2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Toward Efficient Traffic Signal Control: Smaller Network Can Do More

Shuya Li, Hao Mei, Jianwei Li, Hua Wei, Dongkuan Xu

Abstract— **Reinforcement learning (RL)-based traffic signal control (TSC) optimizes signal switches through RL agents, adapting to intersection updates. Yet, existing RL-based TSC methods often demand substantial storage and computation resources, impeding real-world implementation. This study introduces a two-stage approach to compress the network, maintaining performance. Firstly, we identify a compact network via a removal-verification strategy. Secondly, pruning yields an even sparser network. In addition, Multi-task RL is adopted for multi-intersection TSC, reducing costs, and boosting performance. Our extensive evaluation shows a compressed network at 1/1432nd of original parameters, with an 11.2% enhancement over the best baseline. This work presents an efficient RL-based TSC solution for real-world contexts, offering insights into challenges and opportunities in the field.**

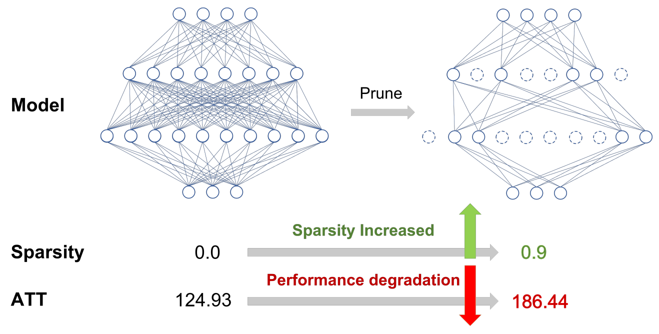


Fig. 1. The illustration of the compression of the TSC model is non-trivial, especially at high sparsity. From dense (left) to compressed (right), a significant sacrifice of ATT occurs when pruning 90% of the parameters.

I. Introduction

Reinforcement learning (RL) is a subfield of machine learning that has recently gained attention in the field of traffic signal control (TSC) which is an important aspect of urban traffic management that regulates the flow of vehicles to increase safety and efficiency at intersections [1]. The RL-based TSC is an emerging approach that has been shown to improve the performance of the TSC systems. Unlike fixed timing plans, RL-based systems dynamically adjust to real-time traffic changes, boosting flexibility and robustness. The RL agent, informed of intersection states, leverages past experiences for signal switch decisions. The RL-based TSC has been explored in a variety of scenarios, including isolated intersections [2], coordinated intersections [3] [4], with promising results. However, the existing RL-based TSC approaches tend to have high costs in terms of computation and storage [5]. This is because these methods rely on complex model structures and a large amount of computation to ensure their performance [3]. This hinders their deployment in real-world scenarios with limited computational resources. As the TSC is a field with a broad application background, it is important and desirable to strike a balance between model performance and cost.

Sparsification is an effective approach to reducing model complexity. However, improper sparsification can result in the pruning of important network components and subsequent degradation of model performance. For instance, Fig. 1 illustrates a popular pruning method, structured pruning, which reduces the number of neurons but also degrades performance in the example shown in Fig. 1. Additionally, when the number of parameters between the initial and target models differs significantly, directly sparsifying the initial network would cause a high sparsity level, leading to increased control and storage and performance degradation [6].

To tackle these challenges, we propose a two-stage strategy: initially identifying a smaller, promising dense network and subsequently pruning it for sparsity.

In addition, we apply a multi-task reinforcement learning approach [7] to address the TSC, enabling the network to control all road intersections simultaneously. This reduces storage and training expenses while enhancing performance through shared intersection knowledge.

The contributions of the paper are summarized as follows:

- We propose an RL-based approach to the TSC that balances model size and performance.
- We propose a practical two-stage approach to compress a neural network while maintaining model performance.
- We perform extensive experiments on multiple datasets. We compress the network up to only **1/1432nd** of the original number of model parameters, while simultaneously improving the results by **11.2%** compared to the best baseline.

