$.*

Lipschitz continuity of the transition function f is required for generalization, otherwise if f changes too abruptly we expect any efficient model-based method which aims to learn it to fail. Moreover, lipschitzess of rewards and policies is not a restrictive assumption since they are typically hand-designed. In addition, GP models lead to Lipschitz continuous posterior standard deviations, according to the kernel metric (cf., Curi et al., 2020, Lemma 13).

The following main theorem provides a dynamic regret bound for H-MARL, as a function of the different game quantities. Its proof is relegated to Appendix A.

Theorem 1 (Dynamic regret bound)**.** *Let Assumptions 1 and 2 be satisfied. After T rounds, the* H-MARL *algorithm ensures that, with probability $1-\delta$, each agent i has bounded dynamic regret:*

$$R^i(T) \leq \bar{L} H^{1.5} \sqrt{T\mathcal{I}_T},$$

where $\bar{L} = \mathcal{O}\big(N^{H/2} L_\pi^{H/2} (\bar{\beta}^H L_\sigma^H + L_f^H) + \log(1/\delta)\big)$, $\bar{\beta} = \max_t \beta_t$, and $\mathcal{I}_T$ is the complexity measure defined in (3).

Theorem 1 shows that the overall regrets' rate, as a function of the interaction rounds T, depends on the growth rate of $\mathcal{I}_T$ and hence on the generalization ability of the statistical model (if the model does not generalize at all, we expect $\mathcal{I}_T \geq \Omega(T)$ and the agents' regrets grow superlinearly). In case of GP models, one has $\bar{\beta} = \mathcal{O}(\sqrt{\gamma_{HT}})$ and $\mathcal{I}_T \leq pH\gamma_{HT}$, yielding overall regret rates of $R^i(T) \leq \mathcal{O}(N^{H/2} H^2 \sqrt{pT}\, \gamma_{HT}{}^{(H+1)/2})$. Substituting the bounds on the maximum information gain γ_{HT}, one obtains *sub-linear* regrets for commonly used kernels such as the linear and squared exponential kernel. We note that the regret rates of Theorem 1 generalize the single-agent optimization guarantee of Curi et al. (2020), with an additional multiplicative factor of $\mathcal{O}(N^{H/2})$ representing the price of dealing with a multi-agent environment. Finally, note that Theorem 1 assumes that an exact CCE is computed at each round (Line 2 of Algorithm 1), but in Appendix A we discuss the more general case where only ϵ_t-CCEs are obtained.

While Theorem 1 concerns the agents' performance throughout the interaction rounds, the next theorem provides an *offline* sample complexity guarantee (i.e., on the performance achieved after T rounds) for H-MARL, characterizing a sufficient number of rounds T to reach an ϵ-CCE of the underlying Markov game. First, for each joint policy $\boldsymbol{\pi}$ and each agent i, we define the *lower* confidence estimate $\mathrm{LCB}_t^i(\boldsymbol{\pi})$ as the solution of Eq. (2) where the outer maximization is replaced by a minimization over $\eta(\cdot)$.

Theorem 2 (Offline performance - Convergence to ϵ-CCE)**.** *Let Assumptions 1 and 2 be satisfied and assume we run* H-MARL *for T rounds. Moreover, consider distribution $\boldsymbol{\mathcal{P}}_{t^\star}$ such that*

$$t^\star = \arg\min_{t\in[T]} \max_{i\in[N]} \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t} [\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) - \mathrm{LCB}_{t-1}^i(\boldsymbol{\pi})].$$

For a given accuracy $\epsilon \geq 0$, if

$$T \geq \Omega\left(\frac{1}{\epsilon^2} \cdot \bar{L}^2 H^3 \mathcal{I}_T\right),$$

then $\boldsymbol{\mathcal{P}}_{t^\star}$ is an ϵ-CCE of the underlying Markov game with probability at least $1-\delta$.

The proof of Theorem 2 can be found in Appendix A. Intuitively, it is obtained by showing that, after T rounds, distribution $\boldsymbol{\mathcal{P}}_{t^\star}$ is an ϵ_T-CCE of the Markov game, where the approximation factor ϵ_T is bounded as $\epsilon_T \leq \mathcal{O}\big(T^{-\frac{1}{2}} \bar{L} H^{1.5} \sqrt{\mathcal{I}_T}\big)$. This implies the result.

3.4. Practical implementation via sampling

Although the optimistic game values in Eq. (2) are hard to compute, we propose the following more practical approximation:

$$\widetilde{\mathrm{UCB}}_t^i(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{h=0}^{H-1} r^i(s_h^\star, \mathbf{a}_h)\right] \tag{4}$$

$$\begin{aligned}
\text{s.t.} \quad & \eta_h^j \sim \mathrm{Unif}([-1,1]^p), \quad j = 1, \dots, Z \\
& s_h^j = \mu_t(s_{h-1}^\star, \mathbf{a}_{h-1}) \\
& \qquad + \beta_t \cdot \Sigma_t(s_{h-1}^\star, \mathbf{a}_{h-1}) \cdot \eta_h^j + w_h \\
& s_h^\star = \arg \max_{s_h^j, j \in \{1,\dots,S\}} r^i(s_h^j, \boldsymbol{\pi}(s_h^j)) \quad (5) \\
& \mathbf{a}_h = \boldsymbol{\pi}(s_h^\star)\,.
\end{aligned}$$

For any given joint policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, function $\widetilde{\mathrm{UCB}}_t^i(\boldsymbol{\pi})$ approximates the optimistic value function $\mathrm{UCB}_t^i(\boldsymbol{\pi})$ (computed as in (2)) via the following sampling method: At each time h, a finite set of Z auxiliary parameters $\{\eta_h^j, j = 1, \dots, Z\}$ is sampled uniformly from $[-1,1]^p$. These are used to compute Z plausible states $s_h^j, j = 1, \dots, Z$, according to the previous state, joint actions, and the hallucinated transition model. Among these, only the state $s_h^\star$ leading to the largest reward is selected and used for the next transition. This process continues up to time $t = H-1$. Further, the outer expectation can be approximated by taking the empirical mean over multiple episodes.

