Improving Lane Level Dynamics for EV Traversal: A Reinforcement Learning Approach

Akanksha Tyagi, Meghna Lowalekar and Praveen Paruchuri
International Institute of Information Technology Hyderabad (IIIT-H), Hyderabad, India

Keywords: Reinforcement Learning, Emergency Vehicles, Lane Level Dynamics.

Abstract: Emergency vehicles (EVs) perform a critical task of attending medical emergencies and delay in their operations can result in loss of lives to long term or permanent health implications. Therefore, it is very important to design strategies that can reduce the delay of EVs caused by slow moving traffic. Most of the existing work on this topic focuses on assignment and dispatch of EVs from different base stations to hospitals or finding the appropriate routes from dispatch location to hospital. However, these works ignore the effect of lane changes when EV is travelling on a stretch of a road. In this work, we focus on lane level dynamics for EV traversal and showcase that a pro-active picking of lanes can result in significant reductions in traversal time. In particular, we design a Reinforcement Learning (RL) model to compute the most optimal lane for an EV to travel at each timestep. We propose RLLS (Reinforcement Learning based Lane Search) algorithm for a general purposes EV traversal problem and perform a series of experiments using the well-known traffic simulator SUMO. Our experimentation demonstrates that our model outperforms the default SUMO algorithm and is also significantly better than the existing state-of-the-art heuristic approach BLS (Best Lane Search) strategy in normal traffic conditions. We also simulate worst case scenarios by introducing slowed down vehicles at regular time intervals into the traffic and observe that our model generalizes well to different traffic scenarios.

1 INTRODUCTION

Emergency vehicles (EV) are a class of vehicles which include ambulances, fire trucks, police cars etc. They are dispatched from their base stations to disaster site to respond to medical emergency, fire disaster among others. Any small delay in their operation can result in loss of life or long term damage or implications to health. Hence, even a small improvement in their traversal time can have significant impact. In the case of any medical emergency or disaster, a call is sent to a helpline such as 911 and upon receiving the call, the helpline connects with the appropriate base station to dispatch the EV. There are multiple critical decisions here which can have an impact on the overall response time. The first decision is, which base station should dispatch the EV vehicle (Ghosh and Varakantham, 2018; Haghani et al., 2003; Joe et al., 2022). The second decision is, what is the route that the EV will take to travel from the base station to emergency site and then from the emergency site to hospital(s) (Giri et al., 2022; Su et al., 2022). The third decision which is mostly overlooked is, when an EV enters a stretch of road in its route, which lane should the EV travel on (Agarwal and Paruchuri, 2016; Cao and Zhao, 2022). As demonstrated in earlier works (Agarwal and Paruchuri, 2016), the lane level dynamics can also play a crucial role in reducing the EV traversal time and hence the overall response time resulting in saving more lives and reduction of long term health implications. The existing work relies on heuristic approaches for lane level dynamics which can be myopic in nature and hence cannot capture the long term effect of decisions. Therefore, in this work we focus on improving the lane level dynamics for EV traversal using a reinforcement learning approach.

Our first contribution is to model the problem of lane level dynamics of EV using Markov Decision Process (MDP) (Puterman, 2014). The modelling of the problem as MDP allows us to use reinforcement learning algorithm to learn the best way to choose the appropriate lane for the EV to travel at each timestep. We propose **RLLS** - a **R**einforcement **L**earning based **L**ane **S**earch which uses the Advantage Actor-Critic method (Mnih et al., 2016) to learn the MDP policy. We compare RLLS with the baseline approaches using the SUMO (Simulation of Urban MObility)

Tyagi, A., Lowalekar, M. and Paruchuri, P.
Improving Lane Level Dynamics for EV Traversal: A Reinforcement Learning Approach.
DOI: 10.5220/0012637200003702

In *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2024)*, pages 134-143
ISBN: 978-989-758-703-0; ISSN: 2184-495X

(Lopez et al., 2018) traffic simulator, which simulates real traffic scenarios. SUMO offers a designated vehicle class known as *emergency*, facilitating the simulation of emergency vehicles and their unique privileges. Vehicles classified as emergency vehicles are automatically assigned default shapes and sizes suitable for rescue operations. They possess special previleges, such as the ability to overtake on the right side in all traffic scenarios. Additionally, these vehicles are permitted to traverse lanes specifically designated for "emergency" use, which may restrict normal passenger traffic. This functionality within SUMO enables an accurate modeling of emergency vehicle behaviors and traffic dynamics. We showcase that RLLS can reduce the EV travel time significantly as compared to this default (SUMO *emergency*) baseline.

Our second contribution is the introduction of experimental settings which can help with evaluating the worst case performance of algorithms. To evaluate the worst case performance of algorithms, we introduce slowing down vehicles at regular time intervals into the traffic. These slowing down vehicles block the traffic and introduce more congestion in the traffic network. This simulation setting helps with evaluation of the robustness of algorithms. Using a wide range of experiments, we showcase that our RLLS model trained using normal traffic scenarios can generalize well to these worst case settings and we do not need to train separate models for the different traffic scenarios.