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

Shuya Li is with the Department of Automation, Tsinghua University, China, lishuya21@mails.tsinghua.edu.cn. Hao Mei and Hua Wei are with the School of Computing and Augmented Intelligence, Arizona State University, USA, {hmei7, hua.wei}@asu.edu. Jianwei Li and Dongkuan Xu are with the Department of Computer Science, North Carolina State University, USA, {jli265, dxu27}@ncsu.edu

II. Related Work

Optimizing the TSC to alleviate traffic congestion has been a challenge in transportation for a long time. Various approaches have been studied in past decades to resolve this problem. Among these, pre-timed methods control traffic lights by employing cyclic control with pre-defined time splits, while adaptive rule-based methods handle it by using techniques such as MaxPressure [8] and SOTL [9]. However, these methods heavily rely on human experience and struggle to adapt to modern life's complex and dynamic traffic conditions. Recently, RL-based approaches [2], [3], [10] have been widely studied in this field, yielding promising outcomes. However, few works emphasize computational efficiency and storage consumption, limiting large-scale implementation. Our EfficientLight achieves competitive traffic efficiency performance while increasing computational efficiency by employing fewer parameters compared to existing works.

Pruning is one of the most popular methods to reduce the model size and computation cost of neuron networks [11] [12]. Specifically, pruning aims to eliminate redundant weights, neurons, and even layers in models. Many works focus on magnitude-based pruning [13] [14], namely to remove the component with the smallest magnitude. Here the magnitude refers to not only the weights but also the output sensitivity, gradients, or Hessian matrices of the training loss [15] [16], while the component refers to not only the weights but also neurons and layers. In our work, we apply iterative magnitude pruning to boost the efficiency of our RL models.

Multi-task learning, involving multiple optimization objectives simultaneously, is a critical challenge in machine learning. It's demonstrated value in robotics and reinforcement learning (RL) systems [17], [18]. However, limited research on the TSC exists in this field. Further research is necessary to reveal potential benefits and challenges, enhancing traffic control systems' efficiency and performance.

III. Preliminaries

Definition 1 (Traffic movement): A traffic movement is defined as the route traffic traveling through intersections. It could be uniquely characterized by a pair of incoming and outgoing lanes. The traffic movement (l, m) means the traffic enters the intersection through lane l and exits it through m.

Definition 2 (Signal phase): Traffic signal phases are a set of pre-defined combinations of traffic movements that do not conflict while vehicles pass through intersections.

Definition 3 (Pressure of each signal phase): A certain combination of traffic movements is enabled at each traffic signal phase. The pressure of the signal phase is the sum of the discrepancies in the number of vehicles between incoming and outgoing lanes in each traffic movement. In phase p, we denote $x(l, m)$ as the discrepancy between incoming lane l and outgoing lane m, the pressure of phase p is represented as $\sum_{(l,m)} x(l, m)$, where $(l, m) \in p$.

Definition 4 (Pressure of intersection): The pressure of an intersection is the difference between the sum of vehicles on incoming lanes and the sum of vehicles on outgoing lanes.

Problem 1 (Multi-task RL for traffic signal control): Following the definition of traditional multi-task reinforcement learning [7], [17], We define our traffic signal control problem in the multi-task RL framework as the following: Given a set of intersections $I = \{I_1, \ldots, I_N\}$ sampled over task distribution $p(\mathcal{T})$, the control process of each task $\mathcal{T}_i$ could be described as a Markov decision process $\mathcal{M}_i = \langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \gamma_i, H_i \rangle$, consisting a finite set of states $\mathcal{S}_i$, a finite set of actions $\mathcal{A}_i$, a reward function $\mathcal{R}_i$, a discounted factor γ_i and a finite episode length H_i.

In multi-task reinforcement learning, we use $\pi_\phi(a_t|s_t, z)$ to represent the policy parameterized by ϕ, where z is the task embedding. Our goal is to find the optimal policy to maximize the expected return across all tasks sampled from $p(\mathcal{T})$, which could be described as:

$$J_\pi(\phi) = \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})}[J_{\pi,\mathcal{T}}(\phi)] \quad (1)$$

and the $J_{\pi,\mathcal{T}}(\phi)$ is the expected return induced by π_ϕ from single task $\mathcal{T}$.

