Smart Edge-Enabled Traffic Light Control: Improving Reward-Communication Trade-offs with Federated Reinforcement Learning

Nathaniel Hudson*, Pratham Oza†, Hana Khamfroush*, and Thidapat Chantem†

*Department of Computer Science, University of Kentucky
†Department of Electrical & Computer Engineering, Virginia Tech

***Abstract*—Traffic congestion is a costly phenomenon of everyday life. Reinforcement Learning (RL) is a promising solution due to its applicability to solving complex decision-making problems in highly dynamic environments. To train smart traffic lights using RL, large amounts of data is required. Recent RL-based approaches consider training to occur on some nearby server or a remote cloud server. However, this requires that traffic lights all communicate their raw data to some central location. For large road systems, communication cost can be impractical, particularly if traffic lights collect heavy data (e.g., video, LIDAR). As such, this work pushes training to the traffic lights directly to reduce communication cost. However, completely independent learning can reduce the performance of trained models. As such, this work considers the recent advent of Federated Reinforcement Learning (FedRL) for edge-enabled traffic lights so they can learn from each other's experience by periodically aggregating locally-learned policy network parameters rather than share raw data, hence keeping communication costs low. To do this, we propose the SEAL framework which uses an intersection-agnostic representation to support FedRL across traffic lights controlling heterogeneous intersection types. We then evaluate our FedRL approach against Centralized and Decentralized RL strategies. We compare the reward-communication trade-offs of these strategies. Our results show that FedRL is able to reduce the communication costs associated with Centralized training by 36.24%; while only seeing a 2.11% decrease in average reward (i.e., decreased traffic congestion).**

***Index Terms*—Smart Traffic, Traffic Light Control, Reinforcement Learning, Edge Computing, Federated Learning**

I. Introduction

According to recent transportation analytics data by INRIX, traffic congestion cost the United States economy $88 billion in 2019 alone [1]. Traffic congestion poses a constant threat to the economy and safety within an urban environment, which can be alleviated by using the compute and communication resources available in smart cities. Urban traffic networks exemplify a typical CPS where data, communication, and connected infrastructure can now jointly optimize traffic operations within a road network. Communication capabilities of the vehicles, traffic lights, and other road-side units (RSUs) powered by vehicle-to-everything (V2X) and vehicular ad-hoc networks (VANETs) provide opportunities for novel strategies to mitigate traffic congestion over large and complex urban road networks [2], [3]. Such strategies may require reliable computing resources for the strict needs of urban traffic networks. The recent advent of *Edge Computing* (EC) [4] pushes compute resources to the network edge via compute node servers, known as "edge servers", that are close to the smart city infrastructure. EC can be used to support more compute-intensive tasks for vehicular networks.

Many recent works trying to support smart decision making for traffic lights (commonly referred to as adaptive traffic signal control) consider *Reinforcement Learning* (RL)-based approaches [5], [6], [7], [8], [9], [10], [11]. RL is a popular technique for training sequential decision-making policies for problems that are highly dynamic and complex. Smart traffic light strategies that incorporate RL typically employ either a centralized [12], [13], [14] or decentralized [15], [10], [16] technique for training policies. In the centralized case, a policy is trained (typically on a roadside server) from the observations collected by detectors and other infrastructural components throughout the system. This central, roadside server then communicates actions to each of the traffic lights. Because the policy is learning over observations throughout the road network, these approaches perform well in terms of maximizing total reward. However, in practice, the amount of communication needed to send all observational data to the server can be costly. Decentralized approaches push the policy training to the traffic lights based on observations local to that traffic light, meaning less communication is needed since training is local to the traffic light itself. However, in decentralized approaches, the performance of the trained policies can be compromised because policies are learning in an isolated and independent manner. Therein lies a natural trade-off between policy performance w.r.t. maximizing reward and the communication cost associated with training. To the best of our knowledge, this trade-off has not been formally studied for smart traffic light control with RL.

To this end, we study the reward-communication trade-off for training smart traffic light control policies in an edge-enabled traffic system. We do this by proposing a *Federated Reinforcement Learning* (FedRL) technique inspired by the recent *Federated Learning* (FL) paradigm [17], [18]. Under our FedRL technique, we train traffic lights in a decentralized manner to reduce overall communication costs. Periodically, traffic lights will communicate their current policy network to a roadside edge server (hereafter referred to as "edge-RSU")

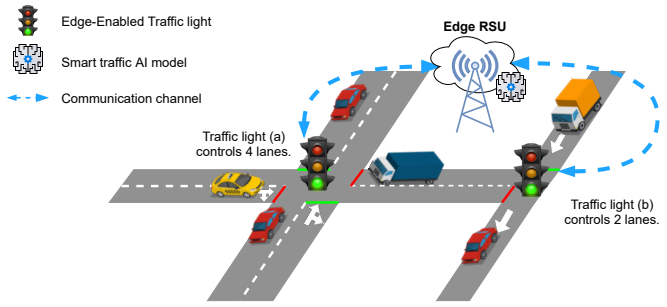

Fig. 1. Example of our traffic system where traffic lights communicate with an edge-enabled roadside unit (Edge-RSU).

