

LSTM vs. Transformer: A Comparative Study in Reinforcement Learning for Traffic Signal Control

Wentao Ding

*Harbin Institute of Technology, Harbin, China
dingwentao2004@126.com*

Abstract: Addressing urban traffic congestion, this paper presents a reinforcement learning (RL)-based traffic signal control (TSC) framework, comparing the effectiveness of Long Short-Term Memory (LSTM) and Transformer architectures for temporal feature extraction. A four-way intersection model is developed, where a Proximal Policy Optimization (PPO) framework is employed with vehicle occupancy ratio as the state, phase switching as the action, and average waiting time as the reward. The proposed models are evaluated through simulations on the SUMO platform. Experimental results indicate that in simple traffic scenarios, the LSTM-based model achieves a significantly lower average waiting time (6.97s) compared to the Transformer-based approach (10.54s) and traditional fixed-phase control (83.13s). These findings highlight the superiority of sequential models in environments dominated by short-term dependencies. Furthermore, this study provides novel insights into the applicability of attention mechanisms versus recurrent neural networks, offering valuable guidance for intelligent traffic systems.

Keywords: Traffic signal control, Reinforcement learning, Long Short-Term Memory, Transformer, PPO

1. Introduction

As urban populations continue to grow and the number of transportation vehicles increases, traffic congestion has become a significant barrier to urban development and daily commuting efficiency [1]. This growing challenge affects not only the productivity of cities but also contributes to environmental degradation and poor public health outcomes. Traditional methods of traffic management, particularly traffic signal control (TSC) at intersections, have been the primary solution for addressing congestion. However, these conventional methods, such as fixed-time and actuated signal control systems, have inherent limitations in their ability to respond dynamically to the ever-changing conditions of urban traffic. Fixed-time systems are often too rigid, operating based on predetermined schedules that may not align with actual traffic flow patterns, while actuated systems, although more responsive, still lack the ability to adapt to complex, multi-intersection networks in real-time [2-4].

Recent advancements in artificial intelligence (AI), specifically in the field of reinforcement learning (RL), have opened new avenues for improving traffic signal optimization. RL-based approaches enable systems to make decisions through continuous learning from the environment, which offers the potential to significantly enhance the flexibility and efficiency of TSC. By leveraging real-time traffic data, such as vehicle count, speed, and density, RL models can continuously adapt

and optimize the signal control strategy, overcoming the shortcomings of traditional approaches [5-7]. These capabilities are particularly important in large urban areas where traffic conditions are highly variable and unpredictable.

A reinforcement learning-based framework for TSC typically operates in the following manner [8-10]: over time sequences, data about vehicle flow and intersection conditions is gathered through various sensors (e.g., loop detectors, cameras, or radar). This data is then used to inform an RL agent, which makes decisions about traffic signal timing to optimize traffic flow and minimize congestion. The RL model learns through interactions with the environment, receiving feedback in the form of rewards or penalties based on the efficiency of the system's traffic management decisions. Over time, the RL model updates its policy to maximize the long-term reward, which in the context of TSC corresponds to minimizing travel time, reducing wait times, and improving overall system efficiency.

The application of RL to TSC presents several advantages, such as the ability to handle complex, high-dimensional data and the capacity to adapt to changing conditions. However, not all RL models are equally effective in capturing the temporal dependencies and long-range relationships that are crucial for optimizing traffic flow. Among the most widely used RL architectures are Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) that excels in preserving long-term temporal information. This feature makes LSTM networks particularly suitable for TSC, as they can capture the dynamics of traffic flow over extended periods. However, recent innovations in deep learning have introduced Transformer models, which have demonstrated superior performance in sequence-based tasks by leveraging a self-attention mechanism. Transformers are particularly adept at modeling both short-term and long-term dependencies in data, making them an attractive option for TSC applications that require rapid and efficient decision-making [7,11].

Our work involves: constructing RL models for TSC by implementing two distinct feature extraction architectures, include LSTM and Transformer, and evaluating these approaches against traditional methods through software simulations to compare their effectiveness.

