Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

CCGN: Centralized collaborative graphical transformer multi-agent reinforcement learning for multi-intersection signal free-corridor

Hamza Mukhtar [a,b], Adil Afzal [a,b,*], Sultan Alahmari [c], Saud Yonbawi [d]

[a] XeroAI, G.T. Road, Lahore, 54890, Punjab, Pakistan
[b] University of Engineering and Technology (UET), Lahore, GT, Road, Lahore, 54890, Punjab, Pakistan
[c] King Abdul Aziz City for Science and Technology, Riyadh, 11442, Kingdom of Saudi Arabia
[d] Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, 21959, Kingdom of Saudi Arabia

ARTICLE INFO

Article history:
Received 5 September 2022
Received in revised form 1 June 2023
Accepted 17 July 2023
Available online 26 July 2023

Keywords:
Collaborative intersection signal control
Graph convolutional network
Multi-agent centralized reinforcement learning
Cooperative traffic signal control
Markov decision processes
Intelligent transportation

ABSTRACT

Tackling traffic signal control through multi-agent reinforcement learning is a widely-employed approach. However, current state-of-the-art models have drawbacks: intersections optimize their own local rewards and cause traffic to waste time and fuel with a start-stop mode at each intersection. They also lack information sharing among intersections and their specialized policy hinders the ability to adapt to new traffic scenarios. To overcome these limitations, This work presents a centralized collaborative graph network (CCGN) with the core objective of a signal-free corridor once the traffic flows have waited at the entry intersection of the traffic intersection network on either side, the subsequent intersection gives the open signal as the traffic flows arrive. CCGN combines local policy networks (LPN) and global policy networks, where LPN employed at each intersection predicts actions based on Transformer and Graph Convolutional Network (GCN). In contrast, GPN is based on GCN and Q-network that receives the LPN states, traffic flow and road information to manage intersections to provide a signal-free corridor. We developed the Deep Graph Convolution Q-Network (DGCQ) by combining Deep Q-Network (DQN) and GCN to achieve a signal-free corridor. DGCQ leverages GCN's intersection collaboration and DQN's information aggregation for traffic control decisions Proposed CCGN model is trained on the robust synthetic traffic network and evaluated on the real-world traffic networks that outperform the other state-of-the-art models.

1. Introduction

Traffic congestion creates various social, environmental and economic challenges, such as travel delay, greenhouse gas emissions and fuel consumption (Rahman, Najaf, Fields, & Thill, 2022; Wen et al., 2020). Traffic signal automation with Adaptive Traffic Signal Control (TSC) has been showing promising results for alleviating the congestion problem (Wang, Cao, & Hussain, 2021). Various interdisciplinary techniques, such as fuzzy logic (Cheng, Wu, Cao, & Li, 2016), genetic algorithms (Odeh, Mora, Moreno, & Merelo, 2015), Immune network (Darmoul, Elkosantini, Louati, & Said, 2017) and neural network (Ghanim & Abu-Lebdeh, 2015), have been applied to enhance the performance of TSC. However, Reinforcement learning (RL) inspired TSC systems are equipped to analyze the dynamic and real-time traffic situation for better management 2017. The TSC model based on the single RL agent controls the isolated intersection where the agent interacts with the environment in a Markov Decision Process (MDP). Mapping of traffic states (e.g., total delay, waiting time and queue length), to corresponding actions (e.g., green time change, signal light shift, cycle length shift, etc.) creates the control policy. Until optimal convergence, RL agent iteratively interacts with the environment and takes optimal action in order to adjust the policy controller network (PCN) for reward maximization. This PCN is a combination of exploitation of learned policies and exploration of new policies (Mousavi, Schukat, & Howley, 2017; Zhang, Ishikawa, Wang, Striner, & Tonguz, 2020). For signal time optimization, a model-free, single-agent Q-learning system was also proposed that used queue length as state and aggregated time delay as reward (Araghi, Khosravi, Johnstone, & Creighton, 2013). Such a state–action-reward mechanism was very effective in the single-agent TSC system for single isolated traffic intersections (El-Tantawy, Abdulhai, & Abdelgawad, 2014; Genders & Razavi, 2019).

Proposed multi-intersection signal control systems as multi-agent settings have one agent at each intersection that enhances the reward by adjusting its policy, and optimizing the whole

[*] Corresponding author at: University of Engineering and Technology (UET), Lahore, GT, Road, Lahore, 54890, Punjab, Pakistan.

E-mail addresses: hamza@xeroai.com, hamza.hm.mukhtar@gmail.com (H. Mukhtar), adil@xeroai.com, adilafzalansari@gmail.com (A. Afzal), sahmari@kacst.edu.sa (S. Alahmari), syonbawi@uj.edu.sa (S. Yonbawi).

https://doi.org/10.1016/j.neunet.2023.07.027

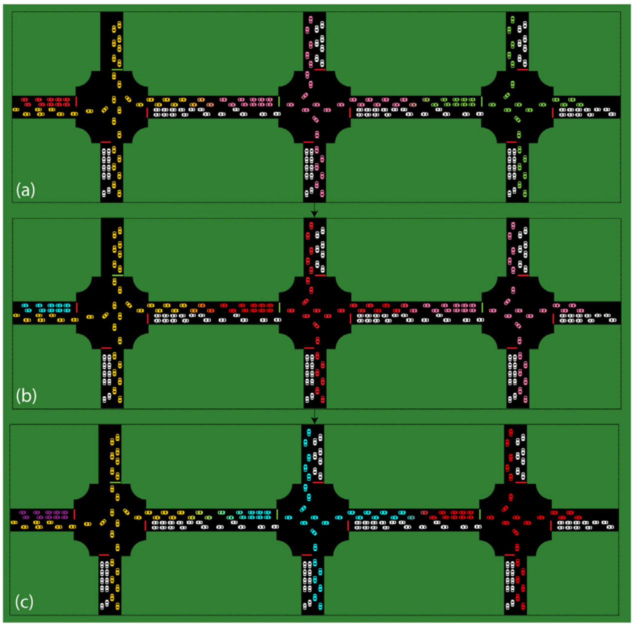


Fig. 1. Signal-free corridor objective: (a) Red traffic flow is stopped on the initial intersection of TIN in part, (b) when the red traffic flow reaches the next intersections, an open signal is available and traffic does not require waiting, and same goes in part (c).

DRL system via reward and policy loss. Generally, vehicles need to stop and wait for the green light at each intersection of the network. So, a vehicle waiting for the green signal at an intersection one has to wait at the subsequent intersections as well. In this work, we have framed the multi-intersection network signal control as a dual, local and global, objective function task. Network wise local objective function is similar to various DRL-based signal control methods (Li, Yu, Zhang, Dong, & Xu, 2021; Wu et al., 2022; Yang, Yang, Kang, & Deng, 2021), however, now their objective is not only to adjust corresponding PCN for reward maximization but also need to contribute for optimizing the global network level objective function. Our global objective is to provide a signal-free corridor and vehicles need to wait only at the initial intersection of the network. So, a flow of traffic needs to wait at the initial intersection of the network like the red traffic flow shown in Fig. 1, meanwhile, the next intersection will manage the traffic in such a manner, once that flow reaches the subsequent intersection, the green signal will be provided so that traffic flows do not have to wait for green signal again. Waiting for a green light at each intersection of the network requires stopping and starting the vehicles that are the major cause of various said traffic congestion problems (Rahman et al., 2022; Wang et al., 2021; Wen et al., 2020). Network-wide local objectives are based on minimizing the waiting time and travel time at every four signals of each intersection, however, when the vehicles of subsequent intersections are approaching the next intersection, a signal-free corridor will be available, which is the network-wide global objective of the TINC.