which will then aggregate the policy network parameters using a weighted averaging method based on total reward. This newly-averaged policy network is then distributed to traffic lights for further training until the next aggregation phase. This aggregation will allow traffic lights to learn from each other without sharing raw observational data. For our FedRL to work, representation of current traffic conditions must be consistent across the road network, even in the face of heterogeneous intersection types. In this way, the representation needs to be *transferable* across road networks and intersections. For this, we design a novel, intersection-agnostic *Markov Decision Process* (MDP) [19] which we refer to as ***S**mart **E**dge-enabled tr**A**ffic **L**ights* (SEAL). The **central contributions** of this work can be summarized as follows:

- Design a novel, intersection-agnostic MDP for representing traffic conditions at traffic lights which we call SEAL. SEAL is designed to have a general representation of traffic conditions at intersections.
- Proposal of a *Federated Reinforcement Learning* (FedRL) approach for training RL decision-making policies for smart traffic light control.
- Improve reward-communication cost trade-off associated with solving SEAL using our proposed FedRL approach by reducing communication costs up to 36.24% on average while losing 2.11% on average when compared to Centralized training.

II. System Description

We now describe the system requirements for traffic infrastructure, data, and communication capabilities for our model. Fig. 1 shows a typical traffic environment where our model could be deployed. Our system considers a road network with one or more intersections (depending on the road topology), each equipped with a traffic light $k \in \mathcal{K}$ where $\mathcal{K}$ denotes the set of traffic lights in the entire system. Each traffic light $k \in \mathcal{K}$ controls the traffic flow entering the intersection through an incoming lane. A set of such *controlled lanes* is denoted by $\mathcal{L}_k$. A traffic controller, either located at each intersection or at a server calculates a "phase state" φ_k^t for the traffic lights at a given traffic light k at time-step t. The assigned phase state is such that a traffic light will be assigned a green, yellow or red "signal state", represented by `G`,`y`,`r`, respectively. Therefore, a phase state is a string representing the signal states of the traffic lights at all controlled lanes at an intersection. For example, the phase state for an intersection with eight controlled lanes would be `GyrrGyrr`. For a visual example of phase states, refer to Fig. 2. Note that the phase states are assigned such that the vehicles with conflicting traffic flows are not allowed to access the intersection at once. The length of the phase state is based on the number of incoming controlled lanes at a given intersection. Our model also expects that the vehicles obey the traffic regulations and do not violate the assigned phase permissions indicated by the traffic lights. Finally, in our system, we enforce the following timing restrictions for phase state changes for each traffic light $k \in \mathcal{K}$: *(i)* 4 seconds must pass since a previous phase state change before a traffic light can change its phase state; and *(ii)* a phase state change must occur within 120 seconds of the most recent phase state change. These timings are in accordance with the U.S. federal highway administration guidelines based on average traffic behavior [20] and can be changed as per traffic regulatory requirements. This is enforced for all training and evaluation.

Traffic infrastructure is equipped either with road-side sensors installed within every controlled lane to measure traffic parameters such as lane occupancy, average traffic flow speed, etc. (detailed in §III) or have connected vehicles to report such data to the traffic lights by utilizing the connected infrastructure. Traffic lights are equipped with edge compute resources to process the data and perform local learning. The edge resources also enable the connectivity among all traffic lights within the traffic network as well as the centralized cloud server to enable global optimization of the learning models. For simplicity, we assume the presence of a *single* deployed *edge-RSU* server in the region that maintains communication channels to all the traffic lights in a given region to support additional processes. Additionally, traffic lights are also equipped with compute resources as well as the edge-RSU server. As a simplifying assumption, we assume compute resources at both the traffic lights and the edge-RSU server are sufficient to train policies for smart traffic decisions. Succinctly, this work aims to improve the implicit reward-communication trade-off associated with distributed learning solutions to support smart traffic systems using FedRL.

III. Proposed SEAL Model Definition

Here, we define the ***S**mart **E**dge-enabled tr**A**ffic **L**ights* (SEAL) system. SEAL is modeled as a *Markov Decision Process* (MDP) [19] with the goal to minimize traffic congestion in road networks. SEAL's novelty is in defining a general state space representation that can describe current traffic conditions at a traffic light in an intersection-agnostic way. This is necessary to support policy aggregation in our FedRL approach (discussed later in §IV-C).

The work most similar to ours is that of Zhou et al.'s DRLE framework in [15]. This work is able to consider a distributed multi-agent RL approach to smart traffic light control with convergence guarantees. However, this does not consider the

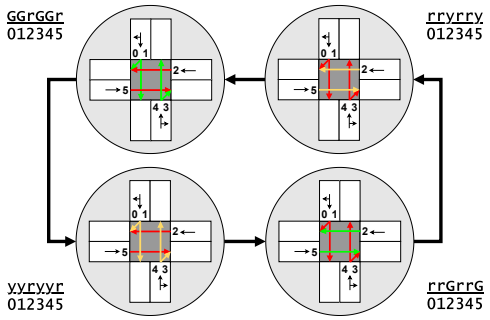

Fig. 2. Example traffic light action transition graph. Consider the given traffic light k's current phase state is `GGrGGR`. If the action $a_k^t = 1$ at time-step t, then the phase state for k will transition to `yyryyr` if sufficient time has elapsed since its last transition. Otherwise, its phase state remains the same, unless too much time has elapsed since its last change.

possibility of traffic lights themselves training their own policy networks. Instead, the DRLE framework sets traffic lights to communicate their local state observations to a roadside server to perform state aggregation to form a "global" state. This global statefulness allows for convergence guarantees, but may not be attractive for future solutions where traffic lights may collect large volumes of data (e.g., hyper-spectral images, videos, LIDAR imaging, etc.) to make decisions. Having large amounts of traffic lights stream these data in real-time to make timely decisions may not scale well. Thus, we consider SEAL. Future works investigate possible convergence bounds of SEAL is of interest but is beyond the scope of this work.