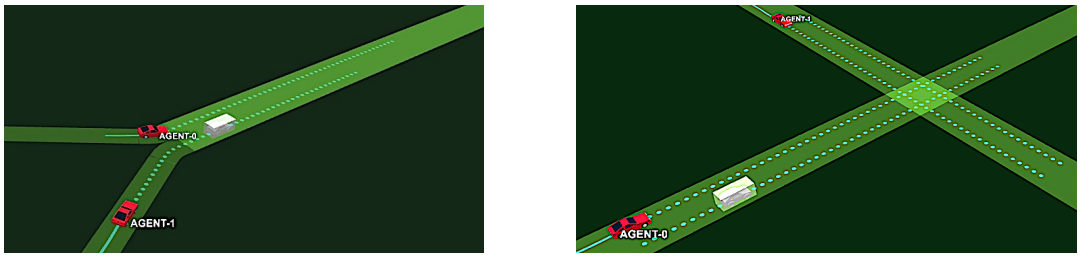
Note that $\widetilde{\mathrm{UCB}}_t^i(\boldsymbol{\pi}) \leq \mathrm{UCB}_t^i(\boldsymbol{\pi})$ for all $\boldsymbol{\pi}$, since the computed state sequences $s_h^\star$ are feasible trajectories for the maximization problem in Eq. (2). However, Eq. (4) offers significant computational advantages: 1) the intractable maximization over functions $\eta(\cdot)$ is replaced by selecting the best out of finitely many samples, 2) the optimization over the cumulative reward is broken down into greedily optimizing the step-wise rewards. While the first approximation can be alleviated by considering a large number of samples, the second one can be problematic in case of sparse rewards (i.e., where the optimal sequence of parameters η_h^j depends on the whole state trajectory). In such a case, instead of greedily selecting state $s_h^\star$ as in Eq. (5), one could keep track of Z parallel trajectories $\{s_h^j\}_{h=0}^{H-1}, j = 1, \dots, Z$, according to the sampled auxiliary variables η_h^j and, only after time H, select $\{s_h^\star\}_{h=0}^{H-1}$ to be the one leading to the largest cumulative reward. In the latter approach, however, a larger Z should be selected to obtain reasonable approximations.

4. Experiments: Autonomous driving SMARTS benchmark

We demonstrate our approach on an autonomous driving application. The goal is to find successful equilibrium policies for Autonomous Vehicles (AVs) when driving on common roads. Crucially, once deployed, AVs must interact with other non-controllable human-driven (HD) vehicles, which can significantly affect the AVs' performance. We make the realistic assumption that at training time the behavior of HD vehicles is a-priori *unknown* and can only be inferred by *online interactions* with the real-world, or with an expensive simulator (e.g., requiring humans-in-the-loop, cf., Reddy et al., 2018). This problem can be naturally formulated according to our multi-agent RL framework. Indeed, we show below that myopically considering it as a single-agent RL problem leads to significantly inferior performance. The transition function $f(\cdot)$ includes the unknown HD vehicles' behavior (e.g., how do they act at any given time, given the current environment state) and can be learned from observed trajectories from past rounds $1, \dots, t-1$. Note that the environment transition function also includes the AV dynamics, which are assumed to be known for simplicity.

Hence, given a pre-designed driving scenario, we seek to evaluate the performance of the policies computed by H-MARL for the AVs after each interaction with the environment. In particular, we are interested in comparing the proposed optimistic exploration strategy with the greedy and perhaps most naive approach of using the predictive posterior mean of $f(\cdot)$ to plan for the next round. We also consider Thompson Sampling (TS) (Russo et al., 2018) as a natural exploration baseline: in our setup TS consists of sampling, at each policy evaluation step, state s_{h+1} from the posterior Gaussian distribution with mean $\mu_t(s_h, \mathbf{a}_h)$ and covariance $\Sigma_t(s_h, \mathbf{a}_h)$. Note that this is substantially different from the proposed optimistic H-MARL where, among plausible next states, we select the ones leading to the highest agents' reward. Moreover, to assess the quality of learning, we compare all these approaches with the idealized benchmark of knowing the HD model in advance. First, we describe the used environment, the agents' specifications, and the HD vehicles' model.

SMARTS environment. We run our experiments using the open-source SMARTS autonomous driving platform (Zhou



(a) Lane merging scenario (b) Intersection scenario

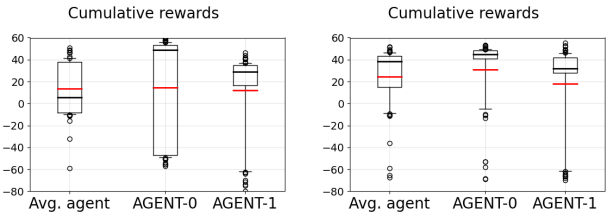
Figure 1. (a) **Lane merging scenario:** both agents want to maximize progress, while AGENT-0 also wants to merge into AGENT-1's lane. The grey car is a human-driven (HD) vehicle with random initial speed. (b) **Intersection scenario:** AGENT-0 wants to turn right, while AGENT-1 wants to proceed straight on its lane. The HD vehicle has a random initial speed and wants to cross the intersection.

et al., 2020), a recently proposed benchmark for MA-RL research. SMARTS provides realistic simulations for autonomous vehicles on configurable road environments, as well as interactions with background traffic vehicles, which for our scope represent HD vehicles. Vehicle dynamics are simulated by the *Bullet physics engine* (Coumans & Bai, 2016–2021), while SUMO (Krajzewicz et al., 2002) is the background traffic provider. At a higher level, SMARTS is integrated with the reinforcement learning RLlib (Liang et al., 2018) library, allowing us to train deep RL policies for the AVs (agents) using existing MA-RL baselines.

Observations, policies, rewards. Each agent is assigned a mission, which is represented by a start and a goal position. Simulation steps are of 0.1s and, at each step, the state observed by agent i consists of: relative position to the goal, the distance to the lane's center, speed, steering, and heading errors, and the states of its neighboring vehicles. Each agent has a discrete action space: $\{\text{keep lane}, \text{slow down}, \text{turn right}, \text{turn left}\}$ and a policy parametrized by a deep neural network with 2 hidden layers of 256 units and $\tanh$ activations (we use default policies from Zhou et al. (2020)). The reward of agent i at each time h is $r^i(s_h, \mathbf{a}_h) = r^i_{\text{bonus}}(s_h, \mathbf{a}_h) - r^i_{\text{penalty}}(s_h, \mathbf{a}_h)$, where r_{bonus} rewards progress (i.e., driven distance) and reaching the goal, while r_{penalty} penalizes acceleration, sharp turns, collisions, and distances from lane center and goal.

HD vehicles' model. At test time, HD vehicles are controlled by the SUMO traffic provider, which is a black box for our purposes. To learn their behavior, we use a GP model which maps the current HD state and its relative position to the other vehicles, to the next HD state (i.e., one-step-ahead prediction). We use a Matérn kernel for predicting the speed change (as we expect it to be rather nonsmooth), and a squared exponential kernel to predict its change in position.

H-MARL implementation. To compute the equilibrium policies at each round t (Line 2 of Algorithm 1), we use independent Deep Q-Networks (DQN, Mnih et al., 2015) and we let $\mathcal{P}_{t+1}$ be the uniform distribution over the last 4 policy iterations (or checkpoints). This avoids



(a) Single-agent optima. (b) Multi-agent equilibrium.

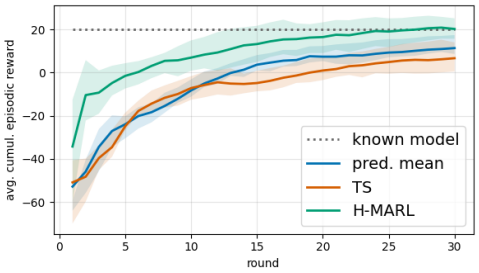
Figure 2. **Lane merging scenario.** MA-RL equilibrium policies (b) lead to higher average and individual rewards for the agents, with respect to using single-agent optimized policies (a). Boxplots of 15-85th percentiles over 50 runs (mean in red, median in black).

non-convergence behaviors and mimics the way CCEs are computed by independent no-regret learning in normal-form games, see, e.g., Cesa-Bianchi & Lugosi (2006). We chose independent DQN for its computational efficiency and because it was shown by Zhou et al. (2020) to find good driving policies. The hallucinated optimistic value functions $\text{UCB}^i_t(\cdot)$ are approximated by the sampling approach of Eq. (4) with $Z = 5$ samples at each time step and $\beta_t = 1.0$. This effectively corresponds to sampling, at each policy evaluation step, Z plausible HD vehicles' states and selecting the one that leads to the highest reward. Additional details of our experimental setup are provided in Appendix B.

