Level-$k$: Meta-Learning for Pedestrian-Aware Self-Driving

Haozhe Lei* and Quanyan Zhu*

Abstract—The potential market for modern self-driving cars is enormous, as they are developing remarkably rapidly. At the same time, however, cases of pedestrian fatalities caused by autonomous driving have been recorded in the case of crossing the road. In this paper, we propose level-$k$ thinking into MAML to create a Level-$k$ Meta Reinforcement Learning (LK-MRL) as a self-driving vehicle model to prepare for heterogeneous pedestrians and improve intersection safety based on the combination of meta reinforcement learning and human cognitive hierarchy framework. In our evaluation, we assign this model to two different cognitive confrontation hierarchy scenarios in an urban traffic simulator to show not only its demonstrate its advantage in road safety but also the producing ability of higher-level thinking strategies.

Index Terms—autonomous vehicles, meta-learning, game theory, reinforcement learning, human safety

I. INTRODUCTION

One challenge for self-driving cars is their interactions with vehicles as well as pedestrians in urban environments. The unpredictability of pedestrian behaviors at intersections can lead to a high rate of accidents. The first pedestrian fatality caused by autonomous vehicles was reported in 2018 when a self-driving Uber vehicle struck a woman crossing an intersection in Tempe, Arizona, in the nighttime [1]. At the same time, many studies have examined the risks of accidents (e.g., [2]) and post-accident measures (e.g., [3]). Considering that human life is only once, it is much more appropriate to avert accidents than to remedy them afterward. To be more precise, there is a need for creating machine intelligence that allows autonomous vehicles to control the car and adapt to different pedestrian behaviors to prevent accidents.

Reinforcement learning is central to the development of end-to-end control algorithms. The extension to the Multi-Agent Reinforcement Learning (MARL) algorithms [4] is suitable for the modeling of vehicle-human interactions and be thought of as a solution for preventing traffic accidents. However, there is a key problem with classical MARL in this area stems from the fundamental differences between machines and human agents. Earlier studies, such as [5], [6], and [7], have created fixed pedestrian models when developing learning-based control algorithms. One of the researchers’ shortcomings is they have not sufficiently captured the variability in pedestrian behaviors, including reaction time, cognitive capabilities, and their dynamic response to the environment. Therefore, autonomous vehicles using these algorithms still have limitations in complex urban environments where vehicles need to interact with different pedestrians. Human agents have limited cognitive and reasoning abilities. There is a need to differentiate the computational and reasoning capabilities between pedestrians and vehicles.

Level-$k$ thinking [8], as a cognitive hierarchy framework, can provide a behavioral approach to capture bounded human reasoning processes in strategic games. It has shown promising accuracy in the prediction of human behaviors in contrast to analytical methods [9]. In recent years, approaches that allow people to use the cognitive hierarchy framework, such as [10] and [11], are also claimed, but they are not well suited to the human-vehicle interaction environment in this paper. According to the scenario considered in this paper, which is a high-dimensional strategy model training task for a single vehicle to a single pedestrian, there are non-negligible benefits of using a vanilla level-$k$ thinking strategy with alternative agents (without loss of generality, we will abbreviate it later as level-$k$ thinking).

In this work, we incorporate the cognitive hierarchy framework into MARL by viewing pedestrians and vehicles as different level-$k$ thinking agents who respond to others with bounded rationality. To capture the variability in human behaviors, we leverage the recent advances in meta reinforcement learning (meta-RL) algorithms, such as Model-Agnostic Meta-Learning (MAML) [12] and Conjectural Online Lookahead Adaptation (COLA) [13], to augment the MARL with the ability of fast adaptation to heterogeneous pedestrian behaviors and changing environments.

A. Related works

This section reviews recent progress in the area of RL-based vehicle-human interactions and RL with level-$k$ thinking.

a) RL-based vehicle-human interactions: Many earlier works have relied on imitation methods to generate the data and have assumed that pedestrians have similar response patterns to cars. [5] and [6] have trained models using robot hardware data and simulation data from static pedestrian agents. However, the models are trained offline, and they lack adaptation ability. In the meantime, [7] uses the sampling data from the real world for a demonstration of Inverse Reinforcement Learning (IRL). It relies on the sensor data from the physical world, which can be costly, limited, and may cause privacy issues.
b) RL with level-k thinking: level-k thinking, as a hierarchical model, has very strong interpretability. The consolidation of level-k thinking and RL has become widely studied in strategic decision-making. [14] and [15] have modeled behavioral predictions of drivers in highway driving scenarios by using level-k thinking and deep Q-learning (DQN). [10] has introduced generalized recursive reasoning (GR2) as a novel framework to model agents with different hierarchical levels of rationality to solve MARL problems, while [11] establishes a generalized cognitive hierarchy (GCH) model which assumes that level-k best-responds to all lower levels. However, it neither concentrates on vehicle-human issues nor deals with the fundamental challenges of human variability and often causes difficulty for high-dimensional reinforcement learning tasks.

