

Article

Coordinated Multi-Intersection Traffic Signal Control Using a Policy-Regulated Deep Q-Network

Lin Ma ¹, Yan Liu ², Yang Liu ², Changxi Ma ^{2,*} and Shanpu Wang ²¹ Linxia Daohe Investment Co., Ltd., Linxia City 731100, China² School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China

* Correspondence: machangxi@mail.lzjtu.cn; Tel.: +86-13109429716

Abstract

Coordinated control across multiple signalized intersections is essential for mitigating congestion propagation in urban road networks. However, existing DQN-based approaches often suffer from unstable action switching, limited interpretability, and insufficient capability to model spatial spillback between adjacent intersections. To address these limitations, this study proposes a Policy-Regulated and Aligned Deep Q-Network (PRA-DQN) for cooperative multi-intersection signal control. A differentiable policy function is introduced and explicitly trained to align with the optimal Q-value-derived target distribution, yielding more stable and interpretable policy behavior. In addition, a cooperative reward structure integrating local delay, movement pressure, and upstream–downstream interactions enables agents to simultaneously optimize local efficiency and regional coordination. A parameter-sharing multi-agent framework further enhances scalability and learning consistency across intersections. Simulation experiments conducted on a 2×2 SUMO grid show that PRA-DQN consistently outperforms fixed-time, classical DQN, distributed DQN, and pressure/wave-based baselines. Compared with fixed-time control, PRA-DQN reduces maximum queue length by 21.17%, average queue length by 18.75%, and average waiting time by 17.71%. Moreover, relative to classical DQN coordination, PRA-DQN achieves an additional 7.53% reduction in average waiting time. These results confirm the effectiveness and superiority of the proposed method in suppressing congestion propagation and improving network-level traffic performance. The proposed PRA-DQN provides a practical and scalable basis for real-time deployment of coordinated signal control and can be readily extended to larger networks and time-varying demand conditions.

Keywords: multi-intersection signal coordination; deep reinforcement learning; policy-regulated and aligned DQN; sustainable transportation

Received: 12 December 2025

Revised: 30 January 2026

Accepted: 31 January 2026

Published: 2 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Urban traffic congestion imposes substantial economic and societal costs worldwide. According to the INRIX 2025 Global Traffic Scorecard, the typical U.S. driver lost 49 h annually to traffic delays, corresponding to approximately \$894 per driver and \$85.8 billion nationwide in lost productivity [1,2]. Signalized intersections are critical control points for congestion mitigation, and field evidence indicates that adaptive signal control can reduce control delay and the number of stops by up to 20% and 30%, respectively [3]. Recent studies have also reported notable improvements under heterogeneous traffic and

peak-demand conditions, including delay reductions exceeding 70% and reductions of around 50% in queue dissipation time using a hybrid Max Pressure and reinforcement learning approach [4], as well as reductions of up to 24% in average travel time and 45% in time loss using Proximal Policy Optimization (PPO)-based deep reinforcement learning [5]. Importantly, traffic demand varies markedly across hour-of-day, day-of-week, and public-holiday versus workday conditions. Prior empirical work shows that public holidays can significantly reshape both traffic volume levels and intraday profiles, highlighting the need to account for such time-varying demand when designing and deploying signal control programs [6].

Traffic-flow theory has described congestion evolution using traffic phase transitions, including congestion boundary approaches that characterize regime changes between undersaturated and oversaturated states [7,8]. In addition, simplified traffic models have interpreted congestion dynamics as a spatiotemporal clustering phenomenon [9]. These theoretical viewpoints imply that congestion propagation is driven by upstream–downstream coupling and spillback, which directly links the operational states of adjacent intersections. These insights motivate coordination mechanisms that explicitly address spillback and inter-intersection interactions in multi-intersection.

Traffic signal control plays a pivotal role in urban traffic management and is widely recognized as a complex sequential decision-making problem under dynamic and uncertain traffic conditions. Traditional rule-based or model-based signal control schemes, such as pre-timed signal plans or actuated signal control, often rely on predefined parameters and simplified assumptions, which limit their adaptability to heterogeneous and time-varying traffic conditions [10]. With the rapid development of sensing, communication, and computing technologies, deep reinforcement learning (DRL) has gained prominence as a promising approach for adaptive traffic signal control, enabling agents to learn control policies directly through interaction with the traffic environment [11].

Early DRL-based studies mainly focused on isolated intersections and showed that deep neural networks enable reinforcement learning to handle high-dimensional traffic states, thereby improving scalability. For example, Shabestary and Abdulhai [12] proposed an adaptive controller that directly processes high-resolution sensory inputs from connected vehicles, thereby avoiding handcrafted feature extraction. Chu et al. [13] developed an end-to-end off-policy DRL framework that takes real-time images of an intersection as input and learns a near-optimal timing policy to minimize average waiting time. Ma et al. [14] further integrated temporal traffic pattern mining with an actor–critic architecture, transforming traffic conditions into sequential image representations to improve the robustness of signal timing. Chen et al. [15] constructed a 3DQN-based model that combines dueling networks, double DQN, and prioritized experience replay to optimize single-intersection timing using floating car data. These studies collectively demonstrate the effectiveness of DRL for adaptive control at individual intersections. As research attention shifted toward network-level performance, multi-agent DRL began to play a central role in multi-intersection signal coordination. Hu and Li [16] proposed a multi-agent Double DQN framework in which local agents are trained independently and then integrated into a global agent to coordinate corridor signals. Zhang et al. [17] designed a fully scalable multi-agent proximal policy optimization (PPO) algorithm with parameter sharing to coordinate arterial corridors under dynamic traffic conditions. Park et al. [18] developed DQN-based models for both isolated and two-intersection coordinated control and showed that learned policies can outperform optimized fixed-time plans. To alleviate scalability and non-stationarity issues in large-scale networks, Jiang et al. [19] introduced a distributed multi-agent reinforcement learning scheme with graph decomposition, while Gu et al. [20] proposed RegionLight, which partitions the network into star-shaped regions and applies an adaptive branching dueling Q-network for regional control.

Related works have also explored knowledge-sharing schemes [21], cooperative group-based Multi-Agent Reinforcement Learning (MARL) [22], and topology-embedding propagation strategies [23] to enhance coordination across large networks.

Beyond basic efficiency objectives, DRL-based traffic signal control has been extended to a wide range of application scenarios, including mixed traffic environments, multi-modal flows, and cooperative vehicle–infrastructure systems. Related predictive control studies in coupled transportation–energy systems further highlight the importance of accurate demand forecasting and multi-stage decision-making [24]. Yang [25] proposed a PPO-based controller with a multi-discrete action space and combined state representation to enhance robustness in mixed connected and non-connected traffic. Wang et al. [26] developed a human-centric multimodal deep signal control scheme to coordinate vehicles, buses, and pedestrians, explicitly targeting per-capita waiting time and social equity. Yazdani et al. [27] designed an intelligent vehicle–pedestrian light controller that minimizes total user delay by jointly considering vehicle–vehicle, vehicle–pedestrian, and pedestrian–pedestrian interactions. Multi-modal and transit-priority settings have also been studied through decentralized MARL and DRL-based Transit Signal Priority (TSP) models, demonstrating the potential of RL to balance bus reliability and car performance in complex networks [28–30]. In the context of connected and automated vehicles (CAVs), several works have investigated cooperative signal and vehicle control to jointly optimize travel time, emissions, and safety, or to integrate signal control with vehicle navigation tasks [31–34]. Other studies have considered privacy-preserving signal control [35], software-defined or 6G-enabled smart transportation architectures [36–38], and democratic multi-goal control schemes based on voting among DRL-trained controllers [39].

At the methodological level, value-based and policy-based DRL algorithms have both been extensively applied to traffic signal control. Beyond standard DQN, various improvements—such as double DQN, dueling architectures, prioritized replay, and hybrid actor–critic schemes—have been introduced to address overestimation, instability, and sample inefficiency [40]. Policy-gradient and PPO-based methods have been proposed to optimize phase sequences and durations in continuous or hybrid action spaces, enhancing flexibility under diverse demand patterns [41–43]. However, several fundamental challenges remain. First, many DQN-based controllers still rely on ϵ -greedy exploration directly on noisy Q-value estimates, which can lead to unstable phase switching and limited interpretability of the learned policies, particularly in multi-agent settings. Second, although reward functions commonly include delay, queue length, or pressure-based terms [44], they often underrepresent upstream–downstream interactions and spillback effects, which are critical for robust multi-intersection coordination. Third, robustness and generalization are not fully resolved: DRL agents are sensitive to abnormal input data, sensor failures, or partially blinded intersections, and improvements via specialized reward designs or robustness-enhancing mechanisms are still an active research topic [45–47].

