EMVLight: A Decentralized Reinforcement Learning Framework for Efficient Passage of Emergency Vehicles

Haoran Su, [1*] Yaofeng Desmond Zhong, [2] Biswadip Dey, [2] Amit Chakraborty [2]

[1]New York University,
[2]Siemens Corporation, Technology
haoran.su@nyu.edu, {yaofeng.zhong, biswadip.dey, amit.chakraborty}@siemens.com

Abstract

Emergency vehicles (EMVs) play a crucial role in responding to time-critical events such as medical emergencies and fire outbreaks in an urban area. The less time EMVs spend traveling through the traffic, the more likely it would help save people's lives and reduce property loss. To reduce the travel time of EMVs, prior work has used route optimization based on historical traffic-flow data and traffic signal pre-emption based on the optimal route. However, traffic signal pre-emption dynamically changes the traffic flow which, in turn, modifies the optimal route of an EMV. In addition, traffic signal pre-emption practices usually lead to significant disturbances in traffic flow and subsequently increase the travel time for non-EMVs. In this paper, we propose EMVLight, a decentralized reinforcement learning (RL) framework for simultaneous dynamic routing and traffic signal control. EMVLight extends Dijkstra's algorithm to efficiently update the optimal route for the EMVs in real time as it travels through the traffic network. The decentralized RL agents learn network-level cooperative traffic signal phase strategies that not only reduce EMV travel time but also reduce the average travel time of non-EMVs in the network. This benefit has been demonstrated through comprehensive experiments with synthetic and real-world maps. These experiments show that EMVLight outperforms benchmark transportation engineering techniques and existing RL-based signal control methods.

1 Introduction

Emergency vehicles (EMVs) are vehicles including ambulances, fire trucks, and police cars, which respond to critical events such as medical emergencies, fire disasters, and criminal activities. Emergency response time is the key indicator of a city's emergency management capability. Reducing response time saves lives and prevents property loss. For instance, the survival rate from a sudden cardiac arrest drops 7% - 10% for every minute without treatment, and there is barely any chance to survive after 8 minutes. EMV travel time, the time interval for an EMV to travel from a rescue station to an incident site, accounts for a majority of the emergency response time. However, overpopulation and urbanization have been worsening road congestion, making

it more and more challenging for fast EMV passage. Analytics (2021) have shown that the average EMV travel time increases from 7.2 minutes in 2015 to 10.1 minutes in 2021 in New York City. Thus, reducing the average EMV travel time on increasingly crowded roads is a significant problem which can help save people's lives.

Existing works have studied strategies to reduce the travel time of EMVs by route optimization and traffic signal pre-emption. *Route optimization* usually refers to the search for a time-based shortest path. The traffic network (e.g., city road map) is modeled as a graph with intersections as nodes and road segments between intersections as edges. Based on the time a vehicle needs to travel through each edge (road segment), route optimization calculates an optimal route such that an EMV can travel from the rescue station to the incident site in the least amount of time. In addition, as the EMV needs to be as fast as possible, the law in most places requires non-EMVs to yield to emergency vehicles sounding sirens, regardless of the traffic signals at intersections. Even though this practice gives the right-of-way to EMVs, it poses safety risks for vehicles and pedestrians at the intersections. To address this safety concern, existing methods have also studied *traffic signal pre-emption*, which refers to the process of deliberately altering the signal phases at each intersection to prioritize EMV passage.

However, as the traffic condition constantly changes, an optimal route returned by route optimization can potentially become suboptimal as an EMV travels through the network. Moreover, traffic signal pre-emption has a significant impact on the traffic flow, which would change the fastest route as well. Thus, the optimal route should be updated with real-time traffic flow information, i.e., the route optimization should be solved in a dynamic (time-dependent) way. As an optimal route can change as an EMV travels through the traffic network, the traffic signal pre-emption would need to adapt accordingly. In other words, the subproblems of route optimization and traffic signal pre-emption are coupled and should be solved simultaneously in real-time. Existing approaches does not address this coupling.

In addition, most of the existing models on emergency vehicle service have a single objective of reducing the EMV travel time. As a result, their traffic signal control strategies have an undesirable effect of increasing the travel time of non-EMVs, since only EMV passage is optimized. In this

---

paper, we aim to perform route optimization and traffic signal pre-emption to not only reduce EMV travel time but also to reduce the average travel time of non-EMVs. In particular, we address the following two key challenges:

- **How to dynamically route an EMV to a destination under time-dependent traffic conditions in a computationally efficient way?** As the congestion level of each road segment changes over time, the routing algorithm should be able to adjust the remaining route as the EMV passes each intersection. Running the shortest-path algorithm each time the EMV passes an intersection is not efficient. A computationally efficient dynamic routing algorithm is desired.

- **How to coordinate traffic signals to not only reduce EMV travel time but reduce the average travel time of non-EMVs as well?** To reduce EMV travel time, only the traffic signals along the route of the EMV need to be altered. However, to further reduce average non-EMV travel time, traffic signals in the whole traffic network need to be operated cooperatively.

To tackle these challenges, we propose **EMVLight**, a decentralized multi-agent reinforcement learning framework with a dynamic routing algorithm to control traffic signal phases for efficient EMV passage. Our experimental results demonstrate that EMVLight outperforms traditional traffic engineering methods and existing RL methods under two metrics - EMV travel time and the average travel time of all vehicles - on different traffic configurations.

