2023 IEEE International Conference on Smart Computing (SMARTCOMP) | 979-8-3503-2281-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMARTCOMP58114.2023.00036

Cooperative Multi-Agent Reinforcement Learning for Large Scale Variable Speed Limit Control

Yuhang Zhang†, Marcos Quinones-Grueiro†, William Barbour†, Zhiyao Zhang†, Joshua Scherer†, Gautam Biswas†, Daniel Work†

***Abstract*—Variable speed limit (VSL) control has emerged as a promising traffic management strategy for enhancing safety and mobility. In this study, we introduce a multi-agent reinforcement learning framework for implementing a large-scale VSL system to address recurring congestion in transportation corridors. The VSL control problem is modeled as a Markov game, using only data widely available on freeways. By employing parameter sharing among all VSL agents, the proposed algorithm can efficiently scale to cover extensive corridors. The agents are trained using a reward structure that incorporates adaptability, safety, mobility, and penalty terms; enabling agents to learn a coordinated policy that effectively reduces spatial speed variations while minimizing the impact on mobility. Our findings reveal that the proposed algorithm leads to a significant reduction in speed variation, which holds the potential to reduce incidents. Furthermore, the proposed approach performs satisfactorily under varying traffic demand and compliance rates.**

***Index Terms*—traffic control, variable speed limit, multi-agent reinforcement learning**

I. Introduction

The recent availability of comprehensive traffic data has paved the way for active traffic management (ATM) strategies in urban areas, which hold the potential to alleviate traffic congestion and enhance road safety. Among the various ATM strategies, variable speed limit (VSL) has emerged as a notable approach by dynamically adjusting speed limits in response to real-time traffic conditions.

Since the first introduction and deployment of VSL in Europe around the 1960s, the performance of VSL on road safety and travel mobility has been widely investigated both in simulation and in the field. Although the effect of VSL systems on traffic mobility is somehow controversial [1], they have proven to be useful in improving road safety by smoothing travel speeds to prevent the abrupt breaking of drivers [2]. In a micro-simulation study [3], the authors computed the real-time crash likelihood for a section on the I-4 freeway in Orlando. Their findings revealed that VSL may enhance traffic safety in medium-to-high-speed conditions. A separate study [4] proposed a proactive VSL control algorithm verifying in a micro-simulator that safety may be improved through speed homogeneity. In [5], the authors conducted a full Bayesian before-after analysis for a VSL system implemented on the I-5 freeway in Seattle. They found that the total crash count decreased significantly after VSL was applied.

In general, control algorithms for VSL systems can be classified into two categories: reactive (typically rule-based) and proactive. The former takes real-time decisions based on traffic characteristics such as volume, speed, and occupancy, with the goal of improving safety [5], [6]. The latter leverages a prediction model for mitigating the further deterioration of traffic conditions accounting for safety and mobility. In [7], the authors formulate VSL as an optimal control problem and show that traffic mobility can be improved substantially. In addition, the authors in [8] develop a fast model predictive control for VSL and illustrate its performance in a simulation study. The results demonstrate a better performance both in reducing travel time and lowering computation time when compared with other MPC methods. The authors in [9] develops a genetic algorithm to solve multi-objective variable speed limit problem and demonstrate that the proposed method outperforms the sequential quadratic programming. However, the execution of evolutionary methods in near real-time conditions is sometimes prohibitive, which makes it challenging to be deployed.

In recent years, the interest in developing data-driven algorithms for VSL control has grown steadily among researchers. As one of the most promising methods for sequential decision-making, reinforcement learning (RL) has been applied to solve VSL control problems [10]–[12]. The authors in [13] propose a Q-learning-based algorithm to reduce travel time considering a simulation setting with a single VSL controller and the assumption of capacity drop. In [14], a similar VSL control problem is framed to reduce traffic congestion through multiple VSL controllers considering Q-learning as the algorithm for decision-making. However, they implement identical speed limits for all VSL controllers ignoring the potential coordination among them. The study [15] considers a relatively large urban network as their testbed and verifies their proposed RL-based VSL control algorithm for mobility improvement and emission reduction. Notwithstanding, coordination among agents (VSL controllers) is not accounted for thus local optimum performance is only achievable.

We formulate VSL control as a multi-agent RL (MARL) problem to promote cooperation among agents aiming to

†Institute for Software Integrated Systems, Vanderbilt University, Nashville TN, 37212, USA. Emails: {yuhang.zhang.1, marcos.quinones.grueiro, william.w.barbour, zhiyao.zhang, joshua.r.scherer, gautam.biswas, dan.work}@vanderbilt.edu

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This work is supported from a grant from the U.S. Department of Transportation Grant Number 693JJ22140000Z44ATNREG3202. However, the U.S. Government assumes no liability for the contents or use thereof. The authors are grateful to Caliper for technical support on the TransModeler micro-simulation software used in this work.

