Multi-Agent Deep Reinforcement Learning with Contrastive Policy Diversification and Hierarchical Graph Networks for Urban Traffic Signal Control

Liping Yan *, Haojie Jia, Shaofeng Wang, Peiran Wu, Wenzhi Zhao

Posted Date: 31 March 2026

doi: 10.20944/preprints202603.2495.v1

Keywords: m-agent reinforcement learning; traffic signal control; unsupervised contrastive learning; credit assignment; graph convolutional network

 doi:10.20944/preprints202603.2495.v1

Article

Multi-Agent Deep Reinforcement Learning with Contrastive Policy Diversification and Hierarchical Graph Networks for Urban Traffic Signal Control

Liping Yan [1,*], Haojie Jia [1], Shaofeng Wang [2], Peiran Wu [1] and Wenzhi Zhao [1]

[1] School of Information and Software Engineering, East China Jiaotong University, Nanchang, Jiangxi 330013, China
[2] MOE Engineering Research Center of Railway Environmental Vibration and Noise, East China Jiaotong University, Nanchang, Jiangxi 330013, China
* Correspondence: yanliping@ecjtu.edu.cn (Liping Yan)

Abstract

Multi-Agent Reinforcement Learning (MARL) provides an effective approach for urban multi-intersection traffic signal control. However, existing methods have faced two fundamental challenges, policy homogenization and inefficient credit assignment. The former led to convergent agent policies that failed to adapt to heterogeneous traffic patterns, while the latter prevented agents from accurately evaluating their individual contributions to system performance. To address these issues, this paper proposes a Multi-Agent Hierarchical Contrastive Learning Traffic Signal Control (MAHCL-TSC) model. The model incorporates an unsupervised contrastive learning module that enhances the discriminative power of state representations, thereby alleviating policy homogenization. Additionally, it designs a hierarchical graph convolutional credit allocation network that leverages road network topology and functional characteristics to enable structure-aware collaborative value estimation, significantly improving the precision of credit assignment. Based on these components, a Contrastive QTRAN with Hierarchical Graph Convolution (CQTRAN-HGC) algorithm is proposed, which jointly optimizes contrastive learning loss and QTRAN constraint loss. Experiments conducted in the SUMO simulation environment on 4×4 and 6×6 grid networks demonstrate that our model outperforms mainstream baseline methods such as QTRAN, MADDPG, and MAPPO in key metrics including average queue length, waiting time, and intersection pressure, validating its effectiveness in improving control efficiency and generalization capability.

Keywords: m-agent reinforcement learning; traffic signal control; unsupervised contrastive learning; credit assignment; graph convolutional network

1. Introduction

Traffic congestion represents a persistent problem in modern urban development, leading to significant time loss, economic costs, and increased energy consumption and environmental pollution [1]. In this context, traffic signal control plays a crucial role in improving road network efficiency. With advances in artificial intelligence, reinforcement learning-based methods have shown considerable promise for signal control. However, in multi-intersection coordination scenarios, challenges such as policy convergence among agents and inefficient global reward allocation continue to limit their effectiveness in large-scale networks [2].

As a fundamental component of urban intelligent transportation systems, traffic signal control is inherently a sequential decision-making problem with high-dimensional state spaces and complex dynamics. Reinforcement learning (RL) has consequently emerged as a prominent model for traffic

 doi:10.20944/preprints202603.2495.v1

signal control [3,4], given its capacity to learn optimal policies through environmental interaction. Initial research mainly focused on single-agent RL architectures [5,6], wherein a centralized controller makes unified decisions for all intersections in a region. Although conceptually straightforward, this centralized paradigm encounters substantial computational complexity and scalability limitations in practical deployment [7,8]. With growing traffic network complexity, multi-agent reinforcement learning (MARL) has become the predominant research direction [9,10]. In this model, each intersection operates as an autonomous agent that makes decentralized decisions based on local observations. However, this distributed approach introduces two critical challenges. First, the credit assignment problem arises because all agents share a global reward signal, making it difficult for individuals to evaluate their specific contribution to system performance. Second, policy homogenization limits the system's adaptability. Conventional MARL methods typically employ identical network architectures and learning rules across all agents, resulting in convergent behavior. Such homogeneous policies cannot adequately accommodate the heterogeneity of intersection topologies, traffic patterns, and functional requirements, ultimately compromising control efficiency.

To address these challenges, this study proposes a Multi-Agent Hierarchical Contrastive Learning (MAHCL-TSC) model for traffic signal control. The regional traffic network is modeled as a graph, where intersections represent nodes and road segments form edges. To effectively capture the complex spatial dependencies within the network, we introduce a hierarchical Graph Convolutional Network (GCN). By stacking multiple GCN layers, each intersection agent can aggregate information from its multi-hop neighbors, thereby extending its perceptual field and capturing non-local traffic dynamics. This architecture not only extracts relational representations between nodes but also reduces the communication load between agents through hierarchical aggregation. Furthermore, we incorporate an unsupervised contrastive learning mechanism. By clustering and optimizing the state representations of the agents, the model's ability to discriminate between diverse traffic patterns is significantly enhanced. This mechanism fundamentally mitigates policy homogenization and fosters the development of efficient, coordinated strategies among agents. The main contributions of this work are as follows.

1) A MAHCL-TSC model under the CTDE paradigm. It integrates four core modules—control environment, data acquisition, network architecture, and contrastive learning—into an intelligent closed-loop system. This design addresses policy coordination and asynchronous decision-making in multi-intersection control.

2) A policy diversification mechanism using unsupervised contrastive learning is designed. It generates regional pseudo-labels via multimodal feature fusion and *K*-means clustering, then refines agent representations with supervised contrastive loss. This enhances the discrimination of heterogeneous traffic patterns, mitigates policy homogenization, and improves generalization in unseen scenarios.

3)A hierarchical graph convolutional credit assignment network is developed. It partitions the road network into functional regions via clustering, while GCNs hierarchically extract intra-region and global features. This explicitly models agent interactions, optimizes credit assignment in QTRAN, and strengthens the global-local reward association, boosting collaborative efficiency and scalability in large networks.

The remainder of this paper is organized as follows. Section 2 reviews related work on multi-agent reinforcement learning for traffic signal control. Section 3 introduces the MAHCL-TSC model and provides its formal formulation based on the Decentralized Partially Observable Markov Decision Process (Dec-POMDP). Section 4 details the CQTRAN-HGC algorithm, with emphasis on its two core innovative components, the Contrastive Policy Diversification Module and the Hierarchical Credit Assignment Network. Section 5 presents the experimental setup and discusses the results. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Related Work

 doi:10.20944/preprints202603.2495.v1

The evolution of traffic signal control technology has undergone a paradigm shift from static planning to dynamic optimization, and from isolated single-point intelligence to multi-agent collaboration. Research in this field aims to overcome the limitations of traditional methods—namely, poor adaptability—and the scalability constraints of single-agent approaches. The developmental trajectory can be broadly categorized into several key directions.

Early research established the foundation for applying reinforcement learning (RL) in this domain. As fixed-time plans became increasingly inadequate for dynamic traffic flows, researchers turned to RL techniques. Beginning with Mikami's [11] pioneering work that demonstrated the feasibility of RL, the subsequent deep integration of deep learning led to the emergence of deep reinforcement learning (DRL) methods [12,13], which effectively addressed the challenges of high-dimensional state spaces. Initial efforts primarily focused on optimizing control for single intersections. However, when extended to multi-intersection networks, single-agent RL methods suffered from severely degraded performance due to the curse of dimensionality in the joint action space, prompting the adoption of multi-agent deep reinforcement learning (MARL) as a necessary evolution.

To achieve effective coordination, subsequent work primarily followed two paths, information sharing and network-aware modeling. VanderPol [14] pioneered the application of DRL to multi-intersection coordination, introducing a novel reward function that integrated multiple metrics. Subsequently, Casas [15] formulated a continuous control model using deep deterministic policy gradients, while Balaji [16] proposed a distributed Q-learning model enabling real-time congestion information sharing among neighboring agents. Liang et al. [17] approached the problem from an action representation perspective, modeling it as a phase duration optimization problem. While these studies improved system scalability, they remained constrained by the limited observational capabilities of individual agents.