A. Action Space

In prior works investigating the use of RL for traffic light control, various kinds of actions have been considered. These include phase switch [15], [16], phase duration [9], and the phase state itself [7]. The phase state considers a discrete space of size n where n is the number of possible states for a traffic light. Since phase state depends on the number of controlled lanes and hence the traffic lights at an intersection, it is infeasible to aggregate knowledge among the intersections with varying topologies. For this work, we consider a simpler phase switch approach in which we consider each traffic light $k \in \mathcal{K}$ in time-step t to take an action $\mathbf{a}_k^t \in \{0, 1\}$ where $\mathbf{a}_k^t = 1$ signifies that traffic light k will attempt to change to the next phase state. Otherwise, $\mathbf{a}_k^t = 0$ signifies no phase state change will be attempted by traffic light k at time-step t.

Note, if a traffic light k attempts to change in some time-step t (i.e., $\mathbf{a}_k^t = 1$), a change can only occur if enough time has elapsed since its last change; further, a traffic light k will be forced to change its phase state regardless of its action if *too* much time has elapsed since its last change. This is due to the phase state timer (discussed in §II) to ensure policies mean mandatory regulations related to road safety [20]. Refer to Fig. 2 for an illustrated example of phase state logic and transitions made when $\mathbf{a}_k^t = 1$.

B. State Space

State space features consist of the following for a traffic light k in time-step t: *lane occupancy* (o_k^t), *halted lane occupancy* (h_k^t), *average speed* (ψ_k^t), and *phase state ratios* ($\varphi_k^t(\cdot)$) for all possible phase states (e.g., green, yellow, red).

1) Lane Occupancy: The average ratio of occupancy across all lanes controlled by a traffic light k in time-step t. Each traffic light k controls some set of lanes. Thus, we consider the occupancy of a lane l to be how much of a lane's length (in meters) is occupied by vehicles (as a ratio). However, we average this across all lanes controlled by traffic light k. The formal definition for lane occupancy is provided below:

$$o_k^t \triangleq \frac{\sum_{l \in \mathcal{L}_k} \sum_{v \in \mathcal{V}_l^t} len(v)}{\sum_{l \in \mathcal{L}_k} len(l)} \tag{1}$$

where $\mathcal{L}_k$ is the set of lanes controlled by traffic light k, $\mathcal{V}_l^t$ is the set of vehicles occupying lane l in time-step t, and $len(\cdot)$ is the length of the vehicle or lane (in meters).

2) Halted Lane Occupancy: SEAL's goal is to minimize congestion in road systems. Thus, we consider how much of a lane is occupied with halted vehicles. As such, we consider h_k^t to be the *halted lane occupancy* of traffic light k in time-step t when we consider a vehicle to be halted if its current speed is ≤ 0.1 meters/second. Thus, we define h_k^t below:

$$h_k^t \triangleq \frac{\sum_{l \in \mathcal{L}_k} \sum_{v \in \mathcal{H}_l^t} len(v)}{\sum_{l \in \mathcal{L}_k} len(l)} \tag{2}$$

where $\mathcal{H}_l^t$ is the set of halted vehicles occupying lane l in time-step t.

3) Average Speed: We also consider the average speed (ψ_k^t) among vehicles occupying lanes controlled by a traffic light k at time-step t as a feature. Similar to the other features, this one is also normalized as a ratio in the range $[0, 1]$. The formal definition is below:

$$\psi_k^t \triangleq \begin{cases} \frac{\sum_{l \in \mathcal{L}_k} \sum_{v \in \mathcal{V}_l^t} \min\left(spd_v^t, spd_l^{\max}\right)}{\sum_{l \in \mathcal{L}_k} \sum_{v \in \mathcal{V}_l^t} spd_l^{\max}} & \left|\bigcup_{l \in \mathcal{L}_k} \mathcal{V}_l^t\right| \geq 1 \\ 1.0 & \text{otherwise} \end{cases} \tag{3}$$

where spd_v^t is the moving speed of vehicle v in time-step t and $spd_l^{\max}$ is the speed limit (or maximum speed allowed) on lane l. The second case in Eq. (3) is for cases when there are no vehicles occupying lanes controlled by traffic light k.

4) Phase State Ratio: The current phase state of a traffic light has been used as feature in prior works (namely, [15]). This is possible because simple road networks are considered with homogeneous intersections where traffic lights have the same sets of possible phase states. To handle *heterogeneous phase state sets* across different intersection types, we instead represent the *ratio* of how each possible traffic light signal (e.g., green, yellow, red) makes up the entire phase state. Thus, we denote the ratio of a traffic light signal for a traffic light k in time-step t by $\varphi_k^k(\cdot) \in [0, 1]$. For instance, given a phase state at traffic light k in time-step t `GGrGGr`, we denote how much

of the phase state are red lights, r, by $\varphi_k^t(r) = 2/6$ (similarly for prioritized green lights, G, $\varphi_k^t(G) = 4/6$). Because we represent the *ratio* rather than assign an arbitrary discrete value to represent the entire phase state, the representation is general and can be used across different road networks with various intersections. It should be noted that $\sum_{p \in \mathcal{P}_k} \varphi_k^t(p) = 1 \ (\forall k, t)$ where $\mathcal{P}_k$ is the set of phase states for traffic light k.