B. Our Contributions

To make self-driving cars better accepted by today’s society, we wanted to create a model that could think deeper than humans. Taking it a step further, we consolidate level-k thinking into MAML to create a Level-k Meta Reinforcement Learning (LK-MRL) as a self-driving vehicle model to prepare for heterogeneous pedestrians and improve intersection safety. Our contribution can be summarised as follows:

1. We develop a Level-k Meta Reinforcement Learning model for the vehicle-human interactions and define its solution concept;
2. We test our LK-MRL structure in level-0 pedestrians interacting with level-1 car scenario, compare the trained policy with multiple baseline methods, and demonstrate its advantage in road safety;
3. Furthermore, based on the properties of level-k thinking, we test our LK-MRL structure in level-1 pedestrians interacting with level-2 car scenario and verify by experimental results that LK-MRL maintains its advantages with the use of reinforcement learning of producing different levels of agents with strategies of the best response of their lower level thinkers, which provides us possible to create higher level scenarios.

II. SYSTEM MODEL AND PRELIMINARY

In this section, we will first give the introduction of the meta-RL method we use in this work and the basic idea of level-k thinking, respectively. Then, we will describe the model and question setting in this paper.

A. Meta-reinforcement Learning

Denote the state and the control inputs according to step $t$ by $s_t \in S$, and $a_t \in A$ in the environment that has a physical law $P_t(s_{t+1} | s_t, a_t)$, where $S$ could be observed input from the sensor and $A$ could be controlling output actions in a self-driving environment. The goal of meta-RL is to find an action choosing policy $\pi(a_t | s_t; \theta)$ embedded by weights $\theta$, where the policy $\pi$ can enlarge the expectation of reward $R(\tau) := \sum_{t=1}^{H} r(s_t, a_t)$ of trajectories $\tau := (s_1, a_1, \ldots, s_H, a_H)$ with maximum $H$ steps. Suppose we have the probability that a trajectory occurs $q(\tau; \theta)$, then we can define the expected cumulative reward $J(\theta) = \mathbb{E}_{\tau \sim q(\cdot; \theta)}[R(\tau)]$ as the loss function.

For the meta-RL, the optimization problem is changed to:

$$\max_{\theta, H} \mathbb{E}_{\tau \sim p} \mathbb{E}_{D_i(\theta) \sim q}[J_i(\theta_i)]$$

subject to $\theta_i = \Phi(\theta, D_i)$.

We introduce $i$ as the different tasks of meta-learning. $p$ is the meta-training distribution, and $p(i)$ denotes the probability that the agent is placed in the environment $i$ with different tasks in the meta-training. $D_i$ is a batch of trajectories in the environment $i$, with the probability of each trajectory appearing being $q_i$. Considering the meta-RL model we choose in this work is the COLA method, $\Phi(\theta, D_i)$ will be obtained by using the Lookahead Adaptation mechanism. The agent in this method forms its belief $b_t$ about the current environment mode, i.e., $b_t \in \Delta([M])$, by its history observation. After that, the agent conjectures that it is interacting with the stationary MDP for $L$ steps with probability $b(i)$. Therefore, we can get the trajectory segment $\tau^L := (s_t, a_t, \ldots, s_{t+L-1}, a_{t+L-1}, s_{t+L})$ follows the distribution:

$$q(\tau^L; b, \theta) = \prod_{l=0}^{L-1} \pi(a_{t+l} | s_{t+l}; \theta) \cdot \prod_{l=0}^{L-1} \left( \sum_{i \in [M]} b(i) P_i(s_{t+l+1} | s_{t+l}, a_{t+l}) \right).$$

In order to maximize its forecast of the future performance in $L$ steps, the adapted policy $\theta_i = \Phi_i(\theta)$ should maximize the forecast future performance:

$$\max_{\theta \in \Theta} \mathbb{E}_{\tau^L | b, \theta'} \sum_{l=0}^{L-1} r(s_{t+l}, a_{t+l}).$$