In addition, we also evaluate the performance of our approaches on real word dataset by using real time speed data from New York City traffic (NYD, 2022) to calibrate traffic in a simulation. In this setting as well, RLLS model outperforms the existing approaches. In all the settings, for purposes of realistic modeling, we also allow the EV to communicate with other vehicles within a communication distance c_d. This is equivalent to communication done by an EV using a siren in real world scenarios.

2 RELATED WORK

The first thread of research focuses on finding strategies for the assignment of EV to incoming requests (emergency calls) (Ghosh and Varakantham, 2018; Haghani et al., 2003; Joe et al., 2022; Schmid, 2012). (Schmid, 2012), formulates the problem of finding the optimal dispatch strategy as an approximate dynamic programming problem and uses value function approximation strategies to find the assignment of EV to emergency calls at each timestep. (Ghosh and Varakantham, 2018) formulate the problem as an integer optimization problem and use Benders decomposition to find a solution to the integer optimization problem.

The second thread of research focuses on finding the best route for EV to travel from base station to the disaster location and from disaster location to hospital(s) (Giri et al., 2022; Su et al., 2022; Jotshi et al., 2009). A sub-thread of this line of work, is the coordination of traffic signal control to mitigate traffic congestion and as a result to allow EV to reach the destination quickly (Asaduzzaman and Vidyasankar, 2017; Chen et al., 2020; Chu et al., 2019; Van der Pol and Oliehoek, 2016).

The last thread of research which is most relevant to this paper is related to improving the lane level dynamics of Emergency vehicles (Agarwal and Paruchuri, 2016; Ismath et al., 2019; Cao and Zhao, 2022). In this thread of work, focus is on understanding and computing the value of each lane so to pick the best feasible lane to optimize on the travel time. (Zhang et al., 2022) focuses on safe lane-changing trajectories for autonomous driving in urban environments to enhance the efficiency as well as safety. (Maleki et al., 2023) studies a real-time optimal cooperative lane change strategy leveraging V2V communication, prioritizing safety and efficiency through constrained optimization. While these approaches primarily address normal traffic scenarios using heuristic methods, our strategy formulates the challenge of minimizing EV traversal time as an MDP and employs RL techniques. Additionally, we introduce scenarios with random slowing vehicles to add complexity and enhance the realism of the simulation. There are different assumptions and aspects of the problem e.g., nature of communication, range of communication, communication protocol to use, privileges of EV etc. that can affect the lane level decision making process. Please note that while traffic simulators will need to handle vehicle routing which involves changing of lanes, decisions to switch lanes are myopic in nature in general even though route planning tends to get optimized in a global sense.

3 BACKGROUND

The handling of lane level dynamics in EV traversal consists of picking the best lane for EV to travel while traversing a multi lane stretch of road. Similar to existing work, we assume the presence of a V2V single hop communication model where EV can obtain the position and speed of a vehicle in any lane up to a fixed communication distance, c_d via V2V communication. It can also send lane change requests to

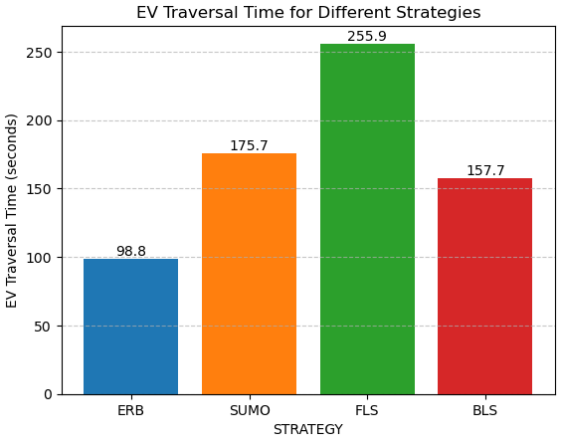

Figure 1: Comparison of EV Traversal Time for ERB, SUMO, FLS and BLS Strategies.

the vehicles ahead of EV in its current lane within this communication distance, c_d, to clear the traffic. Upon receiving the request from EV, vehicles attempt to change the lane and if there is no vehicle present in the destination lane, the lane change action would be successful.

(Agarwal and Paruchuri, 2016) introduced the following two strategies to find the best lane for EV.

- FLS (Fixed Lane Strategy). In this strategy, EV identifies the lane, that is fastest on an average, based on prior information and picks that lane as the fixed lane for its entire journey.
- BLS (Best Lane Strategy). In this strategy, at each timestep, EV identifies the best lane using the utility values for each lane and switches to the best identified lane. BLS computes utility of lane i using the following equation:

$$u_i = w_a * A_i + w_b * B_i + w_c * \mu_i$$

where A_i denotes the normalized speed of the slowest vehicle on lane i, B_i denotes the normalized average speeds of the vehicles on the lane i and μ_i denotes the normalized free space on the lane i. The normalized free space is an approximation and is computed using $\frac{n_i - c_i}{n_i}$, where n_i is the maximum number of vehicles that can be present on lane i within communication distance c_d and c_i is the number of vehicles present on the lane i within communication distance c_d.