2 Related Work

Conventional routing optimization and traffic signal pre-emption for EMVs. Although, in reality, routing and pre-emption are coupled, existing methods usually solve them separately. Many of the existing approaches leverage Dijkstra's shortest path algorithm or A* algorithm to get the optimal route (Wang, Ma, and Yang 2013; Mu, Song, and Liu 2018; Kwon, Kim, and Betts 2003; Jotshi, Gong, and Batta 2009; Nordin et al. 2012). However, as this line of work assumes that the routes and traffic conditions are fixed and static, they fail to address the dynamic nature of real-world traffic flows. Another line of work has considered the change of traffic flows over time. Ziliaskopoulos and Mahmassani (1993) have proposed a shortest-path algorithm for time-dependent traffic networks, but the travel time associated with each road at each time step is assumed to be known in prior. Musolino et al. (2013) propose different routing strategies for different times in a day (e.g., peak/non-peak hours) based on traffic history data at those times. However, in the problem of our consideration, routing and pre-emption strategies can significantly affect the travel time associated with each edge during the EMV passage, and the existing methods cannot deal with this kind of real-time changes. Haghani, Hu, and Tian (2003) formulated the dynamic shortest path problem as a mixed-integer programming problem. Koh et al. (2020) have used RL for real-time vehicle navigation and routing. However, both of these studies have tackled a general routing problem, and signal pre-emption and its influence on traffic have not been modeled.

Once an optimal route for the EMV has been determined, traffic signal pre-emption can be deployed to further reduce EMV travel time. A common pre-emption strategy is to extend the green phases of green lights to let the EMV pass each intersection along a fixed optimal route (Wang, Ma, and Yang 2013; Bieker-Walz and Behrisch 2019; Asaduzzaman and Vidyasankar 2017).

Please refer to Lu and Wang (2019) and Humagain et al. (2020) for a thorough survey of conventional routing optimization and traffic signal pre-emption methods. We would also like to point out that the conventional methods prioritize EMV passage and have significant disturbances on the traffic flow which increases the average non-EMV travel time.

RL-based traffic signal control. Traffic signal pre-emption only alters the traffic phases at the intersections where an EMV travels through. However, to reduce congestion, traffic phases at nearby intersections also need to be changed cooperatively. The coordination of traffic signals to mitigate traffic congestion is referred to as traffic signal control which has been addressed by leveraging deep RL in a growing body of work. Many of the existing approaches use Q-learning (Abdulhai, Pringle, and Karakoulas 2003; Prashanth and Bhatnagar 2010; Wei et al. 2019a,b; Zheng et al. 2019; Chen et al. 2020; Zang et al. 2020). Another line of work has used actor-critic algorithms for traffic signal control (El-Tantawy, Abdulhai, and Abdelgawad 2013; Aslani, Mesgari, and Wiering 2017; Chu et al. 2019; Xu et al. 2021). Please refer to Wei et al. (2019c) for a review on traffic signal control methods. These RL-based traffic control methods, however, focus on reducing the congestion in the traffic network and are not designed for EMV pre-emption. In contrast, our RL framework is built upon state-of-the-art ideas such as max pressure and is designed to reduce both EMV travel time and overall congestion.

Another line of research to reduce EMV travel time focuses on link level EMV traversal, e.g., fixed lane and change lane strategies proposed by Agarwal and Paruchuri (2016); Ismath et al. (2019); Su et al. (2021). However, our paper does not focus on link level traversal and we leave it as a future work.

3 Preliminaries

Definition 1 (traffic map, link, lane) *A traffic map can be represented by a graph $G(\mathcal{V}, \mathcal{E})$, with intersections as nodes and road segments between intersections as edges. We refer to a one-directional road segment between two intersections as a link. A link has a fixed number of lanes, denoted as $h(l)$ for lane l. Fig. 1 shows 8 links and each link has 2 lanes.*

Definition 2 (Traffic movements) *A traffic movement (l, m) is defined as the traffic traveling across an intersection from an incoming lane l to an outgoing lane m. The intersection shown in Fig. 1 has 24 permissible traffic movements. The set of all permissible traffic movements of an intersection is denoted as $\mathcal{M}$.*

Definition 3 (Traffic signal phase) *A traffic signal phase is defined as the permissible combination traffic movements. The intersection with 4 links in Fig. 2 has 8 phases.*
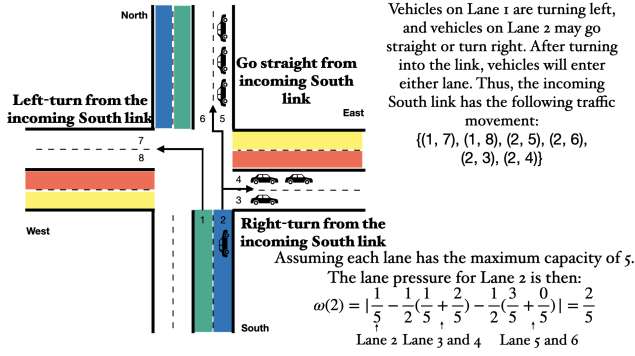
Figure 1: Traffic movements illustration and an example pressure calculation for incoming lane #2.

Figure 2: *Top*: 8 signal phases; *Left*: phase #2 illustration; *Right*: phase #5 illustration.