Unlike the reinforcement learning training, the agent has no access to the distribution $q(\cdot; b, \theta')$ in the online setting and hence, cannot use policy gradient methods to solve for the maximizer. Following the approximation idea in trust region policy optimization (TRPO), the maximization of the future can be reformulated as

$$\max_{\theta \in \Theta} \mathbb{E}_{q(\cdot; b, \theta)} \left[ \sum_{l=0}^{L-1} \frac{\pi(a_{t+l} | s_{t+l}; \theta')}{\pi(a_{t+l} | s_{t+l}; \theta)} \sum_{l=0}^{L-1} r(s_{t+l}, a_{t+l}) \right]$$

subject to $\mathbb{E}_{s \sim q} D_{KL}(\pi(\cdot | s; \theta), \pi(\cdot | s; \theta')) \leq \delta$.

where $D_{KL}$ is the Kullback-Leibler divergence. In the KL divergence constraint, we slightly abuse the notation $q(\cdot)$ to denote the discounted state visiting frequency $s \sim q$. The intuition is that when $\theta'$ is close to the base policy $\theta$ in terms of KL divergence, the estimated objective in [3] using sample trajectories under $\theta$ returns a good approximation to $\mathbb{E}_{q(\cdot; b, \theta')} \sum_{l=0}^{L-1} r(s_{t+l}, a_{t+l})$.

B. Level-k Thinking

Level-k thinking as a cognitive hierarchy theory, which is widely applied in multiplayer games, allows the level-k thinking players to choose their best responses based on the assumption that all other players are level-(k-1) thinkers.
Denote $a^k$ as the action of level-$k$ player and $a^{k-1}$ as all other $(k-1)$-th players’ actions. Suppose we have a function $BR$ as a correspondence maps from an action to the set of its best responses. We could write the logic of level-$k$ thinking by:

$$a^k = BR(a^{k-1})$$  (4)

The relative clarity motivation of using level-$k$ thinking in our model is that it corresponds to reality’s natural human thinking pattern. It makes the model more explicable and rational. Further, using level-$k$ thinking in our method realizes our model is that it corresponds to reality’s natural human thinking, and both have rational. Further, using level-$k$ thinking in our method realizes the model’s foundation, let $\pi_{i,j}$ denote all superscript $k \in \mathbb{R}$ as the index of it belongs to the level-$k$ thinker. Without loss of generality, assuming our model is a Hidden Markov model (HMM), the policy $\pi_i \in \{O \rightarrow \Delta(A_i)\}$ is a mapping from the past observations $\cup_{t=1}^{t}O_k$ to a distribution over the action set $\Delta(A_i)$ which is the probability simplex in $\mathbb{R}^{A_i}$, for finite action set $A_i$. For the optimization and controlling of the policy, assume the policy is generated by some machine learning model, e.g., neural networks, we define $\theta_{i,j}$ as the policy parameters of level-$k$ thinking player $i$ with type $j$. Since we are using the level-$k$ thinking method, we denote all superscript $k \in \mathbb{R}$ as the index of it belongs to the level-$k$ thinker. Without loss of generality, assuming our model is a Hidden Markov model (HMM), the policy $\pi_i \in \{S_i \rightarrow \Delta(A_i)\}$ of player $i$ is only related to the $s_t$ and its own weight $\theta_{i,j}$, i.e., $\pi_i(a_t|s_t;\theta_{i,j}) \in \Delta(A_i)$.

We will first write the general case of the objective function and provide the optimizing algorithm.

### A. General Case of Model

By the definition of level-$k$ thinking, the level-$k$ thinkers conjecture their opponents as level-(k-1) thinkers. To establish the model’s foundation, let $V_{i,j}^k$ be the cumulative reward of level-$k$ thinking player $i$ with type $j$:

$$V_{i,j}^k := \sum_{t=1}^{T} r_{t,i,j} \left[ s_t, \pi_i(a_t|s_{t-1};\theta_{i,j}), \pi_{-i}(a_{t-1}|s_{t-1};\theta_{i,j}^{-1}) \right].$$

And then, we can write (6) as a general form of the objective function $J_k$ for level-$k$ thinkers.

$$\max_{\theta} J_k := \mathbb{E}_{P_{s_1}} \mathbb{E}_{P_{s_{t-1}}|s_{t-1}} \left[ V_{i,j}^k \right].$$

a) **Special case of level-0 thinker:** To simplify the training, we can use some constant policy agents as level-0 thinking players. However, we can also use the equation (6) to obtain some RL policies for the level-1 thinkers and begin the game from level-1. Assuming the car and pedestrians all be level-0 thinkers, they think of themselves as the only player in this environment. Therefore, we could reduce the general form of the cumulative reward function to (7).