In their experiments, the paper (Agarwal and Paruchuri, 2016) compares the above two strategies against the following baseline strategies:

- SUMO. SUMO (Simulation of Urban MObility) (Lopez et al., 2018) is a well known free and open source traffic simulation package which has been used for experimentation purposes in literature (and we also use in this paper). We use SUMO strategy to refer to the default lane change strategy implemented within the simulator.
- ERB (Empty Road Baseline). As the name suggests, it is the time taken by the EV when there are no vehicles on the road for the entire simulation period. This is the minimum possible time the EV can take and hence acts as a lower bound for the EV traversal time.

Figure 1 shows the comparison between EV traversal time of the above mentioned strategies which act as a baseline for our work. As shown in the figure, BLS outperforms FLS and SUMO, therefore, in this work we focus on providing a better strategy than BLS for lane level dynamics. There are two major limitations of BLS which we try to overcome in this work.

- BLS takes the decision of changing lane based on the computed utility values of each lane. These utility values are computed based on the current timestep parameters and as a result can not capture the long term effect of the decision.
- BLS uses static weights for each parameter which need to be pre-decided.

In this work, we propose reinforcement learning based algorithm which can capture the long term effect of the decisions and does not require any weights to be assigned to each parameter.

4 MODEL

As mentioned earlier, we encode the lane changing problem for EV using an RL model. We now describe each of the components of the underlying MDP tuple:

$$< S, A, P, R, N >$$

1. **State Space** (S)**.** The state space comprises below parameter values from each lane i:
   - x_i^{rel}. Relative distance along the x-axis, i.e., the difference between the x-coordinate of the position of vehicle immediately ahead of the EV on the lane i and the x-coordinate of EV.
   - y_i^{rel}. Relative distance along the y-axis, i.e., the difference between the y-coordinate of the position of vehicle immediately ahead of the EV on the lane i and the y-coordinate of EV.
   - v_i^{rel}. Relative speed, i.e., the difference between the current speed of the vehicle immediately preceding the EV on the lane i and the EV's current running speed.

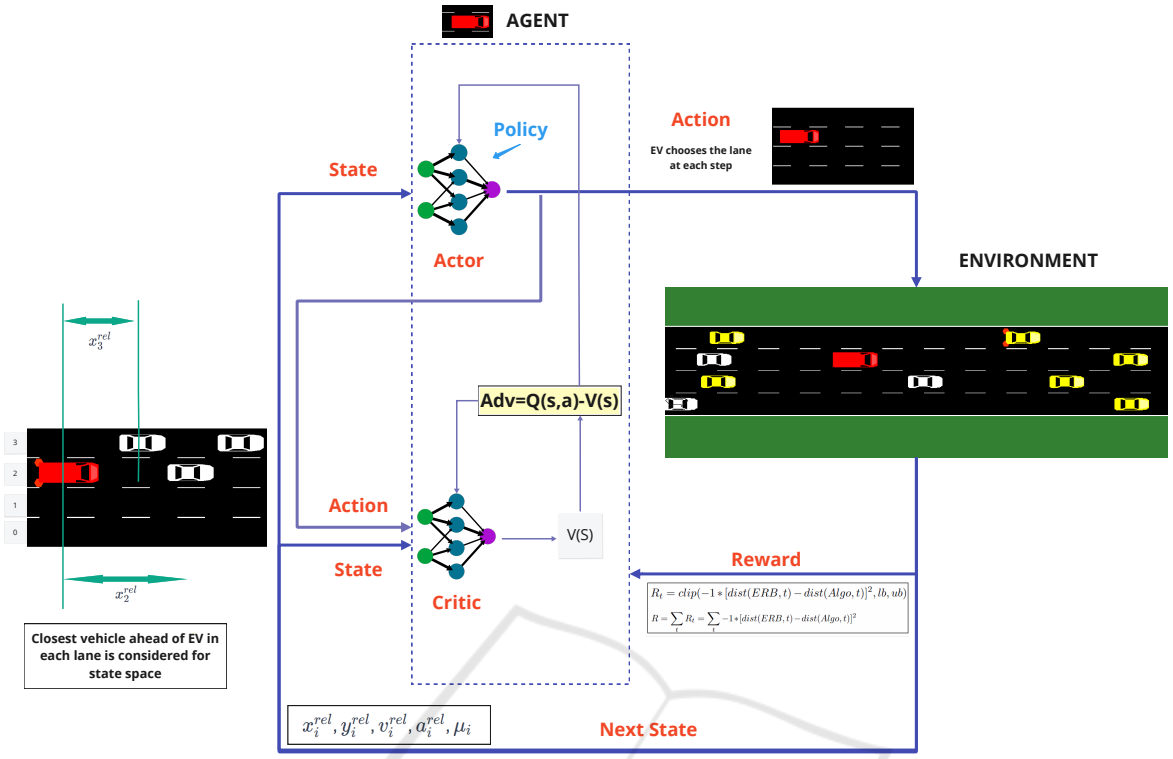

Figure 2: RLLS Algorithm.