Recent studies have explored policy–value separation and stabilization techniques to improve learning robustness, as well as graph-based multi-agent reinforcement learning to encode network topology [25,41]. Centralized-critic architectures under the Centralized Training with Decentralized Execution (CTDE) paradigm and hierarchical reinforcement learning frameworks further address coordination by introducing centralized value estimation or multi-level decision structures [48,49]. In contrast, PRA-DQN focuses on stabilizing value-based learning through explicit policy–Q alignment and parameter sharing, without relying on centralized critics or explicit graph encoders, which distinguishes it from these network-level control paradigms.

Against this background, this study aims to enhance the stability, interpretability, and coordination capability of value-based multi-intersection controllers by proposing a Policy-Regulated and Aligned Deep Q-Network (PRA-DQN). The proposed approach builds upon the classical DQN framework and targets the aforementioned gaps from three perspectives.

- (1) A structured multi-intersection state representation is constructed, integrating waiting time, queue length, and movement-level pressure, so that agents can better capture both local congestion and upstream–downstream interactions.
- (2) A hybrid reward function is designed to combine delay reduction, pressure balancing, and adjacent-intersection influence, thereby encouraging each agent to consider not only its own performance but also the impact on neighboring intersections and network-wide coordination.
- (3) A policy-regulation and Q-alignment mechanism is introduced, in which an explicit, differentiable policy function is trained to align with the Q-value–induced target distribution, smoothing action-selection noise and improving interpretability. Together with a parameter-sharing multi-agent structure, this mechanism improves learning stability and ensures consistent behavior across intersections.

The proposed PRA-DQN is evaluated on a SUMO-based 2×2 grid network and compared against six representative baselines, including classical DQN, distributed DQN, stochastic action selection, Max Pressure, MaxWave, and fixed-time signal control. The experimental results demonstrate that PRA-DQN can effectively reduce maximum queue length, average queue length, and average waiting time, confirming its potential as a practical solution for coordinated control of multi-intersections.

This study focuses on coordinated signal control at the intersection level under coupled traffic dynamics (e.g., queue spillback and upstream–downstream interactions). To ensure clarity and reproducibility, all state variables used for learning are explicitly defined with their measurement units and aggregation rules. The proposed neural-network-based controller learns a mapping from the measured traffic states to discrete signal phase actions, enabling adaptive phase selection under time-varying congestion in the studied multi-intersection setting.

The remainder of this paper is organized as follows. Section 2 describes the problem setting and the proposed multi-intersection state/action formulation. Section 3 presents the PRA-DQN methodology, including the reward design and policy–Q alignment mechanism. Section 4 reports the simulation environment, experimental setup, and comparative results. Section 5 concludes the paper and outlines directions for future work.

2. Multi-Intersection Signal Coordination Model

2.1. State Space

The state space represents the traffic conditions perceived by each agent at every decision step. In the proposed PRA-DQN framework, the state of intersection i is constructed from four key traffic indicators that jointly capture congestion levels, delay accumulation, vehicle demand, and local movement dynamics.

Let L_i denote the set of incoming lanes at intersection i . For each lane $l \in L_i$, let q_l be the queue length, w_l the total waiting time, n_l the number of approaching vehicles, and v_l the total speed of vehicles on that lane.

The intersection-level aggregated features are defined as:

$$Q_i(t) = \sum_{l \in L_i} q_l(t) \quad (1)$$

$$w_i(t) = \sum_{l \in L_i} w_l(t) \quad (2)$$

$$n_i(t) = \sum_{l \in L_i} n_l(t) \quad (3)$$

$$v_i(t) = \sum_{l \in L_i} v_l(t) \quad (4)$$

Accordingly, the complete state vector for intersection i is expressed as:

$$s_i = \{q_i, w_i, n_i, v_i\} \quad (5)$$

This four-dimensional representation provides the necessary information for the PRA-DQN agent to evaluate current intersection conditions and predict short-term traffic evolution. It also forms the basis for constructing structured feature tensors used in the convolutional neural network described in Section 3.2.

2.2. Action Space

The action space specifies the set of signal control operations that can be executed at each intersection. In general, signal controllers may operate under either fixed or variable phase sequences. To ensure safety, avoid phase conflicts, and simplify coordinated decision-making across intersections, this study adopts a fixed four-phase sequence for all intersections in the network. Each cycle consists of four movements: east–west through (EW-T), east–west left-turn (EW-L), north–south through (NS-T), and north–south left-turn (NS-L), as illustrated in Figure 1. The fixed four-phase structure is adopted to ensure safety and comparability in this controlled study. For intersections with heterogeneous geometries or phase requirements, the framework can be extended by defining intersection-specific phase sets and applying action masking to restrict infeasible actions, without altering the core PRA-DQN learning mechanism.

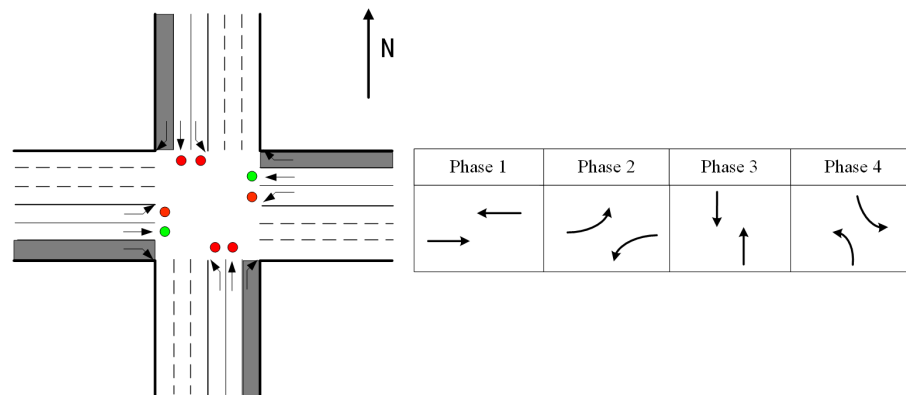


Figure 1. Schematic Diagram of the Four Phases at the Intersection.

For a control region containing N signalized intersections, the action taken at each decision step corresponds to selecting one of the four predefined phases. Let the action set of intersection i be denoted by

$$A_i = \{a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}\} \quad (6)$$

The joint action space of the multi-intersection system is then expressed as the Cartesian product:

$$\mathbf{A} = \mathbf{A}_1 \times \mathbf{A}_2 \times \mathbf{L} \times \mathbf{A}_N \quad (7)$$

with a total size of $|\mathbf{A}| = 4^N$. Each joint action corresponds to a coordinated release pattern across all intersections in the network.

At each time step t , the agent determines whether to maintain the current phase or switch to the next phase in the fixed sequence. A green interval of 10 s is allocated to each activated phase, followed by a 3 s yellow interval to ensure safe clearance. Figure 2 illustrates how different actions correspond to changes in the displayed signal at each intersection.

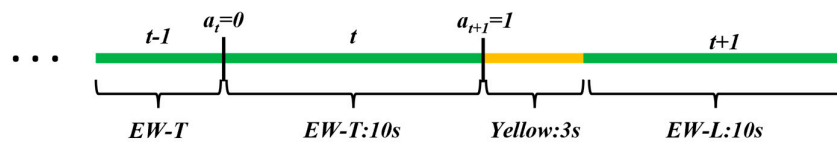


Figure 2. The signal changes corresponding to different action selections.

By constructing the global joint action space in this manner, the control framework supports coordinated phase decisions across the entire multi-intersection network. This design facilitates information sharing and cooperative optimization among agents, ultimately contributing to improved traffic flow efficiency and reduced congestion propagation.

2.3. Reward Function

The reward function provides feedback for learning based on the current traffic state and the selected signal action [50]. In this study, the reward is formulated using lane-level measures of waiting time and queue-related congestion, and all reward terms are computed using the same lane-level measurements and detection segment definitions as the state variables to ensure consistency between the input representation and the optimization objective.

All model parameters are defined in Table 1.

Table 1. Description of Model Parameters.