Large-scale multi-agent networks require interaction between subsequent agents, and interaction brings challenges in adjusting the learning process of the PCN (Gu, Guo, Wei, & Xu, 2021). The PCN learning process should be smooth where the local objectives of each agent of the multi-agent network form and impact the global network-wide objective. Past research work was focused on accelerating the learning of loosely coupled reinforcement multi-agent in adapting the knowledge exchange and game abstraction techniques (Liu, Hu, Gao, Chen, & Fan, 2019). This loosely coupled learning scheme works on the assumption that each agent works independently and has very limited knowledge exchange among agents in multi-agent settings. This assumption makes such a multi-agent learning scheme of limited use where agents have aggressive collaboration and close relationships. Most of the recently proposed multi-agent learning schemes are based on following the game abstraction with pre-defined rules such as distance between connecting agents (Jiang, Dun, Huang, & Lu, 2018). However, such rules are insufficient to define the nature of interaction dependency between agents of multi-agent networks. Game abstraction learns the relationship between agents, so the knowledge exchange mechanism can be streamlined. The attention mechanism, using soft and hard attention, was used to learn the other agents' distribution (Iqbal & Sha, 2019; Jiang et al., 2018; Liu et al., 2020). Their output activation function still has the dependency on the other agents in finding the important weight of each agent which hinders the ability to establish a natural relationship between agents of the network. In this article, we have modelled the multi-agent task for the local objective of each intersection as GCN based on an attention mechanism on top of the transformer network (Janner, Li, & Levine, 2021; Vaswani et al., 2017). Here, the network of multi-agent of multi-intersection is built as a graph, and this graph network learns the information of other agents in a partially observed environment for relation dependency resolution. When it comes to generating node embeddings, GCN is an ideal technique required for aggregating information for a clique of nodes (Chen, Liu, Wang, Zhang, & Liu, 2022) essential to combine local and global knowledge in order to achieve the local and global objectives of TIN. The proposed attention stage is based on hard and soft attention to learn the significance of edges weight to keep related edges only (Shen et al., 2018). Transformer (Janner et al., 2021; Vaswani et al., 2017) can do credit assignment through an attention mechanism that enables better generalization as the transformer can model larger behavioural distribution. TINC's global objective is achieved through a centralized deep Q-network (DQN) collaborative control network based on GCN. The key research contributions of this article are listed as follows:

1. We propose a multi-agent decentralized game abstraction PCN Graph Convolutional Network (GCN) based on the attention mechanism for the local policy network of each intersection.
2. We have Combined Deep Q-network (DQN) and Graph Convolutional Network (GCN) as Deep graph convolution Q network (DGCQ) to achieve a signal-free corridor. GCN provides collaboration between intersections and DQN uses that aggregation of information to control traffic decisions.

The rest of this article is organized as follows: Section 2 explains the related work, while the Problem formulation Section 3 introduces the detailed problem settings including graph modelling, state space, action space, and reward function. Section 4 introduces the proposed methods and explains the model architecture. Section 5 reports the experimental settings, and evaluation and provides a comparative analysis of the proposed model with other state-of-the-art models. Section 6 is the conclusion and draws the limitations of the proposed model, and identifies the prospective future work in traffic intersection automation.

2. Related work

TSC-related literature shows that methods to control traffic signals can be categorized into formal and learning-based optimization processes. Optimization-based policy control builds the signal automation problem as a min–max object function having multiple state constraints, and this kind of problem can be solved by the defined set of control inputs. An isolated intersection, having a local min–max objective function, was controlled by joining the green light speed advisory and adaptive policy control method (Niroumand, Tajalli, Hajibabai, & Hajbabaie, 2020). Collaborative policy control in network intersections, optimization-based control becomes too complex because of the dynamicity of the traffic and the number of intersections involved in the network. This exacerbation of complexity makes the optimization space non-convex that cannot be solved in linear time. Policy controllers based on the optimization scheme faced daunting challenges in the context of multi-agent control (Xu, Ban, et al., 2018).

Learning-based policy controllers leverage Deep Neural Networks (DNNs) because of their universal approximation ability. The combination of RL with DNNs yields the Deep RL (DRL) models that have increased robustness through automatic feature extraction and traffic pattern learning (Mousavi et al., 2017). DRL-inspired trained policy controllers give constant and efficient inference time, and their computational complexity is exacerbated when problem space becomes more complex. Another advantage of DRL controllers is their robustness due to the supply of training data from the simulated environment that can provide massive variational data with ease. DRL policy controllers do well on the single intersection controller or when each agent in a multi-agent setting of the Traffic Intersection Network Controller (TINC) has its local state and action space, and each agent has the objective to minimize or maximize its own objective. These systems do not fulfil the global objective function.

When such single-intersection single-agent DRL systems are extended for the multi-intersection systems, there are two learning schemes currently being used, centralized DRL (CDRL) and decentralized DRL (DDRL) (Abbracciavento, Zinnari, Formentin, Bianchessi, & Savaresi, 2023; Calvo & Dusparic, 2018; Van der Pol & Oliehoek, 2016). In the CDRL, one centralized agent represents a multi-agent to manage the multi-intersection network through the execution of joint policy. The core drawback of a centralized TSC system is that such a system affects the course of state–action scalability when state space grows linearly, and joint action space increases exponentially (Xie, Wang, Chen, & Dong, 2020). Moreover, the challenge of stability also emerges due to the imbalance between exploration and exploitation. Agents are required to explore the state of the environment as well as information about other control units so they can adapt the dynamic behaviours. Excessive exploration hinders stability and makes the learning task difficult (Prashanth & Bhatnagar, 2011).

The DDRL framework focuses on cooperative policies for multi-CDRL agents. Cooperation in DDRL is achieved through a message-passing mechanism (Foerster, Assael, De Freitas, & Whiteson, 2016). This mechanism develops multi-agent settings, such as max-plus decomposition (Van der Pol & Oliehoek, 2016), discount-state (Chu, Wang, Codecà, & Li, 2019) and information about neighbour nodes (Yang, Yang, Wong, & Kang, 2019), for achieving better performance in certain intersection networks. DDRL algorithms based on DNNs learn the embeddings of states of intersection networks (Yang et al., 2019). Such state embeddings are used to acquire better signal control policies. This traffic intersection network (TIN) consists of intersections, signals, road lanes and vehicles (Devailly, Larocque, & Charlin, 2021). As the number of intersections, signals, and road lanes remains fixed, their respective feature vectors can be represented in the DNNs (Xu, Hu, Leskovec, & Jegelka, 2018). However, the vehicle count in the network will always be dynamic. Such dynamicity cannot be handled because the neural network works on the fixed size input, so dynamic vehicle count is not represented in this paradigm (Welling & Kipf, 2016). The DDRL framework faces the moving-target challenge because multiple control agents have to interact with each other in order to adapt optimal PCN, therefore, reward distribution for an agent depends on the other agent in the network. This dependency violates the MDP policy of static reward distribution that makes the convergence impossible and reduces the learning stability (Shou et al., 2020). The network environment is composed of partially observed features, so the agent does not have the access to complete state space and lacks coordination because of the dependency of one agent's action on the other agent's action (Zhang et al., 2019). Consistency and coordination between agents' actions are essential to obtain optimal performance objectives, such as maximum through network-wide and minimum delays.

Graph neural networks (GNNs) are effectively used in various fields from node classification (Schlichtkrull et al., 2018; Welling & Kipf, 2016) to graph representation (Xu, Hu, Leskovec, & Jegelka, 2018). GNNs (Yang, 2023) have the ability to represent the dynamic TIN as a graph. DDRL algorithms based on GNNs can effectively learn the embeddings of the TIN graph and the cooperative control policies. Graph Convolutional Network (GCN) was used to represent the traffic feature vector between roads (Schlichtkrull et al., 2018). For better Cooperative policies, a graph attention network was used to handle the message-passing among CDRL agents (Wei et al., 2019). As compared to DNN, GNN can learn the embedding of existing as well as new traffic. The DDRL algorithm based on homogeneous GCN can jointly learn the TIN graph embeddings and network control policies, after the joint learning, learned network policies are realized to new TIN, where all the TIN are modelled as the heterogeneous graphs (Xu, Hu, Leskovec, & Jegelka, 2018; Yan et al., 2023). Deep graph Q-network (Kim & Sohn, 2022) is developed to overcome the limitations of value-based RL in large-scale networks with many traffic signals. A graph-based Q-network is created to capture spatial–temporal dependencies, along with a parameterized adjacency matrix that considers congestion propagation. MetaST-GAT (Wang et al., 2022) is a spatial–temporal graph attention neural network that considers the spatial–temporal correlations among intersections and utilizes a meta-learning method for GNN. It enables adaptive traffic signal control and can adapt to dynamic traffic flow. Many existing frameworks may not converge in real-world large-scale networks with a higher number of intersections.