2693-8340/23/$31.00 ©2023 IEEE
DOI 10.1109/SMARTCOMP58114.2023.00036

improve safety without sacrificing mobility for long corridors. There are few MARL-based VSL studies in the literature. To the best of our knowledge, the authors in [16] are the first to formulate VSL into a MARL problem. They consider a vehicle-to-infrastructure (V2I) environment in which the connected vehicles will guarantee driving at a specified speed. However, such an assumption makes deployment difficult because of the current lack of a viable V2I setting in which to test. Moreover, the scalability of the algorithm needs to be further investigated as there are only three control segments in the simulation case study considered. In [17], the authors apply W-learning (a variant of Q-learning for multi-objective tasks) to a simulation testbed with two VSL controllers. The centralized action-space design among different VSLs is doable for a small number of VSL controllers but will present issues to allow cooperation among agents for a large number of VSL controllers. Besides, it remains unknown how the algorithm performs in terms of generalizability and scalability.

The main research gap in this area is the absence of a MARL framework for VSL control that scales to large corridors with potential deployment ability. This work aims to address these goals by designing an approach that exclusively incorporates real-world traffic data and possesses the ability to scale up effectively to a long corridor. Specifically, the main contribution of this paper is a novel MARL framework for proactive VSL control in the context of recurring congestion to mainly improve safety with the following features:

- Scalability: designed to scale effectively to large-scale VSL systems.
- Generalizability: robust with respect to varying traffic demands and compliance rates.
- Feasibility: only requires commonly available real-world data which makes it promising to be deployed in near future.

The novelty of this work lies in the intersection of traffic control and MARL. Thus far, recent papers have not shown scalability nor generalizability to varying demand or compliance rate conditions. In addition, we present a novel design for a reward function to multi-agent VSL control considering a balance between safety and mobility plus accounting for real life constraints.

The remainder of the paper is organized as follows: Section II introduces preliminary concepts of MARL and one of the most popular algorithms MAPPO. The problem formulation and MARL framework design for VSL control are presented in Section III. In section IV, we describe the training and testing scenarios, as well as the parameter settings for the algorithm. The results and the respective discussions are included in Section V. Finally, the paper is concluded in Section VI where the future directions of work are also discussed.

II. Multi-agent reinforcement learning

In this section, we first introduce the notation and the general idea of RL. We review the preliminary concepts of MARL and describe one of the state-of-the-art algorithms in MARL called MAPPO.

A. Multi-agent reinforcement learning

Reinforcement learning (RL) focuses on finding a solution to sequential decision-making problems where an agent is expected to learn a policy by interacting with the environment through trial and error. Formally, RL is formulated as a Markov Decision Process (MDP), which can be defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes a reward signal, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes a probability transition function and $\gamma \in [0, 1]$ denotes the discount factor. The goal of RL algorithms is to maximize the cumulative discounted reward:

$$J(\theta) = \mathbb{E}_{s_t, a_t \sim \pi_\theta(a_t|s_t)} \left[\sum_{t=0}^{T} \gamma^t r_t \right] \tag{1}$$

where s_t, a_t, r_t are the state, action, and reward at time step t, respectively. In the context of data-driven models, neural networks are generally used to learn a policy π_θ (a mapping between the state and actions). Here θ denotes the neural network parameters.

Multi-agent RL (MARL) extends the standard RL paradigm to environments with multiple agents. Formally, the MARL problem can be modeled as a Markov Game, defined as a tuple $\langle \{\mathcal{S}^i\}_{i\in\{1,\dots,n\}}, \{\mathcal{A}^i\}_{i\in\{1,\dots,n\}}, \{\mathcal{R}^i\}_{i\in\{1,\dots,n\}}, P, n, \gamma \rangle$, where $\mathcal{S}^i$: denotes the local state space for agent i, $\mathcal{A}^i$: denotes the action space for agent i, $\mathcal{R}^i$: $\{\mathcal{S}^i\}_{i\in\{1,\dots,n\}} \times \{\mathcal{A}^i\}_{i\in\{1,\dots,n\}} \times \{\mathcal{S}^i\}_{i\in\{1,\dots,n\}} \to \mathbb{R}$ denotes the reward function for agent i, P: $\{\mathcal{S}^i\}_{i\in\{1,\dots,n\}} \times \{\mathcal{A}^i\}_{i\in\{1,\dots,n\}} \times \{\mathcal{S}^i\}_{i\in\{1,\dots,n\}} \to [0, 1]$ denotes the transition probability of the environment from a given state s to the next one s'. n denotes the total number of agents in the environment.

Similar to single-agent RL, the goal of MARL for each agent is to learn a policy that maximizes its own cumulative discounted reward:

$$J^i(\theta_1, \dots, \theta_n) = \mathbb{E}_{s_t, A_t} \left[\sum_{t=0}^{T} \gamma^t r_t^i \right], \tag{2}$$

where s_t denotes the global state at time step t and $A_t = (a_t^1, \dots, a_t^n)$ denotes the joint action of all agents at time step t. It is well known that MARL is a more challenging problem since multiple agents are simultaneously interacting with the environment and making decisions that affect each other's rewards, which will raise the issues such as non-stationarity and the credit assignment problem. Therefore, it is hard for the agents to learn to cooperate with each other to achieve a common goal while maximizing their individual rewards at the same time.