Symbol	Definition	Unit	Aggregation/Measurement rule
i, j	Index of intersections, $i, j \in 1, \dots, N$	—	—
N	Number of intersections in the network	—	—
L_i	Set of incoming lanes at intersection i	—	—
$l \in L_i$	Index of an incoming lane of intersection i	—	—
Δt	Decision interval for signal control	s	Fixed to 10 s per decision step (followed by a 3 s yellow clearance)
$q_l(t)$	Queue length on lane l at time t	veh	Instantaneous count per decision step; measured on lane l (same detection segment as other lane-level variables)
$w_l(t)$	Total waiting time on lane l at time t	s	Accumulated waiting time of vehicles on lane l at each decision step
$n_l(t)$	Number of approaching vehicles on lane l at time t	veh	Instantaneous vehicle count per decision step; not time-aggregated flow
$v_l^{\text{sum}}(t)$	Total (summed) speed of vehicles on lane l at time t	m/s	Sum of instantaneous vehicle speeds on lane l at each decision step

$q_i(t)$	Total queue length at intersection i	veh	$q_i(t) = \sum_{l \in L_i} q_l(t)$
$w_i(t)$	Total waiting time at intersection i	s	$w_i(t) = \sum_{l \in L_i} w_l(t)$
$n_i(t)$	Total number of approaching vehicles at intersection i	veh	$n_i(t) = \sum_{l \in L_i} n_l(t)$
$v_i^{\text{sum}}(t)$	Total (summed) speed at intersection i	m/s	$v_i^{\text{sum}}(t) = \sum_{l \in L_i} v_l^{\text{sum}}(t)$
C_l	Maximum capacity of lane l	veh	Used for normalization in ρ_l ; represents the maximum number of vehicles that can be accommodated on lane l (under the adopted lane capacity definition)
$\rho_l(t)$	(Normalized) density of lane l at time t	—	$\rho_l(t) = \frac{n_l(t)}{C_l}$ (dimensionless)
$m \in M_i$	Traffic movement at intersection i	—	A movement corresponds to a specific incoming–outgoing lane pair governed by a phase
M_i	Set of all possible movements at intersection i	—	—
In (m), Out (m)	Incoming and outgoing lanes of movement m	—	—
$P_m(t)$	Pressure of movement m at time t	—	Defined using incoming/outgoing lane densities (see Section 2.3); computed per decision step
$P_i(t)$	Intersection pressure at intersection i	—	$P_i(t) = \sum_{m \in M_i} P_m(t)$
$a_i(t) \in A$	Action (phase selection) at intersection i	—	One action selected per decision step
A	Action set (four-phase scheme)	—	$A = \text{EW} - \text{T}, \text{EW} - \text{L}, \text{NS} - \text{T}, \text{NS} - \text{L}$
$s_i(t)$	State vector of intersection i	—	A four-dimensional state constructed from $\{q_{i'}, w_{i'}, n_{i'}, v_i^{\text{sum}}\}$ (Section 2.1)
$r_i(t)$	Reward of intersection i	—	Computed from waiting-time-related and pressure-related components (Section 2.3)
$Q(s, a)$	Q-value function	—	—
$G(s, a; \theta)$	Adjustable policy function (PRA-DQN)	—	—
θ, θ^-	Trainable parameters and target network parameters	—	—
γ	Discount factor	—	—
α	Learning rate	—	—

All traffic state variables are sampled once per decision step ($\Delta t = 10$ s) with a 3 s yellow clearance. The number of approaching vehicles is defined as an instantaneous count within a predefined upstream detection segment on each lane, rather than a time-aggregated flow. All variables follow standard traffic engineering units and aggregation rules consistent with fundamental-diagram-based and heterogeneous-data traffic analysis [51].

(1) Waiting Time

Waiting time is a critical indicator in signal control systems, impacting traffic flow, vehicle delays, and overall efficiency. The total waiting time across all lanes at each intersection is first computed, and its negative value is then used as the reward to minimize the time vehicles remain stopped at red lights. This encourages smoother traffic flow and higher operational efficiency. Let the set of incoming lanes at intersection i be denoted by L_i , and let w_l denote the total waiting time on lane l . The total waiting time at intersection i is given by

$$W_i = \sum_{l \in L_i} w_l \quad (8)$$

The corresponding waiting-time-related reward is defined as the negative of W_i

$$r_i^{(w)} = -W_i \quad (9)$$

By minimizing the total waiting time W_i through maximizing $r_i^{(w)}$, the time vehicles spend stopped at red lights is reduced, thereby enhancing the dynamic flow of traffic and improving the overall efficiency of the intersection.

(2) Queue Length

Queue length represents the number of vehicles waiting on a signalized approach and is one of the most important indicators of intersection performance. A longer queue typically reflects insufficient traffic discharge during the preceding phases and indicates a higher likelihood of spillback and congestion propagation. To quantify capture this effect, this study adopts the pressure formulation commonly used in multi-intersection signal control.

The density on lane l is defined as

$$\rho_l = n_l / C_l \quad (10)$$

where n_l denotes the number of vehicles on lane l and C_l is the maximum capacity of lane l . For a traffic movement m at intersection i , let $In(m)$ and $Out(m)$ denote its incoming and outgoing lanes, respectively. The movement pressure P_m is then defined as

$$P_m = \rho_{In(m)} - \rho_{Out(m)} \quad (11)$$

The intersection pressure at intersection i is defined as the sum of the absolute values of all movement pressures (Figure 3):

$$P_i = \sum_{m \in M_i} |P_m| \quad (12)$$

Here, M_i denotes the set of all movements at intersection i . A larger value of P_i indicates a more imbalanced distribution of vehicles between incoming and outgoing lanes. Therefore, the pressure-related reward at intersection i is defined as

$$r_i^{(p)} = -P_i \quad (13)$$

By minimizing P_i through maximizing $r_i^{(p)}$, the controller promotes a more balanced traffic distribution across the intersection, helping to reduce congestion and improve overall traffic efficiency.

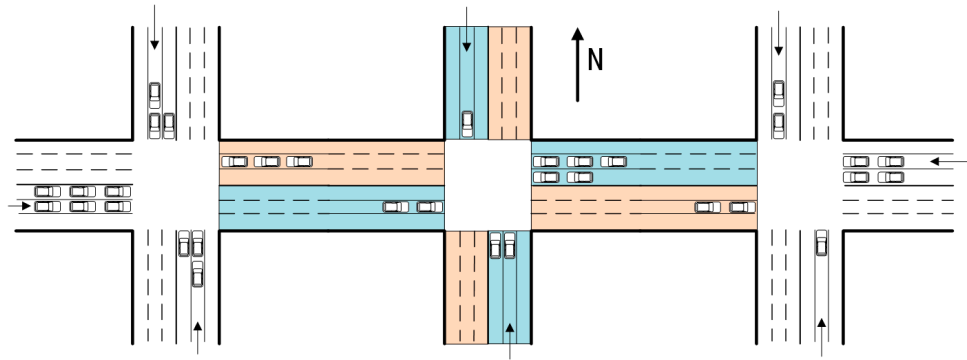


Figure 3. The diagram illustrating the definition of intersection pressure.

(3) Influence of Adjacent Intersections

In addition to local queue length and waiting time, the collaborative influence of adjacent intersections is also incorporated into the reward design. This term penalizes large queues and long waiting times at neighboring intersections so that each agent takes into account upstream–downstream spillback effects and contributes to network-wide coordination.

Let $N(i)$ denote the set of intersections adjacent to intersection i . For each neighboring intersection $j \in N(i)$, let q_j and w_j denote the total queue length and total waiting time at intersection j , respectively. They can be computed as

$$q_j = \sum_{l \in L_j} q_{l,j} \quad (14)$$

$$w_j = \sum_{l \in L_j} w_{l,j} \quad (15)$$

The neighborhood influence term C_i is then defined as

$$C_i = \sum_{j \in N(i)} (q_j + w_j) \quad (16)$$

and the corresponding neighborhood-related reward is given by

$$r_i^{(n)} = C_i \quad (17)$$

By minimizing C_i through maximizing $r_i^{(n)}$, the controller encourages each intersection to reduce the risk of congestion spillback and to support smoother traffic propagation in its surrounding area.

(4) Overall Reward and Objective

Combining the above three components, the total reward for intersection i at each decision step is defined as

$$r_i = r_i^{(w)} + r_i^{(l)} + r_i^{(n)} \quad (18)$$

The overall objective of the multi-intersection signal control problem is to maximize the cumulative reward over all intersections in the control area:

$$\max \sum_{i=1}^N r_i \quad (19)$$

3. Signal Coordination Control Algorithm Based on PRA-DQN

3.1. Policy-Regulated and Aligned DQN (PRA-DQN) Algorithm

In reinforcement learning, the policy function determines how an agent selects actions based on the observed state. Traditional DQN implicitly embeds this policy within the ϵ -greedy mechanism: with probability ϵ , a random action is selected for exploration, and with probability $1 - \epsilon$, the agent exploits the Q-function by choosing the action with the maximal Q-value. While simple, this implicit policy structure often leads to unstable and non-smooth behavior in multi-intersection traffic signal control, where consistency and interpretability are critical.

To overcome these limitations, this study introduces a Policy-Regulated and Aligned DQN (PRA-DQN) framework, which makes the policy explicit and optimizable. PRA-DQN incorporates an adjustable policy function $G(s, a; \theta)$, where s represents the current state (e.g., queue length, waiting time, approaching traffic), a is a candidate signal phase, and θ is the set of trainable policy parameters. The output of G is transformed (e.g., by a Hardmax/Softmax operation) into a probability distribution over actions, enhancing interpretability and reducing noise in action selection.