In recent years, collaborative networks have obtained promising results in enhancing traffic mobility and signal automation (Ding, Dai, Fan, Zhang, & Wu, 2022; Graves, Nelson, & Chakraborty, 2021). It has shown great performance in information sharing between and among traffic intersections to increase signal throughput (Wang, You, Hang, & Zhao, 2023). The spatial scopic nature of the collaborative networks allows the constitution of an environment where traffic and intersectional information are shared through cooperative sensing and manoeuvring (Ma, Yu, Zhang, & Yang, 2022). Cooperative sensing enhances sensing range and awareness, while cooperative manoeuvring enables collaborative operations with centralized or decentralized decision-making (Wang, Liu, Liu, & Sun, 2023). Existing studies (Graves et al., 2021; Wang, You, Hang, & Zhao, 2023; Zhu et al., 2023) on cooperative intersection control mainly focus on improving green time utilization based on pre-defined lane settings, but unbalanced traffic flow due to job-housing disparities is a common issue that intensifies the contradiction between traffic supply and demand. These studies focus on enhancing the performance of each intersection individual through information sharing about other intersections, however, each traffic flow has to suffer from waiting at the subsequent intersections. Furthermore, the proposed methods lack in transferring learned policies to diverse traffic networks, dynamically tackling the time-varying number of vehicles in the network, and capturing heterogeneous features of objects in the network. To provide a signal-free pathway This study proposes a reinforcement learning-based collaborative control method to optimize time-space resources at the intersection via a graphical network for information sharing to provide a signal-free path for traffic flow.

3. Problem formulation

Inspired by the graph-based multi-agent systems (Chen, Dong, Ha, Li, & Labi, 2021; Devailly et al., 2021; Yang et al., 2021), we have modelled the multi-agent TINC as a decentralized graph network (DGN) for the local objective of each intersection in the network, and these decentralized intersections contribute to achieving the global objective as a centralized collaborative graph network (CCGN). For the DGN, multi-agent TINC has been modelled as a Markov Game abstraction; similar to multi-agent graph (Chen et al., 2022; Liu et al., 2020). Nodes in the DGN through game abstraction and graphical representation of the intersection network, collaborate to provide the signal-free corridor and this collaboration network is modelled as a centralized policy controller similar to a centralized graph network (Chen, Dong, Ha, Li, & Labi, 2021).

3.1. TIN as graph network

Signal free corridor, once the vehicles have waited at either entry intersection of the TIN, requires not only intersection management in isolation (local objective), but requires traffic flow information sharing to subsequent upstream and downstream intersections (global objective). Traffic flow information at each intersection is obtained through the traffic pattern and sensory input about the traffic flows, while global information about traffic flow and each intersection of TIN is obtained through the collaborative nature of the graph network that shares the knowledge of each intersection to the CCGN. Local sensory information is used to manage the signal at each intersection to minimize the waiting and travelling time, while global information gives the TINC ability to make the intersection network fully observable which enables the CCGN to provide the signal-free corridor. Both local and global information are critical given the partially observed nature of the TINC for each intersection, so knowledge sharing makes the PCN effective. Local sensory information about the traffic patterns and traffic flow is used by each intersection to optimize the local PCN, DGN, to maximize the local reward, while the information sharing with the CCGN makes the global information. So, the global decision depends on the local information sharing, and the flow of information about the traffic flow goes to downstream and upstream intersections once the traffic flow already has waited at the entry intersection of TINC. The dependency of CCGN on the knowledge of each intersection, and flow stream directions can be modelled as a collaborative graph. The graphical data structure is an effective way to model the network of intersections (nodes), and their natural dependencies (edges). Each intersection is represented as a graph node, and the edge is the communication channel between intersections. When a traffic stream is waiting at the extreme left intersection, information such as distance from the next intersection, green signal timing, expected time of arrival at the next intersection, and length of traffic, are shared through CCGN, and this sharing of information continues from current to next intersection.

GNNs are capable of generating embedding not only on the features of nodes but also on the immediate upstream and downstream nodes. As the graphic form of convolutional neural network (CNN), GCN aggregates the information getting from the node embeddings for a clique of nodes (Chen et al., 2022). The Global objective depends on the information sharing from each intersection of TINC, so GCN uses a fusion model to join the local information of each intersection. Weights of GCN layers behave as the attention mechanism (Chen, Dong, Ha, Li, & Labi, 2021) that enables the model to determine and focus only on the relevant features. The relevance of features determines their impact on the global objective, and the attention mechanism determines this relevance. For example, when a traffic flow is crossing an intersection, the next intersection is required to focus more on the immediate previous intersection for the global objective of providing the flow with a signal-free corridor, while next to next intersection should focus more on the last intersection rather than the farther way intersections. Control decisions need to focus more on the downstream intersection for the current traffic flow, while more focus on the upstream intersection for the next traffic flow. GCN learns this relevance of features and encodes in the weights of the layers, these weights are derived from the successful and unsuccessful intersection control decisions.

The input matrix to GCN contains various features, such as distance between intersections, current states of each intersection, green signal timing, expected time of arrival at the next intersection, and length of traffic, and an adjacency matrix that contains the decision dependencies between nodes. GCN outputs an intersection (node) level embedding matrix that contains an aggregation of local and global information of the TIN environment. These intersection-level embeddings are then used as the key information to make collaborative intersection control decisions in TINC to provide a signal-free corridor. GNN used DRL for multi-agent settings where GNN was employed as an encoder to learn the relational feature embeddings between nodes (agents) and these learned embeddings were further fed to the policy control network for actions (Jiang et al., 2018). Joint training of GNN and PCN networks enhanced the collaborative nature of the DRL agents and enabled them to work for the global objective.

Inspired by joint training schemes of GNN and PCN network (Chen, Dong, Ha, Li, & Labi, 2021; Jiang et al., 2018), we have modified the methodology to make it adaptive to game abstraction and work as a collaborative network to achieve the global objective (e.g. signal-free corridor). The DDRL system requires a separate DGN against each agent that can serve the local objective of each agent (e.g. intersection), however, the global objective needs to have collaboration between and among agents. DDRL does not offer this collaborative nature that can be achieved only through joint training of the embedding encoder (e.g. GNN) and PCN. This study has used a weight-sharing scheme to achieve collaborative nature, and CCGN has developed to output decision actions for each agent of the TINC in order to achieve the global objective (e.g. signal-free corridor). The workflow of the CCGN is shown in Fig. 2, where the traffic and TIN information is fed CCNG-based PCN that predicts control actions for all intersections.

3.2. Multi-intersection TSC as Markov game abstraction

An extension of MDP, Markov game abstraction is widely based for multi-agent RL. Game abstraction reduces the learning complexity to achieve the equilibrium in policy in the multi-agent RL (Markov game). Inspired from the related studies (Chu et al., 2019; Wang et al., 2019; Yang et al., 2021), we have modelled the multi-intersection traffic control as an n-agent ($n \geq 2$) partially observable Markov Game using tuple ($N, A^{n}_{i=0}, S, R^{n}_{i=0}, T, O, \ r, \pi, \gamma$). where N is the number of DDRL agents, A is the action space of agent $t(i = 0, \dots, n)$, S is the state space, $R_i : S \times A \rightarrow R$ is the reward of ith agent, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function, O is the global observation space, and $\pi_I = O_i \times A_i \rightarrow [0, 1]$ that works for maximizing the aggregated discounted reward $R_i = \sum_{t=r}^{T} \lambda^{t-1} r_i^t$ where $\lambda \epsilon [0, 1]$ is the discount factor.