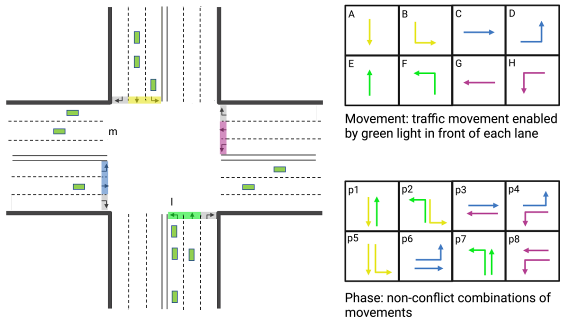

m
l
A
B
C
D
E
F
G
H
Movement: traffic movement enabled by green light in front of each lane
p1
p2
p3
p4
p5
p6
p7
p8
Phase: non-conflict combinations of movements

Fig. 2. An illustration of the ATSC environment.

IV. Method

This paper introduces a novel solution to the traffic signal control challenge using Yang's multi-task reinforcement learning method [7]. This method utilizes a single base policy network with multiple modules. A routing network that takes task identity and observed state as inputs outputs probabilities to weight the modules in the base policy network in a soft manner, allowing task-specific policies to learn and discover which modules to share across tasks. This method also balances learning across tasks by adjusting the training objectives for each task based on the policy's confidence.

A. Simplify the Neural Network

1) Structured pruning: Initially, the network's baseline performance is gauged across various datasets. During pruning, layers with identical functions are removed one by one, and performance is checked after each removal. If performance doesn't drop significantly, the removal proceeds; otherwise, the layer is returned. Subsequently, neuron count in layers is halved and performance is reassessed. If consistent, the change stays; if not, neurons are reinstated. The goal is a compact network with performance mirroring the original, especially for the TSC.

2) Unstructured pruning: Unstructured pruning reduces a neural network's complexity by selectively removing some non-zero weights, leading to a sparser model. This approach is applied to the previous compact, dense model to balance sparsity and performance. The pruning process starts by setting two parameters: target sparsity (s_{target}), indicating the final weight proportion to prune, and pruning frequency (d),

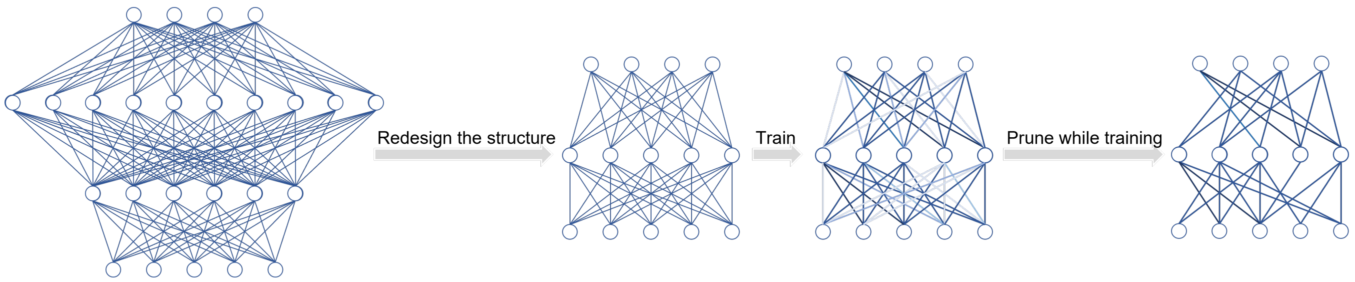


Fig. 3. Visualizing model reduction. Our two-stage method involves reducing layers and neurons to create a smaller network. During training, we remove weights with the smallest absolute values as the weights presented differences, resulting in a sparse network with fewer parameters.

showing how often to increase sparsity during training. We demonstrate the current model sparsity as follows:

$$s_{\text{current}} = \frac{s_{\text{target}}}{\lfloor t_{\text{total}}/d \rfloor} \cdot \lfloor t_{\text{current}}/d \rfloor \tag{2}$$

where t represents the optimization steps. During training, a binary mask with the same shape as the model's weight matrix indicates the sparsity pattern. The mask sets top k weight magnitudes to 1 (retained) and others to 0 (pruned). k is found using $k = s_{\text{current}} \cdot n$, with s_{current} as the current sparsity and n as total weights. The final sparse computation can be described as:

$$f_{\theta}(x) = f(\theta \cdot mask, x) \tag{3}$$

After reaching the desired sparsity, the sparse model is refined for better performance. This optimized network is ideal for environments with limited computational power.