To address the partial observability problem, researchers developed various enhanced architectures and training schemes to improve cooperative efficiency. Chu et al. [18] extended the independent A2C algorithm by incorporating neighbor policy fingerprinting and spatial discount factors. Building on this, Lin et al. [19] designed a model accommodating heterogeneous state observations and employed generalized advantage estimation to enhance policy diversity. Ge [20] proposed a Q-value transfer mechanism facilitating value function sharing among agents. Concurrently, graph neural networks (GNNs) emerged as a powerful tool for structured modeling. Xu [21] identified pivotal nodes using the CRRank algorithm, constructed a bidirectional tripartite graph model, and implemented adaptive control. Nish et al. integrated graph convolutional networks with k-step neural fitted Q-iteration, establishing a distributed update mechanism with global state awareness and validating the effectiveness of graph-structured modeling.

Nevertheless, existing MARL methods still face two critical limitations. First, the widespread use of parameter sharing leads to policy homogenization among agents, severely restricting the system's generalization capability in heterogeneous road networks. Second, most approaches fail to explicitly incorporate the graph-structured nature of road networks during credit assignment, resulting in comparable credit being assigned to both critical and ordinary intersections, which hinders topology-aware coordination and fine-grained collaborative optimization.

To address these challenges, this paper proposes the MAHCL-TSC model. Our approach introduces structural constraints at the representation level of policy networks via unsupervised contrastive learning, encouraging policy differentiation based on regional feature clusters and thereby inducing policy heterogeneity. Simultaneously, the model employs a hierarchical graph convolutional network for credit assignment, partitioning the global road network into collaborative sub-regions and leveraging spatial dependencies within and between these sub-regions to achieve fine-grained credit distribution. The proposed CQTRAN-HGC algorithm further decouples the optimization pathways for policy learning and credit assignment, thereby enhancing generalization and stability in heterogeneous road networks.

 doi:10.20944/preprints202603.2495.v1

3. Problem Definition

3.1. MAHCL-TSC Model

Multi-intersection traffic signal control can be naturally formulated as a cooperative multi-agent decision-making problem. In this setting, each intersection controller is treated as an agent that makes signal phase decisions based on its own local traffic observations, while the overall traffic performance depends on the coordinated behavior of all agents. Since each agent has access only to partial local information, whereas the control objective is defined at the network level, the problem is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) under the Centralized Training and Decentralized Execution (CTDE) paradigm.

Formally, the multi-intersection traffic signal control problem can be characterized by a Dec-POMDP framework $\langle S, A, P, R, \Omega, O, \gamma, N \rangle$, where S denotes the global state space of the traffic network, $A = \prod_{i=1}^{N} A_i$ denotes the joint action space of all agents, $\bar{P}(s'|s,a)$ denotes the state transition probability, $R(s,a)$ denotes the global reward function, Ω denotes the local observation space, $O(o_i|s,i)$ denotes the observation function of agent i, γ is the discount factor, and N is the number of controlled intersections. Under this formulation, each agent i selects its action $a_i \in A_i$ according to its local observation $o_i \in \Omega$, while the environment evolves according to the joint action $a = (a_1, \ldots, a_N)$.

Figure 1 illustrates the overall framework of the proposed MAHCL-TSC model. Following the CTDE paradigm [22], each agent executes its policy independently based on local observations during deployment, whereas centralized information is utilized during training to improve coordination and learning efficiency. Specifically, each intersection agent perceives local traffic features, including queue lengths at incoming lanes, current signal phases, and traffic flow conditions, and then selects the corresponding signal phase action. Meanwhile, the interactions between agents and the traffic environment generate transition trajectories represented as,

$$\tau = (s_t, a_t, r_t, s_{t+1}) \quad (1)$$

where s_t denotes the global traffic state at time step t, a_t denotes the joint action of all agents, r_t denotes the global reward, and $s_t + 1$ denotes the next global state. These trajectories are stored in the replay buffer D and used for offline training of the proposed CQTRAN-HGC algorithm.

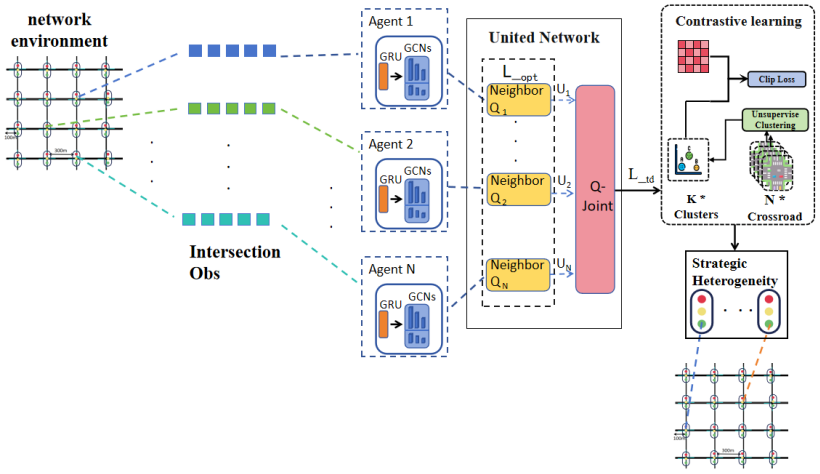


Figure 1. Framework of MAHCL-TSC Model.

The overall workflow of the MAHCL-TSC model forms a closed-loop optimization process. First, the traffic simulator generates dynamic traffic states according to road network topology and traffic flow patterns. Then, each agent collects its local observations and interacts with the environment to produce state-action-reward-state transition data. Based on these collected experiences, the proposed framework integrates three major components for cooperative signal control: a contrastive policy diversification module for enhancing representation discrimination, a

 doi:10.20944/preprints202603.2495.v1

policy network for local decision-making, and a joint value network for global credit assignment. Through the coordinated optimization of these components, the proposed method effectively alleviates policy homogenization and improves credit assignment in large-scale multi-intersection traffic signal control scenarios.

3.2. RL Parameter Configuration

The MAHCL-TSC model centers on the formulation of three fundamental components in traffic signal control (TSC), the state space, action space, and reward function. The sections below elaborate on the representation of traffic environment states and the design of the action space and reward mechanism.

State Space (S): The state space provides each agent with a multi-dimensional representation of the local traffic environment, which serves as the basis for decision-making. In this study, the state representation consists of four key components to characterize real-time traffic conditions at an intersection.

Phase State ($s_{\text{phase}} \in \{0,1\}^k$): The current traffic signal phase is represented by a one-hot encoded vector, where k denotes the number of feasible signal phases at the intersection. In this vector, the active phase is set to 1, while all other elements are set to 0.

Lane Space Occupancy ($s_{\text{lane}} \in \mathbb{R}^{M\times H}$): This component describes the fine-grained spatial occupancy of approach lanes. Here, M denotes the number of approach lanes, and H denotes the number of discretized grid cells along each lane. Each element $s_{\text{lane}}[m,h]$ represents the occupancy status of the h-th grid cell on lane m, indicating the spatial distribution of vehicles.

Lane Traffic Flow ($s_{\text{volume}} \in \mathbb{R}^M$): This vector represents the traffic flow intensity of each inbound lane. Each element $s_{\text{volume}}[m]$ denotes the traffic volume or flow rate observed on lane m within the most recent observation interval.

Lane Queue Length ($s_{queue} \in \mathbb{N}^M$): This non-negative integer vector of dimension M quantifies the queuing conditions on each approach lane. An element $s_{queue}[m]$ indicates the number of vehicles in a queued state—typically defined as those with instantaneous speed below a minimum speed threshold such as 0.1 m/s—in lane m.