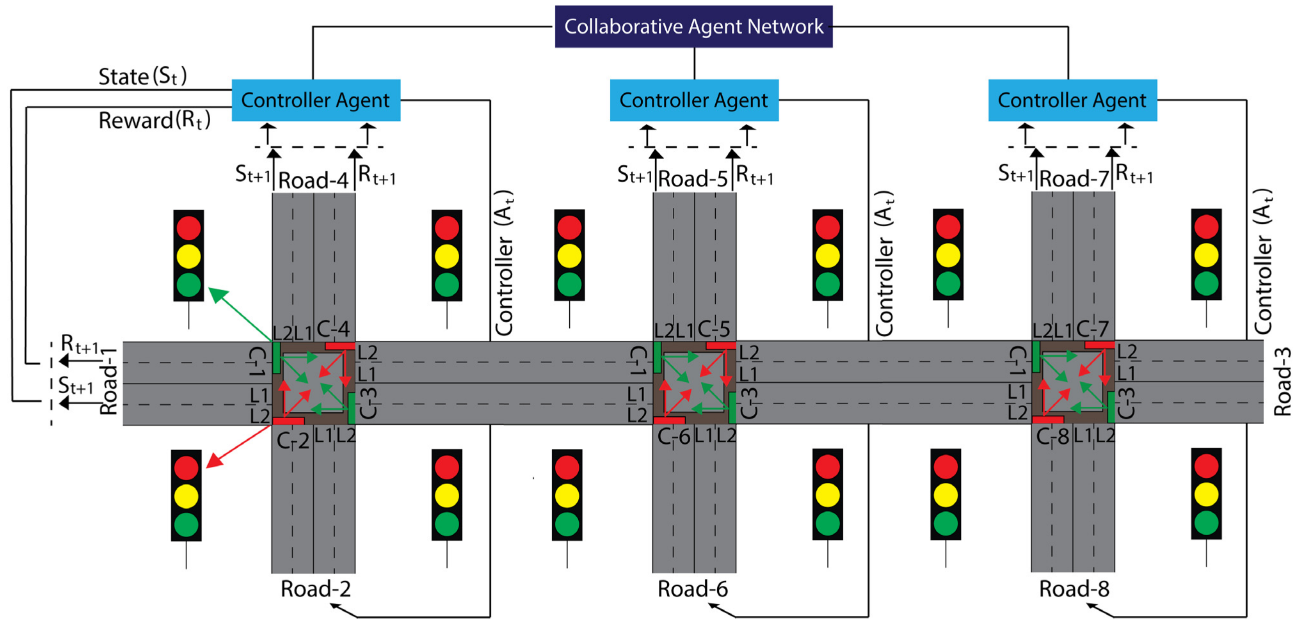


Fig. 2. Graphical representation of traffic intersection network where intersection share knowledge through a collaborative network.

Table 1
Parameters of Synthetic traffic network and Real-world traffic network (Fig. 5).

Parameters	SQ1, SQ2, SQ3
Traffic flow unit	360 vehicle/lane/hour
Multiple of flow unit	C
Vehicle length	4.0 m
Grid cell length	4.0
Maximum vehicle speed (v_max)	48 $\frac{\text{km}}{\text{m}}$
Maximum acceleration	1.2 $\frac{\text{m}}{\text{s}^2}$
Maximum Deceleration	2.5 $\frac{\text{m}}{\text{s}^2}$
Lower green duration limit (G_min)	10 s
Upper green duration limit (G_max)	60
Yellow light durations (d_yellow)	3.30 s

Table 2
Feature set of different node types.

Node type	Attributes
Intersection (F^{IN})	Isopen NSG, Isopen NSLG, Isopen EWG, Isopen EWLG, Distance from previous intersection (P_{in})
Traffic flow (F^{TF})	Intersection's signal position, Real-time speed, location on road
Lane^{FL}	Length, Average traffic speed, Vehicle count, Waiting time

3.2.1. *Global action space A*

For each time step, each intersection of the TINC has a discrete action set, and the intersection can choose an action from this action space (Yang et al., 2021, 2019). Potential action space is as follows:

$$a_i = \left\{(NSG, l_t^1), (NSLG, l_t^2), (EWG, l_t^3), (EWLG, l_t^4)\right\} \tag{1}$$

Where, (NSG, l_t^1) indicates the North-South Green Light, $(NSLG, l_t^2)$ indicates the North-South Left Green Light, (EWG, l_t^3) indicates the East-West Green Light, and $(EWLG, l_t^4)$ indicates East-West Left Green light that TINC agent n turns on for the duration $(l_t^n)(c = 1, 2, 3, 4)$ at time-step t. This action space aggregates all the potential combinations of individual actions available for each agent of the TINC. Duration of $(l_t^n)\epsilon[L_{min}, L_{max}]$, and the duration of yellow light is set to $l_{yellow} = \frac{v_{max}}{a_{dec}}$. The rest of the related parameters are given in Table 1.

3.2.2. *Global observation space O*

Each agent partially observes the global state $s\epsilon S$ as their observation $o\epsilon O$. At time step t, ith agent state is defined as 0_i^t and the agent space state has three blocks: Nodes feature set f_t^X, adjacency matrix M_t^A, and an intersection mask M_t^I. These blocks are computed as follows:

1. Node features set
   There are three node types named: intersections, traffic flow and road, denoted as $f_i^X = (F^I, F^T, F^R)$, and each of the nodes has various attributes as shown in Table 2. The traffic flow node is the core of the global objective, and each of its attributes is calculated as follows:

   (a) N_s^T is the traffic flow position for the intersection signal i, which is defined as a categorical variable, and "one hot" encoding is used to mention the signal position. Traffic flow position is denoted based on the signal of each intersection position as follows: East signal [1, 0, 0, 0], West signal [0, 1, 0, 0], North signal [0, 0, 1, 0] and South signal [0, 0, 0, 1]. Each traffic flow has its own intersection locator vector that is updated by each DGN.

   (b) Actual speed of the incoming traffic flow is estimated as $V_{es} = V_{historic} + V_v.V_{historicis}$ is the speed estimation of the time phases, while V_{vis} speed variation, and is determined from the historical speed variation.

   (c) N_L^T refers to traffic flow location on the road at a given time step t. There is no sensory information about the traffic flow in between intersections, so, this location is inferred from the available information about the traffic flow and intersection environment as follows:

   $$N_L^T = P_{in} - (V_{es}xg) \tag{2}$$

   Overall, feature map F_x for the time step t is the matrix of size $Nx11$ where 11 is the total number of features including intersection (F^I), traffic flow (F^T) and lane features (F^R). This feature vector of size 11 is vertically stacked for all intersections.

$$F_t^X = [f_i^X]_{(i=0)}^N \epsilon R^{(Nx11)} \tag{3}$$

2. Adjacency
   At, an adjacency matrix, M_t^A, is a binary NxN matrix that links the information dependency between and among intersections, where N is the number of intersections in the TIN. All the intersections of TIN are linked with each other through a graphical structure that builds a global collaborative network where each of the intersections shares information with CCGN. Connecting intersections makes the policy decision implicit in the fusion block.
3. Intersection Mask
   M_t^I, intersection mask, filters the intersection graph embeddings after the GCN fusion block. Mask vector is a binary vector of length N where 1s for the included intersection and 0s for the filtered-out intersection embeddings.

3.2.3. Reward function

Our TINC reward has two kinds of reward functions: global and local reward functions. The global reward function guarantees a signal-free corridor for the traffic flow once the traffic flow has waited for the green signal at the entry intersection of the TINC. The local reward function of each intersection enhances the traffic throughput with the minimum waiting time. Each intersection works in a collaborative manner to enhance the global reward. Inspired from the frequently measurable and spatially decomposable principle (Chu et al., 2019; Yang et al., 2021, 2019), local reward, r, function for each intersection of is defined as:

$$r_i = -\sum_{j\epsilon N_i}(q_{i,j}^{t+l_t^n} + w_{i,j}^{t+l_t^n}) \tag{4}$$

where N_i refers to the set of intersection agents i_s neighbours, l_t^n is the action duration at each intersection, $q_{i,j}^{t+l_t^n}$ is the queue length of the waiting vehicle at each intersection, and $w_{i,j}^{t+l_t^n}$ is the aggregated delay of traffic flow.

Each intersection contributes to a global reward function, and this contribution is aggregated into CCGN. The global reward is the sum of all intersections' individual reward minus p^g (global plenty) value. p^g is actually a vector of length $N-1$, so excluding the entry intersection, each index of the vector corresponds to an intersection where the first index refers to the immediate intersection next to the entry intersection for a traffic flow, and subsequent indexes are referred to subsequent intersections. The global objective is to provide signal-free corridors, so the traffic should only need to wait at the entry intersection of the TIN. Therefore, ideally waiting time after the entry intersection should be zero which means traffic will enjoy a green light when approaching to subsequent intersections. To formulate the global reward, p^g vector captures the waiting time for the traffic at each intersection after the entry intersection, and at the end, this vector is subtracted from the individual rewards of the corresponding intersections. So, global reward (R) of TINC aggregates the individual rewards and penalty across all the intersections of TIN as defined:

$$Rg = \frac{1}{n}\sum_{i}^{N}(r_i^t - p_g^t) \tag{5}$$

In order to achieve numerical stability during the training, and keep the reward in finite range, RL setting is defined as the episode finite horizon. The total number of intersections of TIN are specified to control the episode horizon where an episode starts when a traffic flow enters the intersection network and ends when a traffic flow leaves TIN.