Definition 4 (Pressure of an incoming lane) *The pressure of an incoming lane l measures the unevenness of vehicle density between lane l and corresponding out going lanes in permissible traffic movements. The vehicle density of a lane is $x(l)/x_{max}(l)$, where $x(l)$ is the number of vehicles on lane l and $x_{max}(l)$ is the vehicle capacity on lane l, which is related to the length of a lane. Then the pressure of an incoming lane l is*

$$w(l) = \left| \frac{x(l)}{x_{max}(l)} - \sum_{\{m|(l,m)\in\mathcal{M}\}} \frac{1}{h(m)} \frac{x(m)}{x_{max}(m)} \right|, \quad (1)$$

where $h(m)$ is the number of lanes of the outgoing link which contains m. In Fig. 1, $h(m) = 2$ for all the outgoing lanes. An example for Eqn. (1) is shown in Fig. 1.

Definition 5 (Pressure of an intersection) *The pressure P of an intersection is the average of the pressure of all incoming lanes.*

The pressure of an intersection indicates the unevenness of vehicle density between incoming and outgoing lanes in an intersection. Intuitively, reducing the pressure leads to more evenly distributed traffic, which indirectly reduce congestion and the average travel time of vehicles.

4 Dynamic Routing

Dijkstra's algorithm is a general algorithm that finds shortest path between a given node and every other nodes in a graph. It has been widely used for EMV routing. The EMV travel time along each link, which is referred to as the *intra-link travel time.* is estimated based on the number of vehicles on that link. With the traffic graph, the intra-link travel time and a destination, Dijkstra's algorithm calculates the time-based shortest path as well as estimated travel time from each intersection to the destination. The latter is usually referred to as the *estimated time of arrival* (ETA) of each intersection.

However, traffic conditions are constantly changing and so does the EMV intra-link travel time. Moreover, EMV pre-emption techniques alters traffic signal phases, which will significantly change the traffic condition as the EMV travels. The pre-determined shortest path might become congested
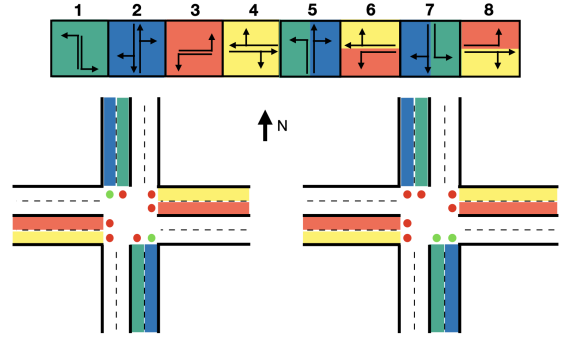

Algorithm 1: Dynamic Dijkstra's for EMV routing

Input :
  $G = (\mathcal{V}, \mathcal{E})$ traffic map as a graph
  $T^t = [T_{ij}^t]$ intra-link travel time at time t
  i_d index of the destination

Output:
  $\mathsf{ETA}^t = [\mathsf{ETA}_i^t]$ ETA of each intersection
  $\mathsf{Next}^t = [\mathsf{Next}_i^t]$ next intersection to go
               from each intersection

  /* pre-population   */
1 $\mathsf{ETA}^0, \mathsf{Next}^0 = \mathtt{Dijkstra}(G, T^0, i_d)$
  /* dynamic routing   */
2 **for** $t = 0 \to T$ **do**
3   **foreach** $i \in \mathcal{V}$ **do (in parallel)**
4       $\mathsf{ETA}_i^{t+1} \leftarrow \min_{(i,j)\in\mathcal{E}}(\mathsf{ETA}_j^t + T_{ji}^t)$
5       $\mathsf{Next}_i^{t+1} \leftarrow \arg\min_{\{j|(i,j)\in\mathcal{E}\}}(\mathsf{ETA}_j^t + T_{ji}^t)$

---

due to stochasticity and pre-emption. Thus, updating the optimal route dynamically can facilitate EMV passage. In theory we can run Dijkstra's algorithm frequently as the EMV travels through the network to take into account the updated EMV intra-link travel time, but this is inefficient.

To achieve dynamics routing, we extend Dijkstra's algorithm to efficiently update the optimal route based on the updated intra-link travel times. As shown in Algorithm 1, first a prepopulation process is carried out where a (static) Dijkstra's algorithm is run to get the ETA from each intersection to the destination. For each intersection, the next intersection along the shortest path is also calculated and stored. We assume this process can be done before the EMV starts to travel. This is reasonable since a sequence of processes, including call-taker processing, are performed before the EMVs are dispatched. Once the pre-population process is finished, we can update ETA and Next for each intersection efficiently in parallel, since the update only depends on information of neighboring intersections. Please see Appendix for how intra-link travel time is estimated in real time.

Remark 1 *In static Dijkstra's algorithm, the shortest path is obtained by repeatedly querying the* Next *attribute of*

each node from the origin until the destination is reached. In our dynamic Dijkstra's algorithm, since the shortest path changes, we only care about the immediate next intersection to go to, which is exactly Next_i for any intersection i.

5 Reinforcement Learning Formulation

While dynamic routing directs the EMV to the destination, it does not take into account the possible waiting times for red lights at the intersections. Thus, traffic signal pre-emption is also required for the EMV to arrive at the destination in the least amount of time. However, since traditional pre-emption only focuses on reducing the EMV travel time, the average travel time of non-EMVs can increase significantly. Thus, we set up traffic signal control for efficient EMV passage as a decentralized RL problem. In our problem, an RL agent controls the traffic signal phases of an intersection based on local information. Multiple agents coordinate the control signal phases of intersections cooperatively to **(1)** reduce EMV travel time and **(2)** reduce the average travel time of non-EMVs. First, we present 3 types of agents. Then we present agent design and multi-agent A2C framework.