2. Preliminaries

We consider a four-way intersection model: two perpendicular roads intersecting at a junction. Vehicles entering from different directions can proceed straight, turn left, or turn right. In countries with right-hand traffic rules, right turns are typically permitted without signal control and thus excluded from our model.

Movements: Each movement represents a possible vehicle trajectory. There are four entry/exit directions: East (E), South (S), West (W), and North (N). For example, a left turn from West to North is denoted as (W, N) . Under our assumptions, there are 8 possible movements: 4 straight movements (E, W) , (N, S) , (W, E) , (S, N) and 4 left-turn movements (E, S) , (N, E) , (W, N) , (S, W) .

Phases: Traffic signals operate by periodically permitting specific movement combinations. Each permitted combination constitutes a phase, with signal operation consisting of phase transitions. Our model uses four phases:

$$P_1 = \{(W, E), (E, W)\}, P_2 = \{(W, N), (E, S)\}, P_3 = \{(N, S), (S, N)\}, P_4 = \{(N, E), (S, W)\} \quad (1)$$

Figure 1 illustrates the vehicle flows corresponding to each phase.

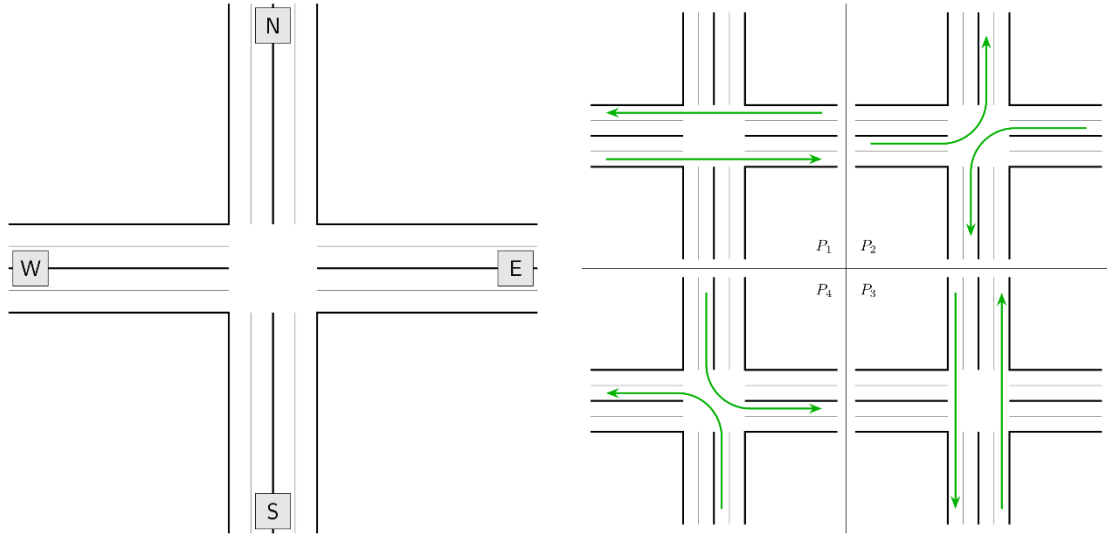


Figure 1: Four-way intersection configuration

3. Model design

Figure 2 presents the overall system architecture. Vehicle detection sensors continuously collect data from the intersection. Sequential observations are stacked to form states, which are then processed by feature extraction modules before being input into value and policy networks. The networks output control decisions that adjust traffic signals, thereby regulating vehicle flows.