4. Methods

The TIN is modelled as the graph network whose intersections are represented by graph nodes Fig. 2. Our TINC is represented as the graphical structure, and decomposed into two layers: The local network, DGN, comprises all four signals of an intersection, and the global network, CCGN, is a central collaborative that connects all intersections included in TIN. Each intersection's local controller acquires not only local sensory information of each of the four signals of the intersection, but also global information of other intersections via a collaborative network. The local network is formed as Mesh topology to share information between signals of an intersection, while the global network is formed through a collaborative network where each intersection shares information. This study uses two types of learning paradigms: Curriculum learning for the local network (Liu et al., 2020; Singh, Jain, & Sukhbaatar, 2018), and centralized learning with decentralized execution for the global network (Chen, Dong, Ha, Li, & Labi, 2021). In multi-agent curriculum learning, agents are gradually increased, and this gradual increase of agents makes the training process stable and simple. This learning paradigm is modelled as multi-agent game abstraction, but we have used Transformer (Vaswani et al., 2017) for embedding extraction instead of LSTM, and GCN is used for the embedding jointment. In multi-agent centralized learning, agents decide on action at each time step for the global objective. This centralization of CCGN for information sharing is modelled with GCN and the global decision controller is modelled as Deep Q-learning.

4.1. Local policy network

Similar work (Iqbal & Sha, 2019; Jiang et al., 2018) tried to learn multi-agent communication with an aggregation scheme where all agent information and communication was accessed into a single aggregated vector, and fed to each of the agents. In this aggregation scheme, each agent received information from every other agent, so ample information flow and frequent communication make policy learning difficult and ambiguous due to flow of unnecessary information and communication linkages. Used softmax function generates relative value that is insufficient in establishing the dependency between agents. This function also assigns the weights to irrelevant agents, so, the number of communication linkages cannot be reduced. Moreover, softmax assigns small nonzero probability values to irrelevant agents which reduces the attention strength given to a few influencer agents. Excess of information also produces overfitting when this information is passed through a Neural Network.

At each time-step, t, each agent i gets local observation o_i^t which is a property of agent i in the agent-Collaboration graph G because of the partially observable nature of the TIN environment. Observed o_i^t is encoded into a feature vector h_i^t which is used to learn the contribution dependencies between the agents through a two-stage, hard and soft, attention mechanism. As shown in Fig. 3, o_i is the agent i observation, the policy is formed as:

$$o_i = \pi(h_i, c_i) \tag{6}$$

Here, π is the action policy of the agent, h_i is the observation feature of agent i and c_i is the contribution of agents for agent i. Large-scale multi-agent settings have a large number of agents, but, in intersection control-like tasks, not each agent depends on all other agents present. One intersection mostly depends on immediate intersections before and after, and the dependency on information sharing reduces with the subsequent intersections. Therefore, each agent does not require to have information on all other agents, and needs to receive only relevant information that influences it.

Recent work (Liu et al., 2020) tried to achieve communication with the attention GNN, where long short-term memory (LSTM) was used as an embedding extractor along with the hard and soft attention mechanism. However, these recurrent modelling techniques factor computations along the indexes of the input and output, and this sequential nature works in a non-parallelization fashion. LSTM has far more parameters as compared to Transformer and this complexity increases in multi-agent settings. Transformer networks (Janner et al., 2021; Vaswani et al., 2017) have the ability to extract global dependencies regardless of the distance between input and output. Inspired by Chen, Lu, et al. (2021), we have used the Transformer encoder network to extract the Position-Enriched feature:

$$h_i' = Transformer(e(o_i)) \tag{7}$$

Where o_i is the agent observation at the time-step t, and $e(-)$ is an encoder function parameterized by a neural network. The two-stage attention mechanism is used to get the agents having contribution on the agent i with minimum dependencies to avoid overflow of dependencies. Self-attention model Transformer produces the one-hot vector that determines the interaction between agents by establishing the edge between the nodes of graph G. Each agent has different contribution strength which is determined by the weight of each edge in graph G. Soft attention is used to learn these weight that forms the sub-graph Gi for agent i which is only connected with the relevant agents.

$$W_h^{i,j} = M_{hd}(h_i, h_j), W_s^{i,j} = M_{st}(W_h, h_i, h_j) \tag{8}$$

Here, W_h and W_s are the output values of the hard and soft-attention models W_{hd}, W_{st}, respectively, calculated by the Transformer output h_i, h_j. With this, contribution c_i is obtained from the other agents by GCN. GCN produces the vector representation that shows the contribution for an agent i from other agents. This contribution is calculated by the weighted sum of the other agents' contribution obtained through a two-stage attention mechanism:

$$ci = \sum w_j h_j = \sum W_h^{i,j} W_s^{i,j} h_j \tag{9}$$

4.2. Collaborative network for global policy

At each time-step t, the centralized agent interacts with the environment to get the current state s_t, takes the at to transition in the next state s_{t+1} and gets a reward r. This transition of the centralized agent from the current to the next state can be shown in the form of quadruplet (s_t, a_t, r_t, s_{t+1}). At the time step t, there is an N intersection in the TIN and each signal of each intersection receives or leaves the flow of traffic. The state s_t is a block of information composed on the node features F^X, adjacency matrix M^a, and intersection mask M^{in}: $s_t = (F^X, M^A, M^I)$. Node features, F_i^X, is a tuple of three kinds of information: (F^I, F^T, F^R). This tuple provides traffic and signal information to each of the intersections. Each intersection shares information with all other information and a central DRL controller aggregates the shared information and determines which part of the shared information is necessary.

At each time step t, node feature matrix, F^X, is fed to Dense Neural Network (DNN) encoder D_e to produce the node embedding matrix E in d dimensional space $E \subset R^{Nxd}$. E is high-dimensional feature matrix output by the DNN, and embedding space contains all the embeddings:

$$E_t = D_e(F_t^X) \in E \tag{10}$$

Then, node embedding E for each intersection is fed to GCN to produce the node embeddings based on the node embedding produced by the D_e and the embeddings of adjacent nodes on either side. These node embedding through GCN are computed in a parallel fashion:

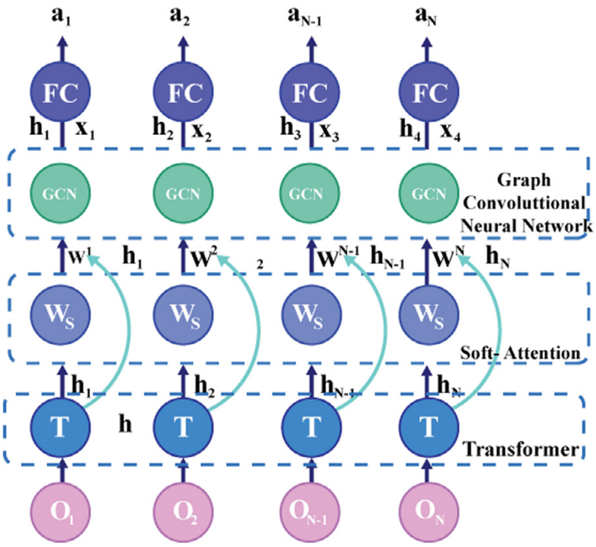


Fig. 3. Model for the local policy controller for each intersection to manage its signals.

$$G_t = g(E_t, M_t^A) = \sigma(D_t^{\frac{-1}{2}} M_t^A D_t^{\frac{-1}{2}} E_t W + b) \tag{11}$$

Here $M_t^A = M^A + I^N$ is the adjacency matrix with self-looping for each node; is the activation function such as Relu, Tanh and Sigmoid to introduce nonlinearity; D^t is the degree matrix calculated from M^a; W and b refer to weight and bias of GCN. GCN can have multiple layers, however, an excess number of layers brings "over-smoothing" (Chen et al., 2020). After the GCN, embeddings map G_t is obtained. Then node embeddings are filtered out to obtain relevant dependencies through the element-wise product of mask M^I and G_t.

$$Z_t^{TIN} = M_i^I.G_t \tag{12}$$

In the end, TIN node embedding is passed to a Q-network, Q_θ, to get the Q value that corresponds to the goodness or badness of the action taken. All the neural network blocks including DNN, GCN, and Q-network can be defined as $\hat{Q}$ function parameterized by θ, and Q_θ refers to the action space for the intersections at time step t.

$$\hat{Q}_\theta(s_t + a_t) = q_Q(Z_t^{TIN}, a_t) \tag{13}$$

Inspired from the techniques for the stable training (Van Hasselt, Guez, & Silver, 2016), the overall neural network is trained with experience replay on mini-batches sampled from the replay buffer R containing transitions of (s_t, a_t, r_t, s_{t+1}). During training, the objective is to minimize the loss at each mini-batch.

$$L_\theta = \frac{1}{b} \sum \beta_t - \hat{Q}_\theta(s_t + a_t) \tag{14}$$

Here, b is the mini-batch size, and β_t is the desired Q-value which defined as $\beta_t = r_t + \beta maxQ(s_{t+1}, a)$. The composition and architecture of each component of the model are shown in Fig. 4.