Action Space (A): The action space is defined as a finite set of feasible signal phases that can be selected by each agent. For the synthetic four-phase intersection considered in this study, the action space is defined as,

$$A = \{a_1, a_2, a_3, a_4\} \quad (2)$$

where each action corresponds to a specific signal phase configuration. As illustrated in Figure 2, these four actions represent the candidate traffic signal phases for intersection control. At each decision step, agent i evaluates the action-value $Q_i(o_i, a_i)$ for all $a_i \in A$, and selects the action with the maximum estimated value for execution at the next time step. For more complex real-world road networks, the number and ordering of feasible phases may vary according to intersection geometry and traffic control requirements. In such cases, the discrete action set can be generalized as,

$$A = \{a_1, a_2, \dots, a_K\} \quad (3)$$

where K denotes the number of feasible signal phase configurations.

 doi:10.20944/preprints202603.2495.v1

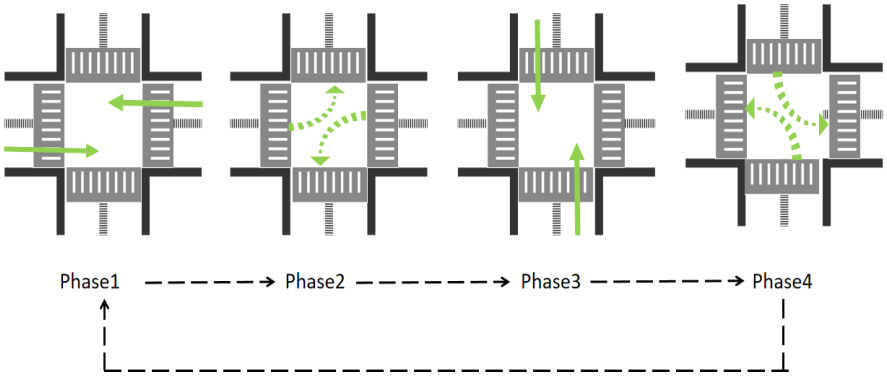


Figure 2. Schematic Diagram of a Four-Phase Intersection.

Reward Function (R): At each discrete time step t, agent i receives an immediate reward r_t from the traffic environment. To accommodate multiple traffic control objectives, we design the reward as a weighted multi-objective combination of three key factors: queue length, vehicle waiting time, and intersection pressure. Accordingly, the reward function is defined as,

$$r_t = -\left(\alpha \sum_{i=1}^{M} q_i + \beta \sum_{j=1}^{V} w_j + \gamma p_t\right) \quad (4)$$

where q_i denotes the queue length of inbound lane i, representing the number of vehicles waiting behind the stop line; w_j denotes the incremental waiting time of vehicle j at the current time step; and p_t denotes the intersection pressure at time step t, defined as the difference between the total queue length of inbound lanes and that of outbound lanes. The coefficients α, β, and γ are weighting parameters, which can be determined through Pareto optimization to balance local congestion mitigation, individual travel efficiency, and global supply-demand coordination. By introducing the negative sign, the minimization of these traffic-related costs is transformed into a reward maximization objective for reinforcement learning.

4. CQTRAN-HGC Algorithm for Multi-Intersection Traffic Signal Control

Building upon the MAHCL-TSC model established in Section 3, this section introduces the core CQTRAN-HGC algorithm, which integrates two fundamental concepts, agent network architecture design and unsupervised contrastive learning for enhanced collaborative decision-making. The agent network architecture acts as the computational core of the system, implementing hierarchical graph convolutions alongside policy-value dual networks to enable distributed decision-making and collaborative value modeling across agents. Simultaneously, the unsupervised contrastive learning model provides dynamic environment perception and feature optimization capabilities. Section 4.1 describes the policy network structure and the contrastive learning strategy. Section 4.2 presents the hierarchical graph convolutional architecture and joint value network design. Finally, Section 4.3 describes the parameter update procedure for the joint value network within the CQTRAN-HGC algorithm mode.

4.1. Contrastive Strategy Diversification Module

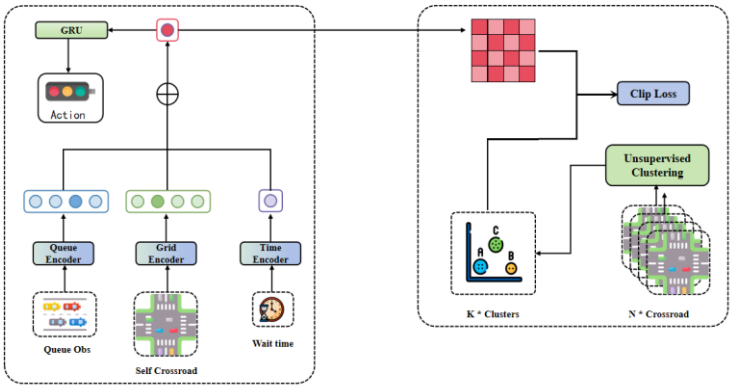
In conventional QTRAN-based methods, each agent typically learns its policy mainly from local observations, which often leads to highly similar latent representations across agents. As a result, different intersections may converge to homogeneous policies, even when they exhibit distinct spatial structures and traffic dynamics. To alleviate this problem, this study introduces a contrastive strategy diversification module, which enhances the discriminability of agent representations through multimodal feature fusion and contrastive learning. The objective of this module is to enable different agents to learn region-sensitive policy representations, thereby improving policy diversity and generalization in heterogeneous traffic environments.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2026 doi:10.20944/preprints202603.2495.v1

As shown in Figure 3(a), the proposed local policy network adopts a multimodal encoding architecture to extract heterogeneous traffic information from each intersection. Specifically, three complementary feature sources are considered: lane queue observations, bird's-eye-view spatial topology, and temporal delay information. These features are first encoded separately and then fused into a unified latent representation, which is further modeled by a recurrent unit to capture temporal dependencies. Based on the resulting latent embedding, each agent outputs its action policy, while contrastive learning is introduced to enhance the separability of latent representations corresponding to different regional traffic patterns. The lane queue encoder processes the queue observation vector $o_{queue} \in \mathbb{R}^M$ through a two-layer fully connected network,

$$h_{queue} = \mathrm{ReLU}\left(W_{q1} o_{queue} + b_{q1}\right), \quad W_{q1} \in \mathbb{R}^{M\times 128} \tag{5}$$

$$e_{queue} = \mathrm{ReLU}\left(W_{q2} h_{queue} + b_{q2}\right), \quad W_{q2} \in \mathbb{R}^{128\times 64} \tag{6}$$



(a) Policy Network Architecture. **(b)** Contrastive Learning Module Architecture.

Figure 3. Contrastive Strategy Diversification Module of the MAHCL-TSC Model.

The Bird's Eye View (BEV) encoder utilizes a ResNet-18 architecture to process spatial features $o_{bev} \in \mathbb{R}^{D\times H\times W}$ and extract topological relationships through convolutional layers,

$$e_{bev} = \mathrm{ResNet}_{\theta_2}(o_{bev}), \quad e_{bev} \in \mathbb{R}^{256} \tag{7}$$

The temporal delay encoder maps temporal statistics $o_{time} \in \mathbb{R}^2$ into a compact temporal embedding,

$$e_{time} = \mathrm{MLP}_{\theta_3}(o_{time}), \quad e_{time} \in \mathbb{R}^{64} \tag{8}$$

To obtain a unified latent representation, the encoded features are fused by linear projection followed by element-wise addition,

$$\hat{e}_{bev} = W_b e_{bev}, \quad \hat{e}_{time} = W_t e_{time}, \quad h_i = e_{queue} + \hat{e}_{bev} + \hat{e}_{time} \tag{9}$$

where W_b and W_t are learnable projection matrices used to align the feature dimensions. This formulation is consistent with the intended element-wise fusion mechanism and avoids ambiguity between summation and concatenation. The fused latent representation is then fed into a gated recurrent unit (GRU) to model temporal dependencies,

$$z_i = \sigma(W_z[h_i, h_{i-1}] + b_z) \tag{10}$$

$$\rho_i = \sigma(W_r[h_i, h_{i-1}] + b_r) \tag{11}$$

$$\tilde{h}_i = \tanh(W_h[\rho_i \odot h_{i-1}, h_i] + b_h) \tag{12}$$

$$\bar{h}_i = (1 - z_i) \odot h_{i-1} + z_i \odot \tilde{h}_i \tag{13}$$

Based on the recurrent representation $\bar{h}_i$, the policy network outputs the action probability distribution of agent i through a fully connected layer with softmax activation,

$$\pi_i(a_i \mid o_i) = \mathrm{softmax}\left(W_\pi \bar{h}_i + b_\pi\right) \tag{14}$$

To address policy homogenization among agents, we incorporate an unsupervised contrastive learning mechanism [23]. This approach groups agents' latent representations through clustering and

 doi:10.20944/preprints202603.2495.v1

utilizes the resulting cluster labels as supervisory signals to enhance the discriminative capability of the policy network's feature representations.