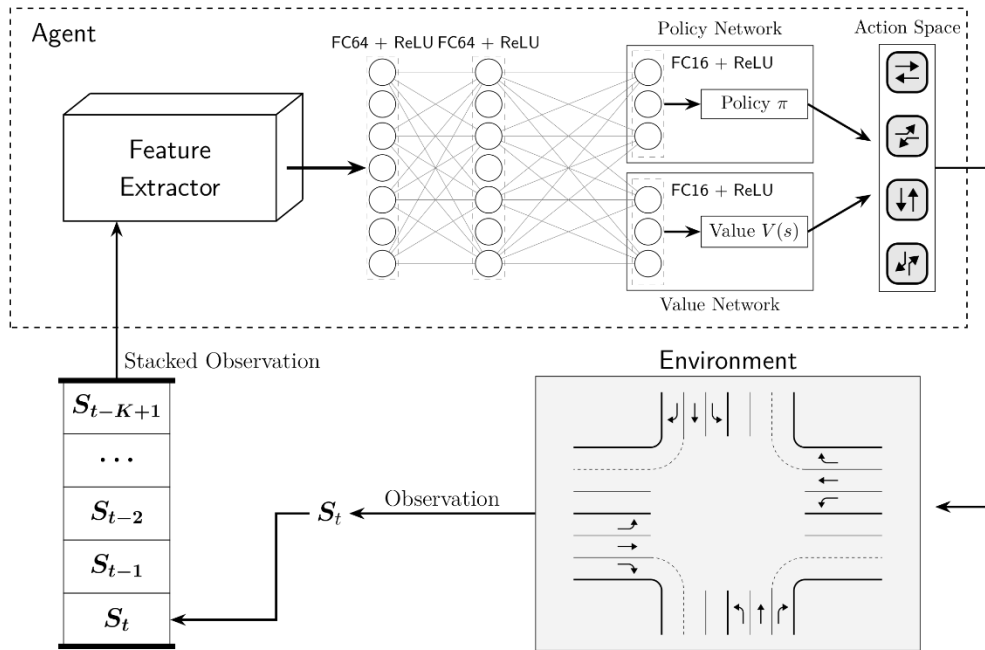


Figure 2: System architecture

3.1. RL design

State: To enable intelligent signal control, we monitor vehicle occupancy near intersections. Sensors measure the length of road segments occupied by vehicles within detection zones (Figure 3). Intersection occupancy is defined as the ratio of occupied area to total detection area.

Let $\mathbf{s}_t = [s_1(t), s_2(t), \dots, s_{12}(t)]^T$ denote occupancy rates at time t , where $[\cdot]^T$ denotes transpose. We stack the last K observations to form the input state $\mathbf{S}_t \in \mathbb{R}^{K \times 12}$:

$$\mathbf{S}_t = [\mathbf{s}_{t-K+1}, \mathbf{s}_{t-K+2}, \dots, \mathbf{s}_t] \quad (2)$$

Action: At each timestep, the agent selects a phase $A_t = a$ where $a \in \{1, 2, 3, 4\}$ (4 possible phases).

Reward: The reward is defined as the negative of the average waiting time across all vehicles:

$$R = -\frac{1}{N} \sum_{i=1}^N w_i \quad (3)$$

where w_i is the waiting time of vehicle i and N is the total number of vehicles.

We employ Proximal Policy Optimization (PPO) [12] for training, which optimizes policies to maximize cumulative rewards while limiting update magnitudes for training stability.

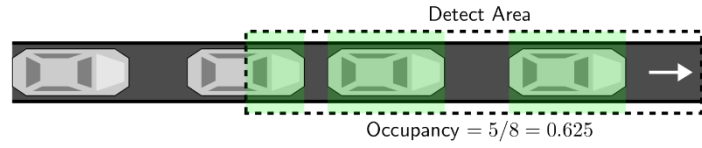


Figure 3: Vehicle detection zones

3.2. Feature extraction

We compare two temporal feature extractors:

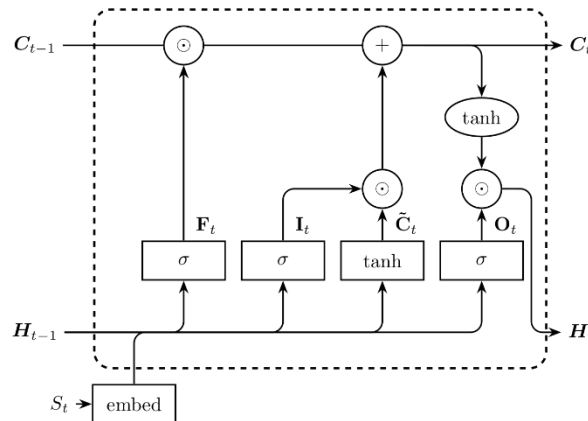


Figure 4: LSTM architecture

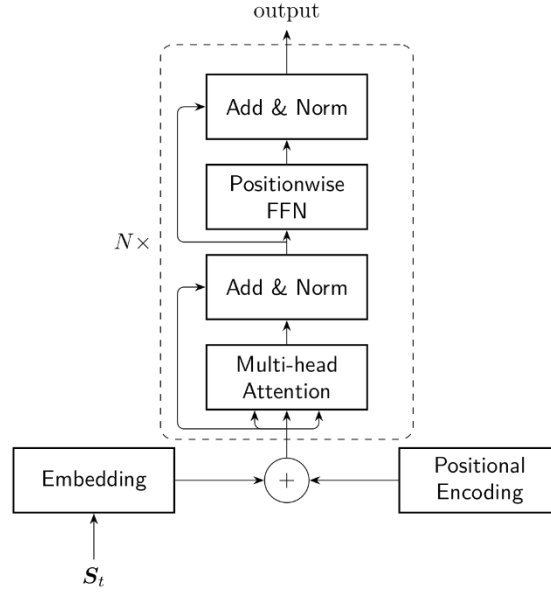


Figure 5: Transformer encoder

1. *LSTM* [13, 15]: The LSTM updates its hidden state $\mathbf{H}_t$ and cell state $\mathbf{C}_t$ through gating mechanisms:

$$\begin{aligned}
 \mathbf{I}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i) \\
 \mathbf{F}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f) \\
 \mathbf{O}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o) \\
 \tilde{\mathbf{C}}_t &= \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c) \\
 \mathbf{C}_t &= \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t \\
 \mathbf{H}_t &= \mathbf{O}_t \odot \tanh(\mathbf{C}_t)
 \end{aligned} \tag{4}$$

2. *Transformer* [14]: Using only the encoder module, the Transformer processes input sequences $\mathbf{S}_t$ through self-attention:

$$f(q) = \sum_{i=1}^n \text{softmax} \left(-\frac{1}{2} (q - k_i)^2 \right) v_i \tag{5}$$

where K , V , and Q represent keys, values, and queries respectively. Inputs are embedded with positional encodings before multi-head attention and feed-forward processing.

4. Experiments

4.1. Experimental setup

We implement our model using the Simulation of Urban MObility (SUMO) platform. The simulated intersection features six-lane approaches, each with dedicated turning lanes (Figure 6). Each 2000-frame simulation generates 1000 vehicles during the first 1800 frames (1 frame = 1 real-world second).

Vehicle spawn times and OD matrices follow real-world distributions, with initial speeds between 3-20 m/s.



Figure 6: SUMO simulation environment

4.2. Implementation details

We implement PPO using stable-baselines3 with a batch size of 50, a learning rate of 10^{-3} , and 500-frame epochs. Evaluation occurs every 5 epochs.

- LSTM: Input dim 32, hidden dim 64
- Transformer: Input dim 32, feedforward dim 128, learned positional encodings

Both models use identical time windows ($K=5$).

4.3. Results

From Figure 5, it can be observed that although the LSTM-based model learns more slowly in the initial stages compared to the Transformer, it eventually outperforms the Transformer, demonstrating superior performance. This may be due to the complexity of the Transformer model, which performs worse than LSTM in simpler scenarios. In Table 1, the LSTM + RL model achieves an average waiting time of 6.97 seconds, significantly outperforming the Transformer + RL model (10.54 seconds) and the baseline RL model (12.3 seconds). This indicates that using feature extraction models can enhance the performance of RL models. The Ruled Method performs the worst, with an average waiting time of 83.13 seconds, highlighting the limitations of fixed, non-adaptive traffic signal control. These results further demonstrate the superiority of RL in optimizing traffic signal control.