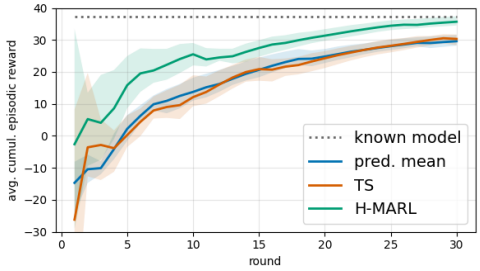
4.1. Results

We consider two different realistic scenarios corresponding to lane merging and intersection.

Lane merging scenario. The lane merging scenario is depicted in Figure 1a. There are two agents (AGENT-0 and AGENT-1) whose goal is to maximize progress (i.e., to reach a goal position at the end of the road), while AGENT-0's goal is also to merge into AGENT-1's lane within a horizon of $H = 150$ steps. In addition, a HD vehicle (grey car) drives on its lane with a random initial speed. We expect that depending on its speed, the agents coordinate to either overtake the HD vehicle and merge, or drive behind it while



(a) Lane merging scenario



(b) Intersection scenario

Figure 3. Average game values (mean and 30-70th percentiles) as a function of the interaction rounds, when equilibrium policies are computed according to the optimistic H-MARL algorithm, Thompson Sampling (TS) exploration strategy, or when using the predictive posterior mean about the transition function (no explicit exploration). We also compare against the idealized benchmark of knowing the HD model in advance.

allowing AGENT-0 to merge.

Intersection scenario. The intersection scenario is depicted in Figure 1b. AGENT-0 (bottom part of the figure) wants to turn right, while AGENT-1's goal is to proceed straight on its lane. The HD vehicle has a random initial speed and wants to cross the intersection. Horizon is of $H = 150$ steps.

First, we demonstrate the superior performance of modeling the problem via MA-RL, as opposed to optimizing single-agent policies. We consider the lane merging scenario and assume the HD model is known. Then, we i) compute a game equilibrium $\boldsymbol{\mathcal{P}}_\star$ as outlined above and ii) optimize single agent policies $\pi^i_\star$. To obtain $\pi^i_\star$, we consider a single-agent environment for agent i, replacing the other agent with a HD vehicle driving on the same lane. We expect ii) to produce inferior policies for the agents, since the learned policies neglect the presence of other AVs and thus miss any opportunity for coordination. In Figure 2 we report agents' reward from 50 scenario evaluations when the used policies come from either of the two approaches. Equilibrium policies lead to higher average and individual rewards for the agents, consistent with our intuition. In particular, the optimal single-agent policy for AGENT-1 often consists of breaking and driving behind the HD vehicle. Instead, we observed that under equilibrium policies both agents learn better coordination maneuvers allowing AGENT-1 to more often overtake (see Appendix B for additional details on this).

Motivated by the above considerations, we evaluate the proposed H-MARL approach in computing equilibrium policies at each round. To evaluate the performance of a joint distribution $\boldsymbol{\mathcal{P}}_t$, we use the average agents' cumulative reward $\bar{V}(\boldsymbol{\mathcal{P}}_t) = \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t} \frac{1}{N} \sum_{i=1}^{N} \sum_{h=0}^{H-1} r^i(s_h, \boldsymbol{\pi}(s_h))$, since we expect this to be high at equilibria (Zanardi et al. (2021) showed that this is typical of driving games under some assumptions).

In Figure 3, we plot the average game values $\frac{1}{T}\sum_{t=1}^{T} \bar{V}(\boldsymbol{\mathcal{P}}_t)$ as a function of the interaction rounds T, when using H-MARL, Thompson Sampling (TS) exploration, or the predictive posterior mean to compute the agents' policies at each round (we denote this approach as "pred. mean"), respectively. After ≈ 30 interaction rounds (corresponding to ≈ 3000 observed input-output data points to learn the HD model), H-MARL returns equilibrium policies that have comparable performance to knowing the exact HD model in advance (see Figure 5a in Appendix B for an illustration of successful merging and crossing maneuvers that result from our approach). Moreover, it displays consistently faster learning curves with respect to the "pred. mean" and TS baselines. This is due to the fact that H-MARL encourages the AVs to optimistically explore different HD vehicle's behaviors, i.e., the diverse ways the HD vehicle can change its speed based on their position. Instead, the "pred. mean" and TS baselines learn about it only indirectly, by greedily choosing the best policies according to the posterior beliefs coming from past observed trajectories. Further differences between the three approaches can also be observed in Table 1. There, we report the averaged completion rates (i.e., when goal positions are reached) and the average episodes' completion time (faster goal reaching corresponds to higher rewards for the agents) experienced by the agents across all the learning

	Avg. completion rate during learning	Avg. completion time during learning
pred. mean	72.0 %	8.90 s
TS	69.9 %	8.87 s
H-MARL	**80.9** %	**8.66** s

(a) Lane merging scenario

pred. mean	88.8 %	9.22 s
TS	87.4 %	9.04 s
H-MARL	**91.9** %	**9.02** s

(b) Intersection scenario

Table 1. Average (across all learning rounds) of the agents' mission completion rates (higher is better) and episodes' completion time (lower is better), when using "pred. mean", TS, or H-MARL.

rounds. The policies computed by H-MARL lead to higher completion rates and lower completion times overall.

In Appendix B, we further compare our model-based approach with *model-free* methods. The latter neglect the underlying problem structure, requiring a significantly larger number of environment interactions to achieve comparable rewards.

5. Conclusions

We have considered a model-based multi-agent reinforcement learning problem, where the environment transition function is unknown and can only be learned by costly interactions with the environment. We have proposed H-MARL (Hallucinated Multi-Agent Reinforcement Learning), a novel sample-efficient algorithm that can provably balance exploration with exploitation. H-MARL constructs statistical confidence bounds around the unknown transition function and uses them to build a hallucinated optimistic game for the agents. We have theoretically analyzed our approach by bounding the agents' dynamic regret and deriving a sufficient number of iterations to converge to approximate equilibria of the underlying Markov game. To the best of our knowledge, ours are the first guarantees in general-sum Markov games with continuous states and actions. We have demonstrated our approach on an autonomous driving simulation benchmark, where it showed fast convergence and outperformed non-optimistic and model-free methods.

Acknowledgments

This project received funding from the Swiss National Science Foundation, under the grant SNSF 200021_172781 and the NCCR Automation grant 51NF40 180545, and by the European Union's ERC grant 815943.

References

Bai, Y. and Jin, C. Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 551–560, 2020.

