Prompt to Transfer: Sim-to-Real Transfer for Traffic Signal Control with Prompt Learning

Longchao Da[1], **Minquan Gao**[2], **Hao Mei** [1], **Hua Wei**[1]*
[1]Arizona State University
[2]Johns Hopkins University
{longchao, hmei7, hua.wei}@asu.edu, mgao40@jh.edu

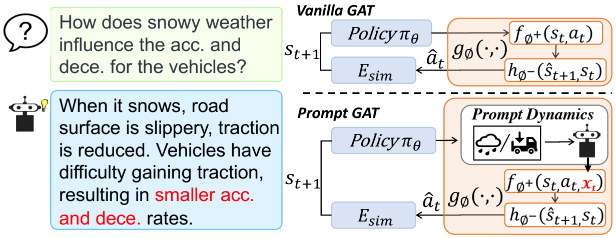
Abstract

Numerous solutions are proposed for the Traffic Signal Control (TSC) tasks aiming to provide efficient transportation and alleviate traffic congestion. Recently, promising results have been attained by Reinforcement Learning (RL) methods through trial and error in simulators, bringing confidence in solving cities' congestion problems. However, performance gaps still exist when simulator-trained policies are deployed to the real world. This issue is mainly introduced by the system dynamic difference between the training simulators and the real-world environments. In this work, we leverage the knowledge of Large Language Models (LLMs) to understand and profile the system dynamics by a prompt-based grounded action transformation to bridge the performance gap. Specifically, this paper exploits the pre-trained LLM's inference ability to understand how traffic dynamics change with weather conditions, traffic states, and road types. Being aware of the changes, the policies' action is taken and grounded based on realistic dynamics, thus helping the agent learn a more realistic policy. We conduct experiments on four different scenarios to show the effectiveness of the proposed PromptGAT's ability to mitigate the performance gap of reinforcement learning from simulation to reality (sim-to-real).

Introduction

Traffic Signal Control (TSC) is a critical task aimed at improving transportation efficiency and alleviating congestion in urban areas (Wei et al. 2021). Reinforcement Learning (RL) methods have shown promising results in tackling TSC challenges through trial and error in simulators (Ghanadbashi and Golpayegani 2022; Mei et al. 2023; Noaeen et al. 2022; Ducrocq and Farhi 2023; Zang et al. 2020; Wu et al. 2020; Haydari and Yılmaz 2020; Du et al. 2023; Vlachogiannis et al. 2023), bringing hope for solving cities' traffic congestion issues. While simulation is a valuable tool for control tasks in the real world with low cost, notable performance gaps arise when deploying simulator-trained policies to real-world environments (Da et al. 2023b,c), mainly due to differences in system dynamics between training simulators and the actual road conditions.

Grounded Action Transformation (GAT) is a framework designed to address the performance gaps that arise when transferring policies learned in simulation to real-world scenarios (sim-to-real) (Hanna and Stone 2017; Da et al. 2023b,c). The key idea behind GAT is to induce simulator dynamics to resemble those of the real world, where policy learning takes place in simulation, and dynamics learning relies on real-world data.


How does snowy weather influence the acc. and dece. for the vehicles?
When it snows, road surface is slippery, traction is reduced. Vehicles have difficulty gaining traction, resulting in smaller acc. and dece. rates.
Vanilla GAT
s_{t+1}
Policy π_θ
E_{sim}
$\hat{a}_t$
$g_\emptyset(\cdot,\cdot)$
$f_{\emptyset^+}(s_t, a_t)$
$h_{\emptyset^-}(\hat{s}_{t+1}, s_t)$
Prompt GAT
s_{t+1}
Policy π_θ
E_{sim}
$\hat{a}_t$
Prompt Dynamics
$g_\emptyset(\cdot,\cdot)$
$f_{\emptyset^+}(s_t, a_t, x_t)$
$h_{\emptyset^-}(\hat{s}_{t+1}, s_t)$

(a) Knowledge about dynamic changes from LLMs (b) Comparison between vanilla GAT and our proposed method

Figure 1: Integrating knowledge from LLMs into Grouned Action Transformation (GAT). (a) LLMs have implicit human knowledge about the change in dynamics. (b) Comparisons between vanilla GAT and our proposed PromptGAT, with GAT integrating a prompt-based dynamics modeling module.

In the GAT framework, the dynamics model, also known as the forward model f_{ϕ^+}, plays a crucial role. It takes the current state s_t and action a_t as inputs and predicts the possible next state s_{t+1} in the real world. Traditional GAT methods focus on learning f_{ϕ^+} solely based on real-world data, while these approaches enable the forward model to be accurately fitted to real-world dynamics, it requires a substantial amount of real-world data covering the entire state distribution to achieve accurate predictions.

One limitation of conventional GAT methods is their struggle to handle unobserved states in the real world. When the policy encounters states that have not been previously observed in the real-world data, the learned forward model may predict s_{t+1} with significant errors. This is particularly evident under extreme weather conditions or rare events that are infrequently represented in the training data. In contrast, *human knowledge* allows us to infer the behavior of the system under such unique conditions. For example, we as humans understand that during extreme weather, vehicles tend to move slower with smaller acceleration and deceleration rates,

[*]Corresponding Author

and the same duration of a green traffic signal may result in a smaller throughput. Moreover, humans can reason that adjusting the duration of green lights from the policy might be necessary to achieve a similar performance as observed in the simulation.

To leverage this implicit human knowledge for more accurate forward models, we propose a prompt-based GAT method, known as PromptGAT by introducing Large Language Models (LLMs) (Da et al. 2023a) into the GAT framework. Specifically, as is shown in Figure 1(b), in the learning of the forward model, we design a prompt-based dynamics modeling module to better understand the real-world dynamic by asking LLMs how weather conditions, traffic states, and road types influence traffic dynamics. Through the inference ability of LLMs in profiling the system dynamics, the agent can learn the grounded action in GAT based on a more accurate and general forward model. This process facilitates the learning of more realistic policies and enhances the transferability of RL models from simulation to reality.