C. Reward Function

The goal of SEAL is to reduce congestion in a given road network. With that in mind, we let reward r_k^t for a traffic light k at time-step t be a function of both *lane occupancy* (o_k^t) and *halted lane occupancy* (h_k^t). We define it below:

$$r_k^t \triangleq -(o_k^t + h_k^t)^2. \tag{4}$$

These state space features are summed to penalize traffic lights with more congestion. We let halted vehicles to incur more penalty since they contribute to both lane occupancy and halted lane occupancy. From there, we define the *total reward*, r^t, over the road whole network at time-step t as

$$r^t \triangleq \sum_{k \in \mathcal{K}} r_k^t. \tag{5}$$

D. Communication Model

As discussed in §II, we require robust communication capabilities between vehicles, traffic lights and edge-enabled RSUs to support smart traffic control. Depending on the training approach (detailed in §IV), a traffic control system must account for different communication channel utilization and their incurred costs. We therefore consider the following 6 different types of possible communications that can take place under the SEAL system: *(i)* policy network parameters from edge-RSU to traffic light, *(ii)* policy network parameters from traffic light to edge-RSU, *(iii)* action from edge-RSU to traffic light, *(iv)* observations from traffic light to edge-RSU, *(v)* vehicle-to-infrastructure (V2I) communication from vehicle to traffic light, and *(vi)* congestion ranks from edge-RSU to traffic light. We will evaluate the associated communication costs while training of our proposed model in §VI. To reiterate, we assume that edge-enabled traffic lights and the edge-RSU have sufficient compute capacity to performing policy training. Thus, we do not consider compute constraints and focus on communication cost instead.

IV. Training Algorithms

The goal of SEAL is to learn optimal traffic light control policies to minimize congestion for a given road network. To solve the SEAL model, we adopt model-free reinforcement learning techniques. More specifically, we will incorporate the recent *Proximal Policy Optimization* (PPO) [21] algorithm. Solutions to SEAL will aim to find a smart traffic light control policy, π, such that

$$Q^\pi(\mathbf{s}, \mathbf{a}) = (1-\gamma) \cdot \mathbb{E}\left[\sum_{t=1}^{\infty} (\gamma)^{t-1} \cdot r^t | \mathbf{s}^1 = \mathbf{s}, \mathbf{a}^1 = \mathbf{a}\right] \tag{6}$$

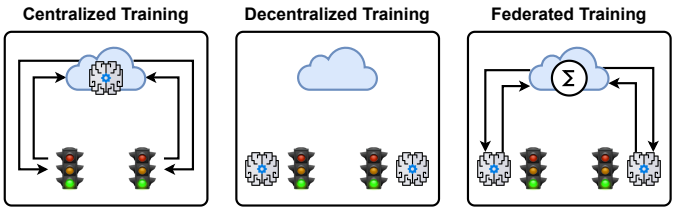

Fig. 3. Training approaches considered for solving SEAL.

is maximized where the policy is a decision-making function $\pi : S \mapsto A$ and γ is the discount factor. Eq. (6) is known as the Q-function. The optimal policy that maximizes the Q-function is defined as $\pi^* = \arg\max_\pi Q^\pi(\mathbf{s}, \pi(\mathbf{s})) \ \forall s$. For the sake of convenience, we denote $Q(\mathbf{s}, \mathbf{a}) = Q^{\pi^*}(\mathbf{s}, \mathbf{a}), \ \forall (\mathbf{s}, \mathbf{a})$ where $\mathbf{s}$ and $\mathbf{a}$ are a state and action, respectively. In RL, the Q-function is commonly approximated with a neural network using parameters ω. RL algorithms can be implemented in real-world systems in various ways. As such, we consider 3 different approaches for facilitating the PPO algorithm to solve SEAL: *(i) centralized training*, *(ii) decentralized training*, and *(iii) federated training*. A visual example of how these approaches compare can be found in Fig. 3. For a comprehensive overview on the theory of RL, please refer to [22].

A. Centralized Training

Under centralized training, there is a single policy network that is hosted on the nearby edge-RSU. At each time-step t, each traffic light $k \in \mathcal{K}$ submits their current state $\mathbf{s}_k^t$ to the edge-RSU which then returns an action $\mathbf{a}_k^t$ to traffic light k. Since a single policy network is learning across *all* observations in the system, it is expected to learn the optimal policy faster than other approaches. However, this is at the expense of incurring a large amount of overhead in terms of communication cost because of the traffic light having nonstop communication with the edge-RSU to take an action. For this work, we view this approach as an upper bound in terms of most quickly learning the optimal policy, π^*.

1) Centralized Training Communication Costs: Decision-making in a centralized manner requires traffic lights to always communicate to the edge-RSU leading to higher communications. Under Centralized training, the following communications take place at each time-step: actions from the edge-RSU to traffic lights, observations from traffic lights to edge-RSUs, and V2I communications from vehicles to traffic lights.