Bai, Y., Jin, C., and Yu, T. Near-optimal reinforcement learning with self-play. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Busoniu, L., Babuska, R., and De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Chen, Z., Zhou, D., and Gu, Q. Almost optimal algorithms for two-player zero-sum linear mixture markov games. In *International Conference on Algorithmic Learning Theory*, pp. 227–261, 2022.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2017.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

Curi, S., Berkenkamp, F., and Krause, A. Efficient model-based reinforcement learning through optimistic policy search and planning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Curi, S., Bogunovic, I., and Krause, A. Combining pessimism with optimism for robust and efficient model-based deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning (ICML)*, pp. 1329–1338, 2016.

Foerster, J. N., Assael, Y. M., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Gronauer, S. and Diepold, K. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, pp. 1–49, 2021.

Iqbal, S. and Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 2961–2970, 2019.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

Jin, C., Liu, Q., and Yu, T. The power of exploiter: Provable multi-agent rl in large state spaces. *arXiv preprint arXiv:2106.03352*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirschner, J. and Krause, A. Information directed sampling and bandits with heteroscedastic noise. In *Conference On Learning Theory (CoLT)*, pp. 358–384, 2018.

Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pp. 183–187, 2002.

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, pp. 2796–2804, 2018.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D. J., and Batra, D. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J., Jordan, M., and Stoica, I. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 3053–3062, 2018.

Liu, I.-J., Jain, U., Yeh, R. A., and Schwing, A. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 6826–6836, 2021a.

Liu, Q., Yu, T., Bai, Y., and Jin, C. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning (ICML)*, pp. 7001–7010, 2021b.

Loftin, R., Saha, A., Devlin, S., and Hofmann, K. Strategically efficient exploration in competitive multi-agent reinforcement learning. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 1587–1596, 2021.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Neural Information Processing Systems (NeurIPS)*, pp. 6382–6393, 2017.

Luo, Y., Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations (ICLR)*, 2019.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Mahajan, A., Samvelyan, M., Mao, L., Makoviychuk, V., Garg, A., Kossaifi, J., Whiteson, S., Zhu, Y., and Anandkumar, A. Tesseract: Tensorised actors for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021.

Mao, W. and Başar, T. Provably efficient reinforcement learning in decentralized general-sum markov games. *Dynamic Games and Applications*, pp. 1–22, 2022.

Marris, L., Muller, P., Lanctot, M., Tuyls, K., and Grapael, T. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. *arXiv preprint arXiv:2106.09435*, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Moulin, H. and Vial, J.-P. Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.

Nash, J. F. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

Pasztor, B., Bogunovic, I., and Krause, A. Efficient model-based multi-agent mean-field reinforcement learning. *arXiv preprint arXiv:2107.04050*, 2021.

Reddy, S., Dragan, A. D., and Levine, S. Shared autonomy via deep reinforcement learning. In *Robotics: Science and Systems*, 2018.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. No-regret learning in unknown games with correlated payoffs. In *Neural Information Processing Systems (NeurIPS)*, December 2019.

Sessa, P. G., Bogunovic, I., Krause, A., and Kamgarpour, M. Contextual games: Multi-agent learning with side information. In *Neural Information Processing Systems (NeurIPS)*, December 2020.

Shapley, L. S. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Van Der Vaart, P., Mahajan, A., and Whiteson, S. Model based multi-agent reinforcement learning with tensor decompositions. *arXiv preprint arXiv:2110.14524*, 2021.

Wang, T., Wang, J., Wu, Y., and Zhang, C. Influence-based multi-agent exploration. In *International Conference on Learning Representations (ICLR)*, 2020.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Xie, Q., Chen, Y., Wang, Z., and Yang, Z. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory (CoLT)*, pp. 3674–3682, 2020.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 5571–5580, 2018.

Zanardi, A., Mion, E., Bruschetta, M., Bolognani, S., Censi, A., and Frazzoli, E. Urban driving games with lexicographic preferences and socially efficient nash equilibria. *IEEE Robotics and Automation Letters*, 6(3):4978–4985, 2021.

Zhang, J., Kailkhura, B., and Han, T. Y.-J. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*, pp. 11117–11128, 2020a.

Zhang, L., Lu, S., and Yang, T. Minimizing dynamic regret and adaptive regret simultaneously. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 309–319, 2020b.

Zheng, L., Chen, J., Wang, J., He, J., Hu, Y., Chen, Y., Fan, C., Gao, Y., and Zhang, C. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Neural Information Processing Systems (NeurIPS)*, 34, 2021.

Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., Huang, A. C., Wen, Y., Hassanzadeh, K., Graves, D., Chen, D., Zhu, Z., Nguyen, N., Elsayed, M., Shao, K., Ahilan, S., Zhang, B., Wu, J., Fu, Z., Rezaee, K., Yadmellat, P., Rohani, M., Nieves, N. P., Ni, Y., Banijamali, S., Rivers, A. C., Tian, Z., Palenicek, D., bou Ammar, H., Zhang, H., Liu, W., Hao, J., and Wang, J. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. In *Conference on Robot Learning (CoRL)*, 11 2020.

A. Proof of Theorems 1 and 2

In this section, we prove Theorems 1 and 2. Their proofs strongly rely on two main lemmas, which we articulate next.

A.1. Confidence Lemma

The following main lemma shows that if the model is calibrated, the functions computed in Eq. (2) are indeed an upper confidence bound on the value functions of each agent i.

Lemma 1. *Under Assumption 1, with probability at least $1-\delta$ we have that for all $t \geq 0$,*

$$UCB_t^i(\boldsymbol{\pi}) \geq V^i(\boldsymbol{\pi}) \quad \forall i \in [N], \quad \forall \boldsymbol{\pi} \in \boldsymbol{\Pi}, \quad \forall t \geq 0\,.$$

Proof. Under Assumption 1, we know that with probability $1-\delta$, for each $s, \mathbf{a}$, and t, $|f(s,\mathbf{a}) - \mu_{t-1}(s,\mathbf{a})| \leq \beta_t \sigma_t(s,\mathbf{a})$ holds coordinate-wise. Hence, there exists a $\eta(s,\mathbf{a}) \in [-1,1]^p$ such that $f(s,\mathbf{a}) = \mu_{t-1}(s,\mathbf{a}) + \beta_t \cdot \Sigma_t(s,\mathbf{a})\eta(s,\mathbf{a})$. That is, the hallucinated transition coincides with the true transition. Therefore, given agent i and joint policy $\boldsymbol{\pi}$, the true states' trajectory (evolving according to $f(\cdot)$) is a feasible solution to (2).

□

A.2. Bounding optimistic trajectories via Lipschitzness

First, the following fact shows that under Assumption 2, the closed-loop transition, reward, and confidence functions are also Lipschitz continuous.