In summary, the contributions of this paper are as follows:

- This paper proposes a novel method, PromptGAT, to mitigate the sim-to-real transfer problem in the context of traffic signal control by incorporating human knowledge with LLMs. To the best of our knowledge, this is the first paper bridging the performance gap between simulation and real-world settings in traffic signal control with LLMs.
- This paper provides the design of prompt generation and dynamics modeling module to understand the change of dynamics in the sim-to-real transfer. Leveraging LLMs with prompt and chain-of-thought (Wei et al. 2022b), PromptGAT provides valuable insights into the system dynamics, which enhances the agent's understanding of real-world scenarios.
- We conduct extensive experiments and case studies to validate the performance of our approach and showcase its potential impact on traffic signal control. All the experiments are conducted under a simulation-to-simulation setting with reproducible experiment settings.

Related Work

This section will introduce the related work from three aspects, regarding the traffic signal control (TSC) methods, simulation-transfer methods, and prompt learning techniques.

Traffic Signal Control Methods Optimizing traffic signal policies to mitigate traffic congestion has posed a significant challenge. Diverse methodologies are proposed, encompassing rule-based methods (Dion and Hellinga 2002; Chen et al. 2020) as well as RL-based methods (Wei et al. 2019b,a, 2018) for enhancing vehicle travel time or reducing delays, most of which yielded notable enhancements over pre-existing time control techniques. Although most of the current RL-based TSC methods do not consider the sim-to-real gap problem, a few recent studies start to tackle the sim-to-real gap by modifying the simulator directly (Müller et al. 2021; Mei et al. 2022), requiring the parameters of the simulator can be easily modified to perfectly match the real world. Rather than modify the simulator, this paper proposes to modify the output actions of the policies learned in the traffic simulator.

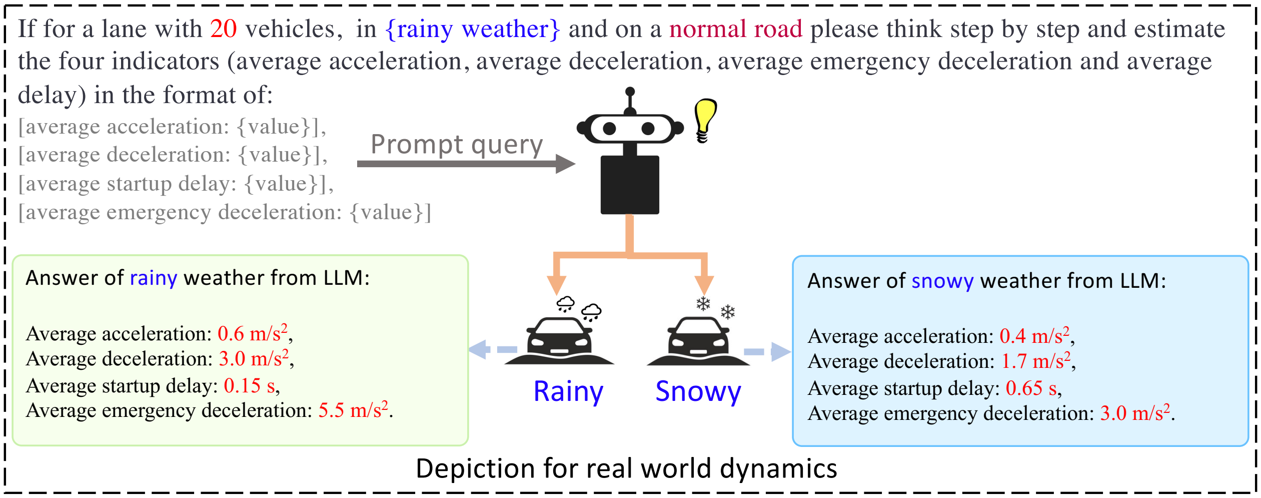
Sim-to-real Transfer Mainly three categorized groups of literature exist in the sim-to-real transfer domain (Zhao, Queralta, and Westerlund 2020). The first group is ***domain randomization*** (Tobin 2019; Andrychowicz et al. 2020; Wei et al. 2022a), with the objective of training policies capable of adapting to environmental variations. This strategy primarily relies on simulated data and proves advantageous when dealing with uncertain or evolving target domains. The second group is ***domain adaptation*** (Tzeng et al. 2019; Han et al. 2019), which is dedicated to addressing the challenge of domain distribution shift by aligning features between the source and target domains. Many domain adaptation techniques focus on narrowing the gap in robotic perception (Tzeng et al. 2015; Fang et al. 2018; Bousmalis et al. 2018; James et al. 2019), whereas in traffic signal control domain, the gap is mainly from the dynamics rather than perception because most TSC methods directly take vectorized representations like lane-level number of vehicles or delays as observations. The third group of approaches, known as ***grounding methods***, intends to improve the accuracy of the simulator concerning the real world by correcting for simulator bias. Unlike system identification approaches (Cutler, Walsh, and How 2014; Cully et al. 2015) that try to learn the precise physical parameters, Grounded Action Transformation (GAT) (Hanna and Stone 2017) does not require a parameterized simulator that can be modified. It induces the dynamics of the simulator to match the real world with grounded action, which has shown promising results for sim-to-real transfer in robotics. Following works (Desai et al. 2020b; Karnan et al. 2020; Desai et al. 2020a) further explore modeling the stochasticity when grounding, applying RL, and imitating from observation techniques to advance grounding. Our PromptGAT is based on GAT, with novel designs on leveraging LLMs to enhance action transformation by better dynamics profiling.

Prompt Learning Prompt learning, first introduced by (Petroni et al. 2019), has been widely studied in NLP with the development of Large Language Models (Jiang et al. 2020; Shin et al. 2020). Prompting means prepending instructions to the input and pre-training the language model so that the downstream tasks can be promoted. (Poerner, Waltinger, and Schütze 2019) use manually defined prompts to improve the performance of language models. To adapt LLMs for specific applications, developers often send prompts (aka, queries) to the model, which can be appended with domain-specific examples for obtaining higher-quality answers. A collection of prompt management tools, such as ChatGPT Plugin, GPT function API call, LangChain, AutoGPT, and BabyAGI, have been designed to help engineers integrate LLMs in applications and services. As far as we know, there is no exploration for prompt learning in sim-to-real transfer or traffic signal control tasks.