(1) Explicit Policy Function

The explicit representation of the policy allows direct optimization instead of relying solely on ϵ -greedy exploitation. During interaction with the environment, PRA-DQN still applies ϵ -greedy exploration—selecting a random action with probability ϵ —while the exploitation step uses the policy distribution generated by $G(s, a; \theta)$. This design preserves exploration capability while enabling a more stable and interpretable exploitation phase.

(2) Policy-Q Alignment Mechanism

A distinguishing feature of PRA-DQN is the policy-Q alignment mechanism, which ensures that the policy function selects the same optimal action as the Q-function. The policy-Q alignment mechanism can be interpreted as training an explicit policy to match a Q-induced target distribution over actions. Compared with direct ϵ -greedy action selection on noisy Q estimates, the explicit policy provides a smoother state-action mapping, which reduces high-frequency switching among signal phases and improves learning stability. For each observed state, the Q-network computes the optimal action:

$$a^* = \arg \max_a Q(s, a; \theta_Q) \quad (20)$$

The alignment target is then defined as:

$$y(s, a) = \begin{cases} 1, & \text{if } a = a^*, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

This target reflects whether each action matches the Q-value-derived optimal choice. The policy parameters θ are optimized by minimizing a log-loss function:

$$L_G = - \sum_a y(s, a) \log G(s, a; \theta) \quad (22)$$

and updated by stochastic gradient descent (SGD). Through repeated updates, the output $G(s, a; \theta)$ progressively aligns with the optimal action dictated by the Q-function. This mechanism enhances policy interpretability and suppresses the action-selection noise typically seen in classical DQN.

Mini-batch training is applied to stabilize updates: samples are drawn from the shared replay buffer, and gradients are computed for each batch before updating θ .

(3) Q-Network Optimization

The Q-network retains the standard DQN structure. The target value for each transition (s, a, r, s') is computed as:

$$y_Q = r + \gamma \max_a Q(s', a'; \theta_Q^-) \quad (23)$$

where θ_Q^- denotes the target network parameters. The Q-network parameters θ_Q are trained by minimizing the Huber loss using the Adam optimizer. This process ensures stable value estimation, while the target network mitigates the non-stationarity of the learning process.

(4) Combined Learning Process

During training, PRA-DQN alternates between two coupled learning steps:

1. Value Learning: Updating the Q-network parameters θ_Q based on sampled transitions.
2. Policy adaptation: updating the policy parameters θ so that $G(s, a; \theta)$ aligns with the Q-value-based optimal action.

This two-step process ensures that the policy remains consistent with value estimation, while reducing the instability and noise that may arise from purely ϵ -greedy action selection.

(5) Practical Training Procedure

The complete algorithmic implementation is shown in Algorithm 1:

Algorithm 1. Policy-Regulated and Aligned DQN (PRA-DQN)

Initialization:

- Initialize replay buffer D with capacity N .
- Initialize Q-network $Q(s, a; \theta_Q)$ with random weights.
- Initialize target Q-network Q^- with $\theta_Q^- = \theta_Q$
- Initialize the adjustable policy function $G(s, a; \theta)$ with parameters $\theta = [1, \dots, 1]$.

Main Loop (for each episode):

- For each episode $t = 1$ to T :
 1. **Action Selection using ϵ -greedy:**
 - With probability ϵ , select a random action a_t ;
 - Otherwise, select action according to the policy function $G(s_t, a; \theta)$.
 2. **Execute action** a_t , observe reward r_t and next state s_{t+1} .
 3. **Store transition** (s_t, a_t, r_t, s_{t+1}) in the experience replay buffer D .

Q-function Update:

- Randomly sample a minibatch from buffer D .
 - For each transition sample (s_j, a_j, r_j, s_{j+1}) :
 1. **Compute Temporal Difference (TD) target:**

$$y_j = r_j + \gamma \max_a Q(s_{j+1}, a'; \theta)$$
-

2. **Update Q-network:** Minimize the Huber loss between y_j and

$$Q(s_j, a_j; \theta_Q).$$

- Every C steps, update the target Q-function $\theta_Q^- = \theta_Q$

Training the Policy Function G:

- For each sample in the minibatch:

1. **Compute alignment target:**

$$y(s, a) = \begin{cases} 1, & \text{if } a = a^*, \\ 0, & \text{otherwise.} \end{cases}$$

2. **Compute log-loss:**

$$L_G = - \sum_a y(s_j, a) \log G(s_j, a; \theta).$$

3. **Update policy parameters:**

- Update θ using stochastic gradient descent.

3.2. Neural Network Architecture Design

The PRA-DQN framework employs a deep neural network to approximate the state–action value function $Q(s, a)$. To exploit spatial relationships among lanes, the lane-level features (q_1, w_1, n_1, v_1) for all incoming lanes $l \in L_i$ are arranged into a two-dimensional tensor $X_i \in \mathbb{R}^{M \times 4}$, where M is the number of incoming lanes.

(1) Input Feature Organization

For each intersection, the lane-level feature (q_1, w_1, n_1, v_1) of all incoming lanes $l \in L_i$ is organised into a two-dimensional tensor of size: $X_i \in \mathbb{R}^{M \times 4}$, where M is the number of incoming lanes.

Each row corresponds to one lane, and each column corresponds to one feature dimension.

(2) Convolutional Feature Extraction

Two convolutional layers with 3×3 kernels and ReLU activation are applied to the feature map. A max-pooling operation follows each convolutional layer to reduce dimensionality and enhance robustness to local noise.

Let $F^{(2)}$ denote the output of the second convolutional layer.

(3) Fully Connected Layers and Output

The final convolutional feature map is flattened and passed through two fully connected layers with ReLU activation:

$$h_1 = \sigma(W_1 z + b_1) \quad (24)$$

$$h_2 = \sigma(W_2 h_1 + b_2) \quad (25)$$

where $z = \text{Flatten}(F^{(2)})$.

W_1, W_2 : weight matrices of the first and second fully connected layers;

b_1, b_2 : bias vectors;

$\sigma(\bullet)$: ReLU activation function;

h_1, h_2 : hidden-layer activation vectors.

The output layer computes the Q-value of each action:

$$Q(s, a; \theta_Q) = W_o h_2 + b_o \quad (26)$$

where

W_o and b_o : weights and bias of the output layer;

$\theta_Q = \{W_1, W_2, W_o, b_1, b_2, b_o\}$: the set of all learnable parameters of the Q-network.

The output layer contains $|A| = 4$ neurons, corresponding to the four signal phases available at each intersection.

(4) Loss Function and Optimization

The Q-network parameters θ_Q are optimized using the Huber loss:

$$L_Q = \begin{cases} \frac{1}{2} (y_Q - Q(s, a; \theta_Q))^2, & \text{if } |y_Q - Q| < 1, \\ |y_Q - Q(s, a; \theta_Q)| - \frac{1}{2}, & \text{otherwise.} \end{cases} \quad (27)$$

The temporal-difference target is computed following the standard DQN formulation:

$$y_Q = r + \gamma \max_a Q(s', a'; \theta_Q^-) \quad (28)$$

where θ_Q^- denotes the parameters of the target network.

Adam is used for optimization, and the target network is updated every C steps to stabilize training.

(5) Experience Replay

Minibatches are uniformly sampled from a shared replay buffer to reduce temporal correlation in samples.

All agents use the same convolutional neural network fully connected (CNN-FC) architecture and shared parameters, enabling consistent and efficient representation learning in the multi-intersection setting.

The network parameters are set as shown in Table 2.

Table 2. Network Parameter Settings.

Parameter	Description	Value
Learning rate (Q-network)	Step size for updating Q-network weights (Adam)	0.001
Learning rate (policy function G)	SGD learning rate for policy alignment	0.01
Discount factor γ	Weight of future rewards	0.99
Replay buffer size	Capacity of stored transitions	10,000
Batch size	Number of samples per update	32
Target network update frequency	Soft update interval	500 steps
Convolutional layers	CNN feature extraction layers	2
Kernel size	Size of convolution kernels	3×3
Pooling	Type of pooling	MaxPooling
Activation function	Nonlinear activation	rectified linear unit (ReLU)
Fully connected layers	Number of FC layers	2

Although the state consists of four feature channels, the lane-level features form a structured representation across incoming lanes. Convolutional layers enable shared kernels to capture local correlations among adjacent lanes while maintaining parameter efficiency. This inductive bias is beneficial for generalization across intersections with similar lane configurations.

3.3. Parameter Sharing Mechanism

To enhance learning efficiency and promote consistent behavior across multiple intersections, a parameter-sharing mechanism is adopted in the PRA-DQN framework. In this design, all agents share a unified Q-network and policy alignment module, while interacting with the traffic environment independently. For each intersection, the local state vector $S_i = \{q_i, w_i, n_i, v_i\}$ is fed into the shared network, which outputs the corresponding Q-values and recommends the signal phase action.