- a_i^{rel}. Vehicle acceleration of the vehicle immediately ahead of the EV on the lane i.
- μ_i. Free space on the lane i within communication distance c_d. It is calculated similar to the BLS strategy as described in the Section 3.

As described above, the state space consists of parameters extracted from vehicles immediately preceding the EV, with one vehicle chosen from each lane. The rationale for selecting a preceding vehicle is to generalize the lane behavior by evaluating the parameters of vehicles ahead of the EV. For N lanes, the state space will consist of 5*N parameters.

2. **Action Space (A).** The action space encompasses the available lane options within the simulation environment,i.e. it is one of the N lanes.
3. **Transition Probability (P).** It is the probability of transitioning from state s_i to state s_j on taking a lane change action a. The transition model is not known and hence we use reinforcement learning to learn the policy for EV.
4. **Reward (R).** The intermediate reward at each timestep is given by:

$$R_t = -1 * [dist(ERB,t) - dist(Algo,t)]^2$$

where $dist(ERB,t)$ is the distance covered by Empty Road Baseline within a Reward Interval t and $dist(Algo,t)$ is the distance covered by the algorithm within the same Reward Interval t. The total episodic reward is the sum of individual rewards and can be represented using following equation:

$$R = \sum_t R_t = \sum_t -1 * [dist(ERB,t) - dist(Algo,t)]^2$$

To mitigate the effect of outliers on the reward computation, hyperparameters such as the lower bound (lb) and upper bound (ub) are employed to clip the intermediate reward. The selection of appropriate values for lb and ub is determined through experimentation, reflecting the adjustment of these parameters to optimize the reward calculation process.

$$R_t = clip(R_t, lb, ub) \quad (1)$$

5. **Total Lanes (N).** Represents the total number of lanes. At any point, each vehicle will be present in one of the N lanes.

5 RLLS: REINFORCEMENT LEARNING BASED LANE SEARCH

Our RLLS approach uses A2C (Advantage Actor-Critic) as an underlying algorithm to solve the model described in Section 4. A2C is a synchronous, deterministic variant of Asynchronous Advantage Actor Critic (A3C) (Mnih et al., 2016). It combines elements of both the actor-critic architecture and advantage estimation to improve training stability and efficiency. This hybrid architecture combines value-based and policy-based methods that help to stabilize the training by reducing the variance. an Actor responsible for controlling the agent's behavior (policy-based method) and a Critic assessing the quality of the actions taken (value-based method). In A2C, the actor component is assigned the role of selecting actions based on the current policy, while the critic part evaluates the value of state-action pairs and provides feedback to the actor. The different components of our RLLS approach are described in Figure 2.

As shown in the figure, the actor network maps each state to a corresponding action. We can update the Actor Network weights after every time step. The actor network outputs a probability distribution corresponding to each action. We sample actions from this probability distribution according to each action's probability. In our case, the action corresponds to the lane on which EV should travel. If the action to take lane1 has a value of .8 and the action to take lane2 has a value of .2, we will only choose the lane1 action 80% of the time and the lane2 action 20% of the time. Because the output is a probability distribution, please note that the agent action will not be deterministic but stochastic.

A2C algorithm uses the Advantage function which plays a crucial role in stabilizing the learning process by estimating how better it is to take an action at a state when compared to the average value of that state. It gauges the additional reward obtained beyond the expected value of that state. If the Advantage function A(s,a) is positive, indicating that our action performs better than the average value of that state, our gradient is encouraged in that direction. Conversely, if $A(s,a)$ is negative, suggesting that our action under performs compared to the state's average value, our gradient is prompted in the opposite direction. This mechanism helps guide the training process towards actions that yield superior outcomes relative to the state's average value.

The critic network maps each state s to its corresponding value $v(s)$. Unlike the Actor Network which outputs a probability distribution of actions, the Critic Network outputs the value of the input state as a floating point number. In the figure 2, the critic network evaluates the input state to have a value $v(s)$.

6 EXPERIMENTS

Goal of the experiments section is to compare the performance of our RLLS approach against the leading baseline strategies across different settings. The metric we use to compare the strategies is the EV traversal time, i.e., the total time taken by the EV to cover the stretch of the road. RLLS is evaluated against the following baseline strategies:

- ERB - Empty Road Baseline
- SUMO - Default strategy used within simulator.
- FLS - Fixed Lane Strategy
- BLS - Best Lane Strategy

We propose three variants of RLLS namely RLLS-A,B and C. Each variant differs from the other based on the type of reward function used in the simulation (episodic,intermediate or both) and the training scenario (trained using normal traffic or the worst case traffic simulation using slowed down vehicles). The difference between these variants is summarized in Table 1.