3) Combination of structured and unstructured pruning: Our two-stage compression merges structured and unstructured pruning benefits. Compared to singular compression methods, our initial structured pruning allows the model to quickly reduce its size and inference time. The subsequent unstructured pruning further reduces the model size and increases its sparsity, ensuring that the model retains its performance even in an extremely sparse state. However, it's essential to acknowledge that the unstructured sparse mode in the second phase does not achieve acceleration under general hardware. Yet, with specially designed sparse hardware, like the Moffett S4 accelerator, this unstructured execution can still be accelerated [19].

B. Agent Design

- **State**. State is conveyed per lane, segmented into k equal parts. State variables encompass total active vehicles (speed $>$ 0.1 m/s) and total waiting vehicles (speed $\leq$ 0.1 m/s) in each segment. Real-world data accessibility underscores this design. State uniformity across intersections adds practicality to our approach.
- **Reward**. We utilize the intersection's negative pressure as the reward [20].
- **Action**. The agent's actions span all possible phases.

C. Process of Learning

Here, we employ a Soft Actor-Critic (SAC) to train our policy. SAC is an off-policy actor-critic deep reinforcement learning method. The actor seeks task success while introducing randomness, fostering exploration for strategy discovery and performance enhancement.

V. Experiments

This section presents experimental studies. Sections A to D provide related settings. We assess our approach through efficiency (Section E) and cost (Section F). Additionally, an ablation study (Section G) shows the performance of the unstructured pruning method employed.

A. Experiments Settings

We test on CityFlow, an open-source simulator for realistic traffic. It generates a road network as per settings. Traffic data guides vehicle movement. The simulator feeds state to signal control, executing actions from the method.

B. Dataset Description

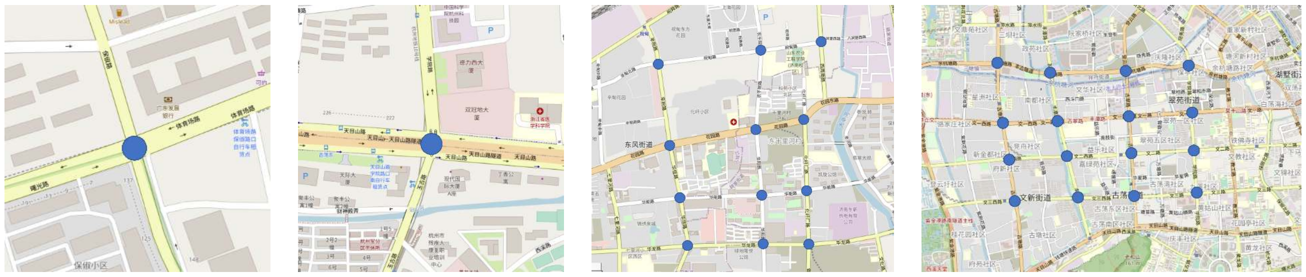
We use four real-world road networks which are abstracted from OpenStreetMap, with traffic flows extracted from camera data. Fig. 4 presents a top view of these road networks. The road networks in Hangzhou 1, Hangzhou 2, and Jinan contain ten traffic flows each. Nine of these flows were obtained by shifting the original vehicle appearance times recorded in real-world data with random noise [5]. We average results across ten flows per network.

C. Methods for Comparison

We compare our model with the following two categories of methods: rule-based solutions and RL-based ones.

Rule-based Methods. Fixedtime [21]: This employs preset cycle and phase times, common for stable traffic scenarios. **MaxPressure** [8]: It selects the highest pressure phase greedily. A state-of-the-art network-level signal control method in transportation. **SOTL** [9]: It will switch to the next phase once the number of waiting vehicles exceeds the predefined threshold. **EcoLight** [22]: EcoLight is a threshold-based model for scenarios with primitive conditions.