5.1 Types of agents for EMV passage

When an EMV is on duty, we distinguish 3 types of traffic control agents based on EMV location and routing (Fig. 3). An agent is a *primary pre-emption agent* i_p if an EMV is on one of its incoming links. The agent of the next intersection $i_s = \mathsf{Next}_{i_p}$ is refered to as a *secondary pre-emption agent*. The rest of the agents are *normal agents*. We design these types since different agents have different local goals, which is reflected in their reward designs, as described below.

5.2 Agent design

- **State**: The state of an agent i at time t is denoted as s_i^t and it includes the number of vehicles on each outgoing lanes and incoming lanes, the distance of the EMV to the intersection, the estimated time of arrival (ETA), and which link the EMV will be routed to (Next), i.e.,

$$s_i^t = \{x^t(l), x^t(m), d_{\mathsf{EMV}}^t[L_{ji}], \mathsf{ETA}_i^{t'}, \mathsf{Next}_i^{t'}\}, \quad (2)$$

  where L_{ji} represents the incoming links of intersection i. With a slight abuse of notation l and m denote the set of incoming and outgoing lanes, respectively. For the intersection shown in Fig. 1, d_{EMV}^t is a vector of four elements. For primary pre-emption agents, one element represents the distance of EMV to the intersection in the corresponding link, while the rest of the elements are set to -1. For all other agents, d_{EMV}^t are padded with -1.

- **Action**: Prior work has focused on using phase switch, phase duration and phase itself as actions. In this work, we define the action of an agent as one of the 8 phases in Fig. 2; this enables more flexible signal patterns as compared to the traditional cyclical patterns. Due to safety concerns, once a phase has been initiated, it should remain unchanged for a minimum amount of time, e.g. 5 seconds. Because of this, we set our MDP time step length to be 5 seconds to avoid rapid switch of phases.
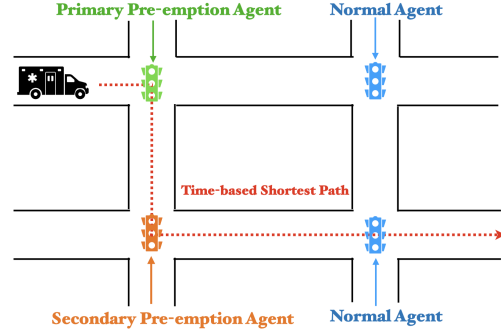

Figure 3: Three types of agents.

- **Reward**: PressLight (Wei et al. 2019a) has shown that minimizing the pressure is an effective way to encourage efficient vehicle passage, we adopt a similar idea for normal agents. For secondary pre-emption agents we additionally encourage less vehicles on the link between the primary and the secondary agent in order to encourage efficient EMV passage. For primary pre-emption agents, we simply assign a unit penalty at each time step to encourage fast EMV passage. Thus, depending on the agent type, the local reward for agent i at time t is

$$r_i^t = \begin{cases} -P_i^t & i \notin \{i_p, i_s\}, \\ -\beta P_{i_s}^t - \frac{1-\beta}{|L_{i_p i_s}|} \sum_{l \in L_{i_p i_s}} \frac{x(l)}{x_{max}(l)} & i = i_s, \\ -1 & i = i_p. \end{cases} \quad (3)$$

Justification of agent design. The quantities in local agent state can be obtained at each intersection using various technologies. Numbers of vehicles on each lane $(x^t(l), x^t(m))$ can be obtained by vehicle detection technologies, such as inductive loop (Gajda et al. 2001), based on the hardware installed underground. The distance of the EMV to the intersection $d_{\mathsf{EMV}}^t[L_{ji}]$ can be obtained by *vehicle-to-infrastructure* technologies such as VANET(Buchenscheit et al. 2009), which broadcasts the real-time position of a vehicle to an intersection. Prior work by Wang et al. (2013) and Noori, Fu, and Shiravi (2016) have explored these technologies for traffic signal pre-emption.

The dynamic routing algorithm (Algorithm 1) can provide $(\mathsf{ETA}, \mathsf{Next})$ for each agent at every time step. However, due to the stochastic nature of traffic flows, updating the route too frequently might confuse the EMV driver, since the driver might be instructed a new route, say, every 5 seconds. There are many ways to ensure reasonable frequency. One option is to inform the driver only once while the EMV is travels in a single link. We implement it by updating the state of an RL agent $(\mathsf{ETA}_i^{t'}, \mathsf{Next}_i^{t'})$ at the time step when the EMV travels through half of a link. For example, if the EMV travels through a link to agent i from time step 11 to 20 in constant speed, then dynamic routing information in s_i^{16} to s_i^{20} are the same, which is $(\mathsf{ETA}_i^{15}, \mathsf{Next}_i^{15})$, i.e., $t' = 15$.

As for the reward design, one might wonder how an agent is able to know its type. As we assume that an agent can

observe the state of its neighbors, agent type can be inferred from the observation. This will become clearer below.

5.3 Multi-agent Advantage Actor-critic

We adopt a multi-agent advantage actor-critic (MA2C) framework similar to Chu et al. (2019). The difference is that our local state includes dynamic routing information and our local reward encourages efficient passage of EMV. Here we briefly introduce the MA2C framework. Please refer to Chu et al. (2019) for additional details.