Table 1: RLLS Settings.

Setting	Training Scenario	Reward Type
RLLS-A	Normal Traffic	Intermediate
RLLS-B	Normal Traffic	Intermediate + Episodic
RLLS-C	3 vehicles slowed for every 10 seconds	Intermediate + Episodic

6.1 Setup

In this section, we present our experimental set-up using the SUMO software to evaluate the different EV strategies. The parameters used for simulation are listed in Table 2. The experimental setup tries to replicate the environment which is prevalent in major city area. The road segments in such cities have multiple lanes that are quite crowded and are in general 1 km to 10 km in length, at a stretch, before hitting an intersection. In our experiments, unless stated otherwise, we use a 2 km one way stretch of road with 4 lanes.

As discussed in Section 5, we employed the Advantage Actor-Critic (A2C) model for training. The

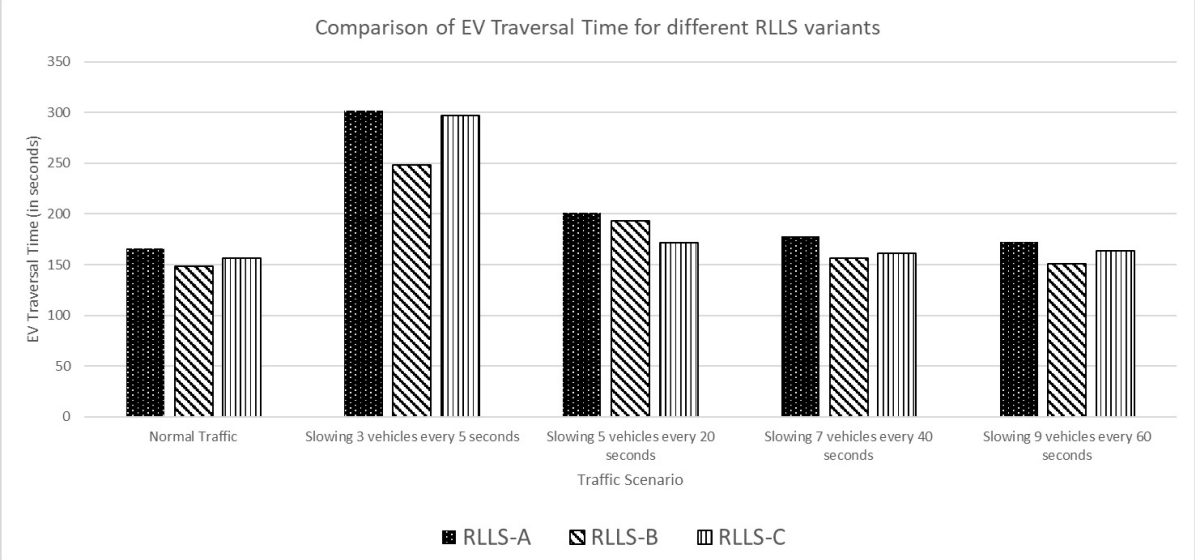

Figure 3: Comparison of EV traversal time for RLLS variants.

Table 2: Experimental Parameters.

Parameters	Description	Defaults
v_i	Vehicle maximum speed	33.33 m/s
vi_p	Preferred maximum speed	variable
a_i	Vehicle acceleration	0.8 m/s^2
d_i	Vehicle deceleration	4.5 m/s^2
c_j	Request delay	1 second
s_{dev}	Speed deviation	0, 0.2
σ	Driver imperfection	0, 0.5
w_a	Lowest speed weight-age	0.4
w_b	Average speed weight-age	0.4
w_c	Free space weight-age	0.2
δ	Min utility difference between lanes	0.20
t	Reward Interval	5 seconds
α	Re-computation Interval	10 seconds
c_d	Communication Distance	100 meters

intermediate reward is subjected to clipping as mentioned in Equation 1. The lower bound (lb)is used as -100 and upper bound (ub) is used as 0 to ensure stability during the reinforcement learning process. The training comprises of 700,000 steps. The training setup is designed in such a way that the EV explores through various traffic distributions. The same simulation environment is used across strategies for uniformity. At every simulation step, RLLS decides whether EV should change to any other lane or continue in the same lane. All the algorithms take lane change decision for EV at every re-computation interval. EV also communicates to other vehicles within communication distance to change the lane at every re-computation interval. The re-computation interval is fixed to 10 seconds in our experiments (refer Table 2).