B. Decentralized Training

Unlike centralized training, decentralized training equips each traffic light $k \in \mathcal{K}$ with a policy network that aims to independently learn an optimal local policy for traffic light k, π_k^*, for optimizing reward using only observations local to that traffic light. In essence, if all traffic lights in the system are able to learn an optimal policy, then that can benefit the entire road network. Zhou et al. in [15] proved that a decentralized training approach using per-traffic light policies for smart traffic light control, can converge to a centralized

approach if given infinite time. In general, this approach can attain good performance if given enough time. While the decentralized approach is bested by the centralized approach in finding an optimal policy, since the latter is learning from global observations, the former approach is of interest as it requires less communication.

1) Decentralized Training Communication Costs: In the decentralized case, since the traffic lights never communicate to the edge-RSU for making decisions, little communication occurs. The only communication that takes place is V21 communication from vehicles to traffic lights.

C. Federated Training

With the expectation that decentralized training will not perform as well as centralized training due to policies learning over fewer observations, but will require less communication, we wish to achieve the best of both worlds. A novel contribution of this work is that we leverage the findings of the recent *federated learning* (FL) paradigm [17], [18] for distributed systems. Here we apply it to decentralized training to allow the traffic lights to learn from each other without needing to communicate raw data. We refer to this notion aptly as *Federated Reinforcement Learning* (FedRL) [23], [24]. FL has shown to reduce communication cost in the literature [25] while providing an immediate layer of privacy because no raw data are communicated. These are crucial advantages for smart traffic light control for future systems. For instance, consider a system that considers live video feed as a feature in the state space representation. Because identifying information (e.g., license plate numbers and faces of pedestrians) may be included, privacy is crucial. Additionally, such data may be very large and incur hefty data transmission costs. As such, we will focus on the benefit of federated training for smart traffic light control w.r.t. the trade-off between communication cost on the system and maximizing reward.

In FedRL, the traffic light agents training their own policy networks will periodically communicate the learned policy network parameters to the edge-RSU. The edge-RSU will then aggregate them using an averaging function. The newly aggregated policy network parameters are then communicated back to the traffic lights for further learning. Aggregation will occur after a number of time-steps occurs. We refer to this time period as a frame and denote it by F. We denote the policy network parameters learned by traffic light k at the end of frame F by ω_k^F.

In [18], the *federated averaging* (FedAvg) technique was proposed. This technique addresses the challenge of non-independent and identically distributed (iid) data distributions across different client devices. FedAvg uses a weighted average of the client's locally-updated model parameters based on the number of data items owned by that client. This weight combats non-iid data distributions common in distributed systems. For the sake of this work, we consider a simplifying assumption that traffic lights have identical data sampling rates — resulting in the same amount of observations. Below is the definition of the averaging we consider,

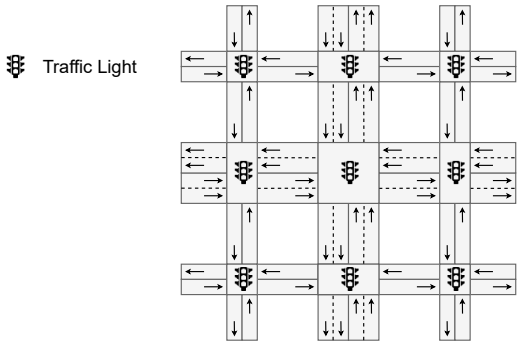

Fig. 4. Considered Grid-3 $\times$ 3 road network with heterogeneous intersection types. Note that the number of lanes increase as roads are more central.

$$\omega^{F+1} \triangleq \sum_{k \in \mathcal{K}} \frac{1}{|\mathcal{K}|} \omega_k^{F+1} \tag{7}$$

where the newly-aggregated, global parameters ω^{F+1} is the average of the parameters collected from all the traffic lights. These parameters are then sent back to the traffic lights at the start of frame $F + 1$ to resume training. Asynchronous aggregation techniques to address heterogeneous data sampling rates among traffic lights is beyond the scope of this work.

1) Federated Training Communication Costs: With federated training, communications that occur at each time-step are mostly identical to that for decentralized training (discussed in §IV-B1). The only difference is at the end of each frame (which occur less frequently than each time-step), 2 additional communications occur: policy network parameters from edge-RSU to traffic lights and policy network parameters from traffic lights to edge-RSU.

V. Experiment Design

We implement the SEAL framework using the Python programming language. Further, we implement the training approaches described in §IV using the SUMO traffic simulator [26] for the traffic simulation and Ray's RLlib [27] toolbox for the RL pipeline. Our software serves as the interface for these tools to fit our work's very specific needs. Thus, we only train the policy networks using PPO using simulations with these tools.

A. Considered Road Network Topologies

For training the policies using Ray's RLlib [27] and performing evaluation via simulation, we consider 3 road network topologies provided in Fig. 4: *(a)* Grid-3 $\times$ 3, *(b)* Grid-5 $\times$ 5, and *(c)* Grid-7 $\times$ 7. Roads on the border of the network have 1 lane going each direction, with the number of lanes going north/south and east/west increasing by 1 when approaching the center north/south and east/west roads. This is to introduce heterogeneous road network topologies. For an example, refer to Fig. 4. For simplicity, we do not allow vehicles to make turns to prevent the vehicles from getting stuck in the simulation. Note that this is a limitation of SUMO and SEAL's design is general enough to support

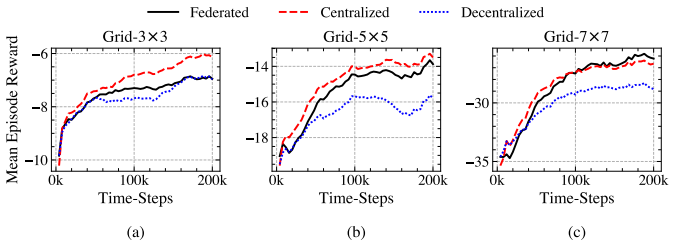

Fig. 5. Learning curves with each training approach on each road network.