To further alleviate policy homogenization, a contrastive learning objective (Figure 3(b)) is imposed on the latent representations of all agents. Specifically, the set of latent embeddings $\{\bar{h}_1, \bar{h}_2, \dots, \bar{h}_N\}$ is clustered by K-means to generate pseudo-labels $y_i \in \{1, \dots, K\}$, where agents assigned to the same cluster are regarded as sharing similar regional traffic patterns. Based on these pseudo-labels, a supervised contrastive objective is used to improve the separability of latent representations,

$$\mathcal{L}_{cont} = -\sum_{i=1}^{N} \log \frac{\exp(\mathrm{sim}(\bar{h}_i, \bar{h}_i^+)/\tau)}{\sum_{j=1}^{N} \exp(\mathrm{sim}(\bar{h}_i, \bar{h}_j)/\tau)} \quad (15)$$

where $\mathrm{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ denotes cosine similarity, $\bar{h}_i^+$ denotes a positive sample belonging to the same cluster as $\bar{h}_i$, and τ is the temperature parameter. Through this pseudo-label-guided contrastive learning process, intersections with different regional traffic characteristics are encouraged to learn more discriminative latent representations, thereby improving policy diversity and adaptation capability in heterogeneous urban traffic scenarios. In addition, the clustering results are reused in the subsequent joint value learning process to organize intersections into functional regional groups. This provides a structural basis for hierarchical feature aggregation and refined credit assignment in the joint value network, which will be described in Section 4.2.

4.2. Credit Allocation Network

While QTRAN provides a theoretically grounded framework for cooperative credit assignment through the optimization terms L_{opt} and L_{nopt}, its joint state-action value estimation still faces substantial challenges in large-scale urban road networks. In particular, the fully connected structure commonly used in joint value modeling becomes computationally expensive as the number of intersections increases, and it does not explicitly exploit the heterogeneous spatial interactions induced by road topology and functional differences across intersections. To address these limitations, this study introduces a Hierarchical Credit Assignment Network (HCAN), which performs structure-aware joint value estimation through regional clustering and hierarchical graph convolution.

The proposed HCAN serves as the credit assignment component of the CQTRAN-HGC framework. Rather than directly relying on a dense global mixing structure, it organizes the traffic network into a hierarchy of regional subgraphs and learns joint value representations in a node-to-cluster-to- global manner. In this way, the network improves both the scalability of joint value learning and the precision of global-to-local credit attribution. Figure 4 illustrates the overall architecture of the proposed joint value network.

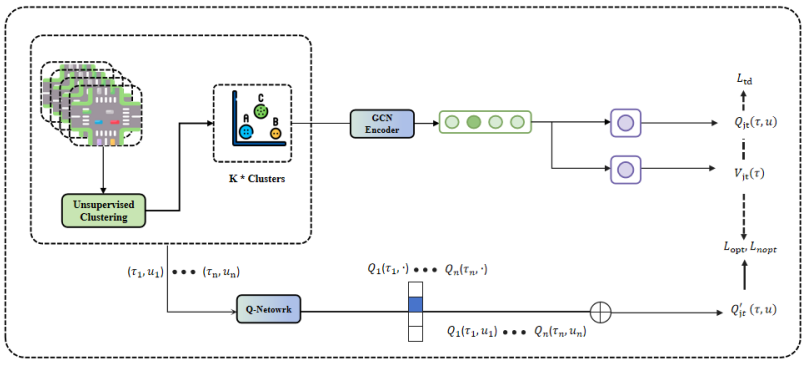


Figure 4. Joint Action Value Network Architecture.

To improve computational efficiency, the global road network is first decomposed into K functional clusters. This regional decomposition reduces the complexity of global interaction

 doi:10.20944/preprints202603.2495.v1

modeling from $O(N^2)$ to $O(\sum_{k=1}^{K}|C_k|^2)$, where C_k denotes the set of intersections in the k-th cluster. Since graph operations are then performed within each cluster independently, the proposed architecture supports parallel processing over regional subgraphs and is therefore more suitable for large-scale urban traffic control. The clustering process is performed according to road topology and traffic flow characteristics. For each intersection i, a regional feature vector is defined as,

$$x_i = [p_i, f_i] \tag{16}$$

where $p_i \in \mathbb{R}^2$ denotes the spatial coordinates of intersection i, and $f_i \in \mathbb{R}^3$ contains statistical traffic features such as mean flow, flow variance, and peak flow. Based on these feature vectors, the K-means algorithm partitions the N intersections into K functional clusters $\{C_1, \dots, C_K\}$ by minimizing the intra-cluster variance,

$$\min \sum_{k=1}^{K} \sum_{x_i \in C_k} \| x_i - \mu_k \|^2 \tag{17}$$

This clustering step groups intersections with similar geographical and traffic characteristics, thereby providing a structural basis for region-aware value learning. After clustering, each functional cluster C_k is treated as a sub-regional graph,

$$G_k = (V_k, E_k) \tag{18}$$

where V_k denotes the set of intersections within cluster k, and E_k denotes the set of intra-cluster connections. Different from conventional graph-based traffic models, the node features are not raw observations, but the multimodal latent representations $\bar{h}_i$ generated by the local policy network in Section 4.1. Since these features have already been refined by contrastive learning, they contain richer semantic information about regional traffic patterns. To model both spatial proximity and traffic interaction intensity, the weighted edge between intersections i and j is defined as,

$$w_{ij} = e^{-d_{ij}} c_{ij} \tag{19}$$

where d_{ij} denotes the Euclidean distance between intersections i and j, and c_{ij} denotes the corresponding road capacity. This formulation assigns larger edge weights to nearby intersections with stronger traffic connectivity, thereby improving the structural fidelity of the subgraph representation.

A two-layer graph convolutional network is then applied to each subgraph G_k,

$$H_k^{(1)} = \mathrm{ReLU}\left(\tilde{D}_k^{-1/2} \tilde{A}_k \tilde{D}_k^{-1/2} H_k^{(0)} W^{(0)}\right) \tag{20}$$

$$H_k^{(2)} = \tilde{D}_k^{-1/2} \tilde{A}_k \tilde{D}_k^{-1/2} H_k^{(1)} W^{(1)} \tag{21}$$

where $H_k^{(0)} \in \mathbb{R}^{|V_k| \times d}$ is the input node feature matrix, $\tilde{A}_k = A_k + I$ is the adjacency matrix with self-loops, and $\tilde{D}_k$ is the corresponding degree matrix. The output $H_k^{(2)}$ represents the region-aware node embeddings within cluster k. To obtain a compact representation for each cluster, max pooling is applied over the node embeddings,

$$g_k = \max_{i \in V_k} H_{k,i}^{(2)} \tag{22}$$

The resulting cluster-level features $\{g_1, \dots, g_K\}$ are then fused to construct the joint value estimation heads. Specifically, the joint action-value function and the joint state-value function are defined as,

$$Q_{\text{joint}}(s, a) = W_Q(\oplus_{k=1}^{K} g_k) + b_Q \tag{23}$$

$$V_{\text{joint}}(s) = W_V \left(\frac{1}{K} \sum_{k=1}^{K} g_k\right) + b_V \tag{24}$$

where $\oplus$ denotes vector concatenation. In this formulation, the concatenated cluster representations are used to model the joint action-value function, while the averaged cluster representation is used to estimate the global state value. Together, these two outputs provide the structural basis for enforcing the QTRAN consistency constraints in the subsequent optimization stage.