1. Encoder D_e : FC(64) + FC(64)
2. GCN layer g: GConv(64)
3. Q-network q_Q: FC(64) + FC(64) + FC(16)
4. Output layer: FC(N)

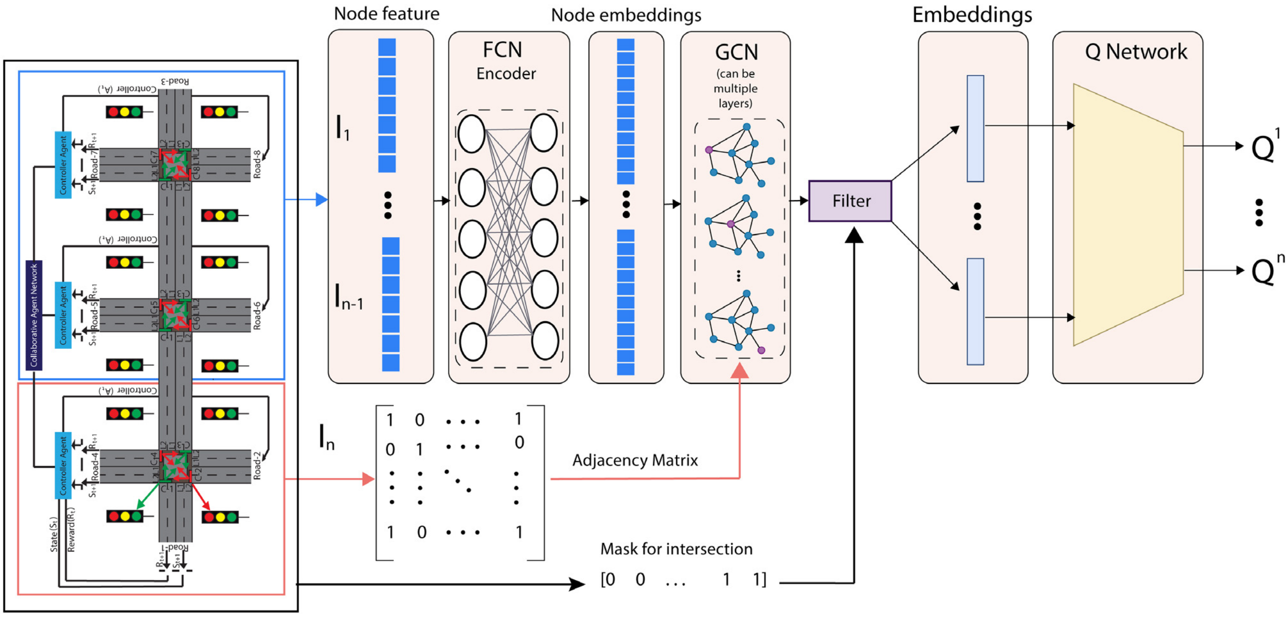


Fig. 4. Model architecture for the centralized collaborative graphical network for a global policy where blue-boxed intersections are reference nodes intersection and red-boxed is a decision node. The reference node feature matrix is fed into a fully connected network (FCN) encoder and then graph convolution (GCN) is performed to get connectivity dependencies. The adjacency matrix of the decision node and graphical representation of reference nodes are filtered and embeddings are passed to a Q-network to output Q values which represent the goodness of a certain action.

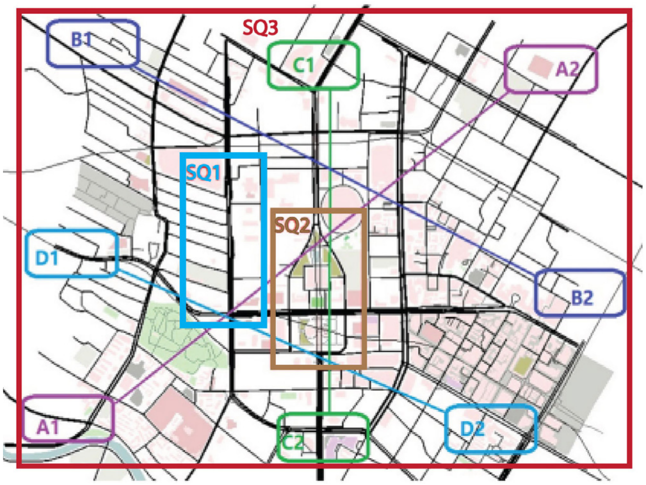


Fig. 5. Overview of 3 real-world traffic scenarios, named SQ1, SQ2, and SQ3.

5. Experiment settings

We have evaluated our method on the SUMO (Codeca & Härri, 2018) for three real-traffic scenarios such as SQ1, SQ2, and SQ3 from Yang and Yang (2021) and Yang et al. (2021, 2019). In SUMO, road, traffic, intersection and training parameters are simulated. For the training, state and action space are defined to achieve the defined reward function to model the MDP for the designed DRL. The performance of our methodology is compared with state-of-the-art algorithms in terms of evaluation metrics such as average waiting time, average travelling time and average stops (Per traffic flow) at intersections.

5.1. Experimental setup

In this section, experimental setup and implementation details are presented. TINC is trained on synthetic traffic environments, while evaluated and compared on the real-world traffic network. SUMO is used as a simulator to perform the experiment.

5.1.1. Training road network

Traffic intersection network TIN^{axb} is a synthetic traffic networks where $a \in [1, 4]$ and $b \in [1, 6]$. The synthetic network utilized in this research is similar to previously studied works (Devailly et al., 2021; Yang et al., 2021). It is composed of 25 intersections that consist of a combination of two shapes: "⊢" and "+ ". Each road in the network has three lanes, with a total length of 2100 metres and a width of 1.8 metres per lane. To train the proposed CCGN, various combinations of the synthetic network are used to provide a signal-free traffic flow once vehicles have arrived at the initial intersection of TIN. The CCGN is trained with both unidirectional and bidirectional traffic sequences, with a range of 60 to 460 vehicles, in order to make the learning process robust, other relevant parameters are specified in Table 1. In the synthetic network, the North-South road has been given a higher priority, allowing for a signal-free corridor with a traffic flow of 420 vehicles per lane per hour (lane/hour), while the West-East road has a lower priority with 140 vehicles lane/hour. The start and end points of the vehicles in the synthetic traffic network are randomly assigned to the edges. To train the CCGN, traffic networks are randomly selected from the synthetic road network and simulations are initiated with random seeds to allow for diverse transitions. The training phase involves 20 parallel simulations, each lasting approximately 1200 episodes (1.224 million steps), with each episode consisting of 1020 steps. The proposed CCGN is trained on a machine equipped with a GTX 3080 Nvidia GPU with 128 GB of RAM and a 6 GB frame buffer. On average, each episode took 7.2 min to complete, and the overall training process took 144.23 h. However, after 12 min, each episode is terminated to prevent the simulation from prolonging due to vehicles taking an excessive amount of time to reach their destinations.

5.1.2. Evaluation road network

Similar to past research (Yang et al., 2021, 2019; Zang et al., 2020), three traffic network scenarios, SQ1 (blue box), SQ2 (brown box), and SQ3 (red box) as shown in Fig. 5 are used to evaluate and compare the performance of CCGN with other state-of-the-art models. The Scenarios SQ1 and SQ2 contain 15 and 20 intersections, respectively, while SQ3 has 121 intersections including intersections of SQ 1 and SQ2 scenarios as well. SQ1 is highly representative of the road system, consisting of seven branches that connect to a main road. This configuration is made up of different parts of the road system. SQ2 is a rare and special configuration, combining a triangular and rectangular road topology. SQ1 and SQ2 are both parts of the overall road system and are being examined due to their unique features. Finally, SQ3 (represented by the red box in Fig. 5) encompasses both SQ1 and SQ2 and is composed of the entire set of intersections. This configuration is characterized by its wide range and complex road conditions, making it a useful scenario to assess the performance of algorithms. Configuration of all three traffic scenarios is similar to studies (Wang et al., 2021; Yang et al., 2021) and given in Table 3. These parameters describe the characteristics of a traffic intersection network (TIN) with different scenarios, SQ1, SQ2, and SQ3. The traffic flow unit is set at 360 vehicles per lane per hour. The multiple of the flow unit is a set of values, [1, 3, 5, 7, 9, 7, 5, 3, 1], used to simulate different levels of traffic volume.