In a multi-agent network $G(\mathcal{V}, \mathcal{E})$, the neighborhood of agent i is denoted as $\mathcal{N}_i = \{j | ji \in \mathcal{E} \text{ or } ij \in \mathcal{E}\}$. The local region of agent i is $\mathcal{V}_i = \mathcal{N}_i \cup i$. We define the distance between two agents $d(i, j)$ as the minimum number of edges that connect them. For example, $d(i, i) = 0$ and $d(i, j) = 1, \forall j \in \mathcal{N}_i$. In MA2C, each agent learns a policy π_{θ_i} (actor) and the corresponding value function V_{ϕ_i} (critic), where θ_i and ϕ_i are learnable neural network parameters of agent i.

Local Observation. In an ideal setting, agents can observe the states of every other agent and leverage this global information to make a decision. However, this is not practical in our problem due to communication latency and it will cause scalability issues. We assume agents can observe its own state and the states of its neighbors, i.e., $s_{\mathcal{V}_i}^t = \{s_j^t | j \in \mathcal{V}_i\}$. The agents feed this observation to its policy network π_{θ_i} and value network V_{ϕ_i}.

Fingerprint. In multi-agent training, each agent treats other agents as part of the environment, but the policy of other agents are changing over time. Foerster et al. (2017) introduce *fingerprints* to inform agents about the changing policies of neighboring agents in multi-agent Q-learning. Chu et al. (2019) bring fingerprints into MA2C. Here we use the probability simplex of neighboring policies $\pi_{\mathcal{N}_i}^{t-1} = \{\pi_j^{t-1} | j \in \mathcal{N}_i\}$ as fingerprints, and include it into the input of policy network and value network. Thus, our policy network can be written as $\pi_{\theta_i}(a_i^t | s_{\mathcal{V}_i}^t, \pi_{\mathcal{N}_i}^{t-1})$ and value network as $V_{\phi_i}(\tilde{s}_{\mathcal{V}_i}^t, \pi_{\mathcal{N}_i}^{t-1})$, where $\tilde{s}_{\mathcal{V}_i}^t$ is the local observation with spatial discount factor, which is introduced below.

Spatial Discount Factor and Adjusted Reward. MA2C agents cooperatively optimize a global cumulative reward. We assume the global reward is decomposable as $r_t = \sum_{i \in \mathcal{V}} r_i^t$, where r_i^t is defined in Eqn. (3). Instead of optimizing the same global reward for every agent, Chu et al. (2019) propose a spatial discount factor α to let each agent pay less attention to rewards of agents far away. The adjusted reward for agent i is

$$\tilde{r}_i^t = \sum_{d=0}^{D_i} \Big(\sum_{j \in \mathcal{V} | d(i,j) = d} (\alpha)^d r_j^t \Big), \tag{4}$$

where D_i is the maximum distance of agents in the graph from agent i. When $\alpha > 0$, the adjusted reward include global information, it seems this is in contradiction to the local communication assumption. However, since reward is only used for offline training, global reward information is allowed. Once trained, the RL agents can control traffic signal without relying on global information.

Temporal Discount Factor and Return. The local return $\tilde{R}_i^t$ is defined as the cumulative adjusted reward $\tilde{R}_i^t := \sum_{\tau=t}^{T} \gamma^{\tau-t} \tilde{r}_i^\tau$, where γ is the temporal discount factor and T is the length of an episode. we can estimate the local return using value function,

$$\tilde{R}_i^t = \tilde{r}_i^t + \gamma V_{\phi_i^-}(\tilde{s}_{\mathcal{V}_i}^{t+1}, \pi_{\mathcal{N}_i}^t | \pi_{\theta_{-i}^-}), \tag{5}$$

where ϕ_i^- means parameters ϕ_i are frozen and θ_{-i}^- means the parameters of policy networks of all other agents are frozen.

Network architecture and training. As traffic flow generates spatial temporal data, we leverage a long-short term memory (LSTM) layer along with fully connected (FC) layers for policy network (actor) and value network (critic). Our multi-agent actor-critic training pipeline is similar to that in Chu et al. (2019). We provide neural architecture details, policy loss expression, value loss expression as well as a training pseudocode in the Appendix.

6 Experimentation

In this section, we demonstrate our RL framework using Simulation of Urban MObility (SUMO) (Lopez et al. 2018) SUMO is an open-source traffic simulator capable of simulating both microscopic and macroscopic traffic dynamics, suitable for capturing the EMV's impact on the regional traffic as well as monitoring the overall traffic flow. A pipeline is established between the proposed RL framework and SUMO, i.e., the agents collect observations from SUMO and preferred signal phases are fed back into SUMO.

6.1 Datasets and map descriptions

We conduct the following experiments based on both synthetic and real-world map.