RL-based Methods. CoLight [3]: It is designed for the multi-intersection scenario. It uses graph attentional networks to facilitate communication. **FRAP** [2]: It is based on the intuitive principle of phase competition in traffic signal control. **MPLight** [4]: It tackles the problem of multi-intersection traffic signal control, especially for large-scale networks. **TinyLight** [5]: It is designed for devices with extremely limited resources.



(a) Hangzhou1 (b) Hangzhou2 (c) Jinan (d) Hangzhou3

Fig. 4. Road networks for real-world datasets. Blue dots are the traffic signals we control. (a) Baochu Rd. and Tiyuchang Rd., Hangzhou; (b) Tianmushan Rd. and Xueyuan Rd., Hangzhou; (c) Dongfeng Dist., Jinan; (d) Gudang Dist., Hangzhou.

TABLE I
PERFORMANCE ON TWO REAL-WORLD SINGLE INTERSECTION ROAD NETWORKS. EFFICIENTLIGHT AND EFFICIENTLIGHT (32X) ACHIEVE ADVANCED PERFORMANCES.

	Hangzhou-1 (1×1)		Hangzhou-2 (1×1)	
	Travel Time (sec./veh.)	Throughput (veh./min.)	Travel Time (sec./veh.)	Throughput (veh./min.)
EcoLight	196.82	29.73	100.45	28.67
FixedTime	282.68	27.59	418.05	23.54
MaxPressure	121.23	31.19	103.98	28.62
SOTL	250.58	28.69	143.35	28.32
FRAP	129.65	30.79	107.74	28.353
MPLight	128.61	30.81	82.48	28.73
TinyLight	102.87	31.36	81.79	28.74
EfficientLight	**87.90**	**31.51**	**71.45**	28.72
EfficientLight (32x)	**101.82**	**31.29**	**72.63**	28.72

TABLE II
PERFORMANCE ON TWO REAL-WORLD MULTIPLE INTERSECTIONS ROAD NETWORKS. EFFICIENTLIGHT-CENTRAL IS THE BEST.

	Jinan (3×4)		Hangzhou-3 (4×4)	
	Travel Time (sec./veh.)	Throughput (veh./min.)	Travel Time (sec./veh.)	Throughput (veh./min.))
EcoLight	384.43	92.08	371.05	45.12
FixedTime	457.21	86.27	689.02	39.75
MaxPressure	340.13	94.76	365.06	48.80
SOTL	424.67	90.02	354.13	48.60
CoLight	865.53	57.96	331.43	48.85
FRAP	327.90	95.10	330.15	45.83
MPLight	297.00	96.21	325.22	45.90
TinyLight	310.62	95.79	345.25	45.53
EfficientLight	**283.95**	**102.88**	334.58	**48.86**
EfficientLight-Central	**270.99**	**102.88**	**321.85**	**48.86**

D. Evaluation Metrics

- **Travel time**: A lower Average travel time (ATT) of all vehicles signifies greater model efficiency.
- **Throughput**: Throughput is the count of completed trips within a time frame. A higher throughput indicates enhanced model efficiency.

Both of these metrics are widely adopted in previous works on traffic signal control [1].

E. Performance Comparison

TABLE I and TABLE II report our experimental results w.r.t. ATT and throughput (sec. = second, veh. = vehicle, min. = minute). There are two versions, **EfficientLight** and **EfficientLight-Central**. They utilize a single policy network that achieves a sparse multiplier of 20 or 16. The key distinction: EfficientLight controls a single intersection's signals, while EfficientLight-Central manages signals across multiple intersections.

Please note that compared with most other RL-based works, our models are implemented with less parameter size.

1) Single Intersection Scenarios: The results of models on Hangzhou1 and Hangzhou2 are presented in TABLE I (Colight is designed for multiple intersections, so it is not covered in TABLE I). The results of the experiment indicate that EfficientLight achieves advanced performances on both metrics. The ATT of EfficientLight was found to be reduced by 14.55% and 12.64% on the two datasets, respectively, in comparison to the best baseline model, indicating a significant reduction. To further compress the model size, We increase the sparse multiplier to 32. Compared to all baselines, EfficientLight (32x) requires the lowest ATT on two datasets with much smaller costs.