Fact 1. *Under the Lipschizness assumption, Assumption 2, it holds:*

$$\begin{aligned}
\|f(s,\boldsymbol{\pi}(s)) - f(s',\boldsymbol{\pi}(s'))\|_2 &\leq L_f\sqrt{1+NL_\pi}\cdot\|s-s'\|_2,\\
\|r^i(s,\boldsymbol{\pi}(s)) - r^i(s',\boldsymbol{\pi}(s'))\|_2 &\leq L_r\sqrt{1+NL_\pi}\cdot\|s-s'\|_2\,, \quad \forall i,\\
\|\sigma_t(s,\boldsymbol{\pi}(s)) - \sigma_t(s',\boldsymbol{\pi}(s'))\|_2 &\leq L_\sigma\sqrt{1+NL_\pi}\cdot\|s-s'\|_2\,, \quad \forall t\,.
\end{aligned}$$

Proof. Using the Lipschitzness of $f(\cdot)$ and $\pi^i(\cdot)$ from Assumption 2, we have:

$$\begin{aligned}
\|f(s,\boldsymbol{\pi}(s)) - f(s',\boldsymbol{\pi}(s'))\|_2 &\leq L_f\|(s-s',\boldsymbol{\pi}(s)-\boldsymbol{\pi}(s'))\|_2\\
&= L_f\sqrt{\|s-s'\|_2^2 + \sum_{i=1}^N \|\pi^i(s)-\pi^i(s')\|_2^2}\\
&\leq L_f\sqrt{\|s-s'\|_2^2 + NL_\pi\|s-s'\|_2^2}\\
&= L_f\sqrt{1+NL_\pi}\cdot\|s-s'\|_2.
\end{aligned}$$

The same derivation is obtained for the functions r^i and σ_t using Lipschitz constants L_r and L_σ, respectively. □

Then, we use the above Lipschitz properties to bound the distance between the optimistic value functions $\mathrm{UCB}_t^i(\boldsymbol{\pi})$ and the true functions $V^i(\boldsymbol{\pi})$. This will depend on the sum of accumulated standard deviations, as stated in the following main lemma.

Lemma 2. *Define $\bar{L}_f = 1 + (L_f + 2\beta_{t-1}L_\sigma)\sqrt{1+NL_\pi}$ and consider any round t. For each agent i and joint policy $\boldsymbol{\pi}$, under Assumption 1 it holds*

$$\left|UCB_{t-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right| \leq 2\beta_{t-1}L_r\sqrt{1+NL_\pi}\bar{L}_f^{H-1}H\cdot\mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2\right], \tag{6}$$

where $\{s_h\}_{h=0}^{H-1}$ is the sequence of environment's states when agents play according to $\boldsymbol{\pi}$ and $\boldsymbol{\omega}$ is the vector of noise realizations $\boldsymbol{\omega} = [\omega_0,\ldots,\omega_{H-1}]$.

Proof. Lemma 2 can be obtained from (Curi et al., 2020, Lemmas 3,4) as follows.

For a given joint policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, let $\{s_h\}_{h=0}^{H-1}$ be the sequence of environment states when agents play according to $\boldsymbol{\pi}$ and the environment transition function is $f(\cdot)$. Note that this sequence is random, depending on the noise realization vector $\boldsymbol{\omega} = [w_0, \dots, w_{H-1}]$. According to this notation, the value function of a generic agent i can be written as:

$$V^i(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\omega}} \left[\sum_{h=0}^{H-1} r^i(s_h, \boldsymbol{\pi}(s_h)) \right].$$

Similarly, consider an hallucinated transition function $\tilde{f}(\cdot)$ and let $\{\tilde{s}_h\}_{h=0}^{H-1}$ be the sequence of environment states visited according to $\tilde{f}(\cdot)$. Also this sequence is random, depending on the noise realizations which we denote with $\tilde{\boldsymbol{\omega}}$. Consider now the hallucinated function $\tilde{f}(\cdot) := \mu_{t-1}(\cdot) + \beta_{t-1} \cdot \Sigma_{t-1}(\cdot)\eta^\star(\cdot)$, where $\eta^\star$ is the auxiliary function that maximizes Eq. (2) at round $t-1$. According to the introduced notation, we have

$$\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) = \mathbb{E}_{\tilde{\boldsymbol{\omega}}} \left[\sum_{h=0}^{H-1} r^i(\tilde{s}_h, \boldsymbol{\pi}(\tilde{s}_h)) \right].$$

Now, we can use the Lipschitz properties of the closed-loop reward functions (Fact 1) to obtain

$$\begin{aligned}
\left|\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right| &= \left| \mathbb{E}_{\tilde{\boldsymbol{\omega}}} \left[\sum_{h=0}^{H-1} r^i(\tilde{s}_h, \boldsymbol{\pi}(\tilde{s}_h)) \right] - \mathbb{E}_{\boldsymbol{\omega}} \left[\sum_{h=0}^{H-1} r^i(s_h, \boldsymbol{\pi}(s_h)) \right] \right| \\
&= \left| \mathbb{E}_{\tilde{\boldsymbol{\omega}}=\boldsymbol{\omega}} \left[\sum_{h=0}^{H-1} r^i(\tilde{s}_h, \boldsymbol{\pi}(\tilde{s}_h)) - r^i(s_h, \boldsymbol{\pi}(s_h)) \right] \right| \\
&\leq L_r \sqrt{1 + NL_\pi} \sum_{h=0}^{H-1} \mathbb{E}_{\tilde{\boldsymbol{\omega}}=\boldsymbol{\omega}} \left[\|s_h - \tilde{s}_h\|_2 \right], \qquad (7)
\end{aligned}$$

where $\mathbb{E}_{\tilde{\boldsymbol{\omega}}=\boldsymbol{\omega}}$ is the expectation over $\boldsymbol{\omega}$ and taking $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}$.

At this point, we are left to bound the accumulated difference between the true and the hallucinated trajectory: $\sum_{h=0}^{H-1} \mathbb{E}_{\tilde{\boldsymbol{\omega}}=\boldsymbol{\omega}} \left[\|s_h - \tilde{s}_h\|_2 \right]$. For this we can invoke (Curi et al., 2020, Lemma 4) which, via a sequence of induction steps and under the calibrated model Assumption 1, shows that

$$\|s_h - \tilde{s}_h\|_2 \leq 2\beta_{t-1} \bar{L}_f^{H-1} \sum_{\tau=0}^{h-1} \|\sigma_{t-1}(s_\tau, \boldsymbol{\pi}(s_\tau))\|_2.$$

Then, we can substitute in Eq. (7) the bound above to obtain

$$\begin{aligned}
|\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})| &\leq 2\beta_{t-1} L_r \sqrt{1 + NL_\pi} \bar{L}_f^{H-1} \sum_{h=0}^{H-1} \mathbb{E}_{\boldsymbol{\omega}} \left[\sum_{\tau=0}^{h-1} \|\sigma_{t-1}(s_\tau, \boldsymbol{\pi}(s_\tau))\|_2 \right] \\
&\leq 2\beta_{t-1} L_r \sqrt{1 + NL_\pi} \bar{L}_f^{H-1} H \cdot \mathbb{E}_{\boldsymbol{\omega}} \left[\sum_{h=0}^{H-1} \|\sigma_{t-1}(s_h, \boldsymbol{\pi}(s_h))\|_2 \right],
\end{aligned}$$

which completes the proof. □

A.3. Proof of Theorem 1

We are now ready to prove Theorem 1.