Synthetic Grid$_{5 \times 5}$ We synthesize a 5×5 traffic grid, where intersections are connected with bi-directional links. Each link contains two lanes. We design 4 configurations, listed in Table 1. The origin (O) and destination (D) of the EMV are labelled in Fig. 4. The traffic for this map has a time span of 1200s. We dispatch the EMV at $t = 600s$ to ensure the roads are compacted when it starts travel.
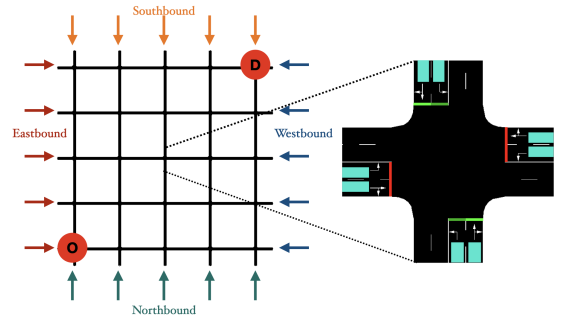

Figure 4: *Left*: the synthetic grid$_{5 \times 5}$. Origin and destination of the EMV are labeled. *Right*: an intersection illustration in SUMO, the teal area are inductive loop detected area.

| Config | Traffic Flow (veh/lane/hr) | | Origin | Destination |
	Non-peak	Peak		
1	200	240	N,S	E,W
2	160	320		
3	200	240	Randomly	
4	160	320	generated	

Table 1: Configuration for Synthetic Grid$_{5\times5}$. Peak flow is assigned from 400s to 800s and non-peak flow is assigned out of this period. For Config. 1 and 2, the vehicles enter the grid from North and South, and exit toward East and West.

Manhattan$_{16\times3}$ This is a 16×3 traffic network extracted from Hell's Kitchen area in Manhattan (Fig. 5) and customized for demonstrating EMV passage. In this traffic network, intersections are connected by 16 one-directional streets and 3 one-directional avenues. We assume each avenue contains four lanes and each street contains two lanes so that the right-of-way of EMVs and pre-emption can be demonstrated. The traffic flow for this map is generated from open-source NYC taxi data. Both the map and traffic flow data are publicly available.[1] The origin and destination of EMV are set to be far away as shown in Fig. 5

Figure 5: Manhattan map: a 16-by-3 traffic network in Hell's Kitchen area. Origin and destination for the EMV dispatching are labeled.

6.2   Baselines

Due to the lack of existing RL methods for efficient EMV passage, we select traditional methods and RL methods for each subproblem and combine them to set up baselines.

For traffic signal pre-emption, the most intuitive and widely used approach is the idea of *Green Wave* (Corman et al. 2009), which extends green light period for EMV passage at each intersection. **Walabi (W)** (Bieker-Walz and Behrisch 2019) is an effective rule-based method that implemented Green Wave for EMVs in SUMO. We integrate Walabi with combinations of routing and traffic signal control strategies introduced below as baselines.

---

[1]https://traffic-signal-control.github.io/

Routing baselines:

- **Static** routing is performed when EMV is dispatched and the route remains fixed as the EMV travels. We adopt A* search as the baseline since it is a powerful extension to the Dijkstra's shortest path algorithm and is used in many real-time applications because of its optimality. [2]
- **Dynamic** routing relies on real-time information of traffic conditions. To set up the baseline, we run A* every 50s as EMV travels. This is because running the full A* to update optimal route is not as efficient as our proposed dynamic Dijkstra's algorithm.

Traffic signal control baselines:

- **Fixed Time (FT)**: Cyclical fixed time traffic phases with random offset (Roess, Prassas, and McShane 2004) is a policy that split all phases with an predefined green ratio. It is the default strategy in real traffic signal control.
- **Max Pressure (MP)**: The state-of-the-art (SOTA) network-level signal control strategy based on pressure (Varaiya 2013). It aggressively selects the phase with maximum pressure to smooth congestion.
- **Coordinated Learner (CL)**: A Q-learning based coordinator which directly learns joint local value functions for adjacent intersections (Van der Pol and Oliehoek 2016).
- **PressLight (PL)**: A RL method aiming to optimize the pressure at each intersection(Wei et al. 2019a).

6.3   Results

We evaluate the performance of EMVLight and baseline models under two metrics: *EMV travel time*, which measures routing and pre-emption ability, and *average travel time*, which indicates the ability of traffic signal control for efficient vehicle passage. The performance of our EMVLight and the baselines in both the synthetic and the Manhattan map is shown in Table 2. The results of all methods are averaged over five independent runs and RL methods adopt different random seeds in each run. We observe that EMVLight outperforms all baseline models under both metrics.

In terms of EMV travel time T_{EMV}, the dynamic routing baselines perform better than static routing baselines. This is expected since dynamic routing considers the time-dependent nature of traffic conditions and updates the optimal route accordingly. EMVLight further reduces EMV travel time by 18% in average as compared to dynamic routing baselines. This advantage in performance can be attributed to the design of secondary pre-emption agents. This type of agents learn to "reserve a link" by choosing signal phases that help clear the vehicles in the link to encourage high speed EMV passage (Eqn. (3)).

As for average travel time T_{avg}, we first notice that the traditional pre-emption technique (W+Static+FT) indeed increases the average travel time by around 10% as compared to a traditional Fix Time strategy without EMV (denoted as "FT w/o EMV" in Table 2), thus decreasing the overall vehicle passage efficiency. Different traffic signal control strategies have a direct impact on overall efficiency. Fixed Time

---

[2]Our implementation of A* search employs a Manhattan distance as the heuristic function.