The vehicle length is 4 m, and the grid cell length used for simulation is also 4 m. The maximum vehicle speed (v_{max}) is set at 48 km/h and the maximum acceleration and deceleration are 1.2 $\frac{m}{s_2}$ and 2.5 $\frac{m}{s_2}$, respectively. The lower green light duration limit (G_{minx}) is 10 s, while the upper green light duration limit (G_{max}) is 60 s. The yellow light duration is set at 3.30 s. These parameters help to define the behaviour and movements of the vehicles in the simulation and provide a realistic representation of real-world traffic.

These traffic networks contain intersections of shapes "⊢" and "+" and make the traffic scenarios complex enough So, evaluation of such a traffic network can exhibit the robustness of the proposed TINC. All the methods are evaluated on these traffic networks with the dynamic traffic flow. As mentioned in Yang et al. (2021), we have used 4 sets of traffic flows $A1 \leftrightarrow A2$, $B1 \leftrightarrow B2$, $C1 \leftrightarrow C2$ and $D1 \leftrightarrow D2$, but, flows of traffic are added at the initial intersection with the dynamic number of vehicles after the random interval.

but, flows of traffic are added at the initial intersection with the dynamic number of vehicles after the random interval. The traffic flows $A1 \leftrightarrow A2$ and $B1 \leftrightarrow B2$ are comprised of multiple unit flows, which are set according to Table 1. These unit flows are initiated every 500 s starting from 0 s. Similarly, the traffic flows $C1 \leftrightarrow C2$ and $D1 \leftrightarrow D2$ follow the same pattern, but with a starting time of 700 s instead. These sets of traffic flow allow for a comprehensive study of the interactions between the different vehicles within the traffic network.

To ensure the robustness of the results, both the training and evaluation experiments are repeated 10 times using different random seeds. This involves conducting 10 parallel simulations for the evaluation, totalling 100 simulations, and then aggregating the results from all 100 simulations. This repetition of the experiments allows for a more reliable and accurate representation of the results, as it helps to minimize the effects of random variability.

5.1.3. Evaluation metrics and compared methods

Performance is evaluated and compared on three metrics:

(1) Average waiting time, used as a metric in Chu et al. (2019) and Yang et al. (2021) is the average of time spent by all vehicles of each traffic flow at all intersections from the origin to destination.

(2) Average travelling time, used as a metric in Devailly et al. (2021) and Zang et al. (2020), is the average time spent by all vehicles to reach the destination.

Table 3
Configuration of Real-world traffic scenarios.

Configuration	SQ1	SQ2	SQ3
No. of Intersection	15	20	121
No. of lanes	3	3	3
Width of lane	1.8 m	1.8 m	1.8 m
Vehicle Length	5 m	5 m	5 m
Grid cell length	5 m	5 m	5 m
Avg. arriving rate	0.1 s^{-1}–1 s^{-1}	0.1 s^{-1}–1 s^{-1}	0.1 s^{-1}–0.5 s^{-1}
Max. acceleration	6.2 $\frac{m}{s^2}$	6.2 $\frac{m}{s^2}$	3.2 $\frac{m}{s^2}$
Max. de-acceleration	3.2 $\frac{m}{s^2}$	3.2 $\frac{m}{s^2}$	3.2 $\frac{m}{s^2}$

(3) Average stops at all intersections by all traffic flows in the entire traffic network, and this is the main objective of CCGN to provide the signal-free corridor once the traffic flows have waited at the initial intersection. Average waiting and travel time are derived from the 3rd metric because the fewer the number of stops, eventually lesser will be the average waiting and travel time.

Algorithms are evaluated on the real-world traffic networks, SQ1, SQ2 and SQ3, never experienced while training for 3 h with dynamic traffic flow. Although a GPU machine is used for the implementation, evaluation is stopped after 3 h due to computations. Each evaluation interaction is repeated 10 times with a dynamic number of vehicles in the traffic flow produced at random time steps. Traffic flows are also generated based on a unidirectional and bidirectional basis. Evaluations on the 20 parallel simulations are repeated 10 times, so 200 simulations are performed with the random seed, and then results are aggregated of all 200 simulation episodes. This process is also followed in the training phase.

To compare the performance of our model, CCGN, we have evaluated several state-of-the-art models of both centralized and decentralized policy schemes. IG-RL (Devailly et al., 2021), Metalight (Zang et al., 2020), MA2C (Chu et al., 2019) and IHG-MA (Yang et al., 2021) are decentralized models, while CGB-MATSC (Wang et al., 2021) is based on a centralized paradigm, other details as shown in Table 4.

5.1.4. Training parameters

Prior to the training of the model, the first 4×10^{10} transitions (200 episodes) serve as the warm-up phase. The model is trained using experience reply with randomly sampled batch size = 128. Overall, training prolongs around 1600 episodes (16×10^{10} steps), including warm-up and actual training phases. An epsilon policy is also set having a probability of 0.25 for random action to provide a robust exploration opportunity. Adam is used as the optimizer with the initial learning rate $\alpha = 10^{-4}$ and model update rate $\tau = 10^{-3}$ for double Q-learning.

5.1.5. Batch computation

As the traffic flow situation is extremely dynamic, different time steps contain different numbers of traffic flows in the TIN, so some intersections may have traffic on one side, and some may not have traffic flow on that side. Therefore, the shape of the input feature matrices $F_t^X(N \times 11)$ is also dynamic, and changes with respect to different time steps. Batch training requires stacking these feature matrices into a tensor for batch training. Although batch size b, could be set 1, however, convergence becomes different due to high variance. For batch size $b > 1$, the input feature tensor (Node feature, adjacency and TIN mask matrix) and output tensor (Q-value) need to have consistent shape across time steps. For this consistent shape, the maximum number of traffic flows N_{max} is specified for the model input, and the tensor is zero-padded when some intersections do not have traffic flows. Thus

Table 4
Summary of state-of-the-art models.

Literature	Model	Year	Policy scheme	Learning paradigm	Reward
MA2C Chu et al. (2019)	MDRL algorithm based on DNN	2019	Decentralized	Neighbourhood policies and spatial-discount -factor	Change of waiting time and queue length
Metalight Zang et al. (2020)	Value and gradient-based algorithm based on DNN	2020	Decentralized	Individual-level and global-level adaptation	Average queue length
IG-RL Devailly et al. (2021)	Q-Learning with Graph convolutional network	2021	Decentralized	Transferable traffic-signal policies	Negative sum of local queues lengths
IHG-MA Yang et al. (2021)	Heterogeneous graph neural network with Bi-GRU	2021	Decentralized	Multi-agent actor–critic	The queue length and waiting time of vehicles
CGB-MATSC Wang et al. (2021)	Collaborative group-based multi-agent Q-learning	2021	Centralized	Multi-agent actor–critic	Average flow of vehicles
IHA-MDG Yang and Pang (2022)	Attention-based multi-agent deep graph infomax algorithm	2022	Decentralized	Multi-agent actor–critic	Quantity of vehicles waiting in entry lane
CI-MA Yang, Yang, Zeng, and Kang (2023)	Causal inference multi-agent reinforcement learning	2023	Decentralized	Cooperative traffic-signal policies and Q-values for multiple agent	Average intersection delays