2) Multiple Intersection Scenarios: The performance of all models on Jinan and Hangzhou3 is presented in TABLE II. Our results show that EfficientLight achieves superior performance on both metrics when compared to baselines. Also, EfficientLight-Central achieves the best results for both metrics on both datasets. The performance of EfficientLight-Central is superior to the performance of EfficientLight, which indicates that sharing learned knowledge from different intersections with traffic signal control can effectively improve the performance of the model.

F. Resource Consumption

The cost of different models is evaluated in terms of storage and computation. The measure of storage is the number of policy network parameters, while the measure of computation is in MACs (multiply-accumulate operations). However, distinct equation implementations can significantly alter additive computation count. To ensure a generalized comparison, only multiplicative computations are considered for MACs comparison. We create two types of scatter plots

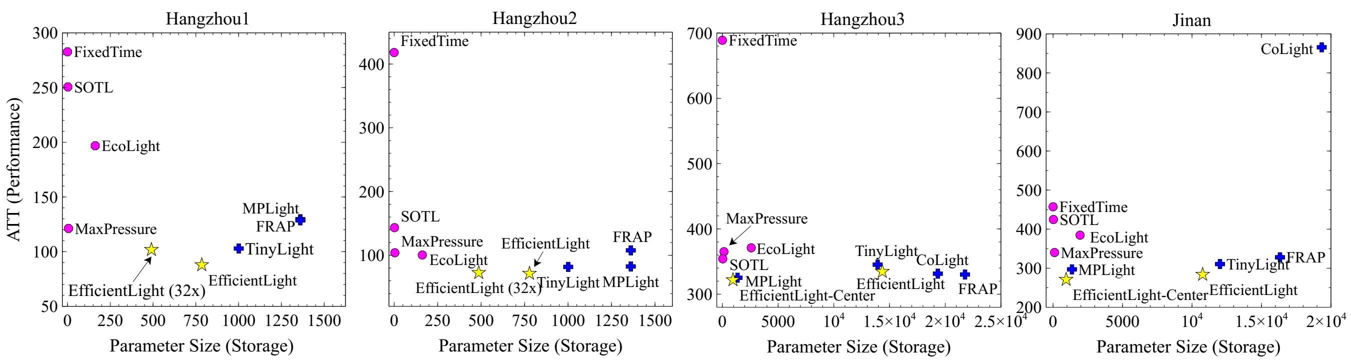


Fig. 5. Storage cost vs performance. Our methods achieve low ATT while utilizing a relatively small number of parameters.

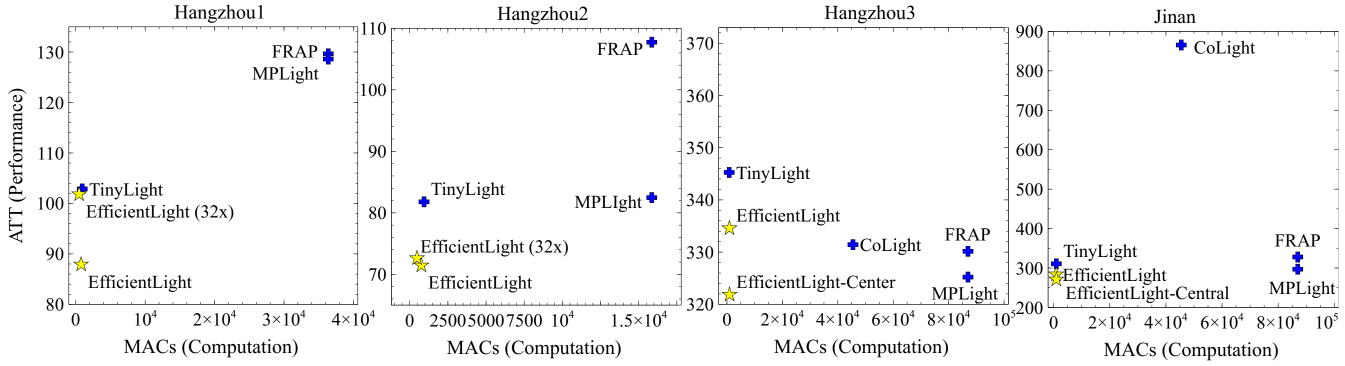


Fig. 6. Computational cost vs performance. Our methods achieve low ATT while requiring minimal computational resources.