$$V_{i,j}^0 := \sum_{t=1}^{T} r_t \left[ s_t, \pi_i(a_t|s_{t-1};\theta_{i,j}) \right].$$

where $a_t$ only contains $a_{c,t}$ since the level-0 player conjectures itself as the unique player in the environment. It is easy to see...
that both players will use a constant maximum strategy based on their reward functions.

b) Case of higher level thinkers: The higher level thinkers should follow the solution of (6). Suppose it is in the conventional logic, the level-1 interact level-1 game should be considered. However, with the LK-MRL model, we have a time-saving trick. If we have level-0 pedestrians, we can use it to obtain the level-1 car policy; If we have level-1 pedestrians, we can use it to obtain the level-2 car policy, and so on. It allows us to solve optimization problems alternately. Some researchers already show that normal humans usually only think about, at most, the second level of recursions in strategic games [8], thus we will not go further than the level-2 car policy, i.e., solve the optimization problem three times (the level-0 thinker’s policy does not need calculation). In this paper, the experiment focuses on the level-1 car interact level-0 pedestrians situation, which is strong enough to support our work. Figure 2 shows two modes of the LK-MRL model structure.

B. Optimizing Algorithm

This section is to find an optimizing algorithm for the constructed general objective function in section III-A. It is straightforward to notice that the objective function, no matter in which level of think, can be divided into two parts—optimizing pedestrians and optimizing cars. In both cases, we can assume their opponent uses a static model because the level-\((k-1)\) thinker’s strategy should be prior knowledge of the level-\(k\) thinker.

For a level-\(k\) (\(k > 0\)) thinking car, the strategy it wants is optimal for all three types of pedestrians. Following the COLA method mentioned in [I-A], we first obtain the car’s base model shown in algorithm 1.

Algorithm 1 RL-base

\[\text{Input} \theta_0 \text{ in level } k, \ t = 0, \ \text{step size } \alpha, \ \text{and the type of pedestrian } j.\]

\[\text{while not converge do}\]

Sample a batch of trajectories \(D_{k-1}^{i,j}\),
\[\theta_{h+1} = \theta_h + \alpha \nabla J_i(\theta_h, D_{k-1}^{i,j}) ;\]
\[h = h + 1.\]

Let \(\theta_{i,j}^h = \theta_{h+1};\)

\[\text{Return } \theta_{i,j}^h\]

The COLA model uses its own belief mechanism to make human-type conjectures. We can use many ways to realize it, e.g., latent type estimation [17] or a simple classification neural network like [18] for image input. Suppose an inference network \(F\), as an approximate to some behavior sequences, is obtained by some machine learning algorithm. Suppose we are in step \(t\). Since type is not directly observable to the vehicle, it is a latent variable to be estimated from the online observations. Based on it, we could claim the human-type conjecture \(P_i = F(s_1, a_1, c, a_1, p; \ldots; s_t, a_t, c, a_t, p)\) is the probability of the vehicle’s current opponent pedestrian’s type. The function could be the belief of our policy to recognize which type of pedestrian it meets and make the Lookahead Adaptation on it. The pseudo-code is shown in algorithm 2.

In order to show the optimal performance of the policy, we use the true type of the pedestrian as the output of human-type conjecture \(P_i\).

IV. EXPERIMENTAL CONFIGURATION

Our experiments use CARLA-0.9.4 [19] as the autonomous driving simulator. On top of the CARLA, we modify the API: Multi-Agent Connected Autonomous Driving (MACAD)
Gym [20] to facilitate communications between learning algorithms and environments. The full structure of state $s_t := \{d_{ct}, d_{pt}, v_{ct}, v_{pt}, a_{ct}, a_{pt}, d_{ct-1}, d_{pt-1}, v_{ct-1}, v_{pt-1}, a_{ct-1}, a_{pt-1}\}$ including 12 different variables. The available actions for pedestrians are: acceleration, deceleration, and cruising at the current speed, and for car are: throttle up, brake, and cruising at the current speed.

To be more relevant to the actual traffic situation, we define three different types of pedestrians as basic level-0 thinkers including: (1) a pedestrian who moves randomly with probability 0.2 maintain the speed, 0.43 move acceleration and 0.37 move backward; (2) a pedestrian accelerates to speed 5 km/h and maintain; (3) a pedestrian who accelerates to speed 3 km/h and maintain. The figure 3(a) shows their speed patterns.