Method	EMV Travel Time T_{EMV} [s]					Average Travel Time T_{avg} [s]				
	Config 1	Config 2	Config 3	Config 4	Manhattan$_{16\times3}$	Config 1	Config 2	Config 3	Config 4	Manhattan$_{16\times3}$
FT w/o EMV	N/A	N/A	N/A	N/A	N/A	353.43	371.13	314.25	334.10	1649.64
W + Static + FT	257.20	272.00	259.20	243.80	487.20	372.19	389.13	342.49	355.05	1811.03
W + Static + MP	255.00	269.00	261.20	245.40	461.80	349.38	352.54	307.91	322.68	708.13
W + Static + CL	281.20	286.20	289.80	277.80	492.20	503.35	524.26	488.12	509.55	2013.54
W + Static + PL	276.00	282.20	271.40	275.00	476.00	358.18	369.45	332.98	338.95	1410.76
W + dynamic + FT	229.60	231.20	228.60	227.20	442.20	370.09	393.40	330.13	345.50	1699.30
W + dynamic + MP	226.20	234.60	224.20	217.60	438.80	345.45	348.43	313.26	325.72	721.32
W + dynamic + CL	273.40	269.60	281.00	270.80	450.20	514.29	536.78	502.12	542.63	1987.86
W + dynamic + PL	251.20	257.80	247.00	268.80	436.20	359.31	342.59	340.11	349.20	1412.12
EMVLight	**198.60**	**192.20**	**199.20**	**196.80**	**391.80**	**322.40**	**318.76**	**301.90**	**321.02**	**681.23**

Table 2: Performance comparison of different methods evaluated in the four configurations of the synthetic traffic grid as well as Manhattan Map. For both metrics, the lower value indicates better performance. The lowest values are highlighted in bold. The average travel time of Manhattan map (1649.64) is retrieved from data. Variations of these values are shown in the Appendix.

is designed to handle steady traffic flow. Max Pressure, as a SOTA traditional method, outperforms Fix Time and, surprisingly, outperforms both RL baselines in terms of overall efficiency. This shows that pressure is an effective indicator for reducing congestion and this is why we incorporate pressure in our reward design. Coordinate Learner performs the worst, probably because its reward is not based on pressure. PressLight doesn't beat Max Pressure because it has a reward design that focuses on smoothing vehicle densities along a major direction, e.g. an arterial. Grid networks with the presence of EMV make PressLight less effective. Our EMVLight improves its pressure-based reward design to encourage smoothing vehicle densities of all directions for each intersection. This enable us to achieve an advantage of 5% over our best baselines (Max Pressure).

Ablation study on pressure and agent types　We propose three types of agents and design their rewards (Eqn. (3)) based on our improved pressure definition and heuristics. In order to see how our improved pressure definition and proposed special agents influence the results, we (1) replace our pressure definition by that defined in PressLight, (2) replace secondary pre-emption agents with normal agents and (3) replace primary pre-emption agents with normal agents.

Table 3 shows the results of these ablations: (1) PressLight-style pressure (see Appendix) yields a slightly smaller EMV travel time but significantly increases the average travel time; (2) Without secondary pre-emption agents, EMV travel time increases by 45% since almost no "link reservation" happened; (3) Without primary pre-emption agents, EMV travel time increases significantly,

which shows the importance of pre-emption.

Ablation study on fingerprint　In multi-agent RL, fingerprint has been shown to stabilize training and enable faster convergence. In order to see how fingerprint affects training in EMVLight, we remove the fingerprint design, i.e., policy and value networks are changed from $\pi_{\theta_i}(a_i^t|s_{\mathcal{V}_i}^t, \pi_{\mathcal{N}_i}^{t-1})$ and $V_{\phi_i}(\tilde{s}_{\mathcal{V}_i}^t, \pi_{\mathcal{N}_i}^{t-1})$ to $\pi_{\theta_i}(a_i^t|s_{\mathcal{V}_i}^t)$ and $V_{\phi_i}(\tilde{s}_{\mathcal{V}_i}^t)$, respectively. Fig. 6 shows the influence of fingerprint on training. With fingerprint, the reward converges faster and suffers from less fluctuation, confirming the effectiveness of fingerprint.
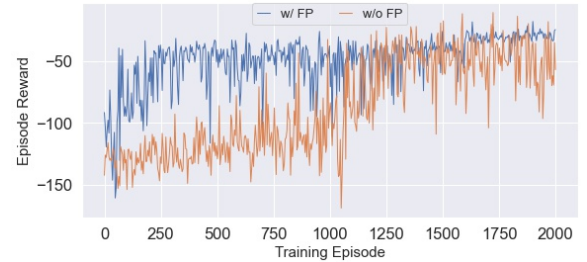

Figure 6: Reward convergence with and without fingerprint. Experiments are conducted on Config 1 synthetic grid$_{5\times5}$.

7　Conclusion

In this paper, we proposed a decentralized reinforcement learning framework, EMVLight, to facilitate the efficient passage of EMVs and reduce traffic congestion at the same time. Leveraging the multi-agent A2C framework, agents incorporate dynamic routing and cooperatively control traffic signals to reduce EMV travel time and average travel time of non-EMVs. Evaluated on both synthetic and real-world map, EMVLight significantly outperforms the existing methods. Future work will explore more realistic microscopic interaction between EMV and non-EMVs, efficient passage of multiple EMVs and closing the sim-to-real gap.

Ablations	(1)	(2)	(3)	EMVLight
T_{EMV} [s]	**197**	289	320	199
T_{avg} [s]	361.05	347.13	359.62	**322.40**

Table 3: Ablation study on pressure and agent types. Experiments are conducted on the Config 1 synthetic grid$_{5\times5}$.

References

Abdulhai, B.; Pringle, R.; and Karakoulas, G. J. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3): 278–285.

Agarwal, A.; and Paruchuri, P. 2016. V2V communication for analysis of lane level dynamics for better EV traversal. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, 368–375.

Analytics, N. 2021. End-To-End Response Times. https://www1.nyc.gov/site/fdny/about/resources/data-and-analytics/end-to-end-response-times.page.