MARL algorithms can be broadly classified into two categories: centralized and decentralized learning. In fully centralized algorithms, there is a central controller that acquires information on the states and actions of all agents and outputs a joint action. This approach reduces the MARL problem to a

single agent but with a much larger state and action space. This method is generally inefficient as the synchronization between agents and the central controller costs time. Meanwhile, it has been shown that a fully centralized method could fail even for simple tasks [18]. In contrast, fully decentralized algorithms allow each agent to make decisions based only on its local observations. However, naive decentralized MARL methods are often unsuccessful because of the non-stationarity issue [18]. Recently, centralized training and decentralized execution (CTDE) framework has gained increasing attention in MARL research [19]–[21]. This framework involves training a centralized policy that observes the states and actions of all agents during training, while each agent executes a decentralized policy based on its local observations during deployment. The CTDE framework can improve the stability and scalability of MARL algorithms, especially in large-scale and complex environments, as it allows the agents to benefit from the centralized information during training while maintaining the flexibility and efficiency of decentralized execution.

B. Policy optimization methods

Policy optimization methods in standard RL focus on learning an explicit policy $\pi_\theta(a|s)$ by optimizing the objective function $J(\theta)$. While it is true that most policy optimization methods suffer from sample inefficiency, they usually demonstrate stable learning with monotonic performance improvement. In general, policy gradient for learning neural networks has the following form:

$$\nabla J(\theta) = \mathbb{E}_{s_t, a_t \sim \pi_\theta(a_t|s_t)} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \Phi_t \right] \quad (3)$$

where Φ_t is usually represented by estimated advantage function in order to reduce variance.

This work applies a policy optimization method called MAPPO [22], which is a variant of PPO [23] in a multi-agent setting. PPO is a widely used policy optimization method that applies a clipped surrogate objective to prevent large policy updates trying to guarantee stable learning. PPO utilizes an actor-critic architecture where the actor learns the policy $\pi_\theta(a|s)$ as a mapping from state to action distribution, while the critic learns the state value function $V_\phi(s_t)$ that estimates the cumulative reward for a given state.

MAPPO extends PPO to the multi-agent setting by including the following features:

- MAPPO adopts CTDE where each agent has its own actor network with local observations as input and individual actions as output while a centralized value function to estimate the global state-value function is implemented to facilitate coordination.
- PopArt [24] is implemented to gain a more stable value function learning.
- A small value for the mini-batch parameter is suggested in order to tackle the non-stationarity issue.

For more details, please refer to [22].

III. Problem formulation and reward design

Here we formulate the VSL control problem over multiple gantries into a cooperative MARL problem. Specifically, we consider a recurring congestion scenario induced by on-ramp merging flow which is commonly observed in the real world. Our goal is to learn a cooperative policy among multiple agents optimizing the following objectives:

- Adaptability: the posted speed limits should not be far too different from the real traffic speed, which aims to improve the drivers' comfort with posted speed limits.
- Safety: the posted speed limits should reduce spatial speed variation thus improving safety.
- Mobility: the posted speed limits should not significantly degrade the mobility.

We note here that our main goal is to improve safety with little deterioration in mobility although we do penalize the agents if they post unreasonable speed limits in terms of adaptability. The definition of agent, state space, action space, and reward function is as follows:

Agent: We consider each VSL controller located in a freeway gantry as a single agent with its own state and action space. A common visibility parameter representing the effective control area is applied to every agent. This setting is more realistic as each agent has the flexibility to make its own decision according to the location-based traffic condition. Meanwhile, as all agents have the same type of state space, action space, and common objectives, we consider this problem as homogeneous where all agents can share the same parameters.

State Space: The goal of state space design is to inform the agent of the traffic conditions upstream and downstream of the gantry's location. Because of the complex traffic dynamics and the lack of full supervision systems in the real world, it is quite complicated to obtain a comprehensive traffic state for each agent. In this paper, we extract the information from commonly available Radar Detection Systems (RDS) on highways to shape the state input of each agent. Specifically, the state for agent i can be defined as a tuple of three elements: $\langle \text{RDS}_{i-1}, \text{RDS}_i, \text{RDS}_{i+1} \rangle$, representing the extracted data including speed, volume, and occupancy from the closest RDS unit to the gantry of agent i, from one downstream RDS unit, and one upstream RDS. This setting informs each agent of the traffic conditions downstream and upstream allowing cooperation among them.

Action Space: We explicitly define the action space of each agent as discrete with values of $30, 50$, and 70 mile/hr, representing speed limits to be posted. We note here that this action space can be generalized to a more complex one by adding more speed limit values which will be the subject of future work. Yet, a small action space enables a decrease in the convergence time of the algorithm during training.

Reward Function: Designing a reward function for cooperative agents with multiple objectives is not trivial, especially when the objectives are conflicting. While multi-objective reinforcement learning (MORL) [25] approaches, which leverage

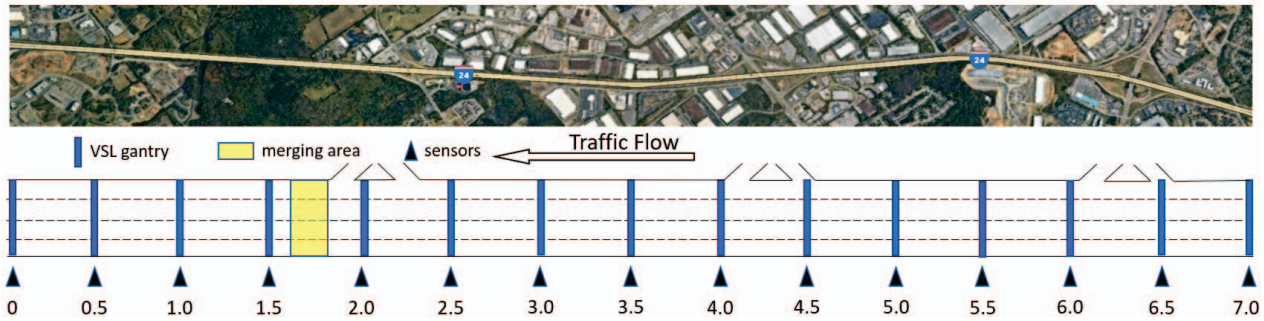

Fig. 1. The simulation network used for MARL training and testing.

a reward vector in place of a scalar, are primarily designed to tackle such complex problems, the overall performance of these methods remains inconclusive due to the scarcity of successful real-world applications in the literature.