Method

Preliminaries

RL-based Traffic Signal Control In Traffic Signal Control (TSC), controllers determine intersection phases. Each phase



If for a lane with 20 vehicles, in {rainy weather} and on a normal road please think step by step and estimate the four indicators (average acceleration, average deceleration, average emergency deceleration and average delay) in the format of:

[average acceleration: {value}],
[average deceleration: {value}],
[average startup delay: {value}],
[average emergency deceleration: {value}]

Answer of rainy weather from LLM:

Average acceleration: 0.6 m/s^2,
Average deceleration: 3.0 m/s^2,
Average startup delay: 0.15 s,
Average emergency deceleration: 5.5 m/s^2.

Answer of snowy weather from LLM:

Average acceleration: 0.4 m/s^2,
Average deceleration: 1.7 m/s^2,
Average startup delay: 0.65 s,
Average emergency deceleration: 3.0 m/s^2.

Figure 2: An example of using LLM with a prompt template for answers to depict real-world dynamics by providing traffic state (vehicle number), weather type, and road type to induce the LLM to infer based on domain knowledge. Given the same vehicle quantity and road type, we could observe that the answers under different weathers abide by the reality situation that snowy weather is more severe than rainy weather.

comprises predefined, non-conflicting traffic movement combinations. Given the current condition of an intersection, the traffic signal controller will choose a phase for the next time interval Δt to minimize the average queue length on lanes around this intersection. Following existing work (Chen et al. 2020; Zheng et al. 2019; Wei et al. 2019b; Li et al. 2023), an agent is assigned to each traffic signal, and the agent will choose the phase as actions in the next Δt. The TSC problem is defined as a Markov Decision Process (MDP) characterized by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$ where $\mathcal{S}$ denotes the system state space $\mathcal{S}$, $\mathcal{A}$ denotes the set of action space, P denotes as the transition dynamics describing the probability distribution of next state $s_{t+1} \in \mathcal{S}$, r denotes the reward, and π_θ as the policy parameterized by θ and γ is the discount factor.

An RL approach solves this problem by maximizing the long-term expectation of discounted accumulation reward adjusted by discount factor γ. The discounted accumulated reward is $\mathbb{E}_{(s_t,a_t)\sim(\pi_\theta,\mathcal{M})}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$. We follow the past work which defines $\mathcal{A}$ as discrete action spaces, and use Deep Q-network (DQN) (Wei et al. 2018) to optimize the RL policy. In the past RL-based TSC works, the above procedure is conducted in the simulation environment E_{sim}.

Grounded Action Transformation Grounded action transformation (GAT) is a framework originally proposed in robotics to improve robotic learning by using trajectories from the physical world E_{real} to modify the actions to take in E_{sim}. Under the GAT framework, MDP in E_{sim} is imperfect and modifiable, and it can be parameterized as a transition dynamic $P_\phi(\cdot|s, a)$. Given real-world dataset $\mathcal{D}_{real} = \{\tau^1, \tau^2, \ldots, \tau^I\}$, where $\tau^i = (s_0^i, a_0^i, s_1^i, a_1^i, \ldots, s_{T-1}^i, a_{T-1}^i, s_T^i)$ is a trajectory collected by running a policy π_θ in E_{real}, GAT aims to minimize differences between transition dynamics by finding ϕ^* as shown in Eq. 1. The $d(\cdot)$ is the distance between two dynamics, P^* is the real world transition dynamics, and P_ϕ is the simulation transition dynamics.

$$\phi^* = \arg\min_{\phi} \sum_{\tau^i \in \mathcal{D}_{real}} \sum_{t=0}^{T-1} d(P^*(s_{t+1}^i|s_t^i, a_t^i), P_\phi(s_{t+1}^i|s_t^i, a_t^i)) \tag{1}$$

To find ϕ efficiently, GAT takes the agent's state s_t and action a_t predicted by policy π_θ as input and generates a grounded action $\hat{a}_t$ as output. Specifically, it uses an action transformation function parameterized with ϕ:

$$\hat{a}_t = g_\phi(s_t, a_t) = h_{\phi^-}(s_t, f_{\phi^+}(s_t, a_t)) \tag{2}$$

which includes two specific functions: a forward model f_{ϕ^+}, and an inverse model h_{ϕ^-}, as is shown in Figure 1.

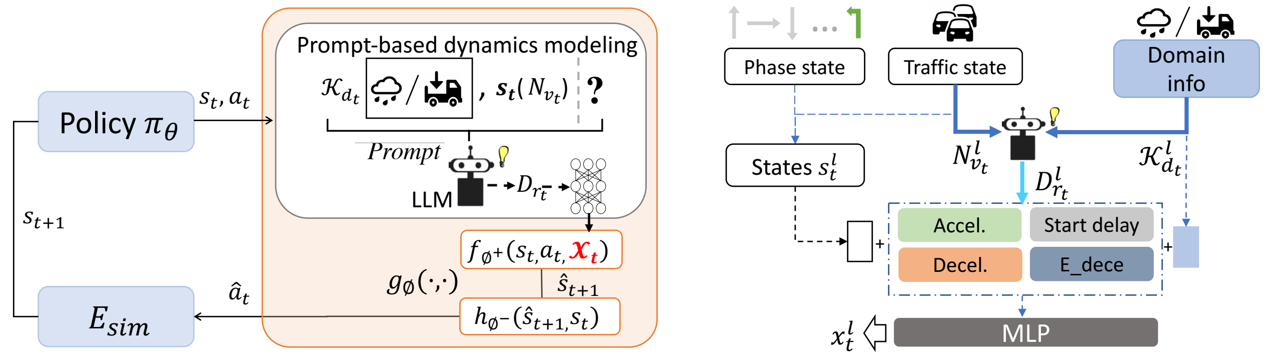
- *The forward model* f_{ϕ^+} is trained with the data from E_{real}, aiming to predict the next possible state $\hat{s}_{t+1}$ given current state s_t and action a_t:

$$\hat{s}_{t+1} = f_{\phi^+}(s_t, a_t) \tag{3}$$

- *The inverse model* h_{ϕ^-} is trained with the data from E_{sim}, aiming to predict the possible action $\hat{a}_t$ that could lead the current state s_t to the given next state. Specifically, the inverse model in GAT takes $\hat{s}_{t+1}$, the output from the forward model, as its input for the next state:

$$\hat{a}_t = h_{\phi^-}(\hat{s}_{t+1}, s_t) \tag{4}$$

Given current state s_t and the action a_t predicted by the policy π_θ, the grounded action $\hat{a}_t$ takes place in E_{sim} will make the resulted s_{t+1} in E_{sim} closer to the predicted next state $\hat{s}_{t+1}$ in E_{real}, which makes the dynamics $P_\phi(s_{t+1}|s_t, \hat{a}_t)$ in simulation closer to the real-world dynamics $P^*(\hat{s}_{t+1}|s_t, a_t)$. Therefore, the policy π_θ learned in E_{sim} with P_ϕ closer to P^* will lead to a smaller performance gap when transferred to E_{real} with P^*.