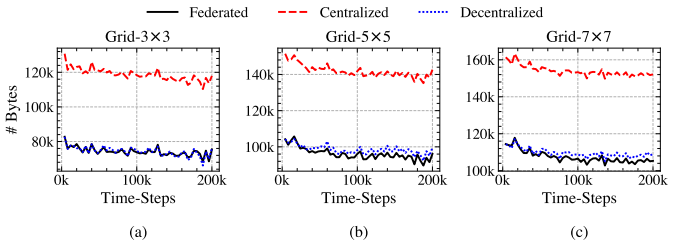

Fig. 6. Communication cost (i.e., data size in bytes) transmitted during training time under each training strategy for each road network.

turning vehicles. Each training approach (discussed in §IV) will learn policies over each road network topology. Vehicles routes for training and evaluation are randomly generated using the `randomTrips.py` module provided by SUMO with 360 *vehicles per lane per hour* (VPLPH) generated.

B. Training Parameters

We use *Proximal Policy Optimization* (PPO) [21] to train policies to solve SEAL. We use the following hyper-parameters. The learning rate is 5×10^{-5}. SGD minibatch size is 128. PPO *CLIP* parameter is set to 0.3. Target value for KL divergence is 0.3. Train batch size is 4000 time-steps. (Note policy network parameter aggregation, described in §IV-C, occurs every 4000 steps.) Roll-out fragment length (size of batches collected from each worker) is 200. We use *Generalized Advantage Estimator* (GAE) and the GAE parameter is set to 1.0. The VF clip parameter is set to 10.

VI. Results & Discussion

A. Reward Evaluation During Training

First, we compare the different training strategies discussed in §IV in terms of the reward achieved by the policy networks during training. In Fig. 5, we can see the learning curves of each training strategy when used on each of the 3 road network topologies described in §V-A. From these results, we find that, in general, make the following observations: *(i)* Centralized training generally achieves the greatest reward, *(ii)* Decentralized training generally achieves the worst reward and, *(iii)* Federated training achieves greater reward than Decentralized training (and often nearly match that of Centralized training). These observations are fairly intuitive. Since Centralized training trains a single policy network over all observations collected in the environment, it has more to learn from. Conversely, with Decentralized training, each traffic light learns independently using its own observations — meaning each traffic light's policy learns over fewer observations. Since Federated training expands on Decentralized training by allowing parameter aggregation among the policy networks learned by the traffic lights, the traffic lights are essentially able to learn from each other without explicitly sharing observations and other raw data. More specifically, we find that Decentralized training suffers from an 8.01% drop in reward compared to the Centralized training. Meanwhile, Federated training only suffers from an 2.11% drop in reward compared to Centralized training.

B. Communication Cost Evaluation During Training

Given Federated training is able to more closely approximate the reward achieved by Centralized training when compared to Decentralized training, we next compare the communication costs associated with each training strategy. We do this by tracking the number of communication that occurs (refer to §III-D) and the number of times each communication type occurs by the amount of bytes needed to transmit the data for that communication. In Fig. 6, we compare the size of the data needed to be communicated through the system during training using each of the training strategies under each of the road network topologies. There is a glaring difference in terms of communication efficiency between Centralized and Decentralized/Federated. Because Centralized training requires constant communication between the edge-RSU and the traffic lights in order to transmit observations, actions, and other data, it naturally incurs much greater communication cost. Meanwhile, Decentralized and Federated training greatly reduce this cost due to them keeping communication mostly between the vehicles and the traffic light. The only communication between the Edge-RSU and the traffic lights under Federated training is when policy network parameters are aggregated after each frame concludes. It is interesting to note that Federated is able to best Decentralized training in terms of communication cost in these results. This is due to the Federated training strategy producing better policy networks and removing vehicles from the system more efficiently than the Decentralized model — resulting in less vehicle-to-infrastructure communication. More numerically speaking, from our results Decentralized and Federated training are able to achieve a communication cost reduction of 34.65% and 36.24%, respectively, when compared to Centralized training.