Here we apply a linear combination of reward terms to optimize different objectives plus an optional term used to penalize invalid speed profiles. In order to encourage agents to cooperate, we define a global reward term shared among all agents. Local terms are computed for each agent individually. The formal definition for each reward term is presented below:

a) Adaptability term: The adaptability term is a **local** term used to penalize an agent posting high-speed limits when the traffic is in congestion. It can be described as:

$$r_{1t}^{i} = \begin{cases} -0.5 & \text{if } a_t^i = 50 \text{ and } \nu_t^i \leq 35 \\ -1 & \text{if } a_t^i = 70 \text{ and } \nu_t^i \leq 35 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where a_t^i denotes the selected action of agent i at timestep t, and ν_t^i denotes the real average traffic speed recorded by the RDS unit of agent i.

b) Mobility term: The mobility term is a **local** term used to encourage the agent to post a high-speed limit when the traffic condition is free-flow. It can be described as:

$$r_{2t}^{i} = \frac{\nu_t^i}{\nu_{\max}^i} \tag{5}$$

where $\nu_{\max}^i$ is a normalizer set to 70 in our experiments.

c) Safety term: The safety term is a **global** term used to encourage agents to coordinate in order to improve safety measured as a reduction in the coefficient of variation in speed (CVS) [26]. We first calculate CVS_t^i for agent i based on the collocated RDS unit and the closest upstream RDS unit at timestep t, and then a transformation function is applied to CVS_t^i with values less or equal to 0.2 as 0 and values greater than 0.6 as 0.6. The value after transformation is then defined as CVS_t^i. This transformation aims to remove noise from small values and make reward more sensitive. We then normalize CVS values among all agents at timestep t using the following equation:

$$\overline{\text{CVS}_t} = \begin{cases} \frac{\sum_{i=1}^{n}\left(\text{CVS}_t^i\right)_{>0.2}}{\sum_{i=1}^{n}\left(\mathbb{1}_{\text{CVS}_t^i>0.2}\right)} & \text{if } \sum_{i=1}^{n}\left(\mathbb{1}_{\text{CVS}_t^i>0.2}\right) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $(x)_{>0.2}$ returns the value x if $x > 0.2$ and 0 otherwise, $\mathbb{1}_{x>0.2}$ returns the value of 1 if $x > 0.2$ and 0 otherwise. The safety term can then be defined as:

$$r_{3t}^{i} = \begin{cases} \alpha\left(\frac{0.6-\overline{\text{CVS}_t}}{0.6-0.2}\right) & \text{if } \overline{\text{CVS}_t} \neq 0 \\ \alpha & \text{otherwise} \end{cases} \tag{7}$$

where α is a hyperparameter that determines the relative importance of this term compared to the others. Here we set $\alpha = 5$ to emphasize the importance of safety.

d) Invalid speed profile penalization term: In the real world, there are usually hard constraints for the speed limit drop between any two consecutive VSLs following the traffic direction (step-down rule). This optional **global** term aims to penalize invalid speed profiles where we define a speed profile with the differential of any two consecutive VSLs greater than 20 as invalid. We note here that this penalization term can be generalized to a more complicated speed limit set with a smaller allowable differential value. In addition, this term also penalizes the scenario with a bounce, where the joint policy is trying to speed up the traffic and then slow down. We use this additional term as one way for constraint implementation in Section V. This term can be described as:

$$r_{4t}^{i} = \sum_{i=1}^{n-1}\left(P_t^i\right) + Q_t \tag{8}$$

where,

$$P_t^i = \begin{cases} -0.1 & \text{if } a_t^{i+1} - a_t^i > 20 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$Q_t = \begin{cases} -0.2 & \text{if } \exists i \text{ such that } a_t^{i+1} < a_t^i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where P_t^i is used to penalize large differential speed limits and Q_t to penalize the bounce in speed limit profile. a_t^{i+1} denotes the selected action of one upstream agent for agent i.

Finally, the reward function for agent i at timestep t is as following:

$$r_t^i = wL_t^i + (1-w)G_t^i \tag{11}$$

$$L_t^i = r_{1t}^i + r_{2t}^i \tag{12}$$

$$G_t^i = r_{3t}^i + r_{4t}^i \tag{13}$$

where L_t^i and G_t^i denote the summation of all local terms and the summation of all global terms for agent i at timestep t, respectively. w is a hyperparameter aiming to balance the effect of local and global terms. Here we set $w = 0.5$.

IV. Experiments

In this section we discuss the experiment settings for both training and testing scenarios. The implementation details of the algorithm will be listed at the end of this section. We use TransModeler [27], [28] as our microscopic traffic simulator which allows customization of speed limits from Python through GISDK API.