6.1.1 Simulation Model Setup for Real World Dataset

We model the traffic patterns corresponding to cities with relatively faster moving traffic using the data available from the City of New York Department of Transportation(NYCDOT) (NYD, 2022). The NYCDOT data feed contains real-time traffic information from sensor feeds, mostly from major arterials and highways of New York City (NYC). This data feed is updated every minute for each road over a total of 137 roads. We developed our simulation model using the following steps:

1. We collected data for about a week (9670 minutes) from the real-time data feed.
2. For any road simulated, a segment of 2 kms is used, irrespective of the actual length of the road segment.
3. For (a few) roads with varying number of lanes (merges or splits), we use the maximum number of lanes for the entire length we simulate.
4. We included 131 roads having 4 lanes each (roads having other number of lanes were excluded).
5. The road speed data we have is converted into individual lane speed using the procedure described below that would capture the salient features of NYC traffic.
6. **Modeling Groups.** Firstly, we classify roads into groups based on number of lanes in the road.

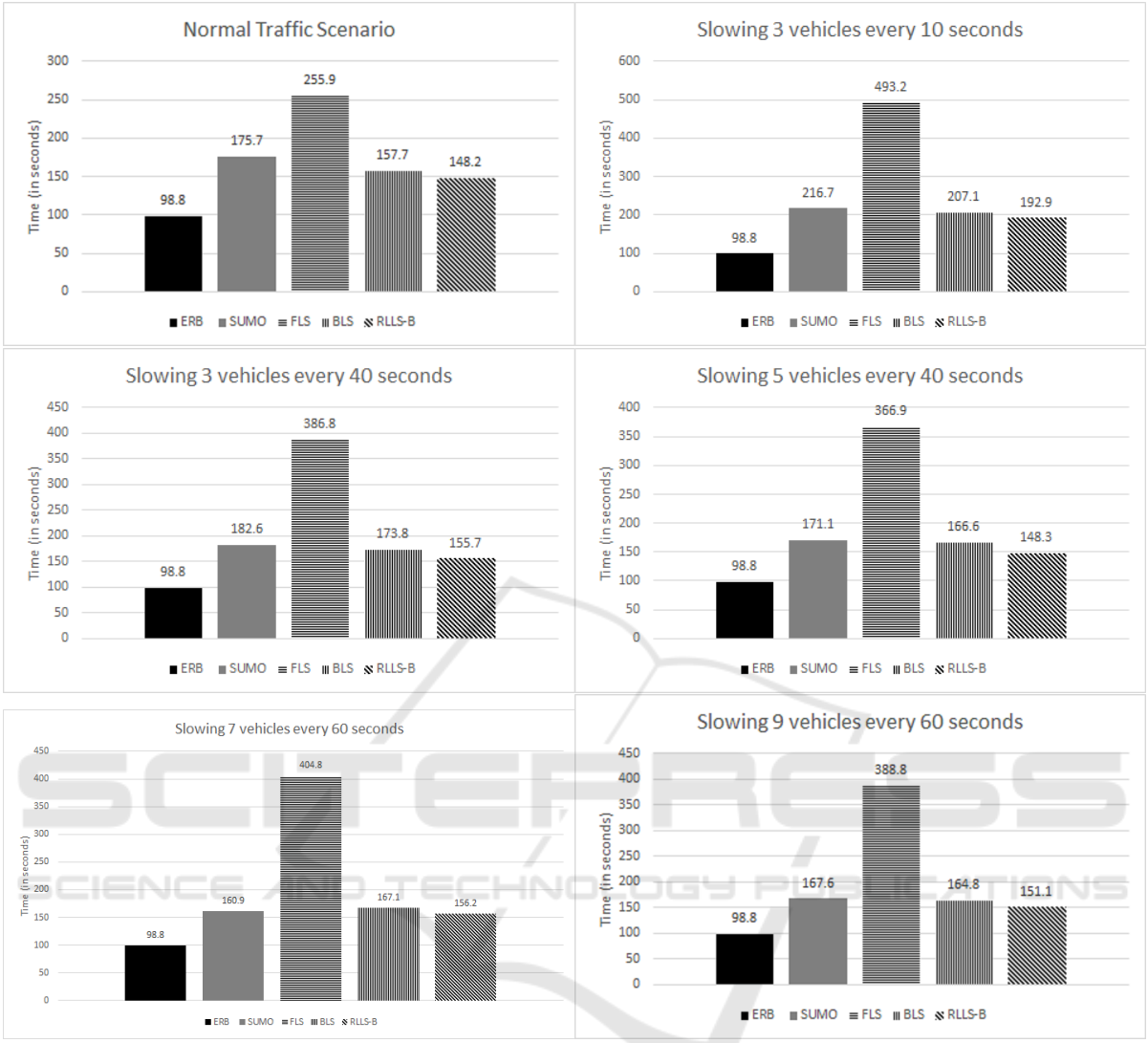

Figure 4: Comparison of EV traversal time of RLLS with baseline strategies in different traffic scenarios.