During each interaction cycle, intersection i executes the action suggested by the shared network, observes the resulting next state and reward, and stores the transition (s_i, a_i, r_i, s'_i) into a common replay buffer. This shared replay memory allows agents to benefit from one another's experience, providing richer and more diverse samples that enhance training stability and accelerate value propagation throughout the network.

Parameter sharing offers two important advantages. First, it significantly reduces the number of trainable parameters compared to maintaining separate networks for each agent, thereby improving computational efficiency and scalability. Second, it encourages partially synchronized updates across intersections, which is beneficial for coordinated signal control in a grid network where local traffic states influence one another. The overall architecture of the parameter-sharing framework is illustrated in Figure 4.

To ensure stable convergence, key hyperparameters are chosen based on widely adopted practices in DQN-based traffic signal control. Specifically, the discount factor is uniformly set to $\gamma = 0.99$ for both the Q-network and training environment. The Q-network parameters are updated using the Adam optimizer with a learning rate of 0.001, while the policy alignment module $G(s, a; \theta)$ is optimized using stochastic gradient descent with a learning rate of 0.01. These parameter settings help maintain numerical stability and prevent divergence during training.

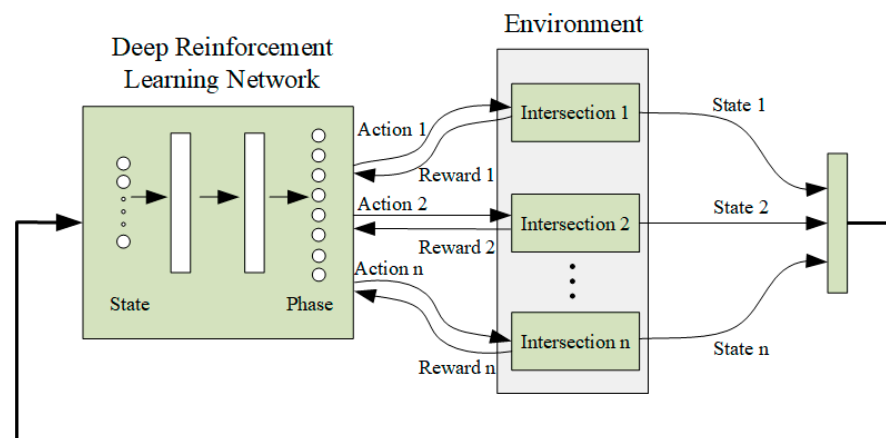


Figure 4. Architecture of Multi-Intersection Traffic Signal Control Algorithm.

3.4. Signal Control Based on Parameter-Sharing PRA-DQN Algorithm

Figure 5 presents the complete workflow of the proposed parameter-sharing PRA-DQN signal control algorithm. At each decision step, the agent at intersection i observes the current state vector $s_i = \{Q_i, w_i, n_i, v_i\}$. The shared PRA-DQN network evaluates the state and outputs estimated Q-values for all possible signal phases. The adjustable policy function $G(s, a; \theta)$ then selects the action based on the policy-aligned probability distribution, ensuring consistency between the learned policy and the Q-value-derived optimal action.

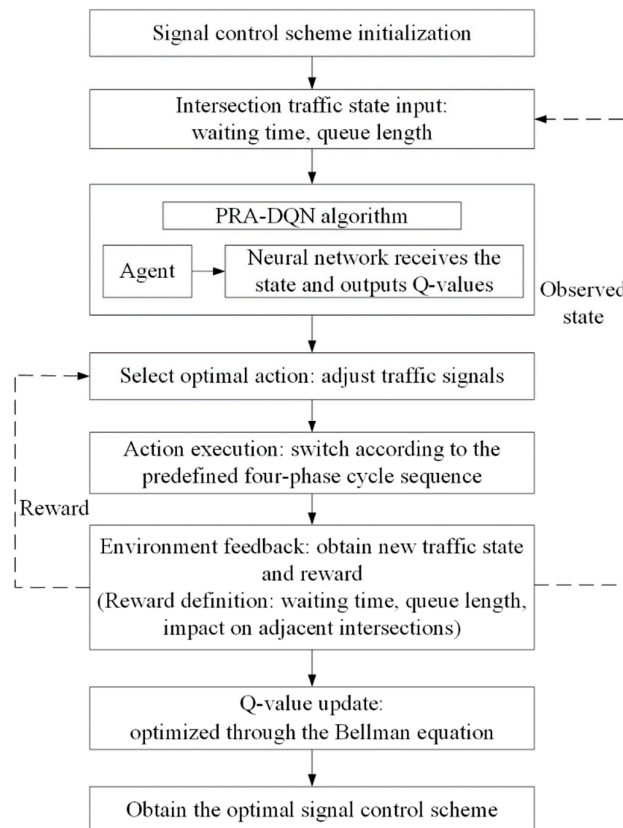


Figure 5. Signal Coordination Control Flowchart Based on Parameter-Sharing PRA-DQN Algorithm.

Once an action is selected, the corresponding traffic phase is executed for a fixed time interval, after which the environment returns the next state and reward. This experience is appended to the shared replay buffer for batch sampling and network updates. The Q-network is trained using the Huber loss and Adam optimizer, while the policy alignment module is trained using a log-loss objective and SGD. By alternating between Q-value updates and policy alignment updates, PRA-DQN simultaneously improves value estimation accuracy and action selection stability.

Through repeated interaction with the traffic environment, the PRA-DQN framework gradually converges to an effective coordinated control strategy. The shared structure enables agents to implicitly learn spatial dependencies between intersections, considering factors such as traffic spillback, movement pressure, and local delay. As a result, the algorithm not only optimizes each intersection's phase transitions but also fosters network-level cooperation, reducing queue accumulation and preventing congestion propagation across the grid.

The parameter-sharing PRA-DQN algorithm provides four main advantages:

Local decision-making: Each agent reacts promptly to its own traffic conditions using a unified model.

Pressure-aware coordination: By incorporating intersection pressure into the reward, the agents balance inflows and outflows to maintain smooth traffic progression.

Efficient learning: Shared parameters reduce computational cost and improve model generalisation across different intersections.

Collaborative adaptation: Shared experience enables agents to learn cooperative behaviors that mitigate spillback and enhance overall traffic flow in the multi-intersection network.

4. Simulation Experiments and Results Analysis

4.1. Simulation Environment

All experiments are conducted using the open-source traffic simulation platform SUMO. The simulation environment is implemented on a Windows 10 (64-bit) system equipped with an Intel Core i5-7200U Central Processing Unit (CPU) (2.50 GHz) and 8 GB of Random-Access Memory (RAM). SUMO version 1.20.0 is used, together with the Traffic Control Interface (TraCI) interface, to retrieve traffic states and apply signal control actions in real time. The PRA-DQN algorithm is implemented in Python 3.7 with TensorFlow 1.19.0. The reinforcement learning agent communicates with SUMO at each decision step through TraCI to perform signal-phase adjustments and collect transition samples. Signal control is executed at a fixed decision interval of $\Delta t = 10$ s with a 3 s yellow clearance, and all learning-related traffic measurements are collected at each decision step to align the simulation interface with the defined state and reward formulations.

Unless otherwise specified, vehicles follow SUMO's default microscopic models, namely the Krauss car-following model and the LC2013 lane-changing model. The test network is a 2×2 bidirectional grid with four signalized intersections spaced 500 m apart. Each approach is configured with two incoming lanes (a dedicated through lane and a dedicated left-turn lane) to match the four-phase signal plan. Vehicles are generated from boundary entry links located 250 m upstream of each stop line. The maximum speed on all links is set to 50 km/h, and vehicles are inserted with an initial speed of 35 km/h.

Traffic demand follows a time-varying peak-period profile, starting from a low level, increasing to a peak, and then gradually decreasing, with an average arrival rate of 1.25 veh/s. For each vehicle, the origin is sampled from boundary entry links and the destination is randomly selected from boundary links, yielding diversified turning movements and inter-intersection interactions. The peak-demand period produces congested (over-saturated) conditions and potential spillback. Signal control uses a fixed four-phase definition (east–west through, east–west left-turn, north–south through, and north–south left-turn). The minimum and maximum green times are set to 15 s and 60 s, respectively; a 3 s yellow clearance is applied between phases; and the all-red time is set to 0 s. All intersections start with a zero initial offset to ensure consistent initialization across runs.

4.2. Road Network Configuration

The simulation experiments are conducted on a 2×2 grid network, as illustrated in Figure 6. The network consists of four signalized intersections with identical geometric structures and phase plans. Each intersection supports four movement phases: east–west through, east–west left-turn, north–south through, and north–south left-turn.