A. Training scenario

This study considers a 7-mile corridor stretch with four lanes on I-24 westbound, Nashville. Three ramps present along this stretch are included in the experiment network. Most existing RL-basd VSL studies only focus on few number of VSLs thus lacking the ability to generate the full potential of VSL control. In this work, we set up VSL at 0.5-mile intervals, which results in 15 VSLs in general. Moreover, one RDS unit aggregating traffic data for 60s is assigned as collocated with each VSL to detect traffic conditions. The simulation starts at 7:50 AM and ends at 10:00 AM with the first 10-min as a warm-up period.

To introduce recurring congestion induced by on-ramp traffic weaving behavior, we set two lanes for the first downstream on-ramp with a flow around 1000 veh/lane/hr, which is fixed over the full simulation period. The mainstream inflow is set high enough as 1850 veh/lane/hr for the first hour to generate recurring congestion and reduces to half for the second hour to clear congestion. This setting mimics a full cycle of recurring congestion from the formation to the full clearance and therefore is selected as a training scenario for MARL-VSL system. Moreover, we consider a compliance rate of 100% during training. The simulation network is present in Figure 1.

We implement 5 agents right upstream of the on-ramp merging location in the training scenario with an expectation that 5 VSLs are enough for agents to learn cooperation while tackling different traffic conditions.

B. Testing scenario

The testing scenario is used to evaluate the generalizability and scalability of the learned policy. By customizing the mainstream flow values and the compliance rate, we can test the robustness of the learned policy when under variant traffic scenarios. Similarly, we can test the scalability of the learned policy by changing the number of implemented agents. Specifically, we have the following three scenarios:

TABLE I
Implementation details for training scenario

Parameters	Values
Episode length:	120
Training batch size:	120
Number of mini-batch:	1
Number of hidden layer:	1
Hidden layer size:	64
Actor learning rate:	7e-4
Critic learning rate:	5e-4
Number of PPO epochs:	15
Entropy coefficient:	0.01
Value loss coefficient:	1
Discount factor:	0.99

- *Scenario A:* Mainstream flow of **1750** veh/l/hr, **5%** compliance rate, **10** agents.
- *Scenario B:* Mainstream flow of **1850** veh/l/hr, **5%** compliance rate, **10** agents.
- *Scenario C:* Mainstream flow of **1950** veh/l/hr, **5%** compliance rate, **10** agents.

We note here that the on-ramp flow is the same as the training scenario and the mainstream flow reduces to half during the second hour to clear congestion. The performance of the learned policy is evaluated using two metrics: the Coefficient of Variation in Speed (CVS) and the Vehicle Hours Delay (VHD) in veh·hr, which are defined as follows:

$$\overline{\text{CVS}} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n}\left(\text{CVS}_t^i\right)_{>0.2}}{\sum_{t=1}^{T}\sum_{i=1}^{n}\left(\mathbb{1}_{\text{CVS}_t^i>0.2}\right)} \tag{14}$$

$$\text{VHD} = \sum_{t=1}^{T}\sum_{i=1}^{n}\left(\frac{L_i}{\nu_t^i} - \frac{L_i}{\nu_f}\right)V_t^i \tag{15}$$

where i is the index of VSL segment, L_i is the segment length, ν_f is the free-flow speed and is determined as 70 in this study. V_t^i is the volume recorded by RDS unit.

C. Implementation details

Table I shows the parameter settings for MAPPO algorithm in the training scenario. Since we consider this problem as a homogeneous problem, we share the parameters among all agents for both actor and critic network. All experiments are performed on a local machine with a 16-core CPU, 16 GB RAM and a NVIDIA GeForce RTX 3060 GPU. The default car following model is selected in TransModeler.

V. Results and Discussion

In this section, we present the training performance of the framework proposed in Section III and the evaluation of the generalizability and scalability of learned policy.

A. Training performance

The training takes about 9.5 hr of wall-clock time on the aforementioned machine to finish. The training curve is presented in Figure 3. On the one hand, we can observe