Hence all roads with 2 lanes are classified into one group g2, roads with 3 lanes into group g3 and so on. In this work, we consider only roads with 4 lanes so we have a single group g4.

7. **Modeling Buckets.** For the group g4, the average road speeds are classified into buckets using intervals of 5 m/s (18 km/h) each. Therefore we have buckets b_{1j} for 0-5 m/s, b_{2j} for 5-10 m/s till b_{14j} for 65-70 m/s.The average speed for each bucket, b_{avgj} , is taken as the mean of the bucket. For example, for the bucket 0-5 m/s, the average bucket speed, b_{avgj} is considered 2.5 m/s. In a similar way the average road speed for 5-10 m/s bucket is considered 7.5 m/s.
8. **Obtaining Weights for Buckets.** We then obtain the weights for each bucket in the following fashion: We have information for 131 roads having 9670 minutes of data collected per minute. Therefore, for each road, we have 9670 data points corresponding to average speed of the road at that minute. Each of these 9670 points are then classified into buckets depending on the speed the data point represents. The weight of a bucket is incremented by 1 for each data point that falls under this bucket. For example, if the average speed of a road (with j lanes) is 0-5 m/s for 500 minutes (out of the 9670 minutes) then we increment the weight w_{1j} by 500. This procedure is repeated for all the roads to obtain the total weight for each bucket.
9. **The SUMO Simulation.** Each b_{ij} has an average bucket speed b_{avgj} and a weight w_{ij}. b_{avgj}

is taken as the average road speed in simulation. To simulate lanes, b_{avgj} is converted into lane speeds: Each lane is set a maximum speed, picked using a uniform distribution between $b_{ij} \pm 40\%$. Hence, the mean of maximum speed across lanes is the average road speed on expectation. We then let the EV traverse on a 2 km stretch of road in the SUMO simulation and calculate it's run time. Each setting is run for 100 times, hence the same lane will have different maximum speeds across the 100 runs and we obtain 100 different EV run times.

10. **EV Run Time per Bucket.** We average the 100 different EV run times obtained to compute the EV run time EV_{rij} for a bucket i and group j. It represents the time an EV would take (on average) if the number of lanes is j and the average road speed corresponds to b_{avgj} . This is repeated for all buckets in every group.
11. **Computing Mean Run Time.** For each group j, we compute $gr_j = EV_{r1j} * w_{1j} + EV_{r2j} * w_{2j} + ... + EV_{r14j} * w_{14j}$. $\frac{gr_j}{(w_{1j}+...+w_{14j})}$ represents the average time an EV needs to travel a 2 km road with j lanes. As we consider only a single group with 4 lanes, the mean run time is obtained as $\frac{gr_4}{(w_{14}+...+w_{144})}$. This mean run time represents the average time an EV needs to cover a 2 km stretch of road with speeds corresponding to NYC roads and is used as run time for different strategies.

6.1.2 Traffic Scenarios

In our experiments we consider different traffic scenarios to evaluate the performance of our approach.

- Normal Traffic: This is the regular traffic scenarios where vehicles are travelling on the road. In the simulation, at each second one vehicle enters the road with a probability of 60%.
- Slowing down m vehicles every t seconds: These are the specialized traffic scenarios which we introduce to simulate traffic congestion on the road. In this scenario, every t seconds, we randomly slow down m vehicles on the road. This can be considered as worst case scenarios as it is difficult to move in a congested road where vehicles are moving at a very slow pace. Therefore, if an algorithm performs well in such scenarios it can be considered robust to random traffic congestion's which can occur in real-world.

6.2 Results

Our first experiment is to compare the results of different variants of our approach i.e., RLLS in different traffic scenarios to identify the best performing variant. Figure 3 shows the comparison of EV traversal times in different traffic conditions. As shown in the figure, RLLS-B outperforms the other variants and has lowest EV traversal time across different settings. We can make following observations from these results:

- Providing both intermediate and episodic reward helps in learning a better model than providing only the intermediate reward.
- We can train a single model for normal traffic scenarios and it generalizes well to different traffic conditions.

Next, we provide the comparison of RLLS-B model against baseline approaches on the synthetic dataset. Figure 4 show the comparison of the time taken by EV using RLLS-B and other approaches. As shown in the figure, RLLS-B consistently performs better than existing approaches for different scenarios. Here are the key observations:

- BLS outperforms SUMO in minimizing the EV traversal time through its utility function computation in normal traffic conditions. However, when the environment is made more complex, like slowing vehicles; the performance of BLS gets limited with only a little improvement over SUMO. This could be due to its deterministic nature. In contrast, by utilizing the adaptability of the RL algorithm, the RLLS-B algorithm can dynamically respond to these varying traffic scenarios using the state space information.
- In normal traffic scenarios, RLLS-B obtains 6% improvement over BLS but in specialized scenarios of slowing down vehicles this improvement goes up to 10.9% (Slowing down 5 vehicles every 40 seconds)

Finally, in Figure 5, we present the comparison of RLLS-B approach with the baseline strategies on the real-world dataset. The data is processed and the runtime of EV is computed as mentioned in the Section 6.1.1. In this case as well, RLLS-B outperforms BLS and other baseline strategies. In the normal traffic scenario, RLLS-B obtains 2.5% improvement over BLS which increases to 4% when we slow down vehicles.