C. Trained Policy Network Performance

Here, we are interested in *two* questions: (1) Can RL-based traffic lights trained with SEAL improve traffic conditions? (2) Can policy networks trained with SEAL perform well when used on road networks they were not trained on? To answer the first question, we compare our trained policy networks against a standard traffic light control baseline: a *pre-timed control* [20] where traffic lights cycle through phase states at fixed time intervals. We support this comparison using real-world traffic metrics to evaluate the experience of drivers in the system. Namely, we consider both "Travel Time" and "Waiting

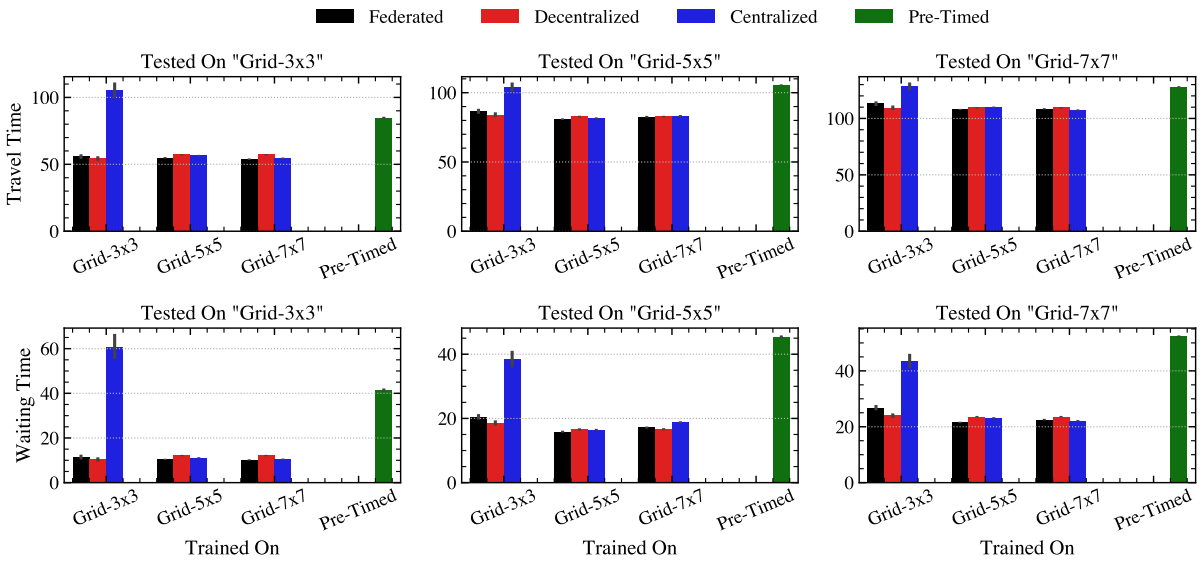

Fig. 7. Evaluation of trained policy networks on each road network using trip metrics, namely Travel Time (top row) and Waiting Time (bottom row). We compare the results to a Pre-Timed phase transition model as a baseline. Results confirm the RL-based solutions generally outperform the baseline.

Time". The former is the total amount of (simulation) time taken for vehicles to reach their destination; the latter is the amount of (simulation) time vehicles are waiting to move at a traffic light. The results of this evaluation are shown in Fig. 7. We see that in nearly all cases, the RL-based training strategies outperform that of the Timed-Phase baseline. The only outlier is the Centralized trainer when learning in the Grid-3 $\times$ 3 road network. As for the second question regarding possible transferability of the policy networks, we observe in Fig. 7 that the policy networks are generally able to perform comparable to one another (ignoring the Centralized trainer when trained on Grid-3 $\times$ 3). This generally holds true for policy networks being tested on the same road network they were trained on when compared to policy networks trained on other networks. These results serve to motivate the use of RL-based approaches for future smart traffic applications. We find that (on average) Centralized, Decentralized, and Federated reduce *travel time* compared to Pre-Timed by 11.63%, 18.16%, and 18.14%, respectively. Also, we find that (on average) Centralized, Decentralized, and Federated reduce *waiting time* compared to Pre-Timed by 42.81%, 58.92%, and 58.93%, respectively. The underperformance of Centralized here, compared to Decentralized and Federated, is likely due to the outlier scenarios when its trains on Grid-3 $\times$ 3. We attribute these anamolies to potential overfitting, though further experiments are needed.

VII. Related Works

Improving traffic light signal control in road networks has been a widely studied subject. Much work is being done to improve traffic conditions by developing *adaptive traffic signal control* (ATSC) where traffic lights adapt intelligently based on current traffic demands [28]. Many different techniques have been considered for realizing ATSC. Early works considered linear optimization frameworks [29].While linear programming is straightforward, it is not an appropriate match for ATSC because of the highly dynamic nature of real-world traffic systems — making accurate objective functions and constraints difficult to define. Genetic (or evolutionary) algorithms have also been considered in prior works [30]. In the early 2000s, initial works focusing on the application of *Reinforcement Learning* (RL) techniques for ATSC were published [6], [12]. While seminal, these initial works considered very simple road network scenarios. With advancements in both vehicular communication [2], [3] and RL algorithms [22], interest in RL for ATSC (or smart traffic) has been renewed. However, recent RL algorithms use more complex policy networks that require more compute resources to train.

Works considering RL for smart traffic light signal control have greatly increased over the years [7], [9], [10]. Because of the large number of entities in a traffic system (e.g., multiple traffic lights, multiple vehicles), *multi-agent* RL techniques have been applied to smart traffic light control [16], [11]. El-Tantawy et al. in [16] propose a multi-agent RL framework where agents can either be independent or collaborative in how they make decisions with other traffic light agents. Chu et al. in [8] propose a decentralized, multi-agent RL framework to provide robust learning with using a scalable framework. Chen et al. in [5] propose a decentralized actor-critic model and a difference reward method to accelerate the convergence of the trained policies for smart traffic light control. Mousavi et al. in [13] study both, policy- and value-based deep RL approaches for smart traffic light control. However, they only consider a single intersection, where the state space is a screenshot of the intersection provided by a traffic simulator. These works focus on improving training first and foremost, the communication cost for training these policies is ignored.