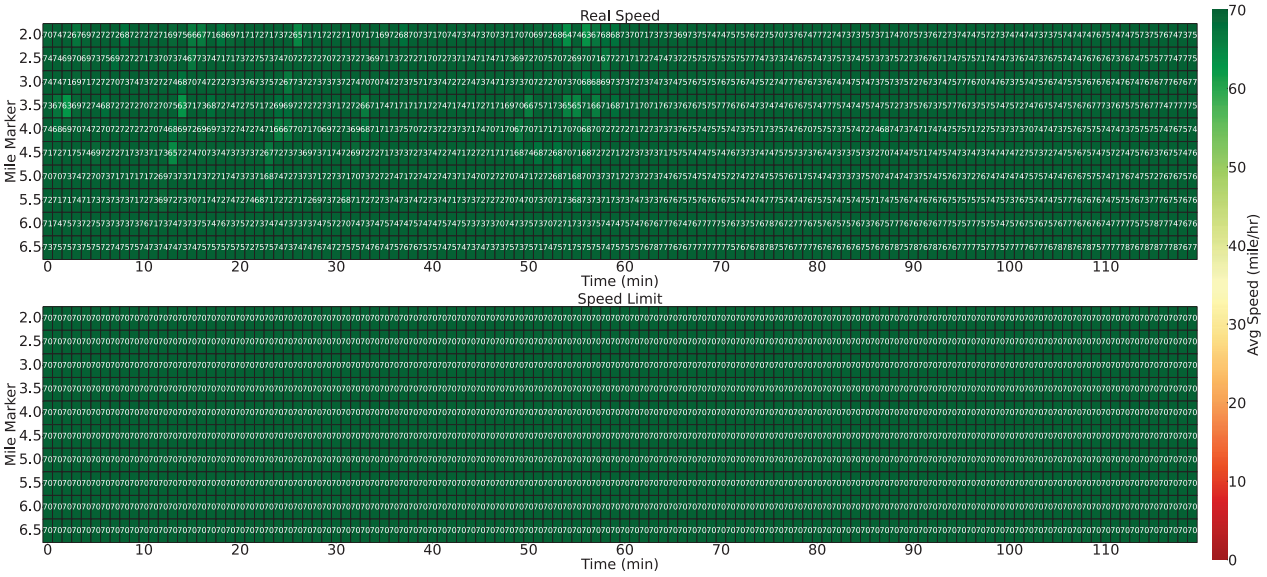

Fig. 2. Real traffic speed (top) and speed limits (bottom) generated by MARL-VSL in Scenario A.

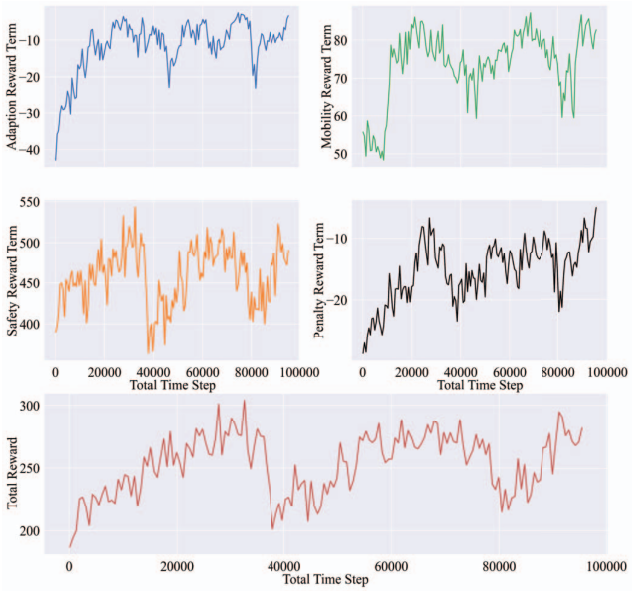

Fig. 3. The learning curve of the total reward, adaption reward term, mobility reward term and safety reward term.

a relatively stable learning process for the adaptation and mobility reward terms, indicating that they allow learning harmoniously without hindering each other's learning ability for the agents. On the other hand, the safety reward term shows a more fluctuating pattern during the training phase, which can be attributed to the inherent complexities in learning coordination and the contradicting nature of the safety and mobility objectives. Despite the oscillations, an overall increasing trend in the safety reward term's learning pattern can be appreciated. This trend means that the agents gradually improve their ability to balance safety and mobility over time, demonstrating the effectiveness of our designed reward function.

These insights from the training curve reinforce the validity of our approach and highlight its potential for addressing more complex, real-world scenarios where adaptation, mobility, and safety are all crucial.

B. Testing performance: generalizability and scalability analysis

Figure 2, Figure 4 and Figure 5 present the time-space diagram of real traffic speed and the corresponding speed limits generated by MARL-VSL in testing Scenarios A, B, and C, respectively. In Scenario A, we have a mainstream inflow rate much lower than the one in the training scenario which does not trigger any congestion. Correspondingly, the MARL-VSL agents post max speed limits during the whole period which is the best for mobility, adaption, and safety as well. Conversely, we appreciate a high mainstream inflow rate in Scenario C, which leads to further congestion upstream. Surprisingly, MARL-VSL generalizes well to this scenario where the agents post low-speed limits for the congested area, high-speed limits for the freeflow area, and speed limits in between for the congested tail area to prevent abrupt breaking. Scenario B has similar traffic demand settings as the training scenarios considered.

These results demonstrate that MARL-VSL generalizes satisfactorily to different traffic demand conditions. In addition, the compliance rate in all three testing scenarios is around 5% which is far lower than the 100% considered in the training scenario. This emphasizes the robustness of MARL-VSL with respect to varying compliance rates, which implies that it potentially may be applied to deployment sites where drivers have different compliance backgrounds.