It is important to note that even a very small improvement in the travel time of EV can help in saving human lives, so improvement of 4% is of significant importance.

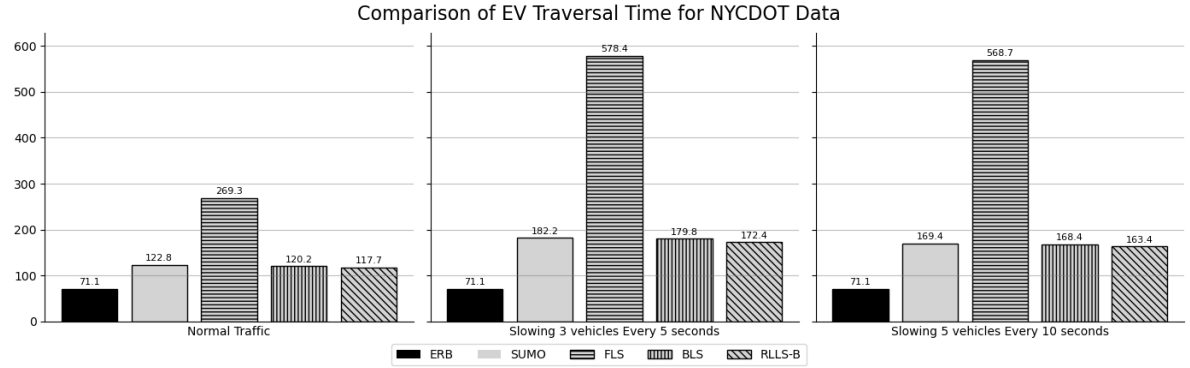

Figure 5: Comparison of EV Traversal Time for ERB, SUMO, FLS, BLS, RLLS-B strategies for Real-World dataset.

We would also like to highlight that the performance of RLLS-B can be improved by choosing a better state space. Initially we experimented with a bigger state space including the details of 20 closest vehicles around EV but this state space resulted in sub-optimal results and also increased the training time. The algorithm performance has been enhanced by adjusting this state space to include the positional difference and speed, acceleration differences of the immediate vehicle only in each of the four lanes; as well as the free space. This reduction in the dimension resulted in not only reducing the noise from the initial state space, but also focusing on more relevant factors, leading to improved performance of the RLLS-B lane changing strategy.

7 CONCLUSION

In this paper we presented a reinforcement learning strategy to improve lane level dynamics for emergency vehicles. Through detailed experiments using SUMO simulator, we showed that our approach outperforms the baseline strategies in different traffic scenarios. We considered a straight stretch of road in this work. In future, we would like to extend this work to combine the route planning and lane level dynamics which can further help in reducing the overall EV travel time.

REFERENCES

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)*, pages 368–375. IEEE.

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)*, pages 1–7. IEEE.

Cao, W. and Zhao, H. (2022). Lane change algorithm using rule-based control method based on look-ahead concept for the scenario when emergency vehicle approaching. *Artificial Life and Robotics*, 27(4):818–827.

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, pages 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*, 21(3):1086–1095.

Ghosh, S. and Varakantham, P. (2018). Dispatch guided allocation optimization for effective emergency response. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Giri, A. R., Chen, T., Rajendran, V. P., and Khamis, A. (2022). A metaheuristic approach to emergency vehicle dispatch and routing. In *2022 IEEE International Conference on Smart Mobility (SM)*, pages 27–31. 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.

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. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1981–1986. IEEE.

Joe, W., Lau, H. C., and Pan, J. (2022). Reinforcement learning approach to solve dynamic bi-objective police patrol dispatching and rescheduling problem. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 453–461.

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.

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)*, pages 2575–2582. IEEE.

Maleki, M., Taghavipour, A., and Azadi, S. (2023). A real-time optimal cooperative lane change strategy via v2v communication. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 237(13):3094–3107.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR.

NYD (2022). New york department of transportation - real-time traffic speed data. https://data.cityofnewyork.us/transportation/real-time-traffic-speed-data/xsat-x5sa.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European journal of operational research*, 219(3):611–621.

Su, H., Zhong, Y. D., Dey, B., and Chakraborty, A. (2022). Emvlight: A decentralized reinforcement learning framework for efficient passage of emergency vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4593–4601.

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)*, 8:21–38.

Zhang, S., Deng, G., Yang, E., and Ou, J. (2022). Optimal vehicle lane change trajectory planning in multi-vehicle traffic environments. *Applied Sciences*, 12(19):9662.