(a) The structure of PromptGAT

(b) Details of prompt-based dynamics modeling module

Figure 3: The overall framework of our proposed PromptGAT. (a) The structure of PromptGAT, with a prompt0based dynamics modeling integrating the knowledge of LLMs into the learning of forward model f_{ϕ^+}. (b) Details of prompt-based dynamics modeling module that infer and integrate the change of dynamics with traffic states.

Prompt-based GAT

In-context Learning for Dynamics Knowledge LLMs are known to be capable of in-context zero-shot or few-shot inference, which adapt these models to diverse tasks without gradient-based parameter updates (Alayrac et al. 2022). This allows them to rapidly generalize to unseen tasks and even exhibit apparent reasoning abilities with appropriate prompting strategies. Here is a set of system dynamic descriptions in the natural language as in Equation (5), provided to LLMs (GPT-4.0) as context. They stand for the weather information, road condition, and lane-level traffic state respectively:

$$\textit{Context} : \langle \textit{Weather} \rangle \langle \textit{Road Type} \rangle \langle \textit{Traffic State} \rangle \tag{5}$$

These contexts are organized and filled into the designed prompt template as below:

$$\langle \textit{Task} \rangle \langle [\textit{Context}] \rangle \langle \textit{Output Restriction} \rangle \tag{6}$$

where $\langle \textit{Task} \rangle$ provides task intention explanation to LLMs, and $\langle \textit{Output Restriction} \rangle$ induces the LLMs to infer the possible dynamics change based on the $\langle [\textit{Context}] \rangle$ provided in Equation (5) for the language model to understand the current perceptible information. As shown in Figure 2, the resulting output of dynamics knowledge could be then utilized by the forward model f_{ϕ^+}.

Prompt-based Dynamics Modeling In GAT, the learning of the forward model f_{ϕ^+} and inverse model h_{ϕ^-} is crucial. The forward model $f_{\phi^+}(s_t, a_t)$ in GAT predicts the next RL state $\hat{s}_{t+1}$ in the real world given taken action a_t and the current state s_t as in Equation (3), however, the prediction only using (s_t, a_t), and omits the consideration of domain knowledge $\mathcal{K}_d$, such as weather or road conditions, but the state transition $s_{t+1} = T_r(s_t, a_t)$ in the real world is a joint consequence related to this perceptible domain knowledge and real-time traffic states (vehicle quantities), e.g., In the snowy days, vehicles normally act with larger startup delay than in fine weather time, and for heavily loaded vehicles on the high loading allowed roads, the acceleration and emergency deceleration will be lower than those on the low loading allowed roads, which is mainly decided by the vehicles' standard machine characteristics. Therefore, we propose leveraging the $\mathcal{K}_d$ to provide a hint on the concrete real-world system dynamics D_r such as *acceleration*, *deceleration*, *emergency deceleration* and *startup delay* reflected by transition T_r. For $\forall\ s_t \in S$ on lane l, we employ LLMs (GPT-4.0) to realize inference by the prompt organized in Equation (6):

$$D^l_{r_t} = LLM(Prompt(\mathcal{K}^l_{d_t}, N^l_{v_t})) \tag{7}$$

where $\mathcal{K}^l_{d_t} = (\textit{weather}, \textit{road})$ and $N^l_{v_t}$ is the number of vehicles. Based on this, we incorporate the current lane state, lane level the domain knowledge $\mathcal{K}^l_{d_t}$ and dynamics knowledge $D^l_{r_t}$ from LLM together into the forward model through a fusion module of the model's network design as in Figure 3. Please note that the *road* description in $\mathcal{K}^l_{d_t}$ can vary for lanes but keep the same for the whole complete trajectory, and *weather* holds on same for one complete trajectory as well.

$$\dot{x_t}^l = \oplus\{s^l_t, D^l_{r_t}, \mathcal{K}^l_{d_t}\} \tag{8}$$

$$x_t{}^l = ReLU(Linear(\dot{x_t}^l)) \tag{9}$$

where $\dot{x}$ is a temporary calculation step after the operation $\oplus$, which is implemented as $Concatenate(\cdot)$, the derived $x_t{}^l$ represents the feature space for specific lane l at time step t. The input for f_{ϕ^+} is an integrated information $\mathcal{X}$ from all n lanes: $\mathcal{X}_t = (x^1_t, x^2_t, \ldots, x^n_t)$. Now we could represent the forward model into:

$$\hat{s}_{t+1} = f_{\phi^+}(s_t, a_t, \mathcal{X}_t) \tag{10}$$

We approximate f_{ϕ^+} with a neural network and optimize ϕ^+ by minimizing the Mean Squared Error (MSE) loss:

$$\mathcal{L}(\phi^+) = MSE(\hat{s}^i_{t+1}, s^i_{t+1}) = MSE(f_{\phi^+}(s^i_t, a^i_t), s^i_{t+1}) \tag{11}$$

where s_t^i, a_t^i, s_{t+1}^i are sampled from the trajectories collected from E_{real}.

Different from f_{ϕ^+}, the inverse model $h_{\phi^-}(\hat{s}_{t+1}, s_t)$ predicts the grounded action $\hat{a}_t^i$ that can lead to the same traffic states $\hat{s}_{t+1}$ in simulation E_{sim}. The h_{ϕ^-} could be learned through the interactions within simulation with lower cost than f_{ϕ^+}, therefore in this paper, we did not incorporate the dynamics knowledge in the inverse model for a more accurate model. We approximate h_{ϕ^-} with a deep neural network and optimize ϕ^- by minimizing the Categorical Cross-Entropy (CE) loss since the target a_t^i is a discrete value in traffic signal control problem defined by existing work (Wei et al. 2018):

$$\mathcal{L}(\phi^-) = CE(\hat{a}_t^i, a_t^i) = CE(h_{\phi^-}(s_{t+1}^i, s_t^i), a_t^i) \qquad (12)$$

where s_t^i, a_t^i, s_{t+1}^i are sampled from the trajectories collected from E_{sim}.