All links are bidirectional, and the distance between adjacent intersections is set to 500 m. Vehicle entry points are located 250 m upstream of each stop line, enabling the simulation to capture traffic demand propagation and potential queue spillback within

and across intersections under dynamic traffic conditions. Accordingly, the 2×2 grid network is adopted as a controlled experimental setting to examine congestion propagation across adjacent intersections, including spillback and upstream–downstream coupling. This configuration provides a coupled multi-intersection environment while retaining a well-defined and reproducible topology for systematic evaluation of coordination mechanisms.

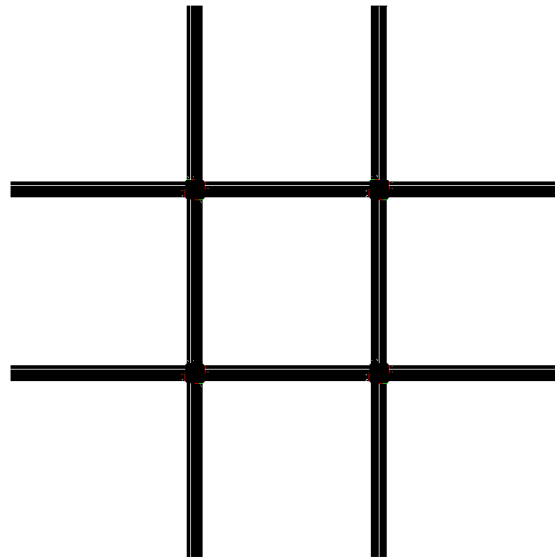


Figure 6. 2×2 SUMO Simulation Road Network.

4.3. Traffic Flow Simulation Setup

To emulate realistic urban peak-hour traffic, a dynamically varying traffic flow is generated for all incoming approaches. Vehicle arrivals follow a stochastic process with variable intensities to reflect fluctuations commonly observed during congested periods. This configuration captures the inherent randomness of traffic conditions and aims to learn control strategies that remain effective under variations in demand.

The experiments focus on ordinary vehicles without the presence of connected or prioritized vehicle types. This ensures that the learned policy is evaluated under general traffic conditions and that performance improvements can be attributed solely to the proposed signal control method.

4.4. Simulation Parameter Settings

The simulation and algorithm training are performed over 300 episodes, with each episode consisting of 400 simulation steps. Before training begins, 6000 pre-training steps are executed to populate the replay buffer with initial experience samples. Each decision step corresponds to a fixed signal duration of 10 s, followed by a 3-s yellow interval. Minimum and maximum green times are enforced for each phase to ensure safety and operational feasibility.

The discount factor γ is set to 0.99 to emphasize long-term traffic performance, consistent with the settings in Section 3. The Q-network is trained using the Adam optimizer with a learning rate of 0.001, while the policy alignment function $G(s, a; \theta)$ is trained using stochastic gradient descent with a learning rate of 0.01. These settings follow standard practices in deep reinforcement learning for traffic signal control and contribute to stable convergence.

A summary of all simulation and training parameters is presented in Table 3.

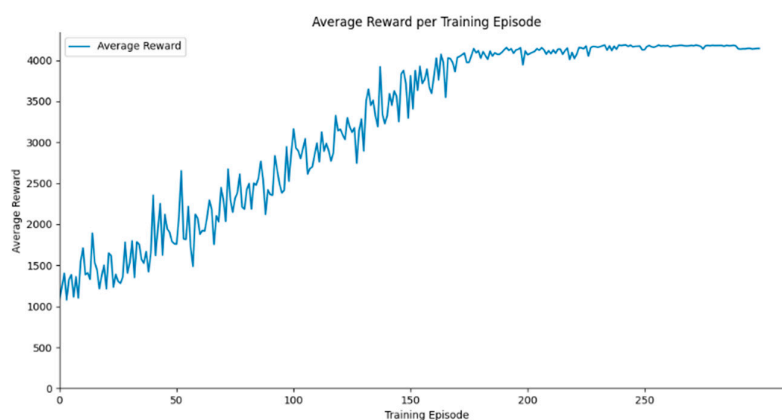
Table 3. Simulation Model Parameter Settings.

Parameter	Description	Value
Number of episodes	Total training episodes	300
Steps per episode	Simulation steps per episode	400
Pre-training steps	Steps for warm-up/memory initialization	6000
Discount factor γ	Same as Table 2	0.99
ϵ -initial	Initial exploration rate	1.0
ϵ -final	Minimum exploration rate	0.05
ϵ -decay	Linear decay steps	220
Phase duration	Decision interval	10 s
Yellow time	Yellow phase duration	3 s

4.5. Simulation Setup

To evaluate the learning performance and convergence stability of the proposed PRA-DQN algorithm, ten independent simulation runs were conducted under identical settings. During each run, the cumulative reward obtained in every training episode was recorded. After all runs were completed, the episode rewards were averaged to eliminate randomness and highlight the overall training trend.

As shown in Figure 7, the average reward exhibits a clear upward trend as training progresses, indicating continuous policy improvement. In the early stages of training, the reward fluctuated considerably due to high exploration and limited samples in the replay buffer. As more experience is collected and both the Q-network and the policy function become better optimized, the reward curve gradually stabilizes and converges toward a higher value.

**Figure 7.** Average Reward Function Curve of the PRA-DQN Algorithm.

These results demonstrate that the explicit policy–Q alignment mechanism and parameter-sharing architecture adopted by PRA-DQN effectively enhance training stability and improve learning efficiency. The stable convergence trend also indicates that the algorithm successfully learns coordinated phase decisions across intersections, reducing congestion and improving network performance.

Benchmark Algorithms

To provide a comprehensive and fair performance evaluation, PRA-DQN is compared with six representative baseline controllers. These baselines include reinforcement learning-based approaches, rule-based algorithms, and classical fixed-time coordination.

- (1) DQN-based Coordinated Signal Control (DQN-CoC): This method approximates the Q-value function using a deep neural network and employs an ϵ -greedy strategy for action selection. In coordinated mode, agents share information and jointly optimize signal timings to enhance network-wide performance.
- (2) DQN-based Independent Signal Control (DQN-IC): Similarly to DQN-CoC, this method also relies on a deep neural network and ϵ -greedy exploration. However, each agent makes decisions solely based on its own local state without sharing information, serving as a baseline to assess the benefits of coordination.
- (3) Stochastic Control (STOCHASTIC): In this approach, the signal phase is selected randomly without considering traffic state or historical patterns. It serves as a lower-bound benchmark for assessing learning-based strategies.
- (4) Max-Pressure Control (MAXPRESSURE): This rule-based controller selects the phase with the highest pressure, computed as the density difference between incoming and outgoing lanes. It aims to resolve local flow imbalance and reduce congestion.
- (5) Max-Wave Pressure Control (MAXWAVE): This algorithm estimates shockwave propagation and calculates wave-front pressure. The phase associated with the highest wave pressure is selected to mitigate spillback and delays.
- (6) Fixed-Time Control: This classical method assigns predetermined cycle lengths and green splits at all intersections. By synchronizing signal timings, it aims to create a green-wave effect and provide a reference for evaluating adaptive control methods.

These baselines collectively cover classic, stochastic, pressure-based, and Reinforcement Learning (RL)-based strategies, ensuring a comprehensive comparison with the proposed PRA-DQN controller.

4.6. Analysis of Simulation Results

In addition to lane-level performance indicators derived from the local reward function, network-level metrics are reported to evaluate system-wide effects. Specifically, average travel time is computed as the mean entry-to-exit duration of completed trips, and throughput is measured as the number of vehicles discharged from the network per unit time. These metrics help verify whether improvements obtained under local control objectives are associated with system-level benefits.

To assess the effectiveness of PRA-DQN, all six baseline algorithms described in Section Benchmark Algorithms were implemented and tested under identical traffic conditions. Three key performance indicators were used for quantitative comparison: maximum queue length, average queue length, and average waiting time.

All results in Table 4 are averaged over ten independent simulation runs with different random seeds. For each metric, mean values and corresponding standard deviations are reported to reflect performance variability across runs.

Table 4. Performance Comparison of Different Control Algorithms.

Model	Maximum Queue Length		Average Queue Length		Average Waiting Time	
	Value	Optimization Ratio	Value	Optimization Ratio	Value	Optimization Ratio
PRA-DQN-CoC	17.60	21.17%	16.30	18.75%	8.60	17.71%
DQN-CoC	19.20	13.88%	17.50	12.50%	9.30	11.43%
DQN-IC	20.10	9.87%	18.30	8.50%	9.80	6.67%

STOCHASTIC	21.50	3.59%	19.80	1.00%	10.20	2.86%
MAXWAVE	18.70	16.14%	17.10	14.50%	9.00	14.29%
MAXPRESSURE	20.00	10.31%	18.00	10.00%	9.70	7.62%
Fixed-Time	22.30	—	20.00	—	10.50	—

Table 4 summarizes the results. The performance improvement from independent DQN to coordinated DQN reflects the effect of inter-intersection coordination, whereas the additional gain from coordinated DQN to PRA-DQN indicates the contribution of the policy–Q alignment mechanism. Parameter sharing further supports consistent representation learning across intersections by exposing the shared network to diverse but structurally similar traffic states. Overall, PRA-DQN demonstrates superior performance across all evaluation metrics. Compared with Fixed-Time control, PRA-DQN reduces:

- maximum queue length by 21.17%,
- average queue length by 18.75%,
- and average waiting time by 17.71%.