Through this hierarchical feature extraction process from node level to cluster level and finally to the global level, the proposed HCAN refines implicit credit assignment in QTRAN. By explicitly modeling dependencies among geographically adjacent intersections with similar traffic patterns, the network generates high-fidelity regional features that allow the joint value estimator to identify more accurately the contributions of individual intersections and regional substructures. As a result, gradients can be propagated more precisely to the corresponding policy networks, which improves

 doi:10.20944/preprints202603.2495.v1

coordination efficiency and alleviates the scalability bottleneck of fully connected joint value modeling in large-scale traffic control scenarios.

4.3. CQTRAN-HGC Algorithm

Building upon the contrastive strategy diversification module and the hierarchical credit assignment network introduced in Sections 4.1 and 4.2, this section presents the overall optimization procedure of the proposed CQTRAN-HGC algorithm. Within the proposed framework, the local policy network generates agent-specific action decisions from local observations, while the hierarchical joint value network estimates the global cooperative value through region-aware graph representations. The interactions between agents and the traffic environment are stored in an experience replay buffer and used to jointly optimize the QTRAN [24] objective and the contrastive learning objective. Let a transition sample be denoted by (s, u, r, s'), where s and s' are the current and next global states, $u = (u_1, \ldots, u_N)$ is the joint action of all agents, and r is the global reward. Based on the QTRAN framework, the overall objective of CQTRAN-HGC is formulated as,

$$L_{\text{total}} = L_{\text{QTRAN}}(s, u, r, s'; \theta) + \lambda_{\text{cont}} L_{\text{cont}} \tag{25}$$

where L_{QTRAN} denotes the value decomposition loss derived from QTRAN, L_{cont} denotes the contrastive learning loss introduced in Section 4.1, and λ_{cont} is the balancing coefficient. The QTRAN loss consists of three components,

$$L_{\text{QTRAN}}(s, u, r, s'; \theta) = L_{\text{td}} + \lambda_{\text{opt}} L_{\text{opt}} + \lambda_{\text{nopt}} L_{\text{nopt}} \tag{26}$$

where L_{td} is the temporal-difference loss, L_{opt} is the optimality constraint loss, and L_{nopt} is the non-optimality constraint loss. The temporal-difference loss is used to train the joint action-value network,

$$L_{td} = \left(Q_{\text{joint}}(s, u) - y^{dq_n}\right)^2 \tag{27}$$

where the target value is defined as,

$$y^{dq_n} = r + \gamma Q_{\text{joint}}(s', \bar{u}; \theta^-) \tag{28}$$

and $\bar{u}$ denotes the greedy action selected by the target network. The optimality constraint loss ensures the consistency between the transformed joint action-value and the jointly optimal value,

$$L_{opt} = \left(\hat{Q}_{\text{joint}}(s, u) - \bar{Q}_{\text{joint}}(s, \bar{u}) + V_{\text{joint}}(s)\right)^2 \tag{29}$$

where $\bar{Q}_{\text{joint}}(s, \bar{u})$ is the fixed target estimate of the joint action-value, and $V_{\text{joint}}(s)$ is the global state-value function produced by the hierarchical credit assignment network.

The non-optimality constraint loss is defined as,

$$L_{nopt} = \left(\min\left[Q_{\text{joint}}(s, u) - \hat{Q}_{\text{joint}}(s, u) + V_{\text{joint}}(s), 0\right]\right)^2 \tag{30}$$

This term constrains non-optimal joint actions to satisfy the decomposition condition required by QTRAN, thereby improving the consistency between local greedy actions and the global cooperative objective. To further enhance representation learning, each agent produces a latent embedding $\bar{h}_i$ through the local policy network. The set of latent representations $\{\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_N\}$ is clustered by K-means to generate pseudo-labels y_i, and the contrastive learning loss is computed as,

$$L_{cont} = -\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(\bar{h}_i, \bar{h}_i^+)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\bar{h}_i, \bar{h}_j)/\tau)} \tag{31}$$

where $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ denotes cosine similarity, $\bar{h}_i^+$ denotes a positive sample belonging to the same cluster as $\bar{h}_i$, and τ is the temperature parameter. By minimizing L_{cont}, the policy network learns more discriminative latent representations, which improves policy diversity and stabilizes collaborative learning in heterogeneous traffic environments.

The training procedure of CQTRAN-HGC is summarized as follows. First, the replay memory D is initialized to store interaction trajectories, and the parameters of the local policy network, the hierarchical joint value network, and the target network are randomly initialized. During each episode, each agent selects its action according to an ϵ-greedy strategy. After executing the joint action, the environment returns the next state and the global reward, and the transition (s, u, r, s') is stored in the replay buffer. Mini-batches are then sampled from D to compute L_{td}, L_{opt}, L_{nopt}, and

 doi:10.20944/preprints202603.2495.v1

L_{total} is minimized to update the network parameters. To improve training stability, the target network parameters θ^- are periodically synchronized with the online network parameters.

The pseudo code of the CQTRAN-HGC algorithm is described as follows.

Algorithm 1: *Training Procedure of CQTRAN-HGC*

Input: *Agent set* N, *replay buffer* D, *discount factor* γ, *batch size* B, *exploration rate* ε, *loss weights* λ_{opt}, λ_{nopt}, *and* λ_{cont}.

Output: *trained network parameters* θ.

Initialize replay memory D
Initialize online network parameters θ.
Initialize target network parameters $\theta^- = \theta$.
For *episode = 1 to* M ***do:***
  Observe the initial state s *and local observations* $\{o_i\}$ *for all agents.*
  For $t = 1$ *to* T ***do:***
    With probability ε, *each agent selects a random action* u_i.
    Otherwise, each agent selects the greedy action according to its policy.
    Execute the joint action $u = (u_1, u_2, \ldots, u_N)$.
    Observe the reward r, *next state* s', *and next observations* $\{o'_i\}$.
    Store transition (s, u, r, s') *in* D.
    Sample a mini-batch of transitions from D.
    Compute latent representations $\{\bar{h}_i\}$ *for all agents.*
    Apply K-means to $\{\bar{h}_i\}$ *and generate pseudo-labels* y_i.
    Compute the contrastive loss L_{cont}.
    Compute the target value y^{dqn}.
    Compute the temporal-difference loss L_{td}.
    Compute the optimality loss L_{opt}.
    Compute the non-optimality loss L_{nopt}.
    Compute the total loss:

$$L_{total} = L_{td} + \lambda_{opt} L_{opt} + \lambda_{nopt} L_{nopt} + \lambda_{cont} L_{cont}.$$

    Update θ *by minimizing* L_{total}.
    Periodically update the target network parameters θ^-.
    Set $s = s'$.
  End for
End for

5. Experiments

To evaluate the effectiveness of the proposed MAHCL-TSC model and CQTRAN-HGC algorithm, we established a high-fidelity traffic simulation environment using SUMO (Simulation of Urban Mobility) [25] and developed a systematic experimental protocol. The experimental design incorporates a classic Manhattan grid network topology with two distinct configurations, a 4×4 grid layout comprising 16 intersections (with 10 agent-controlled nodes) for medium-scale network validation, and a 6×6 grid layout containing 36 intersections (with 20 agent-controlled nodes) for large-scale network scalability assessment. This hierarchical experimental structure enables comprehensive evaluation of the model's control performance, coordination efficiency, and system stability across varying traffic density conditions, from conventional to high-density scenarios.

5.1. Comparative Benchmarks

To comprehensively evaluate the performance of the proposed MAHCL-TSC model and CQTRAN-HGC algorithm, we selected three representative multi-agent reinforcement learning algorithms as baseline comparisons, QTRAN, MADDPG [26], and MAPPO [27]. All baseline methods

 doi:10.20944/preprints202603.2495.v1

were trained under identical experimental conditions, including the same state spaces and reward functions, to ensure a fair comparison. The QTRAN algorithm follows a value function decomposition approach, whose core strength lies in its ability to accurately decompose the joint action-value function into individual agents' local value functions under provable constraints, thereby theoretically ensuring consistency between individual and global optima. MADDPG builds upon the actor-critic model and is distinguished by its centralized-training-with-decentralized-execution architecture. The algorithm uses a centralized critic network that accesses state-action information from all agents for global value estimation, while each agent maintains its own actor network for distributed decision-making. This design enables full utilization of global information during training while preserving execution-time independence. MAPPO extends proximal policy optimization to multi-agent settings, combining a centralized value function with trust region optimization. It coordinates policy updates across agents via the centralized value function, while the clipping mechanism from proximal policy optimization ensures training stability and mitigates policy degradation in multi-agent coordination.