that visualize the relationship between the cost and performance. In both plots, our method is represented by yellow five-pointed stars, other RL-based methods are represented by blue crosses and Rule-Based methods are represented by pink solid circles. Points closer to the origin represent lower cost and smaller ATT. Fig. 5 shows the relationship between storage cost and performance. On the multi-intersection datasets, the number of parameters for the method in which a neural network controls only one intersection is the sum of the number of all network parameters. Fig. 6 shows the relationship between computational cost and performance. Note that this second plot does not include the three Rule-Based methods, as they do not involve multiplication operations.

1) Storage Costs:

- **Single Intersection Scenarios**. In Fig. 5, it can be seen that on the single-intersection datasets of Hangzhou1 and Hangzhou2, our two models achieve the smallest model size and ATT among RL-based methods.
- **Multiple Intersection Scenarios**. The EfficientLight-Central method achieves the best tradeoff between model size and performance on the datasets Hangzhou3 and Jinan, as indicated by the smallest parameter size and ATT in Fig. 5. As for EfficientLight, on the Hangzhou3 dataset. EfficientLight's model size and ATT are similar to those of TinyLight. And EfficientLight's ATT is slightly higher (0.95% and 1.34% respectively) than CoLight and FRAP, while with significantly fewer model parameters (25.76% and 34.07% fewer respectively). It has the fewest parameters and lowest ATT on the Jinan dataset compared to the RL-based methods other than MPLight and lower ATT than MPLight.

2) Computational Costs: In Fig. 6, our two models are the closest to the origin for the single-junction datasets of Hangzhou1 and Hangzhou2, and the multi-junction dataset of Jinan. On the Hangzhou3 dataset, the ATT of EfficientLight is 0.95%, 1.34%, and 2.88% higher than CoLight, FRAP, and MPLight, respectively, but the computational effort is only 1.94%, 1.02%, and 1.02% of these three methods, respectively.

G. Ablation Study

In this ablation study, we extensively evaluate three variants of the EfficientLight method: EL-Stochastic, EL-FixedK, and EL-PostPrune. These variants were introduced to prove the efficiency of the unstructured pruning by making modifications to specific steps.

- EL-Stochastic randomly selects unchanged weights.
- EL-FixedK fixes k value.
- EL-PostPrune utilizes a one-time pruning approach, after the completion of the training process.

Fig. 7 presents the training process of three methods on the Hangang1 dataset. EfficientLight's training curve steadily rises with stability, in contrast to EL-Random's erratic behavior. This underscores EfficientLight's efficacy in preserving stability by eliminating weights with the smallest

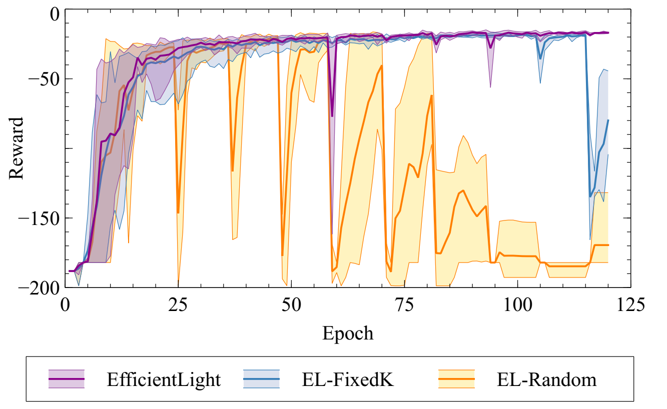


Fig. 7. Training curves of EfficientLight, EL-FixedK, and EL-Random. The Reward of EfficientLight exhibits a consistent and steady increase over time.