These improvements highlight PRA-DQN’s ability to dynamically adjust signal phases based on real-time state observations, thereby mitigating congestion buildup.

Compared with classical DQN (DQN-IC), PRA-DQN achieves an additional 7.53% reduction in average waiting time. This improvement arises from the explicit policy function and the policy–Q alignment mechanism, which enhance action-selection stability and reduce oscillatory behavior typical in standard DQN.

Moreover, PRA-DQN outperforms both MAXPRESSURE and MAXWAVE. Although these heuristic controllers react to immediate congestion or wave-front changes, they lack long-term optimization capability. PRA-DQN, through reinforcement learning, captures both short-term conditions and long-term network effects, resulting in better overall coordination and smoother traffic operations.

These comparative results confirm that PRA-DQN effectively balances local delay, intersection pressure, and spillback mitigation, making it a robust and efficient solution for multi-intersection signal control tasks.

5. Conclusions

This study developed a cooperative multi-intersection signal control framework based on deep reinforcement learning. A comprehensive mathematical formulation was established, including the state space, action space, and reward function, which provides a rigorous foundation for the proposed control strategy. Based on this formulation, we designed a Policy-Regulated and Aligned Deep Q-Network (PRA-DQN) to alleviate learning instability and strengthen coordination in DQN-based multi-intersection signal control. The proposed method integrates explicit policy regulation, policy–Q alignment, and parameter sharing, enabling stable and interpretable phase selection across multiple intersections. In particular, the cooperative reward explicitly integrates local delay, movement pressure, and upstream–downstream interactions to better capture spillback-coupled dynamics in multi-intersection networks. SUMO-based simulations were conducted on a controlled 2×2 grid network under congested and coupled traffic dynamics. The results show that PRA-DQN consistently improves maximum queue length, average queue length, and average waiting time compared with multiple baseline controllers, including Fixed-Time, classical DQN, distributed DQN, and pressure/wave-based methods. Compared with fixed-time control, PRA-DQN reduces maximum queue length by 21.17%, average queue length by 18.75%, and average waiting time by 17.71%. Moreover, relative to classical DQN coordination, PRA-DQN achieves an additional 7.53% reduction in average waiting time, indicating improved coordination performance beyond standard

value-based learning. Overall, these results indicate that PRA-DQN can mitigate congestion and suppress spillback propagation, thereby improving multi-intersection operational efficiency in the tested setting. Methodologically, the results suggest that aligning an explicitly parameterized policy with the Q-induced target distribution can reduce action-selection noise and mitigate unstable phase switching, while incorporating upstream–downstream coupling into the reward design helps agents explicitly account for spillback effects rather than optimizing each intersection in isolation.

Several directions remain for future research. Future work will validate the proposed approach on larger-scale and non-uniform urban networks, beyond the controlled 2×2 testbed used in this study. Improving real-time performance and robustness under highly dynamic and stochastic traffic conditions is also important; robust training, adaptive reward shaping, and multi-objective formulations are promising directions. In addition, integrating cooperative perception or communication mechanisms among intersections and connected vehicles may further strengthen coordination. In practical deployments, incorporating additional context information (e.g., incident reports, temporary restrictions, or priority rules) into the state representation may further improve decision-making. Finally, extending PRA-DQN to irregular intersection geometries, heterogeneous phase structures, and mixed traffic environments is another valuable avenue for investigation.

Author Contributions: L.M.: conceptualization, formal analysis, writing—review & editing; Y.L. (Yan Liu): methodology, software, Funding acquisition, writing—original draft, writing—review & editing; Y.L. (Yang Liu): conceptualization, methodology, software, writing—original draft; C.M.: validation, visualization, writing—review and editing; S.W. writing—original draft, software. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Industry Support Plan Project of the Department of Education of Gansu Province (No. 2024CYZC-28); the Jinchang Science and Technology Program Project (No. 2025SF006); the Youth Project of Humanities and Social Sciences Research of the Ministry of Education of China (No. 25XJCZH011); the Basic Research Program of the Gansu Provincial Science and Technology Plan (No. 25JRRA221); and the Lanzhou Jiaotong University–Tianjin University Joint Innovation Fund Project (No. LH2025006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors thank the reviewers and editors for their valuable comments and efforts in improving the manuscript.

Conflicts of Interest: Author Lin Ma was employed by Linxia Daohe Investment Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. INRIX. 2025 Global Traffic Scorecard. 2025. Available online: <https://inrix.com/scorecard/> (accessed on 10 December 2025).
2. INRIX. Traffic Is Back: Insights from the 2025 INRIX Global Traffic Scorecard. 2025. Available online: <https://inrix.com/blog/traffic-is-back-insights-from-the-2025-inrix-global-traffic-scorecard> (accessed on 10 December 2025).
3. Wang, X.; Jerome, Z.; Wang, Z.; Zhang, C.; Shen, S.; Kumar, V.V.; Bai, F.; Krajewski, P.; Deneau, D.; Jawad, A.; et al. Traffic light optimization with low penetration rate vehicle trajectory data. *Nat. Commun.* **2024**, *15*, 1306.
4. Agarwal, A.; Sahu, D.; Mohata, R.; Jeengar, K.; Nautiyal, A.; Saxena, D.K. Dynamic traffic signal control for heterogeneous traffic conditions using Max Pressure and Reinforcement Learning. *Expert Syst. Appl.* **2024**, *254*, 124416.

5. Wang, L.; Zhang, G.; Yang, Q.; Han, T. An adaptive traffic signal control scheme with Proximal Policy Optimization based on deep reinforcement learning for a single intersection. *Eng. Appl. Artif. Intell.* **2025**, *149*, 110440.
6. Macioszek, E.; Kurek, A. Road traffic distribution on public holidays and workdays on selected road transport network elements. *Transp. Probl.* **2021**, *16*, 127–138.
7. Laval, J.A. Traffic Flow as a Simple Fluid: Toward a Scaling Theory of Urban Congestion. *Transp. Res. Rec.* **2024**, *2678*, 376–386.
8. Kerner, B.S. *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
9. Jha, A.; Wiesenfeld, K.; Lee, G.; Laval, J. Simple traffic model as a space-time clustering phenomenon. *Phys. Rev. E* **2025**, *112*, 054104.
10. Chen, X.; Wu, S.; Shi, C.; Huang, Y.; Yang, Y.; Ke, R.; Zhao, J. Sensing Data Supported Traffic Flow Prediction via Denoising Schemes and ANN: A Comparison. *IEEE Sensors J.* **2020**, *20*, 14317–14328.
11. Chen, X.; Li, Z.; Yang, Y.; Qi, L.; Ke, R. High-Resolution Vehicle Trajectory Extraction and Denoising from Aerial Videos. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3190–3202.
12. Shabestary, S.M.A.; Abdulhai, B. Adaptive Traffic Signal Control with Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 20021–20035. <https://doi.org/10.1109/tits.2022.3179893>.
13. Chu, K.-F.; Lam, A.Y.S.; Li, V.O.K. Traffic Signal Control Using End-to-End Off-Policy Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 7184–7195. <https://doi.org/10.1109/tits.2021.3067057>.
14. Ma, D.; Zhou, B.; Song, X.; Dai, H. A Deep Reinforcement Learning Approach to Traffic Signal Control with Temporal Traffic Pattern Mining. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11789–11800. <https://doi.org/10.1109/tits.2021.3107258>.
15. Chen, D.; Xu, T.; Ma, S.; Gao, X.; Zhao, G. Research on Intelligent Signal Timing Optimization of Signalized Intersection Based on Deep Reinforcement Learning Using Floating Car Data. *Transp. Res. Rec. J. Transp. Res. Board* **2024**, *2678*, 1126–1147. <https://doi.org/10.1177/03611981241249736>.
16. Hu, T.; Li, Z. A multi-agent deep reinforcement learning approach for traffic signal coordination. *IET Intell. Transp. Syst.* **2024**, *18*, 1428–1444. <https://doi.org/10.1049/itr2.12521>.
17. Zhang, W.; Yan, C.; Li, X.; Fang, L.; Wu, Y.-J.; Li, J. Distributed Signal Control of Arterial Corridors Using Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 178–190. <https://doi.org/10.1109/tits.2022.3216203>.
18. Park, S.; Han, E.; Park, S.; Jeong, H.; Yun, I. Deep Q-network-based traffic signal control models. *PLoS ONE* **2021**, *16*, e0256405. <https://doi.org/10.1371/journal.pone.0256405>.
19. Jiang, S.; Huang, Y.; Jafari, M.; Jalayer, M. A Distributed Multi-Agent Reinforcement Learning with Graph Decomposition Approach for Large-Scale Adaptive Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14689–14701. <https://doi.org/10.1109/tits.2021.3131596>.
20. Gu, H.; Wang, S.; Ma, X.; Jia, D.; Mao, G.; Lim, E.G.; Wong, C.P.R. Large-Scale Traffic Signal Control Using Constrained Network Partition and Adaptive Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 7619–7632. <https://doi.org/10.1109/tits.2024.3352446>.
21. Ran, Q.; Liang, C.; Liu, P. A safe lane-changing strategy for autonomous vehicles based on deep Q-networks and prioritized experience replay. *Digit. Transp. Saf.* **2025**, *4*, 170–174. <https://doi.org/10.48130/dts-0025-0013>.
22. Wang, T.; Cao, J.; Hussain, A. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103046. <https://doi.org/10.1016/j.trc.2021.103046>.
23. Wang, X.; Taitler, A.; Smirnov, I.; Sanner, S.; Abdulhai, B. eMARLIN: Distributed Coordinated Adaptive Traffic Signal Control with Topology-Embedding Propagation. *Transp. Res. Rec.* **2024**, *2678*, 189–202. <https://doi.org/10.1177/03611981231184250>.
24. Li, Y.; Pu, Z.; Liu, P.; Qian, T.; Hu, Q.; Zhang, J.; Wang, Y. Efficient predictive control strategy for mitigating the overlap of EV charging demand and residential load based on distributed renewable energy. *Renew. Energy* **2025**, *240*, 122154.
25. Yang, G.; Wen, X.; Chen, F. Multi-Agent Deep Reinforcement Learning with Graph Attention Network for Traffic Signal Control in Multiple-Intersection Urban Areas. *Transp. Res. Rec.* **2025**, *2679*, 880–898. <https://doi.org/10.1177/03611981241297979>.
26. Wang, T.; Zhu, Z.; Zhang, J.; Tian, J.; Zhang, W. A large-scale traffic signal control algorithm based on multi-layer graph deep reinforcement learning. *Transp. Res. Part C-Emerg. Technol.* **2024**, *162*, 104582. <https://doi.org/10.1016/j.trc.2024.104582>.
27. Yazdani, M.; Sarvi, M.; Bagloee, S.A.; Nassir, N.; Price, J.; Parineh, H. Intelligent vehicle pedestrian light (IVPL): A deep reinforcement learning approach for traffic signal control. *Transp. Res. Part C-Emerg. Technol.* **2023**, *149*, 103991. <https://doi.org/10.1016/j.trc.2022.103991>.