This study adopts a systematic hyperparameter configuration to ensure model stability and reproducibility. Key hyperparameters include, a policy network hidden layer dimension of 256 and hidden dimension of 128 to balance representational capacity and computational efficiency; a learning rate of 0.003, discount factor $\gamma = 0.99$, target network update parameter $\tau = 0.01$, and a two-layer GCN structure with 64 hidden units per layer to effectively capture spatial dependencies in the road network. Detailed parameter settings are summarized in Table 1, with selections following common practices in deep reinforcement learning and preliminary experimental validation.

Table 1. Algorithm Parameter Table.

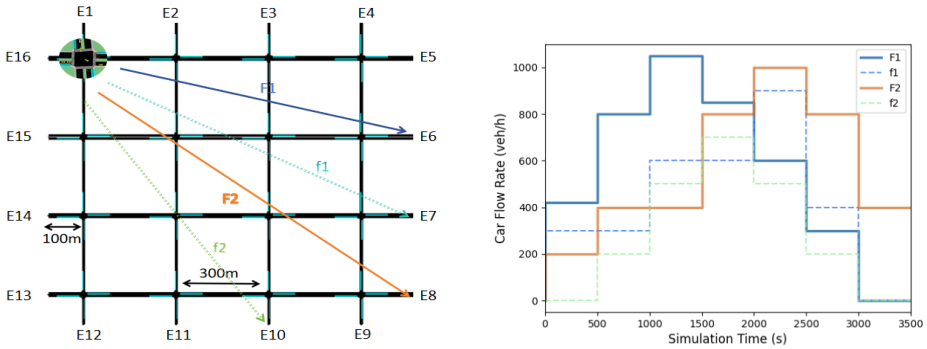
Parameter Category	CQTRAN-HGC Parameter Value
Policy Network Hidden Dimension	128
Value Network Hidden Dimension	256
Learning Rate	3×10^{-4}
Discount Factor (γ)	0.99
Target Network Update (τ)	0.01
Replay Buffer Size	1×10^{6}
Batch Size	512
Exploration Noise Variance	0.1
Number of Clusters (K)	3 (10 agents)/ 6 (20 agents)
Contrastive Loss Weight (λ)	0.3
GCN Layers	2
GCN Hidden Units	128

5.2.4×4. Synthetic Grid Network

Figure 5(a) illustrates the 4×4 grid traffic network used in this study, consisting of 16 signal-controlled intersections. Each intersection is configured with six approach lanes, the east-west arterial roads are designed as four-lane bidirectional roadways with a speed limit of 70 km/h, while the north-south roads are two-lane bidirectional roadways with a speed limit of 40 km/h. To simulate realistic traffic flow patterns, four main vehicle routes were established, Path 1 (F1) comprises traffic from E16 to E6 (represented by blue lines in Figure 5(a)), Path 2 (f1) covers traffic from E16 to E7 (Light Blue lines), Path 3 (F2) includes traffic from E16 to E8 (Orange lines), and Path 4 (f2) contains traffic from E16 to E10 (Light green lines). Fifteen minutes after the simulation begins, the traffic volumes on Paths 1 and 2 gradually decrease, while Paths 3 and 4—begin to generate traffic flow. Figure 5(b)

 doi:10.20944/preprints202603.2495.v1

details the dynamic evolution of these four traffic flow types throughout the simulation cycle, capturing the generation, dissipation, and temporal variation patterns of each route. This configuration effectively replicates the spatiotemporal distribution characteristics of traffic demand in real urban road networks, providing a reliable test environment for evaluating the performance of multi-agent cooperative control algorithms.



(a) 4×4 simulated road network structure. **(b)** Traffic flow in the simulated road network.

Figure 5. 4×4 Simulation Network Experiment Setup.

In the training performance analysis, Figure 6 presents the training curves of three multi-agent reinforcement learning algorithms and MAHCL-TSC model, all trained for 200,000 steps on the same 4×4 grid network. The solid lines indicate the moving average of the mean reward, while the shaded areas represent the standard deviation ranges. Generally, as training progresses, the agents gradually improve their policies through accumulated experience, reflected in a consistent increase in mean reward values. Specifically, the MAPPO algorithm demonstrates rapid convergence during the initial training phase but reaches a performance plateau after approximately 80,000 steps. Its final performance is limited by the fully connected network architecture's inability to effectively model complex spatial relationships. The MADDPG algorithm shows substantial training instability due to its inherent challenges in adapting to discrete action spaces, exhibiting notably higher variance in training rewards compared to other methods. The QTRAN algorithm maintains relatively stable training progress through its constrained optimization mechanism, though with a comparatively slower convergence rate. In contrast, the proposed MAHCL-TSC model demonstrates rapid reward improvement during early training and sustains a stable upward trajectory throughout the entire training process.

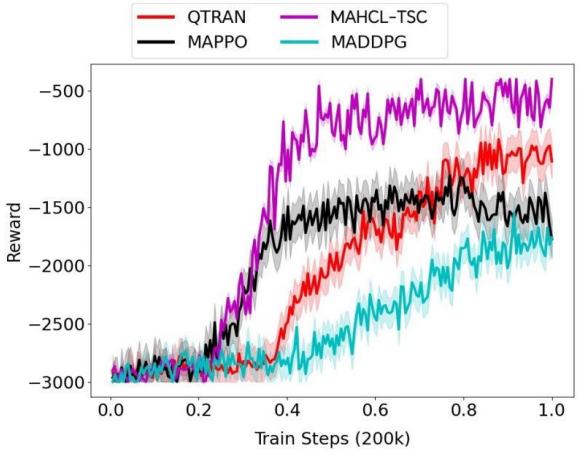


Figure 6. Training Rewards on 4×4 Simulated Road Network.

 doi:10.20944/preprints202603.2495.v1

The average queue length is calculated by dividing the total number of queuing vehicles across all intersection approaches by the number of intersections, providing an intuitive measure of overall network congestion. Figure 7 shows the evolution of average queue lengths over the simulation period for four control methods—three baseline algorithms (QTRAN, MADDPG, and MAPPO) and the proposed MAHCL-TSC model—in the 4×4 grid network. As the simulation progresses, the network experiences significantly increased load when traffic flows (Paths 3 and 4) begin operating after 15 minutes. Under these conditions, all three baseline reinforcement learning methods exhibit continuously growing queue lengths, indicating their limited adaptability to dynamically changing traffic demand. By contrast, the proposed MAHCL-TSC model maintains the lowest queue levels throughout the simulation period, demonstrating particular effectiveness in stabilizing queue lengths during high-load phases.

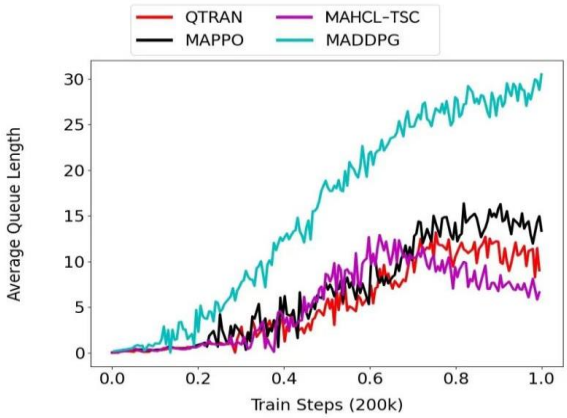


Figure 7. Average Queue Length Variation.

To comprehensively evaluate the overall performance of various signal control methods in practical traffic efficiency, a multidimensional analysis of four approaches is performed using vehicle trajectory data. Table 2 summarizes the values of key metrics—including average delay (seconds), average waiting time (seconds), and intersection pressure—for each method in the 4×4 synthetic road network scenario.

Table 2. Experimental Results for 4×4 Network.