Proof. According to H-MARL, the joint policy $\boldsymbol{\pi}_t$ is sampled from distribution $\boldsymbol{\mathcal{P}}_t$ which, by construction, is a CCE of the game defined by the optimistic value functions $\mathrm{UCB}_{t-1}^1(\cdot), \dots, \mathrm{UCB}_{t-1}^N(\cdot)$ (see Line 2 in Algorithm 1). Hence, by definition of CCE (Def. 1), for each player i and any policy $\pi^i \in \Pi^i$,

$$\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t} [\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi})] \geq \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}_t^{-i}} [\mathrm{UCB}_{t-1}^i(\pi^i, \pi^{-i})]. \qquad (8)$$

Equation (8) implies that, for each player i and and any policy $\pi^i \in \Pi^i$, under Assumption 1 with probability at least $1-\delta$,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}[V^i(\boldsymbol{\pi})] &= \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}[\mathrm{UCB}^i_{t-1}(\boldsymbol{\pi})] - \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\mathrm{UCB}^i_{t-1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right] \\ &\geq \mathbb{E}_{\pi^{-i}\sim\boldsymbol{\mathcal{P}}_t^{-i}}[\mathrm{UCB}^i_{t-1}(\pi^i,\pi^{-i})] - \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\mathrm{UCB}^i_{t-1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right] \\ &\geq \mathbb{E}_{\pi^{-i}\sim\boldsymbol{\mathcal{P}}_t^{-i}}[V^i(\pi^i,\pi^{-i})] - \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\mathrm{UCB}^i_{t-1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right], \end{aligned} \tag{9}$$

where the second inequality follows from (8), and the third one from the model being calibrated (Assumption 1) and Lemma 1. Using this, we can bound the dynamic regret for agent i as:

$$R^i(T) := \sum_{t=1}^{T} \max_{\pi\in\Pi^i} \mathbb{E}_{\pi^{-i}\sim\boldsymbol{\mathcal{P}}_t^{-i}}\left[V^i(\pi,\pi^{-i})\right] - \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[V^i(\boldsymbol{\pi})\right] \tag{10}$$

$$\leq \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\mathrm{UCB}^i_{t-1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right]$$

$$\leq \sum_{t=1}^{T} \underbrace{2\beta_{t-1}L_r\sqrt{1+NL_\pi}\bar{L}_f^{H-1}}_{\leq \bar{L}} H \cdot \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2\right] \tag{11}$$

$$\leq \bar{L}H\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2\right]$$

$$\leq \bar{L}H\sqrt{TH\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2^2\right]}$$

where $\bar{L} = 2\bar{\beta}L_r\sqrt{1+NL_\pi}\left(1+(L_f+2\bar{\beta}L_\sigma)\sqrt{1+NL_\pi}\right)^{H-1}$. The first inequality follows from Eq. (9). The second one follows from Lemma 2, the third one by definition of $\bar{L}$ and $\beta_t \leq \bar{\beta}$ for all t, while the fourth one by Cauchy-Schwartz.

Now, applying standard concentration arguments (see, e.g., (Kirschner & Krause, 2018, Lemma 3)), with probability at least $1-\delta$, it holds

$$\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2^2\right] \leq \mathcal{O}\left(\sum_{t=1}^{T}\sum_{(s,\mathbf{a})\in\mathcal{D}_t}\|\sigma_{t-1}(s,\mathbf{a})\|_2^2 + \log(\frac{1}{\delta})\right). \tag{12}$$

This is because at each round t, the term $\mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1}\|\sigma_{t-1}(s_h,\boldsymbol{\pi}(s_h))\|_2^2\right]$ is the expected value of $\sum_{(s,\mathbf{a})\in\mathcal{D}_t}\|\sigma_{t-1}(s,\mathbf{a})\|_2^2$, given the history up to round $t-1$.

Hence, by taking the union bound over the events of Lemma 1 and Eq. (12), the regret of each agent i is bounded, with probability at least $1-\delta$ by:

$$\begin{aligned} R^i(T) &\leq \mathcal{O}\left(\bar{L}H\sqrt{TH\sum_{t=1}^{T}\sum_{(s,\mathbf{a})\in\mathcal{D}_t}\|\sigma_{t-1}(s,\mathbf{a})\|_2^2 + \log(1/\delta)}\right) \\ &\leq \mathcal{O}\big(\bar{L}H\sqrt{TH\mathcal{I}_T}\big), \end{aligned}$$

since $\sum_{t=1}^{T}\sum_{(s,\mathbf{a})\in\mathcal{D}_t}\|\sigma_{t-1}(s,\mathbf{a})\|_2^2 \leq \mathcal{I}_T$ by definition of $\mathcal{I}_T$. □

A.4. When only approximate CCEs are computed at each round

Theorem 1 assumes that an exact CCE is computed at each round of H-MARL (Line 2 of Algorithm 1). However, in practice one may obtain only ϵ_t-CCE at each round t. The next theorem shows that in such a case, the agents' dynamic regret suffers an additive factor which corresponds to the sum of approximation errors ϵ_t. As a result, even if equilibria are not computed exactly, agents' dynamic regret can still be sublinear provided that ϵ_t decreases sufficiently fast.

Theorem 3. *Let Assumptions 1,2 be satisfied. Moreover, consider the case where* H-MARL *computes a ϵ_t-CCE at each round (Line 2 of Algorithm 1). After T rounds, with probability $1-\delta$, each agent i has bounded dynamic regret:*

$$R^i(T) \leq \bar{L} H^{1.5} \sqrt{T \mathcal{I}_T} + \sum_{t=1}^{T} \epsilon_t$$

where $\bar{L} = \mathcal{O}\big(N^{H/2} L_\pi^{H/2} (\bar{\beta}^H L_\sigma^H + L_f^H) + \log(1/\delta)\big)$, $\bar{\beta} = \max_t \beta_t$, and $\mathcal{I}_T$ is the complexity measure defined in (3).

Proof. At each round t, H-MARL computes an ϵ_t-CCE of the game associated to value functions $\text{UCB}^1_{t-1}(\cdot), \dots, \text{UCB}^N_{t-1}(\cdot)$. Hence, for each player i and any policy $\pi^i \in \Pi^i$,

$$\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t}[\text{UCB}^i_{t-1}(\boldsymbol{\pi})] \geq \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}_t^{-i}}[\text{UCB}^i_{t-1}(\pi^i, \pi^{-i})] - \epsilon_t \,.$$