In our experiments, we create a pedestrian 30 meters in front of the car on the sidewalk scenario with every step containing 0.05 seconds. The vehicle will keep accelerating until its speed reaches around 5 km/h. The episode will end with two events: the collision happens, training terminal state reach the horizon length, which is 300 steps, or the vehicle gets to its destination 40 meters far away.

The model structure for both pedestrians and car is $12 \times 64 \times 32 \times 3$ neural network with the rectified linear unit (ReLU) as its activation function. For pedestrians and the base policy of the car, the learning rate begins at $1 \times 10^{-4}$, and the policy gradient update is performed end of every episode. The entropy regularized method [21] is used. Once episode rewards stabilize, the learning rate will be changed to $1 \times 10^{-5}$ and $1 \times 10^{-6}$. The gradient buff of the COLA method will collect the gradient for every 10 steps and store 1000 episodic data for each type of pedestrian.

The main RL structure we use is the Asynchronous Advantage Actor-Critic algorithm (A3C) with Adam optimizer mentioned in [22].

### Algorithm 2 COLA

**Input** The base policy $\theta_1 = \theta_{i,j}^k$, human-type conjecture $P_i$, and gradient buffer $B$, gradient sample batch size $D$ lookahead horizon length $L$ and $b_t$ is the belief related to $P_i$ of step $t$, step size $\alpha$.

**for** $t \in \{1, 2, \ldots, H\}$ **do**

**Obtain** the state input $s_t$ from the environment;

**Implement** the action using $\pi_v(a_t|s_t, \theta_t);$

**Obtain** the probability output from $\pi(c_t|s_t, \theta_t);$

**Update** the belief $\theta_t$;

**Sample** $D$ gradients under different pedestrians’ types from $B$ according to $b_t$;

$\theta_{t+1} = \theta_t + \alpha \cdot$ gradients sample mean;

**else**

$\theta_{t+1} = \theta_t;$

**Let** $\theta_{i,j} = \theta_{t+1};$

**Return** $\theta_{i,j}$

---

**V. Simulation studies**

In this section, we divide our experiments into two kinds—mode 1 and mode 2. As we define in Figure 2, mode 1 is a level-1 car interacting level-0 pedestrian scenario, which is more concentrate on the analysis of the model performance in each hierarchy of level-$k$ thinker. And mode 2 focuses on proving the retentivity ability for level-$k$ thinking properties of LK-MRL by simulating a level-1 pedestrian and a level-2 car.

**A. Mode 1**

We introduce mode 1 as we already mentioned in Figure 2. It includes a level-1 car interacting with level-0 pedestrians. In this setting we only use constant policy describe in section II-C includes: (1) Randomly select with probability 0.2 maintain the speed, 0.43 move acceleration and 0.37 move backward; (2) Accelerating to 5 km/h and maintain; (3) Accelerating to 3 km/h and maintain.

The figure 3(a) shows their speed patterns. We compared the level-1 COLA agent (using learning rate $1 \times 10^{-3}$ and gradient buffer size equals to 500 update every 10 step) with a vanilla A3C RL agent trained in the same environment. In our experiments, we give the COLA agent the actual type of pedestrians it meets since we want to test its optimal performance and reduce the complexity of
The mean and variance of step speed are shown in figure 4(b). which means it can react with varying car actions. And since of cars. Its speed patterns shown in figure 4(a) are various, pedestrian performs further confrontations with different levels

function based on a level-

0

1

ability for level-

B. Mode 2

In mode 2 experiments, we want to show the retentivity ability for level-\(k\) thinking properties of the LK-MRL structure. We first train a level-1 pedestrian with a given reward function based on a level-0 car with a given constant policy. The mean and variance of step speed are shown in figure 4(b). Unlike constant level-0 pedestrians, the trained level-1 pedestrian performs further confrontations with different levels of cars. Its speed patterns shown in figure 4(a) are various, which means it can react with varying car actions. And since the level-1 car has the same level as the level-1 pedestrian, the result obtained for this situation is much worse than in mode 1. However, with 1000 episodes of fine-tuning training with a learning rate \(1 \times 10^{-4}\), we get a prototype level-2 car, which shows significant performance improvement in the right-hand side of figure 4(b). As the results, we obtain a pedestrian that can think as level-1 thinker by conjecture.