Overall Algorithm In this part, we will provide the pseudocode to show how the process is implemented.

Algorithm 1: Algorithm for PromptGAT

```
Input: Initial policy $\pi_\theta$, forward model $f_{\phi^+}$, inverse
       model $h_{\phi^-}$, domain info set $\mathcal{K}_d$, real-world
       dataset $\mathcal{D}_{real}$, simulation dataset $\mathcal{D}_{sim}$
Output: Policy $\pi_\theta$, $f_{\phi^+}$, $h_{\phi^-}$
Pre-train policy $\pi_\theta$ for M iterations in $E_{sim}$
for i = 1,2, . . . , I do
    Rollout policy $\pi_\theta$ in $E_{sim}$ and add data to $\mathcal{D}_{sim}$
    Load corresponding $\mathcal{K}_d^i$ (weather, road condition)
    Rollout policy $\pi_\theta$ in $E_{real}$ and add data to $\mathcal{D}_{real}$
    # Forward model update
    for l = 1, 2, . . . , n do
        Acquire $D_r^l$ dynamics by Equation (7)
        Feature fusion $x_t^l$ follow Equation (8), (9)
        Construct $\mathcal{X}_t = (x_t^1, x_t^2, \ldots, x_t^n)$
        Predict $\hat{s}_{t+1}$ by $f_{\phi^+}$ as Equation (10)
    end
    Update $f_{\phi^+}$ with Equation (11)
    # Inverse model update
    Update $h_{\phi^-}$ with Equation (12)
    # Policy training
    for e = 1, 2, . . . , E do
        # Policy update step
        Improve policy $\pi_\theta$ with reinforcement learning
    end
end
```

Experiments

In this section, we introduce the experimental setup and analysis results of PromptGAT. The implementation is based on a cross-simulator platform, LibSignal [1] and Pytorch.

Setting	accel (m/s^2)	decel (m/s^2)	eDecel (m/s^2)	sDelay (s)	Description
V0	2.60	4.50	9.00	0.00	Default setting
V1	1.00	2.50	6.00	0.50	Lighter loaded vehicles
V2	1.00	2.50	6.00	0.75	Heavier loaded vehicles
V3	0.75	3.50	6.00	0.25	Rainy weather
V4	0.50	1.50	2.00	0.50	Snowy weather

Table 1: Real-world Configurations for E_{real}

Experiment Settings

In this section, we introduce the overall environment setup for our experiments, commonly used metrics, important hyperparameters and model structures.

Environment Setup In our study, we leverage LibSignal (Mei et al. 2022), an open-source framework that incorporates multiple simulation environments. Our implementation involves using Cityflow (Zhang et al. 2019) as the simulation environment E_{sim} and SUMO (Lopez et al. 2018) as the real-world environment E_{real}. Throughout the paper, we refer to E_{sim} and E_{real} as our default simulation and real-world environments, respectively. Note that this simulation-to-simulation setting not only serves as a representative sim-to-real scenario but also allows for replicable and reproducible results in our experiments. All the hyperparameters, prompts, and codes can be found in repository[2].

To simulate real-world scenarios, we consider four different configurations in SUMO, representing two types of real-world scenarios: heavy industry roads and special weather-conditioned roads, with specific parameter settings detailed in Table 1. The four configurations are as follows:

- *V0: Default setting*[3]. This represents the default parameters for SUMO and CityFlow, capturing the normal settings of vehicle movement in E_{sim}.
- *V1 & V2: Heavy industry roads.* In this configuration, we model areas where the majority of vehicles are heavy trucks. In Table 1, for vehicles in $V1$ and $V2$, their accelerating, decelerating, and emergency decelerating rates are set to be slower than the default settings. We also consider a larger average startup delay for the vehicles (greater than the default assumption of $0s$). Additionally, $V1$ describes roads with lighter-loaded vehicles, while $V2$ represents the same roads with heavier-loaded vehicles, differing in startup delay.
- *V3 & V4: Special weather-conditioned roads.* For this configuration, we consider areas with special weather conditions. In Table 1, $V3$ and $V4$ represent rainy and snowy weather conditions, respectively. In these settings, the vehicles' accelerating, decelerating, and emergency decelerating rates are smaller than the default values, and the startup delays are larger. In snowy weather, the first three rates are smaller than in rainy conditions, and the discrepancy in startup delays for snowy conditions is extended to emulate tire slip.

[1] https://darl-libsignal.github.io/

[2] https://github.com/DaRL-LibSignal/PromptGAT.git

[3] https://sumo.dlr.de/docs/Definition_of_Vehicles, _Vehicle_Types,_and_Routes.html

Evaluation Metrics The goal of this work is to mitigate the performance gap of the trained policy π_θ in the simulation environment E_{sim} and in the real-world environment E_{real}. We calculate the performance difference Δ for commonly used traffic signal control metrics: average travel time, throughput, reward, average queue length, and average delay following work (Da et al. 2023c), and denote their differences as *ATT*$_\Delta$, *TP*$_\Delta$, *Reward*$_\Delta$, *Queue*$_\Delta$, and *Delay*$_\Delta$. For a given metric ψ:

$$\psi_\Delta = \psi_{real} - \psi_{sim} \tag{13}$$

Since in real-world settings, policy π_θ tends to perform worse than in simulation, the values of *ATT*, *Queue*, and *Delay* in E_{real} are typically larger than those in E_{sim}. Based on our goal of mitigating the gap and improving the performance of π_θ in E_{sim}, we expect that for *ATT*$_\Delta$, *Queue*$_\Delta$, and *Delay*$_\Delta$, smaller values are better, while for *TP*$_\Delta$ and *Reward*$_\Delta$, larger values are better. For a fair comparison, ψ_{sim} of all methods in E_{sim} are trained to be similar and reported in Table 2: with the similar ψ_{sim}, we can also compare ψ_{real} from different methods to know which method performs the best.