By following the same steps of proof of Theorem 1, this implies that

$$\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t}[V^i(\boldsymbol{\pi})] \geq \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}_t^{-i}}[V^i(\pi^i, \pi^{-i})] - \epsilon_t - \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t}\big[\text{UCB}^i_{t-1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\big], \tag{13}$$

where we have used Assumption 1 and Lemma 1. Then, we can use Eq. (13) to bound agents' regrets obtaining,

$$\begin{aligned} R^i(T) &:= \sum_{t=1}^{T} \max_{\pi \in \Pi^i} \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}_t^{-i}} \left[V^i(\pi, \pi^{-i})\right] - \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t} \left[V^i(\boldsymbol{\pi})\right] \\ &\leq \sum_{t=1}^{T} \epsilon_t + \bar{L} H \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t, \boldsymbol{w}} \left[\sum_{h=0}^{H-1} \|\sigma_{t-1}(s_h, \boldsymbol{\pi}(s_h))\|_2\right] \\ &\leq \bar{L} H \sqrt{T H \mathcal{I}_T} + \sum_{t=1}^{T} \epsilon_t, \end{aligned}$$

where we have used the same derivation done after Eq. (10). □

A.5. Proof of Theorem 2

Proof. At round $t^\star$, where $t^\star$ is selected according to Theorem 2, the proof steps of Theorem 1 (in particular Eq. (9)) imply that

$$\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}^\star_t}[V^i(\boldsymbol{\pi})] \geq \mathbb{E}_{\pi^{-i} \sim \boldsymbol{\mathcal{P}}^{\star -i}_t}[V^i(\pi^i, \pi^{-i})] - \underbrace{\mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}^\star_t}\big[\text{UCB}^i_{t^\star - 1}(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\big]}_{:= \epsilon^i_T}, \tag{14}$$

That is, distribution $\boldsymbol{\mathcal{P}}^\star_t$ is an ϵ-CCE of the underlying Markov game, where the approximation factor ϵ is bounded by $\max_i \epsilon^i_T$ and ϵ^i_T is the quantity defined above. In what follows, we show that

$$\max_i \epsilon^i_T \leq 2T^{-\frac{1}{2}} \bar{L} H^{1.5} \sqrt{\mathcal{I}_T}. \tag{15}$$

Then, Theorem 2 follows by selecting an accuracy level $\epsilon > 0$ and use Eq. (15) to solve for T.

To prove (15), recall that $t^\star$ is selected according to:

$$t^\star = \arg\min_{t \in [T]} \max_{i \in [N]} \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mathcal{P}}_t} [\text{UCB}^i_{t-1}(\boldsymbol{\pi}) - \text{LCB}^i_{t-1}(\boldsymbol{\pi})], \tag{16}$$

where the function $\text{LCB}^i_t(\boldsymbol{\pi})$ is the solution of Eq. (2) with the outer maximization replaced by a minimization over $\eta(\cdot)$.

By following the same steps of the confidence Lemma 1, with probability at least $1-\delta$ it holds

$$\text{LCB}^i_t(\boldsymbol{\pi}) \leq V^i(\boldsymbol{\pi}) \quad \forall i \in [N], \quad \forall \boldsymbol{\pi} \in \boldsymbol{\Pi}, \quad \forall t \geq 0, \tag{17}$$

that is, $\text{LCB}^i_t(\boldsymbol{\pi})$ is a lower confidence bound on the agents' value functions. Moreover, the distance $\big|\text{LCB}^i_t(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\big|$ can be bounded exactly as was done in Lemma 2 for $\big|\text{UCB}^i_t(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\big|$. The only difference in its proof consists of

considering $\{\tilde{s}_h\}_{h=0}^{H-1}$ to be the sequence of environment states resulting from the pessimistic transition model $\tilde{f}(\cdot) = \mu_{t-1}(\cdot) + \beta_{t-1} \cdot \Sigma_{t-1}(\cdot)\eta^{lcb}(\cdot)$ where η^{lcb} is the minimizer of Eq. (2).

Then, according to (16) and (17) we can bound $\max_i \epsilon_T^i$ as

$$\begin{aligned}
\max_i \epsilon_T^i &:= \max_i \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t^\star}\left[\mathrm{UCB}_{t^\star-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right] \\
&\leq \max_i \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t^\star}\left[\mathrm{UCB}_{t^\star-1}^i(\boldsymbol{\pi}) - \mathrm{LCB}_{t^\star-1}^i(\boldsymbol{\pi})\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \max_i \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) - \mathrm{LCB}_{t-1}^i(\boldsymbol{\pi})\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \max_i \mathbb{E}_{\boldsymbol{\pi}\sim\boldsymbol{\mathcal{P}}_t}\left[\,\left|\mathrm{UCB}_{t-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right| + \left|\mathrm{LCB}_{t-1}^i(\boldsymbol{\pi}) - V^i(\boldsymbol{\pi})\right|\,\right] \\
&\leq \frac{2}{T}\sum_{t=1}^{T} 2\beta_{t-1} L_r \sqrt{1+NL_\pi}\,\bar{L}_f^{H-1} H \cdot \mathbb{E}_{\boldsymbol{\pi}_t,\boldsymbol{w}}\left[\sum_{h=0}^{H-1} \|\sigma_{t-1}(s_h, \boldsymbol{\pi}_t(s_h))\|_2\right] \\
&\leq \frac{2}{T}\bar{L}H\sqrt{TH\mathcal{I}_T} = 2T^{-\frac{1}{2}}\bar{L}H^{1.5}\sqrt{\mathcal{I}_T}.
\end{aligned}$$

In the last equality we have used the same bound from Eq. (11). □

B. Supplementary material for Section 4

In this section, we provide additional details concerning our experimental setup illustrated in Section 4.

Multi-agent equilibria vs. single-agent optimal policies. We compare the performance of multi-agent equilibria, with respect to computing single-agent optimal policies for the AVs. We consider the lane merging scenario and assume the HD vehicles' model is known. Then, we obtain driving policies for the AVs by:

i) Computing equilibrium policies using independent DQN learning. AVs' policies are trained simultaneously using the DQN algorithm, for 600 iterations. Then, we select the joint policy leading to the highest sum of agents' reward.

ii) Computing single-agent optimal policies. These are obtained as follows: We consider one agent at a time, by removing the other agent from the environment and replacing it with a HD vehicle driving on the same lane. For each agent, we obtain an optimal policy running DQN algorithm for 600 iterations and selecting the checkpoint with highest reward.

We evaluate the policies coming from either i) or ii) for 50 episodes and plot the corresponding rewards in Figure 2. Equilibrium policies lead to higher average and individual rewards for the agents. In particular, we observe that the single-agent optimal policy of AGENT-1 is often to break and drive behind the HD vehicle, especially when the HD vehicle drives at moderate/high speed. Instead, when using equilibrium policies, both AGENT-0 and AGENT-1 often coordinate in overtaking the HD vehicle, yielding higher rewards. This is confirmed by Table 2 which shows that agent's distance to their goal position (especially for AGENT-1) is lower when using equilibrium policies.

	Avg. agent	AGENT-0	AGENT-1
Single-agent optima	57.49 m	52.43 m	62.55 m
Multi-agent equilibria	**56.09** m	**52.17** m	**60.02** m

Table 2. Agents' distance to their goal position (lower is better) averaged over 50 runs, when using single-agent optimal policies versus equilibrium policies.