VI. CONCLUSION

This work proposes a level-\(k\) meta reinforcement learning (LK-MRL) structure based on the combination of speculative online lookahead adaptation (COLA) and hierarchical model works in urban traffic environments where vehicles need to interact with a diverse population of pedestrians. LK-MRL provides a solution by reducing the multi-agent question to alternate single-agent optimization and showing its ability for retentive level-\(k\) thinking properties with the using of reinforcement learning.

REFERENCES

[1] D. Wakabayashi, “Self-driving uber car kills pedestrian in arizona, where robots roam,” The New York Times, vol. 19, no. 03, 2018.

[2] X. Shi, Y. Wong, M. Li, and C. Chai, “Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory,” Accident Analysis & Prevention, vol. 117, pp. 346–356, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000145751830191X

[3] P.ator, J. Alam, M. Islam, and A. S. Salleh, “An emergency rescue dispatch system for road vehicles for instant notification of road accidents and post crash analysis,” in 2014 International Conference on Informatics, Electronics & Vision (ICIEV), 2014, pp. 1–6.

[4] A. OroojlooyJadid and D. Hajinezhad, “A review of cooperative multi-agent deep reinforcement learning,” 2019. [Online]. Available: https://arxiv.org/abs/1908.03965

[5] M. Everett, Y. F. Chen, and J. P. How, “Collision avoidance in pedestrian-rich environments with deep reinforcement learning,” IEEE Access, vol. 9, pp. 10 357–10 377, 2021.

[6] B. Lütjens, M. Everett, and J. P. How, “Safe reinforcement learning with model uncertainty estimates,” in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8662–8668.

[7] R. Alsakeh and T. Sayed, “Modeling pedestrian-cyclist interactions in shared space using inverse reinforcement learning,” Transportation Research Part F: Traffic Psychology and Behaviour, vol. 70, pp. 37–57, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847819306552

[8] C. F. Camerer, T.-H. Ho, and J.-K. Chong, “A Cognitive Hierarchy Model of Games,” The Quarterly Journal of Economics, vol. 119, no. 3, pp. 861–898, 08 2004. [Online]. Available: https://doi.org/10.1162/00335530415022225

[9] M. Kosinski, D. Stillwell, and T. Graepel, “Private traits and attributes are predictable from digital records of human behavior,” Proceedings of the National Academy of Sciences, vol. 110, no. 15, pp. 5802–5805, 2013. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1218772110

[10] Y. Wen, Y. Yang, R. Luo, and J. Wang, “Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning,” 2019. [Online]. Available: https://arxiv.org/abs/1901.09216

[11] J.-K. Chong, T.-H. Ho, and C. Camerer, “A generalized cognitive hierarchy model of games,” Games and Economic Behavior, vol. 99, pp. 257–274, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0899825616300847

[12] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135. [Online]. Available: https://proceedings.mlr.press/v70/finn17a.html

[13] T. Li, H. Lei, and Q. Zhu, “Self-adaptive driving in nonstationary environments through conjectural online lookahead adaptation,” 2022. [Online]. Available: https://arxiv.org/abs/2210.03209

[14] B. M. Albaba and Y. Yildiz, “Driver modeling through deep reinforcement learning and behavioral game theory,” IEEE Transactions on Control Systems Technology, vol. 30, no. 2, pp. 885–892, 2022.

[15] M. Yuan, J. Shan, and K. Mi, “Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles,” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 818–825, 2022.

[16] T. Li, Y. Zhao, and Q. Zhu, “The role of information structures in game-theoretic multi-agent learning,” Annual Reviews in Control, vol. 53, pp. 296–314, 2022.
[17] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, “Efficient off-policy meta-reinforcement learning via probabilistic context variables,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, PMLR, 09–15 Jun 2019, pp. 5331–5340. [Online]. Available: https://proceedings.mlr.press/v97/rakelly19a.html

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” arXiv, 2015.

[19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” vol. 78, pp. 1–16, 2017. [Online]. Available: https://proceedings.mlr.press/v78/dosovitskiy17a.html

[20] P. Palanisamy, “Multi-agent connected autonomous driving using deep reinforcement learning,” 2019.

[21] J. Schulman, X. Chen, and P. Abbeel, “Equivalence between policy gradients and soft q-learning,” 2017. [Online]. Available: https://arxiv.org/abs/1704.06440

[22] P. Palanisamy, Hands-On Intelligent Agents with OpenAI Gym: Your Guide to Developing AI Agents Using Deep Reinforcement Learning. Packt Publishing, 2018.