Experimental Results and Analysis

We analyze the proposed method in the following way: First, we verify if the prompt result from the Large Language Model is giving the rational inference based on the context description. Second, we compare to the Direct-Transfer to verify if the PromptGAT can mitigate the performance. Then, we discuss the performance improvement competing with baseline models. Furthermore, we demonstrate our method's contribution to the forward model's accuracy and how it is correlated to the final E_{real} performance gap mitigation.

Prompt Intention Analysis In this section, we conduct an analysis on the verification of whether the LLM Prompt infers the expected information following the practical laws in reality. We construct prompts in the format of $\langle$*Task*$\rangle\langle$[*Context*]$\rangle\langle$*Output Restriction*$\rangle$ as defined in Equation (6). We first introduce the $\langle$*Task*$\rangle$ description below:

> The indicators describing the traffic dynamics include the average acceleration (AC) of the vehicles (m/s²), the average deceleration (AD) (m/s²), the average emergency deceleration (AED) (m/s²) and the average startup delay (ADL) describing the average time needed for the waiting vehicles to start moving with the unit (s), and the above might vary based on weather or road type. Please assume the above indicators based on the traffic perceptive information below:

Then specifically, for the $\langle$[*Context*]$\rangle$, we have the following implementation (the colored content is replaceable based on the actual situation, *weather*, *road*, and *traffic state (vehicle quantity)* in correspondence with Equation (5):

> V1: In *sunny* day, on a *light industry road* with *8* vehicles,
> V2: In *sunny* day, on a *heavy industry truck road*, *5* vehicles,
> V3: In *rainy* day, on a *normal road* with *10* vehicles,
> V4: In *snowy* day, on a *normal road* with *7* vehicles.

And for $\langle$*Output Restriction*$\rangle$ we design as below:

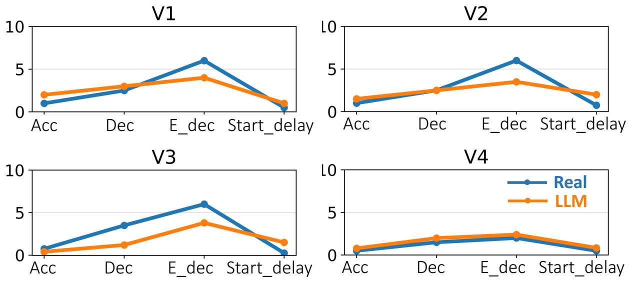


Figure 4: Comparison of LLM prompt answers and real-world settings reflects the same tendency across four settings.

> Please answer by replacing {value} in the format below:
> [average acceleration: value],
> [average deceleration: value],
> [average emergency deceleration: value],
> [average startup delay: value].

We take the real-world setting dynamics values as the ground truth and compare the LLM inferred value outputs to the ground truth to analyze their relationship, the results are shown in Figure 4. We could observe that within each sub-figure, LLM's inference results show a similar curve across metrics, and from v1 to v4, the LLM also reflects the same tendency as shown by Real settings. This proves the LLM's ability to provide a realistic inference based on the given information, thus guaranteeing the rationality to apply Prompt in our task of approximating real-world dynamics.

Env	*ATT*	*TP*	*Reward*	*Queue*	*Delay*
E_{sim}	**111.23**$_{\pm 3.5}$	**1978**$_{\pm 5}$	**-39.44**$_{\pm 2.23}$	**26.11**$_{\pm 1.15}$	**0.62**$_{\pm 0.10}$

Table 2: Overall performance in E_{sim}

Ability to Mitigate the Performance Gap We first train policies using DQN in E_{sim} to well-converged as in Table 2. and apply to four E_{real} settings described in Table 1. This is taken as the 'direct transfer', which exists a large gap compared to E_{sim} training performance (Da et al. 2023c). Then we leverage the proposed method PromptGAT to train policies under the same four settings. Following the metrics in Section 21, we could show a comparison in Figure 5: the center blue area is the metrics connection reported when policies are well trained in E_{sim}, these are the most ideal achievement that a policy could acquire. When the well-trained policy directly applies to E_{real}, the performance is shown in the orange area, we could obviously observe that large gaps commonly exist in four different real-world settings. Even though the severeness varies on settings, still none of the gaps is trivial. The PromptGAT shows promising results by effectively shrinking the gap to a much lower level (as shown in the purple area).

Comparison to Baseline Models In this part, we analyze how the proposed PromptGAT competes with other methods at a quantity level, including Direct Transfer and VanillaGAT.