Metric	*MADDPG*	*MAPPO*	*QTRAN*	*MAHCL-TSC*
Average Delay (s)	*109.0 ± 3.10*	*97.50 ± 1.30*	*85.40 ± 2.10*	*62.30 ± 1.30*
Average Wait (s)	*3.79 ± 0.11*	*2.93 ± 0.22*	*2.69 ± 0.18*	*2.11 ± 0.09*
Intersection Pressure	*8.85 ± 0.41*	*7.19 ± 0.33*	*5.35 ± 0.12*	*3.87 ± 0.07*

The experimental results indicate that the MADDPG algorithm fails to effectively capture complex spatial dependencies within its fully connected network structure, leading to moderate performance across all evaluation metrics. The MAPPO algorithm achieves reasonable performance in average waiting time; however, its adaptation mechanism for discrete action spaces results in decision instability, causing considerable fluctuations in average queue length measurements. While the QTRAN algorithm attains certain advantages in average delay metrics through its stable policy optimization process, it still exhibits deficiencies in handling dynamic traffic flow variations during peak periods. In comparison, the proposed MAHCL-TSC model achieves optimal results across all evaluation metrics, demonstrating particularly significant improvements in average waiting time and intersection pressure. This superior performance originates from the model's unsupervised contrastive learning mechanism, which enables effective discrimination between different traffic

 doi:10.20944/preprints202603.2495.v1

patterns, combined with the precise credit assignment facilitated by the hierarchical graph convolutional network.

5.3.6×6. Synthetic Road Network

To verify the scalability and generalization capability of the proposed model in large-scale traffic networks, extended experiments were conducted on a 6×6 Manhattan grid network, as shown in Figure 8. The network contains 36 signal-controlled intersections, each using a standard four-phase scheme. Both east-west and north-south directions are configured with dual four-lane roads with speed limits of 70 km/h and 50 km/h, respectively. Through this carefully designed extension experiment, we primarily assess the control performance and stability of the MAHCL-TSC model under conditions of higher intersection density and more complex traffic flow interactions. Twenty minutes after the simulation begins, the traffic flow in each path undergoes dynamic adjustments according to preset patterns, simulating the spatiotemporal evolution of travel demand during the evening commute peak.

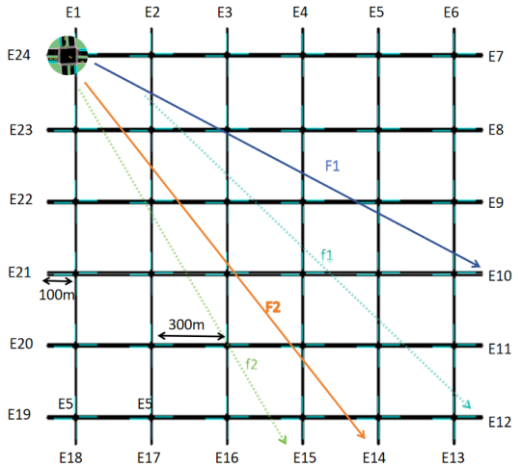


Figure 8. 6×6 Simulation Network.

In the 6×6 large-scale network environment, the training processes and convergence characteristics of each algorithm are clearly reflected in their reward function curves, as shown in Figure 9. It can be observed that with the significant expansion of state and action spaces, all baseline algorithms face varying degrees of training challenges. The QTRAN algorithm exhibits considerable fluctuations in its reward curve during later stages, primarily due to imprecise credit assignment; MADDPG shows the slowest convergence speed, affected by environmental non-stationarity; while MAPPO maintains better stability, though its final converged reward value remains at a relatively low level. In stark contrast, the proposed MAHCL-TSC model demonstrates remarkable scalability and learning efficiency. Its reward curve not only rises at a significantly faster rate during early training—indicating the algorithm's capability to quickly identify high-performance policy directions—but also maintains steady growth throughout the entire training cycle, eventually stabilizing at a reward level substantially higher than other benchmark methods.

 doi:10.20944/preprints202603.2495.v1

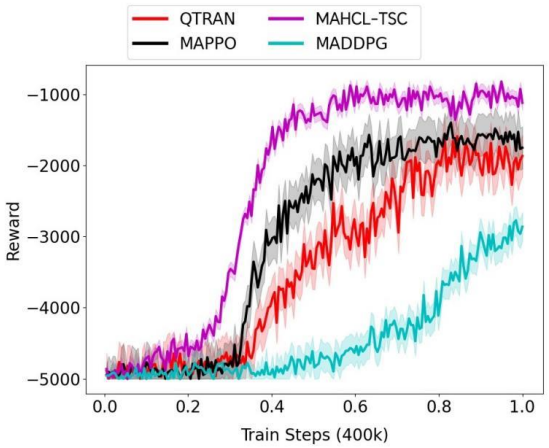


Figure 9. Training Rewards in 6×6 Simulated Road Network.

In the 6×6 road network environment, the scalability and stability of the MAHCL-TSC model were further validated. Figure 10 shows the trends in average queue length for each method under these complex conditions. As network scale increases and traffic patterns become more complex, the control performance of baseline methods—including QTRAN, MADDPG, and MAPPO—degrades to varying degrees. Limited by its fully connected structure in capturing complex spatial dependencies, MADDPG shows a rapid rise in queue length during mid-simulation. MAPPO displays considerable fluctuations in control performance due to its challenges in adapting to discrete action spaces. Although QTRAN maintains relative stability, it still underperforms when handling dynamic traffic flows during peak periods. In comparison, MAHCL-TSC model demonstrates excellent scalability, maintaining the lowest queue levels throughout the simulation while showing consistent advantages in handling peak traffic conditions.

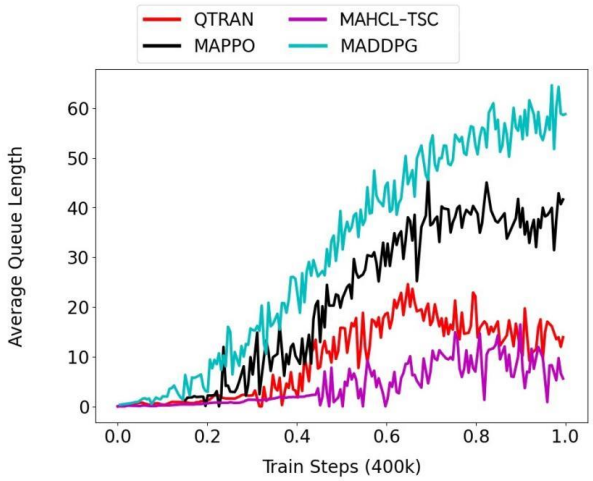


Figure 10. Average Queue Length in 6×6 Simulation Network.

The quantitative results in Table 3 further confirm these advantages. In the 6×6 network, MAHCL-TSC model outperformed all baseline methods across key metrics, including average queue length and intersection pressure. Specifically, it reduced average waiting time by 18.1% compared to MAPPO and by 28.7% compared to QTRAN. These results demonstrate that through the coordinated operation of hierarchical credit assignment and contrastive policy diversification, MAHCL-TSC model effectively addresses the credit assignment challenges in large-scale networks. This demonstrates that through the synergistic combination of hierarchical credit assignment and contrastive policy diversification, MAHCL-TSC model can effectively address the credit assignment challenges posed by large-scale road networks. It achieves global cooperative optimization while

 doi:10.20944/preprints202603.2495.v1

maintaining the policy specificity of individual agents, thereby providing an effective solution for signal control in complex urban traffic scenarios.

Table 3. Experimental Results on 6×6 Simulation Network.

Metric	*MADDPG*	*MAPPO*	*QTRAN*	*MAHCL-TSC*
Average Delay (s)	*189.9 ± 4.30*	*108.30 ± 2.70*	*116.90 ± 3.50*	*74.30 ± 1.80*
Avg. Wait (s)	*6.51 ± 0.50*	*3.33 ± 0.28*	*3.81 ± 0.32*	*2.72 ± 0.11*
Intersection Pressure	*12.11 ± 0.32*	*6.56 ± 0.31*	*7.16 ± 0.20*	*4.86 ± 0.13*

The MAHCL-TSC model demonstrates notable scalability advantages in the following aspects. In terms of control effectiveness, as presented in Tables 2 and 3, MAHCL-TSC model maintains superior performance over other multi-agent reinforcement learning methods on key metrics such as average queue length and intersection pressure as the road network scales from a 4×4 grid (16 intersections) to a 6×6 grid (36 intersections), with the performance advantage becoming more pronounced with increasing network size.