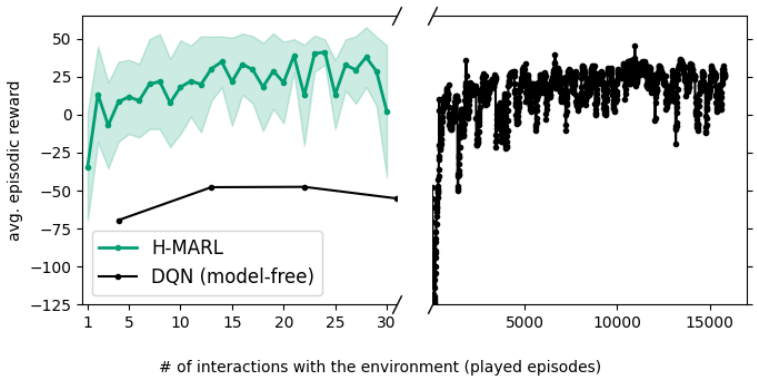
HD vehicles' model. To learn the behavior of HD vehicles, we use a GP model. The model maps the current HD vehicle's state (position and velocity) and its relative position to the other vehicles, to the next state for the HD vehicle. More specifically, we train our GP model on the *changes* (i.e., increase or decrease) of the HD vehicle's position and velocity. Hence, the next HD state is obtained by summing the current state's coordinates with the GP predictions. We use a Màtern kernel for predicting the speed change (as we expect this to be quite nonsmooth), and a squared exponential kernel to predict its change in position. However, we have observed different kernel choices (e.g., only using squared-exponential kernels)

to produce similar performance. At the end of each round t, GP inference is performed on the whole set of past observed trajectories $\{\mathcal{D}_\tau\}_{\tau=1}^{t}$ using GPyTorch (Gardner et al., 2018) with Adam (Kingma & Ba, 2014) optimizer for 50 iterations with learning rate $l = 0.1$. At round 0, we initialize the HD model with 2 random samples taken from past simulations.

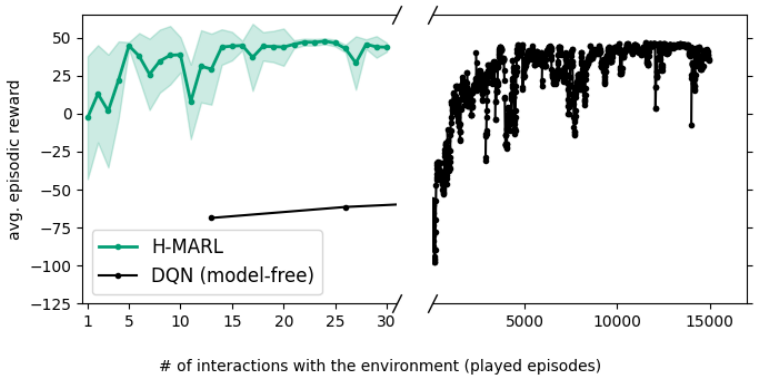
H-MARL implementation. To compute equilibrium policies at each round t (Line 2 of Algorithm 1), we run 50 iterations of independent DQN (Mnih et al., 2015) learning (initialized by the 50^{th} checkpoint from round $t-1$) and select distribution $\boldsymbol{\mathcal{P}}_{t+1}$ to be the uniform distribution over checkpoints $\{35^{th}, 40^{th}, 45^{th}, 50^{th}\}$. As mentioned in Section 4, this alleviates non-convergence behaviors and mimics the way CCEs are computed by independent no-regret learning dynamics. The hallucinated optimistic value functions $\text{UCB}_t^i(\cdot)$ are approximated by the sampling approach of Eq. (4) with $Z = 5$ samples at each time step and $\beta_t = 1.0$. This is effectively achieved as follows. During each policy evaluation step, we sample Z plausible next states for the HD vehicle and keep only the one that leads to the largest reward for the agents. Then, this process continues until the end of the episode. We have observed that changes in HD vehicles' position are accurately predicted by the GP model even with few data (this is because, given current position and speed, they only depend on the kinematics of the car and are invariant to the position of the AVs). Hence, for computational efficiency, we always use the posterior mean about the next HD vehicle's position, and sample $Z = 5$ plausible values for its next speed. We have also observed that uniformly spaced samples lead to better performance, compared to random i.i.d. samples.

Thompson Sampling (TS) implementation. We follow the exact same implementation of H-MARL with the important difference on how HD vehicle's states are computed during policy evaluation. Indeed, according to TS the next HD vehicle's state s_{h+1} is sampled from the posterior Gaussian distribution with mean $\mu_t(s_h, \mathbf{a}_h)$ and covariance $\Sigma_t(s_h, \mathbf{a}_h)$. Note that this is substantially different from the proposed optimistic H-MARL where, among plausible next states, we select the ones leading to the highest agents' reward.

Model-based vs. model-free. In this section, we compare our model-based H-MARL approach with the more general *model-free* DQN (Mnih et al., 2015) algorithm. In DQN, agents are trained directly on the rewards observed from the 'true' environment (i.e., interacting with the true HD vehicle's model). Instead, in H-MARL we assume a known model structure and agents interact with a hallucinated version of the environment where the HD vehicles are simulated according to the current (optimistic) model estimate. Indeed, in H-MARL we utilize a DQN subroutine at each round (see our implementation above) which, however, interacts only with the hallucinated model and not with the true one. Hence, we expect the model-free DQN to require a significantly higher number of environment interactions to achieve comparable performance. This is confirmed in Figure 4, where we plot agents' average reward as a function of the interaction rounds, both for the lane merging and intersection scenario. Perhaps not surprisingly, the model-free DQN requires interactions of several higher orders of magnitudes to achieve similar rewards.

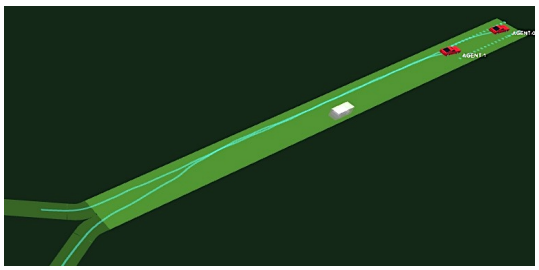


(a) Lane merging scenario

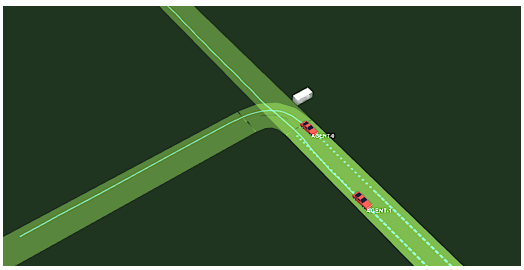


(b) Intersection scenario

Figure 4. Average agents' reward as a function of the interactions with the true environment (i.e., with the true HD vehicle's model), when agents' policies are computed according to the model-based H-MARL, or when using the model-free DQN (Mnih et al., 2013) algorithm. Model-free DQN requires a significantly higher number of interactions to achieve comparable performance. We remark that H-MARL utilizes a DQN subroutine at each round which, however, interacts with a hallucinated model for the HD vehicle and not with the true one.



(a) Overtake and merge maneuver



(b) Intersection crossing

Figure 5. (a) **Lane merging scenario** (see Fig. 1 (a)). Example of a successful overtake and merge maneuver: Both agents accelerate so that AGENT-1 overtakes the HD vehicle, and AGENT-0 successfully merges. (b) **Intersection scenario** (see Fig. 1 (b)). Example of a successful crossing: AGENT-1 crosses before both AGENT-0 and the HD vehicle, while AGENT-0 waits for the HD vehicle to cross, and then turns right.