Table 1: Average waiting time (s)

Model	RL	LSTM+RL	Transformer + RL	Ruled Method
Waiting time	12.3	6.97	10.54	83.13

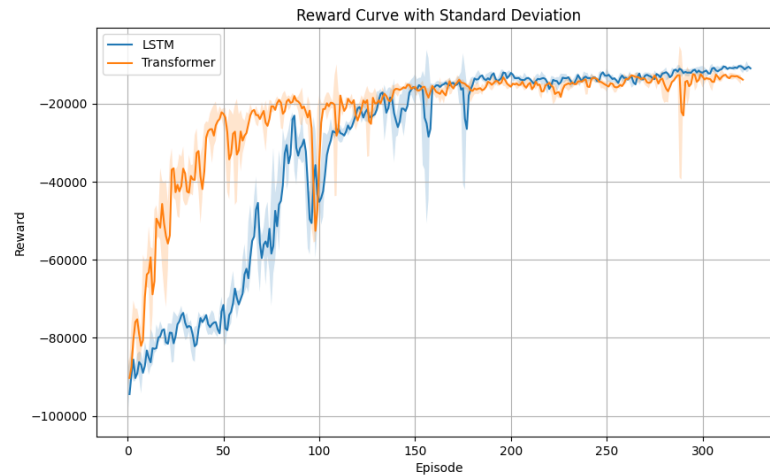


Figure 7: Reward progression during training

5. Conclusion

We proposed an RL-based solution for TSC using a four-way intersection model. Our PPO-based framework compared LSTM and Transformer feature extractors. Experimental results demonstrate RL's superiority over traditional methods, with LSTM achieving better performance than Transformer in this scenario. This suggests that while attention mechanisms show potential, traditional sequential models may be more effective for simple TSC tasks.

References

- [1] Pengjun Zhao and Haoyu Hu. *Geographical patterns of traffic congestion in growing megacities: Big data analytics from beijing*. *Cities*, 92:164–174, 2019.
- [2] Zhao H, Dong C, Cao J, et al. *A survey on deep reinforcement learning approaches for traffic signal control*[J]. *Engineering Applications of Artificial Intelligence*, 2024, 133: 108100.
- [3] Rasheed F, Yau K L A, Noor R M, et al. *Deep reinforcement learning for traffic signal control: A review*[J]. *IEEE Access*, 2020, 8: 208016-208044.
- [4] Noaeen M, Naik A, Goodman L, et al. *Reinforcement learning in urban network traffic signal control: A systematic literature review*[J]. *Expert Systems with Applications*, 2022, 199: 116830.
- [5] Yu Du, Wei ShangGuan, Dingchao Rong, and Linguo Chai. *Ra-tsc: Learning adaptive traffic signal control strategy via deep reinforcement learning*. In *2019 IEEE intelligent transportation systems conference (itsc)*, pages 3275–3280. IEEE, 2019.
- [6] Aoyu Pang, Zixiao Xu, Maonan Wang, Man-On Pun, and Yuheng Kan. *Reinforcement learning-based traffic signal control using delayed observations for v2x*. In *ICC 2023-IEEE International Conference on Communications*, pages 4020–4025. IEEE, 2023.
- [7] Du Y, ShangGuan W, Rong D, et al. *RA-TSC: Learning adaptive traffic signal control strategy via deep reinforcement learning*[C]//2019 IEEE intelligent transportation systems conference (itsc). IEEE, 2019: 3275-3280.
- [8] Mannion P, Duggan J, Howley E. *An experimental review of reinforcement learning algorithms for adaptive traffic signal control*[J]. *Autonomic road transport support systems*, 2016: 47-66.
- [9] Prabuchandran K J, AN H K, Bhatnagar S. *Multi-agent reinforcement learning for traffic signal control*[C]//17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014: 2529-2534.
- [10] Chu K F, Lam A Y S, Li V O K. *Traffic signal control using end-to-end off-policy deep reinforcement learning*[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(7): 7184-7195.
- [11] Aoyu Pang, Maonan Wang, Yirong Chen, Man-On Pun, and Michael Lepech. *Scalable reinforcement learning framework for traffic signal control under communication delays*. *IEEE Open Journal of Vehicular Technology*, 2024.
- [12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal policy optimization algorithms*. *arXiv preprint arXiv:1707.06347*, 2017.

- [13] *Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735– 1780, 11 1997.*
- [14] *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.*
- [15] *Karimzadeh M, Aebi R, de Souza A M, et al. Reinforcement learning-designed LSTM for trajectory and traffic flow prediction[C]//2021 IEEE wireless communications and networking conference (WCNC). IEEE, 2021: 1-6.*