Edge Computing (EC) [4] is a recent enabling technology that pushes compute resources to the network edge. This has become an increasingly popular context for deploying AI (e.g., machine learning, deep learning, and RL) services to the network edge to provide low-latency intelligence. A significant recent work by Zhou et al. in [15] studied the applicability of edge computing for decentralized RL for smart traffic lights. A central contribution of this work is the theoretical guarantees that show that their proposed decentralized framework can provide a near-optimal guarantee on reduced traffic if given enough time. Different from this work, we design a framework that allows heterogeneous traffic lights to train policy networks in a federated manner to reduce communication costs.

The *central gap* in the literature related to RL for smart traffic light control is that the trade-off between reward and communication cost has been neglected. Additionally, recent advancements in the realm of *Federated Learning* (FL) or, more specifically, *Federated Reinforcement Learning* (FedRL) has yet to be applied to the smart traffic control problem.

VIII. Conclusions

In closing, this work to the best of our knowledge, is the first to approach smart traffic light control using *Federated Reinforcement Learning* (FedRL) in an edge computing-enabled system. We do this by proposing SEAL, which is an intersection-agnostic Markov Decision Problem for smart traffic light control to support aggregating learned policy network parameters across heterogeneous intersection types. This allows traffic lights to learn from each other's experiences without sharing raw experience data which reduces communication workloads (while providing some level of privacy). Our experiments demonstrate that SEAL combined with FedRL approach is able to closely match the rewards provided by a Centralized training approach (only a 2.11% decrease) when compared to the Decentralized approach that shows a 8.01% drop in reward. Further, our FedRL approach reduces the communication cost by 36.24% when compared to Centralized training. Hence, FedRL improves the implicit reward-communication trade-off for distributedly training smart traffic systems. In the future, we aim to extend our work to further analyze the theoretical bounds of SEAL and to study its effectiveness in small robotic testbed systems.

References

[1] INRIX, "Congestion costs each american nearly 100 hours, $1,400 a year," Mar 2020. [Press release]. Retrieved from https://inrix.com/press-releases/2019-traffic-scorecard-us/.

[2] M. S. Anwer and C. Guy, "A survey of VANET technologies," *Journal of Emerging Trends in Computing and Information Sciences*.

[3] S. K. Bhoi and P. M. Khilar, "Vehicular communication: a survey," *IET networks*, vol. 3, no. 3, pp. 204–217, 2014.

[4] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, 2017.

[5] Y. Chen, C. Li, W. Yue, H. Zhang, and G. Mao, "Engineering a large-scale traffic signal control: A multi-agent reinforcement learning approach," *IEEE INFOCOM 2021 Workshops*, pp. 1–6, 2021.

[6] M. A. Wiering, "Multi-agent reinforcement learning for traffic light control," in *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158, 2000.

[7] L. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412–421, 2010.

[8] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 1086–1095, 2020.

[9] M. Aslani, M. S. Mesgari, and M. Wiering, "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*, vol. 85, 2017.

[10] X. Wang, L. Ke, Z. Qiao, and X. Chai, "Large-scale traffic signal control using a novel multiagent reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 51, pp. 174–187, 2021.

[11] T. Tan, T. Chu, and J. Wang, "Multi-agent bootstrapped deep q-network for large-scale traffic signal control," *2020 IEEE CCTA*, 2020.

[12] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering-asce*, vol. 129, pp. 278–285, 2003.

[13] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 417–423, 2017.

[14] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *Iet Intelligent Transport Systems*, vol. 4, pp. 177–188, 2010.

[15] P. Zhou, X. Chen, Z. Liu, T. Braud, P. Hui, and J. Kangasharju, "DRLE: Decentralized reinforcement learning at the edge for traffic light control in the iov," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2262–2273, 2020.

[16] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "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*, vol. 14, pp. 1140–1150, 2013.

[17] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017.

[19] R. Bellman, "A Markovian decision process," *Journal of mathematics and mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[20] P. Koonce and L. Rodegerdts, "Traffic signal timing manual.," tech. rep., United States. Federal Highway Administration, 2008.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. abs/1707.06347, 2017.

[22] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[23] H. H. Zhuo, W. Feng, Q. Xu, Q. Yang, and Y. Lin, "Federated reinforcement learning," 2019.

[24] B. Liu, L. Wang, and M. Liu, "Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems," *IEEE Robotics and Automation Letters*, pp. 4555–4562, 2019.

[25] N. Hudson, M. J. Hossain, M. Hosseinzadeh, H. Khamfroush, M. Rahnamay-Naeini, and N. Ghani, "A framework for edge intelligent smart distribution grids via federated learning," in *2021 IEEE ICCCN*, pp. 1–9, 2021.

[26] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-Simulation of Urban MObility," *International journal on advances in systems and measurements*, vol. 5, no. 3&4, 2012.

[27] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2018.

[28] Z. Liu, "A survey of intelligence methods in urban traffic signal control," *IJCSNS International Journal of Computer Science and Network Security*, vol. 7, no. 7, 2007.

[29] M. Dotoli, M. P. Fanti, and C. Meloni, "A signal timing plan formulation for urban traffic control," *Control engineering practice*, 2006.

[30] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004.