Setting	Methods	Metrics				
	-	*ATT*($\Delta\downarrow$)	*TP*($\Delta\uparrow$)	*Reward*($\Delta\uparrow$)	*Queue*($\Delta\downarrow$)	*Delay*($\Delta\downarrow$)
$V1$	Direct-Transfer	158.93 (47.69)$_{\pm 55.02}$	1901 (-77)$_{\pm 52.21}$	-71.55 (-32.11)$_{\pm 22.51}$	47.71 (21.59)$_{\pm 14.98}$	0.73 (0.11)$_{\pm 0.03}$
	Vanilla-GAT	156.10 (44.87)$_{\pm 4.81}$	1905 (-73)$_{\pm 13.00}$	-70.03 (-30.59)$_{\pm 3.80}$	46.61 (20.50)$_{\pm 1.97}$	0.71 (0.09)$_{\pm 0.01}$
	PromptGAT	**154.97 (43.74)$_{\pm 6.09}$**	**1918 (-60)$_{\pm 9.62}$**	**-66.88 (-27.44)$_{\pm 4.47}$**	**44.56 (18.45)$_{\pm 2.96}$**	**0.71 (0.09)$_{\pm 0.01}$**
$V2$	Direct-Transfer	177.27 (66.03)$_{\pm 82.63}$	1898 (-80)$_{\pm 102.25}$	-87.71 (-48.27)$_{\pm 26.18}$	58.59 (32.47)$_{\pm 17.46}$	0.76 (0.14)$_{\pm 0.02}$
	Vanilla-GAT	180.58 (69.35)$_{\pm 11.72}$	**1908 (-69)$_{\pm 12.00}$**	-89.69 (-50.25)$_{\pm 9.13}$	59.93 (33.82)$_{\pm 8.01}$	0.74 (0.12)$_{\pm 0.07}$
	PromptGAT	**174.31 (63.08)$_{\pm 13.11}$**	1904 (-73)$_{\pm 21.63}$	**-84.71 (-45.27)$_{\pm 18.89}$**	**56.64 (30.53)$_{\pm 12.62}$**	**0.72 (0.10)$_{\pm 0.02}$**
$V3$	Direct-Transfer	205.86 (94.63)$_{\pm 64.49}$	1877 (-101)$_{\pm 100.86}$	-101.26 (-61.82)$_{\pm 20.10}$	67.62 (41.51)$_{\pm 13.37}$	0.76 (0.14)$_{\pm 0.03}$
	Vanilla-GAT	214.29 (103.06)$_{\pm 40.59}$	1846 (-131)$_{\pm 56.74}$	-91.15 (-51.71)$_{\pm 15.13}$	60.93 (34.82)$_{\pm 10.10}$	0.73 (0.11)$_{\pm 0.02}$
	PromptGAT	**198.48 (87.25)$_{\pm 7.27}$**	**1879 (-98)$_{\pm 6.02}$**	**-89.25 (-49.81)$_{\pm 5.51}$**	**59.65 (33.54)$_{\pm 3.70}$**	**0.72(0.10)$_{\pm 0.01}$**
$V4$	Direct-Transfer	332.48 (221.25))$_{\pm 109.00}$	1735 (-252)$_{\pm 151.91}$	-126.71 (-87.23)$_{\pm 14.79}$	84.53 (58.42)$_{\pm 9.86}$	0.83 (0.21)$_{\pm 0.01}$
	Vanilla-GAT	318.70 (207.47)$_{\pm 12.35}$	1750 (-227)$_{\pm 16.93}$	-115.01 (-75.57)$_{\pm 9.27}$	76.74 (50.63)$_{\pm 5.10}$	0.81 (0.19)$_{\pm 0.08}$
	PromptGAT	**310.29 (199.06)$_{\pm 22.57}$**	**1750 (-227)$_{\pm 16.47}$**	**-113.55 (-74.11)$_{\pm 6.68}$**	**75.77 (49.66)$_{\pm 4.48}$**	**0.81 (0.19)$_{\pm 0.01}$**

Table 3: The performance using Direct-Transfer, Vanilla-GAT compared with using PromptGAT method. The $(\cdot)$ shows the metric gap ψ_Δ from E_{real} to E_{sim} and the $\pm$ shows the standard deviation with 5 runs. The $\uparrow$ means that the higher value for the metric indicates a better performance and $\downarrow$ means that the lower value indicates a better performance.

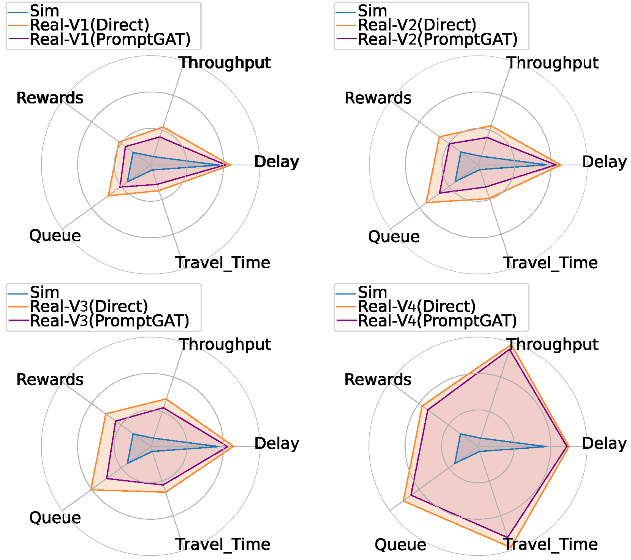


Figure 5: The performance in the E_{real} using Direct-Transfer and PromptGAT comparing to the performance in E_{sim}.

We apply all three approaches in four settings and compare their performance under five metrics. Each performance is represented as mean value and standard deviation after conducting five runs of tests. As shown in Table 3 that most of the time, the PromptGAT performs better than other baselines across various settings and metrics.

Correlation Analysis We conduct a case study on setting $V4$ to show the contribution of PromptGAT to the forward model accuracy and its relation to the performance gap mitigation in E_{real}. We first compare the prediction error of our method to Vanilla GAT as in Figure 6 (left), proving our method provides a better prediction of system dynamics. To understand how would dynamic profiling ability influence the model's performance gap in E_{real}, we conduct correlation analysis across the metrics gap of throughput, and waiting queue length. As shown in Figure 6 (right), the improvement of the mitigated gap (absolute values) across multiple metrics is positively correlated to the improvement of the forward model's accuracy. This indicates PromptGAT mitigates the real-world gap by better profiling the realistic dynamics.

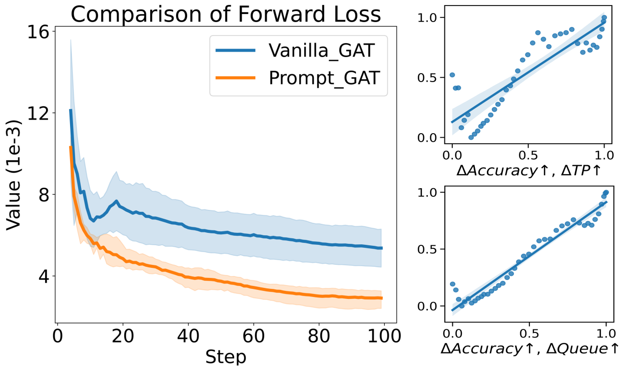


Figure 6: Left: Prediction error from the forward model using PromptGAT *vs* VanillaGAT, our method consistently approximates the true system dynamics and reduces the loss. Right: The correlation between improvement of accuracy and improvement of mitigated performance gap Δ in E_{real}.

Conclusion

In this paper, we propose a prompt-based grounded action transformation method, PromptGAT, for reinforcement learning paradigm to mitigate the sim-to-real performance gap. By leveraging the inference ability of pre-trained LLMs, and incorporating the perceptible domain knowledge, PromptGAT manages to better profile the system dynamics and increases the forward model's accuracy, further mitigating the sim-to-real gap by rectifying the action through action grounding.

Acknowledgments

The work was partially supported by NSF award #2153311. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736.

Andrychowicz, O. M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1): 3–20.

Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, 4243–4250. IEEE.

Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; and Li, Z. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3414–3421.

Cully, A.; Clune, J.; Tarapore, D.; and Mouret, J.-B. 2015. Robots that can adapt like animals. *Nature*, 521(7553): 503–507.

Cutler, M.; Walsh, T. J.; and How, J. P. 2014. Reinforcement learning with multi-fidelity simulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 3888–3895.

Da, L.; Liou, K.; Chen, T.; Zhou, X.; Luo, X.; Yang, Y.; and Wei, H. 2023a. Open-TI: Open Traffic Intelligence with Augmented Language Model. *arXiv preprint arXiv:2401.00211*.

Da, L.; Mei, H.; Sharma, R.; and Wei, H. 2023b. Sim2Real Transfer for Traffic Signal Control. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 1–2. IEEE.

Da, L.; Mei, H.; Sharma, R.; and Wei, H. 2023c. Uncertainty-aware Grounded Action Transformation towards Sim-to-Real Transfer for Traffic Signal Control. *arXiv preprint arXiv:2307.12388*.

Desai, S.; Durugkar, I.; Karnan, H.; Warnell, G.; Hanna, J.; and Stone, P. 2020a. An Imitation from Observation Approach to Transfer Learning with Dynamics Mismatch. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Desai, S.; Karnan, H.; Hanna, J. P.; Warnell, G.; and Stone, P. 2020b. Stochastic Grounded Action Transformation for Robot Learning in Simulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2020)*.

Dion, F.; and Hellinga, B. 2002. A rule-based real-time traffic responsive signal control system with transit priority: application to an isolated intersection. *Transportation Research Part B: Methodological*, 36(4): 325–343.

Du, W.; Ye, J.; Gu, J.; Li, J.; Wei, H.; and Wang, G. 2023. Safelight: A reinforcement learning method toward collision-free traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 14801–14810.

Ducrocq, R.; and Farhi, N. 2023. Deep reinforcement Q-learning for intelligent traffic signal control with partial detection. *International Journal of Intelligent Transportation Systems Research*, 21(1): 192–206.

Fang, K.; Bai, Y.; Hinterstoisser, S.; Savarese, S.; and Kalakrishnan, M. 2018. Multi-task domain adaptation for deep learning of instance grasping from simulation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3516–3523. IEEE.

Ghanadbashi, S.; and Golpayegani, F. 2022. Using ontology to guide reinforcement learning agents in unseen situations: A traffic signal control system case study. *Applied Intelligence*, 52(2): 1808–1824.

Han, T.; Liu, C.; Yang, W.; and Jiang, D. 2019. Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions. *ISA transactions*, 93: 341–353.

Hanna, J.; and Stone, P. 2017. Grounded action transformation for robot learning in simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Haydari, A.; and Yılmaz, Y. 2020. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1): 11–32.

James, S.; Wohlhart, P.; Kalakrishnan, M.; Kalashnikov, D.; Irpan, A.; Ibarz, J.; Levine, S.; Hadsell, R.; and Bousmalis, K. 2019. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12627–12637.

Jiang, Z.; Xu, F. F.; Araki, J.; and Neubig, G. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438.

Karnan, H.; Desai, S.; Hanna, J. P.; Warnell, G.; and Stone, P. 2020. Reinforced Grounded Action Transformation for Sim-to-Real Transfer. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2020)*.

Li, S.; Mei, H.; Li, J.; Wei, H.; and Xu, D. 2023. Toward Efficient Traffic Signal Control: Smaller Network Can Do More. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 8069–8074. IEEE.

Lopez, P. A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.-P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; and Wießner, E. 2018. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, 2575–2582. IEEE.

Mei, H.; Lei, X.; Da, L.; Shi, B.; and Wei, H. 2022. LibSignal: An Open Library for Traffic Signal Control. *arXiv preprint arXiv:2211.10649*.

Mei, H.; Li, J.; Shi, B.; and Wei, H. 2023. Reinforcement Learning Approaches for Traffic Signal Control under Missing Data.

Müller, A.; Rangras, V.; Ferfers, T.; Hufen, F.; Schreckenberg, L.; Jasperneite, J.; Schnittker, G.; Waldmann, M.; Friesen, M.; and Wiering, M. 2021. Towards real-world deployment of reinforcement learning for traffic signal control. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 507–514. IEEE.

Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z. S. H.; Bazzan, A. L.; and Far, B. 2022. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, 199: 116830.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473. Hong Kong, China: Association for Computational Linguistics.

Poerner, N.; Waltinger, U.; and Schütze, H. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4222–4235. Online: Association for Computational Linguistics.

Tobin, J. P. 2019. *Real-World Robotic Perception and Control Using Synthetic Data*. University of California, Berkeley.

Tzeng, E.; Devin, C.; Hoffman, J.; Finn, C.; Peng, X.; Levine, S.; Saenko, K.; and Darrell, T. 2015. Towards adapting deep visuomotor representations from simulated to real environments. *arXiv preprint arXiv:1511.07111*, 2(3).

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2019. Deep domain confusion: Maximizing for domain invariance. arXiv 2014. *arXiv preprint arXiv:1412.3474*.

Vlachogiannis, D. M.; Wei, H.; Moura, S.; and Macfarlane, J. 2023. HumanLight: Incentivizing Ridesharing via Human-centric Deep Reinforcement Learning in Traffic Signal Control. *arXiv preprint arXiv:2304.03697*.

Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; and Li, Z. 2019a. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1290–1298.

Wei, H.; Chen, J.; Ji, X.; Qin, H.; Deng, M.; Li, S.; Wang, L.; Zhang, W.; Yu, Y.; Linc, L.; et al. 2022a. Honor of kings arena: an environment for generalization in competitive reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 11881–11892.

Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; and Li, Z. 2019b. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1913–1922.

Wei, H.; Zheng, G.; Gayah, V.; and Li, Z. 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(2): 12–18.

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; and Wu, D. O. 2020. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8): 8243–8256.

Zang, X.; Yao, H.; Zheng, G.; Xu, N.; Xu, K.; and Li, Z. 2020. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1153–1160.

Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; and Li, Z. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*, 3620–3624.

Zhao, W.; Queralta, J. P.; and Westerlund, T. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, 737–744. IEEE.

Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; and Li, Z. 2019. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1963–1972.