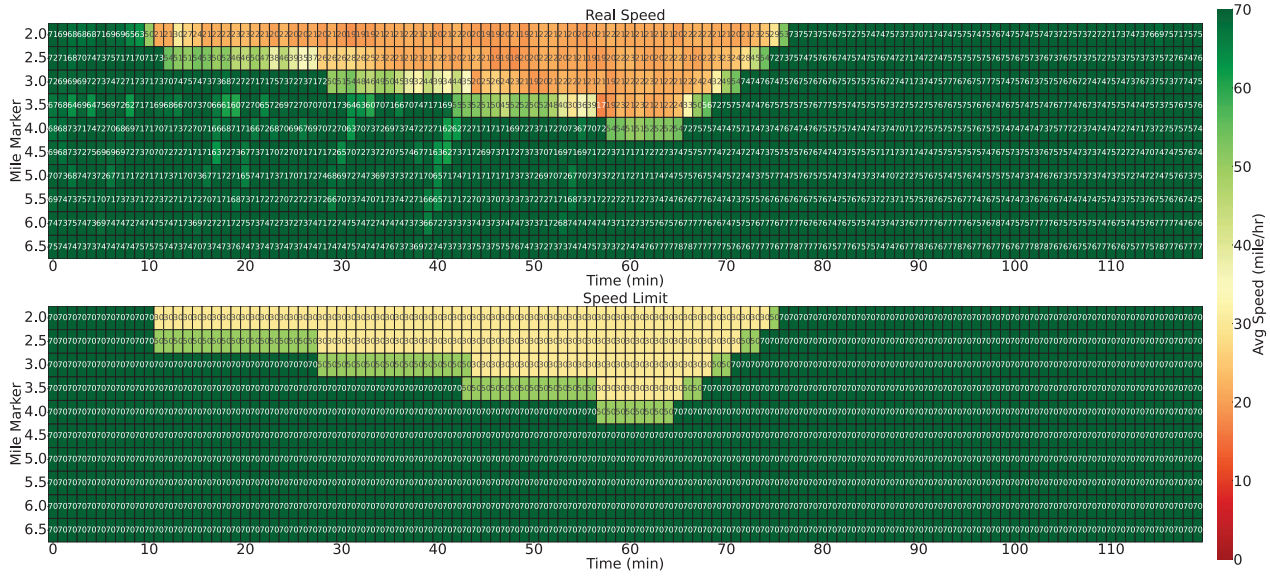

Fig. 4. Real traffic speed (top) and speed limits (bottom) generated by MARL-VSL in Scenario B.

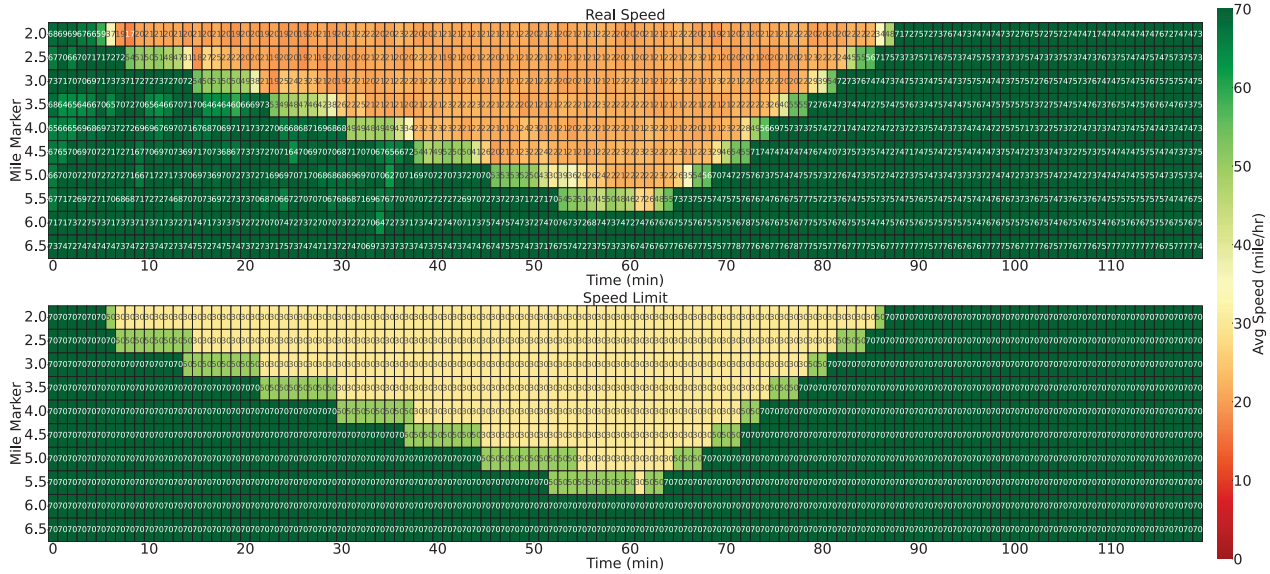

Fig. 5. Real traffic speed (top) and speed limits (bottom) generated by MARL-VSL in Scenario C.

Finally, it is worth highlighting that the number of agents has been increased to 10 in all testing scenarios, compared to 5 agents utilized during the training phase. The successful scalability with respect to the number of agents can be ascribed to the homogeneous training setting, along with the effective communication and reward design. This demonstrates the potential applicability of the proposed MARL-VSL framework to address more complex, large-scale VSL control systems, paving the way for tackling more complex traffic scenarios in real-world conditions.

TABLE II
THE MEAN (AND STD) OF TRAFFIC MEASURES UNDER NO CONTROL AND MARL-VSL CONTROL.