28. Yu, J.; Laharotte, P.-A.; Han, Y.; Leclercq, L. Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach. *Transp. Res. Part C-Emerg. Technol.* **2023**, *154*, 104281. <https://doi.org/10.1016/j.trc.2023.104281>.
29. Hu, W.X.; Ishihara, H.; Chen, C.; Shalaby, A.; Abdulhai, B. Deep Reinforcement Learning Two-Way Transit Signal Priority Algorithm for Optimizing Headway Adherence and Speed. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7920–7931. <https://doi.org/10.1109/tits.2023.3266461>.
30. Long, M.; Zou, X.; Zhou, Y.; Chung, E. Deep reinforcement learning for transit signal priority in a connected environment. *Transp. Res. Part C-Emerg. Technol.* **2022**, *142*, 103814. <https://doi.org/10.1016/j.trc.2022.103814>.
31. Guo, J.; Cheng, L.; Wang, S. CoTV: Cooperative Control for Traffic Light Signals and Connected Autonomous Vehicles Using Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10501–10512. <https://doi.org/10.1109/tits.2023.3276416>.
32. Song, L.; Fan, W.D. Performance of State-Shared Multiagent Deep Reinforcement Learning Controlled Signal Corridor with Platooning-Based CAVs. *J. Transp. Eng. Part A-Syst.* **2023**, *149*, 04023072. <https://doi.org/10.1061/jtepbs.Teeng-7768>.
33. Wang, L.; Zhang, W.; Yan, Z. Vehicle-Infrastructure Cooperation Framework for Vehicle Navigation and Traffic Signal Control using Deep Reinforcement Learning. *Transp. Res. Rec.* **2025**, *2680*, 568–583. <https://doi.org/10.1177/03611981251359285>.
34. Li, Y.; Zhang, H.; Zhang, Y. Traffic Signal and Autonomous Vehicle Control Model: An Integrated Control Model for Connected Autonomous Vehicles at Traffic-Conflicting Intersections Based on Deep Reinforcement Learning. *J. Transp. Eng. Part A-Syst.* **2025**, *151*, 04024107. <https://doi.org/10.1061/jtepbs.Teeng-8572>.
35. Ying, Z.; Cao, S.; Liu, X.; Ma, Z.; Ma, J.; Deng, R.H. PrivacySignal: Privacy-Preserving Traffic Signal Control for Intelligent Transportation System. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16290–16303. <https://doi.org/10.1109/tits.2022.3149600>.
36. Kumar, N.; Mittal, S.; Garg, V.; Kumar, N. Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2411–2421. <https://doi.org/10.1109/tits.2021.3095161>.
37. Zhou, S.; Chen, X.; Li, C.; Chang, W.; Wei, F.; Yang, L. Intelligent Road Network Management Supported by 6G and Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 17235–17243. <https://doi.org/10.1109/tits.2024.3451193>.
38. Yang, J.; Zhang, J.; Wang, H. Urban Traffic Control in Software Defined Internet of Things via a Multi-Agent Deep Reinforcement Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3742–3754. <https://doi.org/10.1109/tits.2020.3023788>.
39. Sun, Z.; Jia, X.; Cai, Y.; Ji, A.; Lin, X.; Liu, L.; Wang, W.; Tu, Y. Joint control of traffic signal phase sequence and timing: A deep reinforcement learning method. *Digit. Transp. Saf.* **2025**, *4*, 118–126. <https://doi.org/10.48130/dts-0025-0008>.
40. Zhu, Y.; Lv, Y.; Lin, S.; Xu, J. A Stochastic Traffic Flow Model-Based Reinforcement Learning Framework for Advanced Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 714–723. <https://doi.org/10.1109/tits.2024.3494251>.
41. Mao, F.; Li, Z.; Lin, Y.; Li, L. Mastering Arterial Traffic Signal Control with Multi-Agent Attention-Based Soft Actor-Critic Model. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3129–3144. <https://doi.org/10.1109/tits.2022.3229477>.
42. Huang, L.; Qu, X. Improving traffic signal control operations using proximal policy optimization. *IET Intell. Transp. Syst.* **2023**, *17*, 588–601. <https://doi.org/10.1049/itr2.12286>.
43. Luo, H.; Bie, Y.; Jin, S. Reinforcement Learning for Traffic Signal Control in Hybrid Action Space. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 5225–5241. <https://doi.org/10.1109/tits.2023.3344585>.
44. Wang, Z.; Yang, K.; Li, L.; Lu, Y.; Tao, Y. Traffic signal priority control based on shared experience multi-agent deep reinforcement learning. *IET Intell. Transp. Syst.* **2023**, *17*, 1363–1379. <https://doi.org/10.1049/itr2.12328>.
45. Xu, D.; Li, C.; Wang, D.; Gao, G. Robustness Analysis of Discrete State-Based Reinforcement Learning Models in Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 1727–1738. <https://doi.org/10.1109/tits.2022.3221107>.
46. Laval, J.; Zhou, H. Congested Urban Networks Tend to Be Insensitive to Signal Settings: Implications for Learning-Based Control. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24904–24917. <https://doi.org/10.1109/tits.2022.3208236>.
47. Jiang, Q.; Qin, M.; Zhang, H.; Zhang, X.; Sun, W. BlindLight: High Robustness Reinforcement Learning Method to Solve Partially Blinded Traffic Signal Control Problem. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 16625–16641. <https://doi.org/10.1109/tits.2024.3416154>.
48. Shen, J. Hierarchical reinforcement learning-based traffic signal control. *Sci. Rep.* **2025**, *15*, 32862.
49. Zhou, Y.; Liu, S.; Qing, Y.; Zheng, T.; Chen, K.; Song, J.; Song, M. CADP: Towards Better Centralized Learning for Decentralized Execution in MARL. In Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, Detroit, MI, USA, 19–23 May 2025; pp. 2838–2840.

50. Chu, T.S.; Wang, J.; Codecà, L.; Li, Z.J. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1086–1095. <https://doi.org/10.1109/tits.2019.2901791>.
51. Alonso, B.; Musolino, G.; Rindone, C.; Vitetta, A. Estimation of a fundamental diagram with heterogeneous data sources: Experimentation in the city of santander. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 418.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.