Asaduzzaman, M.; and Vidyasankar, K. 2017. A Priority Algorithm to Control the Traffic Signal for Emergency Vehicles. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 1–7.

Aslani, M.; Mesgari, M. S.; and Wiering, M. 2017. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, 85: 732–752.

Bieker-Walz, L.; and Behrisch, M. 2019. Modelling green waves for emergency vehicles using connected traffic data. *EPiC Series in Computing*, 62: 1–11.

Buchenscheit, A.; Schaub, F.; Kargl, F.; and Weber, M. 2009. A VANET-based emergency vehicle warning system. In *2009 IEEE Vehicular Networking Conference (VNC)*, 1–8. 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 3414–3421.

Chu, T.; Wang, J.; Codecà, L.; and Li, Z. 2019. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*.

Corman, F.; D'Ariano, A.; Pacciarelli, D.; and Pranzo, M. 2009. Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, 17(6): 607–616.

El-Tantawy, S.; Abdulhai, B.; and Abdelgawad, H. 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3): 1140–1150.

Foerster, J.; Nardelli, N.; Farquhar, G.; Afouras, T.; Torr, P. H.; Kohli, P.; and Whiteson, S. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, 1146–1155. PMLR.

Gajda, J.; Sroka, R.; Stencel, M.; Wajda, A.; and Zeglen, T. 2001. A vehicle classification based on inductive loop detectors. In *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics*, volume 1, 460–464. IEEE.

Haghani, A.; Hu, H.; and Tian, Q. 2003. An optimization model for real-time emergency vehicle dispatching and routing. In *82nd annual meeting of the Transportation Research Board, Washington, DC*. Citeseer.

Humagain, S.; Sinha, R.; Lai, E.; and Ranjitkar, P. 2020. A systematic review of route optimisation and pre-emption methods for emergency vehicles. *Transport reviews*, 40(1): 35–53.

Ismath, I.; Samarasinghe, T.; Dias, D.; Wimalarathna, M.; Rasanga, W.; Jayaweera, N.; and Nugera, Y. 2019. Emergency Vehicle Traversal using DSRC/WAVE based Vehicular Communication. 1981–1986.

Jotshi, A.; Gong, Q.; and Batta, R. 2009. Dispatching and routing of emergency vehicles in disaster mitigation using data fusion. *Socio-Economic Planning Sciences*, 43(1): 1 – 24.

Koh, S.; Zhou, B.; Fang, H.; Yang, P.; Yang, Z.; Yang, Q.; Guan, L.; and Ji, Z. 2020. Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing*, 96: 106694.

Kwon, E.; Kim, S.; and Betts, R. 2003. Route-based dynamic preemption of traffic signals for emergency vehicle operations. In *Transportation Research Board 82nd Annual MeetingTransportation Research Board*.

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.

Lu, L.; and Wang, S. 2019. Literature Review of Analytical Models on Emergency Vehicle Service: Location, Dispatching, Routing and Preemption Control. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 3031–3036.

Mu, H.; Song, Y.; and Liu, L. 2018. Route-Based Signal Preemption Control of Emergency Vehicle. *Journal of Control Science and Engineering*, 2018: 1–11.

Musolino, G.; Polimeni, A.; Rindone, C.; and Vitetta, A. 2013. Travel time forecasting and dynamic routes design for emergency vehicles. *Procedia-Social and Behavioral Sciences*, 87: 193–202.

Noori, H.; Fu, L.; and Shiravi, S. 2016. A connected vehicle based traffic signal control strategy for emergency vehicle preemption. In *Transportation Research Board 95th Annual Meeting*, 16-6763.

Nordin, N. A. M.; Zaharudin, Z. A.; Maasar, M. A.; and Nordin, N. A. 2012. Finding shortest path of the ambulance routing: Interface of A-star algorithm using C programming. In *2012 IEEE Symposium on Humanities, Science and Engineering Research*, 1569–1573. IEEE.

Prashanth, L.; and Bhatnagar, S. 2010. Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 12(2): 412–421.

Roess, R. P.; Prassas, E. S.; and McShane, W. R. 2004. *Traffic engineering*. Pearson/Prentice Hall.

Su, H.; Shi, K.; Chow, J. Y. J.; and Jin, L. 2021. Dynamic Queue-Jump Lane for Emergency Vehicles under Partially Connected Settings: A Multi-Agent Deep Reinforcement Learning Approach. arXiv:2003.01025.

Van der Pol, E.; and Oliehoek, F. A. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

Varaiya, P. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36: 177–195.

Wang, J.; Ma, W.; and Yang, X. 2013. Development of degree-of-priority based control strategy for emergency vehicle preemption operation. *Discrete dynamics in nature and society*, 2013.

Wang, Y.; Wu, Z.; Yang, X.; and Huang, L. 2013. Design and implementation of an emergency vehicle signal preemption system based on cooperative vehicle-infrastructure technology. *Advances in Mechanical Engineering*, 5: 834976.

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.; 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. 2019c. A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117*.

Xu, B.; Wang, Y.; Wang, Z.; Jia, H.; and Lu, Z. 2021. Hierarchically and Cooperatively Learning Traffic Signal Control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 669–677.

Zang, X.; Yao, H.; Zheng, G.; Xu, N.; Xu, K.; and Li, Z. 2020. MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01): 1153–1160.

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.

Ziliaskopoulos, A. K.; and Mahmassani, H. S. 1993. Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. In *Transportation Research Record 1408*, 94–100.