	Scenario B	Scenario C
	CVS	CVS
No control	0.52 (0.01)	0.52 (0.01)
MARL-VSL	**0.35 (0.01)**	**0.36 (0.01)**
	VHD	VHD
No control	**289.98 (22.27)**	**671.65 (36.27)**
MARL-VSL	304.59 (20.01)	715.26 (41.81)

C. Testing performance: quantitative analysis

We run 10 experiments with and without MARL-VSL control for both Scenarios B and C to evaluate the safety and mobility performance of MARL-VSL (Scenario A is not included since it does not trigger congestion thus showing no differences between no control and MARL-VSL control). The statistics of traffic metrics are reported in Table II. It is observed that when using MARL-VSL control, the safety metric can be improved up to 32.7% with a little negative effect on the mobility measure, i.e., up to 6.5%. It should be noted that despite the observed slight increase in VHD under MARL-VSL control, a potential reduction in the number of primary or secondary incidents may contribute to an overall enhancement in mobility metrics. Furthermore, this finding aligns with the contentious conclusion regarding the impact of VSL on mobility, as its effectiveness is strongly influenced by factors such as network geometry and the presence of capacity drops.

VI. Conclusions and future work

In this paper, we propose a MARL framework for VSL control with traffic safety improvement as the primary objective. The proposed algorithm is evaluated under different traffic conditions and performs satisfactorily, especially under varying traffic demand and compliance rates. In addition, we demonstrate that the learned policy scales to more VSL controllers than those considered during training. Although the proposed approach causes a slight delay, it significantly smooths traffic speed thus improving safety. In future work (1) we will extend the action space to a more realistic setting and then compare with the state-of-the-art algorithm, (2) we will analyze the impact on mobility by testing different reward functions under different traffic settings, and (3) we will explore other methods to encourage agents satisfying real-world constraints such as step-down speed rules.

References

[1] X.-Y. Lu and S. E. Shladover, "Review of variable speed limits and advisories: Theory, algorithms, and practice," *Transportation Research Record*, vol. 2423, no. 1, pp. 15–23, 2014.

[2] E. De Pauw, S. Daniels, L. Franckx, and I. Mayeres, "Safety effects of dynamic speed limits on motorways," *Accident Analysis & Prevention*, vol. 114, pp. 83–89, 2018.

[3] M. Abdel-Aty, J. Dilmore, and A. Dhindsa, "Evaluation of variable speed limits for real-time freeway safety improvement," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 335–345, 2006.

[4] R. Yu and M. Abdel-Aty, "An optimal variable speed limits system to ameliorate traffic safety risk," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 235–246, 2014.

[5] Z. Pu, Z. Li, Y. Jiang, and Y. Wang, "Full bayesian before-after analysis of safety effects of variable speed limit system," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 2, pp. 964–976, 2020.

[6] B. Khondaker and L. Kattan, "Variable speed limit: an overview," *Transportation Letters*, vol. 7, no. 5, pp. 264–278, 2015.

[7] R. C. Carlson, I. Papamichail, M. Papageorgiou, and A. Messmer, "Optimal mainstream traffic flow control of large-scale motorway networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 2, pp. 193–212, 2010.

[8] Y. Han, A. Hegyi, Y. Yuan, S. Hoogendoorn, M. Papageorgiou, and C. Roncoli, "Resolving freeway jam waves by discrete first-order model-based predictive control of variable speed limits," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 405–420, 2017.

[9] M. Yu and W. Fan, "Optimal variable speed limit control at a lane drop bottleneck: Genetic algorithm approach," *Journal of Computing in Civil Engineering*, vol. 32, no. 6, p. 04018049, 2018.

[10] K. Kušić, E. Ivanjko, M. Gregurić, and M. Miletić, "An overview of reinforcement learning methods for variable speed limit control," *Applied Sciences*, vol. 10, no. 14, p. 4917, 2020.

[11] Y. Wu, H. Tan, and B. Ran, "Differential variable speed limits control for freeway recurrent bottlenecks via deep reinforcement learning," *arXiv preprint arXiv:1810.10952*, 2018.

[12] T. Schmidt-Dumont and J. H. van Vuuren, "Decentralised reinforcement learning for ramp metering and variable speed limits on highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 8, p. 1, 2015.

[13] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3204–3217, 2017.

[14] E. Walraven, M. T. Spaan, and B. Bakker, "Traffic flow optimization: A reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, 2016.

[15] F. Zhu and S. V. Ukkusuri, "Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach," *Transportation research part C: emerging technologies*, vol. 41, pp. 30–47, 2014.

[16] C. Wang, J. Zhang, L. Xu, L. Li, and B. Ran, "A new solution for freeway congestion: Cooperative speed limit control using distributed reinforcement learning," *IEEE Access*, vol. 7, pp. 41947–41957, 2019.

[17] K. Kušić, I. Dusparic, M. Guériau, M. Gregurić, and E. Ivanjko, "Extended variable speed limit control using multi-agent reinforcement learning," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, IEEE, 2020.

[18] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[19] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[20] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[21] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7234–7284, 2020.

[22] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with popart," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3796–3803, 2019.

[25] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, p. 26, 2022.

[26] Y. Zhang, M. Quinones-Grueiro, W. Barbour, C. Weston, G. Biswas, and D. Work, "Quantifying the impact of driver compliance on the effectiveness of variable speed limits and lane control systems," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3638–3644, IEEE, 2022.

[27] R. Balakrishna, D. Morgan, H. Slavin, and Q. Yang, "Large-scale traffic simulation tools for planning and operations management," *IFAC Proceedings Volumes*, vol. 42, no. 15, pp. 117–122, 2009.

[28] Q. Yang and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996.