Regarding training efficiency, we compare the computational time required by each method to complete an equal number of training steps (1×10^6) across different network scales. MAHCL-TSC model achieves training efficiency comparable to MAPPO while substantially reducing training time compared to MADDPG and QTRAN. This efficiency primarily originates from its hierarchical graph convolutional architecture—through cluster-based region partitioning, the global computational complexity is reduced from $O(N^2)$ to $O(\sum_{k=1}^{K}|C_k|^2)$, allowing most graph convolution operations to execute in parallel within individual clusters.

6. Conclusion and Outlook

To address the core challenges in multi-intersection cooperative control—particularly policy homogenization and inefficient credit assignment, this paper proposes a Multi-Agent Hierarchical Contrastive Learning Traffic Signal Control (MAHCL-TSC) model. It incorporates an unsupervised contrastive learning mechanism to enhance the diversity of agent state representations, effectively mitigating policy homogenization. A hierarchical graph convolutional credit assignment network is designed to explicitly model road network topology, thereby optimizing the distribution of global rewards to local agents. Based on these components, a CQTRAN-HGC algorithm is proposed, which jointly optimizes contrastive learning loss and QTRAN constraint loss. Comprehensive experiments demonstrated that the proposed MAHCL-TSC model achieved significant improvements in key performance metrics, compared to the baseline method such as QTRAN, MADDPG, and MAPPO, which validates the model's advantages in control performance and generalization capability.

In spite of the work achieved in this paper, future research could be explored in the following aspects. Firstly, introducing more advanced learning mechanisms and designing progressive training environments and tasks that escalate from simple to complex scenarios. Secondly, incorporating external disturbance factors such as weather conditions, pedestrian activity, and special events into the state space would help establish a more comprehensive environment perception and decision-making system. This will significantly enhance model robustness and practical applicability in real-world traffic environments.

Author Contributions: Conceptualization, L.Y. and H.J.; methodology, L.Y.; software, W.Z.; validation, P.W. and S.W.; resources, L.Y.; data curation, H.J.; writing—original draft preparation, H.J.; writing—review and editing, H.J.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grants No. 62362031, 52268066 and 62262022) and the Jiangxi Provincial Natural Science Foundation (Grants No. 20252BAC240353).

 doi:10.20944/preprints202603.2495.v1

Data Availability Statement: The datasets analyzed during the current study are available in the.

OpenStreetMap repository: https://www.openstreetmap.org/ (accessed on 15 March 2026). simulation network and traffic flow settings are available from the corresponding author upon reasonable request.

Acknowledgments: The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MAHCL-TSC	*Multi-Agent Hierarchical Contrastive Learning Traffic Signal Control*
CQTRAN-HGC	*Contrastive QTRAN with Hierarchical Graph Convolution*
SUMO	*Simulation of Urban Mobility*
GCN	*Graph Convolutional Network*
MARL	*Multi-Agent Reinforcement Learning*
Dec-POMDP	*Decentralized Partially Observable Markov Decision Process*
CTDE	*Centralized Training with Decentralized Execution*
DRL	*Deep Reinforcement Learning*

References

1. Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Abad, Z.S.H.; Bazzan, A.L.; Far, B. Reinforcement learning in urban network traffic signal control: A systematic literature review. Expert Syst. Appl. 2022, 199, 116830.
2. Wei, H.; Zheng, G.; Gayah, V.; Li, Z. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. ACM SIGKDD Explor. Newsl. 2021, 22, 12–18
3. B. Abdulhai, R. Pringle, and G. J. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” J. Transp. Eng., vol. 129, no. 3, pp. 278–285, May 2003.
4. E. Camponogara and W. Kraus, “Distributed learning agents in urban traffic control,” in Proceedings of the Program on Artificial Intelligence, F. M. Pires and S. Abreu, Eds. Berlin, Germany: Springer, 2003, pp. 324–335.
5. K. Wen, S. Qu, and Y. Zhang, “A stochastic adaptive control model for isolated intersections,” in Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), December 2007, pp. 2256–2260.
6. L. Shoufeng, L. Ximin, and D. Shiqiang, "Q-learning for adaptive traffic signal control based on delay minimization strategy," in Proceedings of the IEEE International Conference on Networks, Sensors, and Control, Apr. 2008, pp. 687–691.
7. M. Wiering, “Multi-agent reinforcement learning for traffic light control,” in Proceedings of the 17th International Conference on Machine Learning (ICML), 2000, pp. 1151–1158.
8. M. Steingrover et al., "Reinforcement learning of traffic light controllers adapting to traffic congestion," in Proceedings of BNAIC, 2005, pp. 216–223.
9. T. Brys, T. T. Pham, and M. E. Taylor, "Distributed learning and multi-objectivity in traffic light control," Connection Sci., vol. 26, no. 1, pp. 65–83, Jan. 2014.
10. M. E. Taylor, M. Jain, P. Tandon, M. Yokoo, and M. Tambe, “Distributed on-line multi-agent optimization under uncertainty: Balancing exploration and exploitation,” Adv. Complex Syst., vol. 14, no. 03, pp. 471–528, Jun. 2011.
11. Mikami, S.; Kakazu, Y. Genetic reinforcement learning for cooperative traffic signal control. In Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, June 27–29, 1994; pp. 223–228.
12. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. IEEE Signal Process. Mag. 2017, 34, 26–38.

 doi:10.20944/preprints202603.2495.v1

13. M. Abdoos, N. Mozayani, and A. L. C. Bazzan, “Hierarchical control of traffic signals using Q-learning with tile coding,” Int. J. Speech Technol., vol. 40, no. 2, pp. 201–213, Mar. 2014
14. E. Van der Pol and F. A. Oliehoek, “Coordinated deep reinforcement learners for traffic light control,” in Proceedings of the International Conference on Learning, Inference, and Control of Multi-Agent Systems (NIPS), 2016.
15. N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, arXiv:1703.09035. [Online]. Available: http://arxiv.org/abs/1703.09035
16. P. G. Balaji, X. German, and D. Srinivasan, “Urban traffic signal control using reinforcement learning agents,” IET Intell. Transp. Syst., vol. 4, no. 3, pp. 177–188, Sep. 2010.
17. X. Liang, X. Du, G. Wang, and Z. Han, “A deep reinforcement learning network for traffic light cycle control,” IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.
18. T. Chu, J. Wang, L. Codecà, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.
19. Y. Lin, X. Dai, L. Li, and F.-Y. Wang, “An efficient deep reinforcement learning model for urban traffic control,” 2018, arXiv:1808.01876. [Online]. Available: http://arxiv.org/abs/1808.01876
20. H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, “Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control,” IEEE Access, vol. 7, pp. 40797–40809, 2019.
21. M. Xu, J. Wu, L. Huang, R. Zhou, T. Wang, and D. Hu, "Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning," Journal of Intelligent Transportation Systems, vol. 24, no. 1, pp. 1–10, Jan. 2020.
22. Sunehag P, Lever G, Gruslys A, Czarnecki WM, Zambaldi V, Jaderberg M, Lanctot M, Sonnerat N, Leibojz, Tuyls K et al. (2017) Value-decomposition networks for cooperative multi-agent learning. arXiv:1706.05296.
23. He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-973
24. Son K. Learning to factorize with regularization for cooperative multi-agent reinforcement learning[J]. 2019.
25. Lopez PA, Behrisch M, Bieker-Walz L, Erdmann J, Flotteröd Y-P, Hilbrich R, Lücken L, Rummel J, Wagner P, Wießner E (2018) Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 2575–2582.
26. Lowe R, Wu Y I, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in neural information processing systems, 2017, 30.
27. Yu C, Velu A, Vinitsky E, et al. The surprising effectiveness of PPO in cooperative multi-agent games[J]. Advances in Neural Information Processing Systems, 2022, 35: 24611-24624.