TABLE III
RESULTS OF SMALL-DENSE, EFFICIENTLIGHT AND EL-POSTPRUNE ON DIFFERENT DATA SETS. EL-POSTPRUNE'S PERFORMANCE DROPS CONSIDERABLY.

	Hangzhou-1 (1×1)		Hangzhou-2 (1×1)	
	ATT (sec./veh.)	Throughput (veh./min.)	ATT (sec./veh.)	Throughput (veh./min.)
Small-Dense	90.89	31.57	69.43	28.83
EL-PostPrune	500.25	24.45	922.02	16.43
EfficientLight	90.93	31.50	73.51	28.63

absolute values each epoch. EL-FixedK maintains stability initially, with fewer weight cuts than EfficientLight. Yet, as cuts rise, rewards plummet suddenly, recovering slowly. This emphasizes EfficientLight's gradual pruning for sustained training stability.

TABLE III reports the results of three methods, Small-Dense, EfficientLight, and EL-PostPrune, regarding one traffic flow in Hangzhou1 and Hangzhou2. Small-dense refers to the model obtained in the first stage. The results underscore the benefit of in-training weight pruning, allowing the model to learn using retained weights. Contrastingly, post-training pruning can reduce accuracy.

VI. CONCLUSION

In this paper, we introduce EfficientLight, an RL-based approach to traffic signal control that balances cost and performance. Our approach follows a two-stage process, involving model structure compression and weight reduction. We also introduce EfficientLight (32x), which sacrifices performance for a smaller model, and EfficientLight-Central, which controls multiple intersections with one neural network. In single intersection settings, EfficientLight and EfficientLight (32x) achieve minimal ATT with minimal storage and computational costs. In multi-intersection scenarios, EfficientLight-Central surpasses all baseline models in performance, with fewer parameters and relatively lower computation costs than other RL-based methods. These results illustrate our method's capacity to deliver strong performance at a reasonable cost.

REFERENCES

[1] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, pp. 12–18, 2021.

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

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

[4] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3414–3421.

[5] D. Xing, Q. Zheng, Q. Liu, and G. Pan, "Tinylight: Adaptive traffic signal control on devices with extremely limited resources," *arXiv preprint arXiv:2205.00427*, 2022.

[6] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 10 882–11 005, 2021.

[7] R. Yang, H. Xu, Y. Wu, and X. Wang, "Multi-task reinforcement learning with soft modularization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4767–4777, 2020.

[8] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," *Advances in dynamic network modeling in complex transportation systems*, pp. 27–66, 2013.

[9] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," *Advances in applied self-organizing systems*, pp. 45–55, 2013.

[10] S. G. Rizzo, G. Vantini, and S. Chawla, "Time critic policy gradient methods for traffic signal control in complex and congested scenarios," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1654–1664.

[11] A. Brutzkus, A. Globerson, E. Malach, and S. Shalev-Shwartz, "Sgd learns over-parameterized networks that provably generalize on linearly separable data," *arXiv preprint arXiv:1710.10174*, 2017.

[12] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[13] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," *ArXiv*, vol. abs/1902.09574, 2019.

[14] G. Thimm and E. Fiesler, "Evaluating pruning methods."

[15] S. P. Singh and D. Alistarh, "Woodfisher: Efficient second-order approximation for neural network compression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 098–18 109, 2020.

[16] X. Dong, S. Chen, and S. Pan, "Learning to prune deep neural networks via layer-wise optimal brain surgeon," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: a hierarchical bayesian approach," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 1015–1022.

[18] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing solving sparse reward tasks from scratch," in *International conference on machine learning*. PMLR, 2018, pp. 4344–4353.

[19] I. E.-H. Yen, Z. Xiao, and D. Xu, "S4: a high-sparsity, high-performance ai accelerator," *arXiv preprint arXiv:2207.08006*, 2022.

[20] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "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*, 2019, pp. 1290–1298.

[21] A. J. Miller, "Settings for fixed-cycle traffic signals," *Journal of the Operational Research Society*, vol. 14, no. 4, pp. 373–386, 1963.

[22] S. Chauhan, K. Bansal, and R. Sen, "Ecolight: Intersection control in developing regions under extreme budget and network constraints," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 027–13 037, 2020.