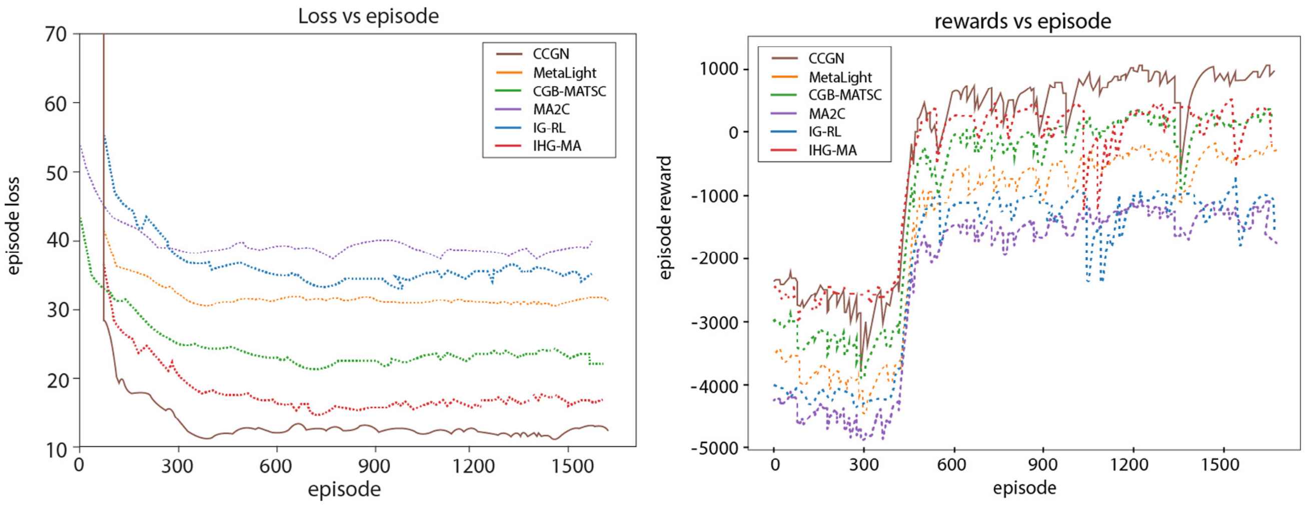


Fig. 6. Loss and reward against episode training graph.

N_{max} is the maximum number of traffic that can appear in the TIN. When the input tensor is zero-padded, there are some dummy values that can be filtered through the TIN mask. Thus, the final shape of the node feature tensor is (b, N_{max}, 12), similarly, the adjacency matrix and TIN mask is of shape (b, N_{max}, N_{max}) and (b, N_{max}) respectively. Finally, the Q-value output tensor is of shape (b, N_{max}, A), where A is the action space for the TNIC.

5.2. Results

5.2.1. Training results

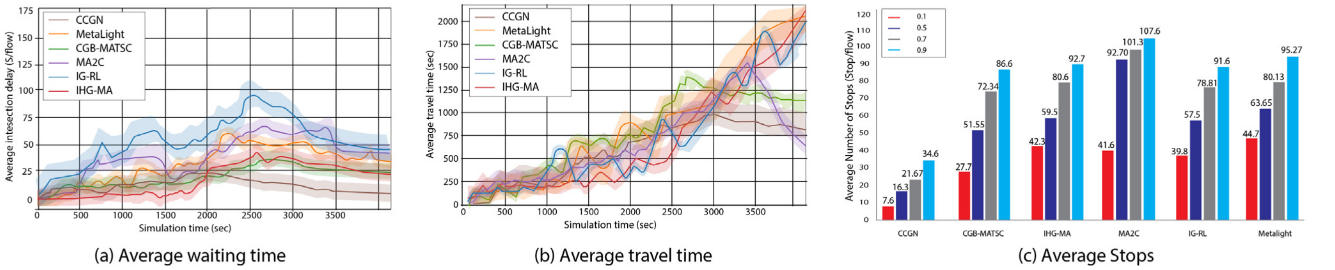
Training loss and reward curves are shown in Fig. 6 (loss and reward). The first 200 episodes represent the "Warming-up" phase for exploring the action space. After 200 'warm-up' episodes, the actual training phase starts and continues till 1600 episodes, so actual training is continued for 1200 episodes. Average waiting time is used as the unified evaluation measure, and the negative of average waiting time which is regarded as the reward plotted in Fig. 6, where solid lines represent the average reward, while shad represents the standard deviation. Training loss and reward are based on the synthetic-TIN, and our CCGN outperformed as compared to the state-of-the-art network in terms of convergence rate and reward against each episode after reaching the convergence. CCGN training curve gradually rises and achieves convergence rapidly, and when it reached stability, the reward value is −188.10±20. IHG-MA, IG-RL and MA2C have similar oscillational trends throughout the training, so these models are very dynamic in performance with respect to the different phases of the training. CGB-MATCS (Odeh et al., 2015) and Metalight have shown stability at the early stages of training and reward goes upward sharply, however, this trend becomes reversed and both models have started oscillating abruptly which shows that these models do not learn well-generalized initialization.

5.2.2. Comparative analysis

The generalization and transferability CCGN model is compared with the state-of-the-art model on the three evaluation metrics. All models are evaluated on the three traffic scenarios: SQ1, SQ2 and SQ3. Evaluation metrics are plotted in Fig. 7 8 9 against all three traffic scenarios shown in Fig. 5. The performance of CCGN shows that it generalizes on all three scenarios more effectively than other models. To evaluate the robustness, models are tested on variational traffic density and dynamically added traffic flows.

5.2.3. Evaluation on the SQ3 traffic scenario

Average waiting time (WT), plotted curves Fig. 7(a) of various models show that CCGN has surpassed other models. CCGN, CGB-MATSC and IHG-MA have shown similar and pretty close trends at the beginning of training, peaking at 17.34, 27.11, and 28.20 at the simulation time from 1750 to 2100, 2800 to 2900,

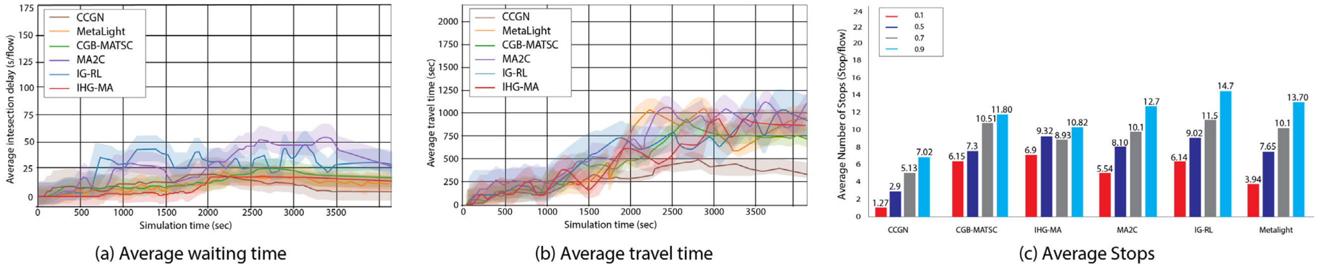


(a) Average waiting time

(b) Average travel time

(c) Average Stops

Fig. 7. Comparison of CCGN performance with other state-of-the-art models on SQ3.

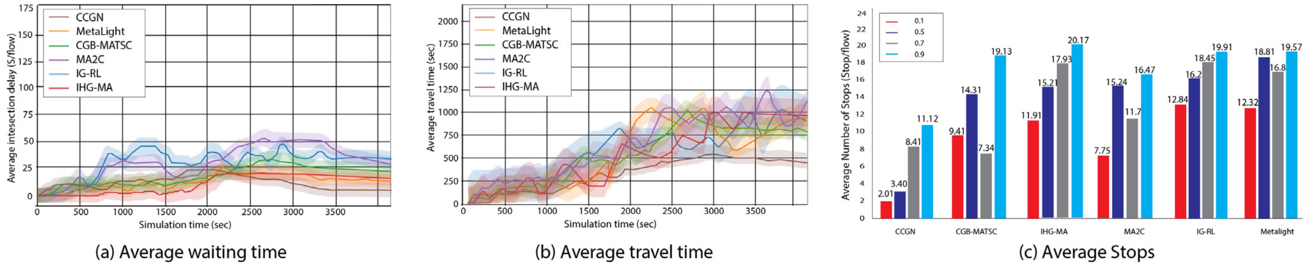


(a) Average waiting time

(b) Average travel time

(c) Average Stops

Fig. 8. Comparison of CCGN performance with other state-of-the-art models on SQ1.



(a) Average waiting time

(b) Average travel time

(c) Average Stops

Fig. 9. Comparison of CCGN performance with other state-of-the-art models on SQ2.

and 2000, 3150 respectively. Reported in Table 5, the standard deviation (Std) and mean μ, approximately 8.70 and 10.20, of CCGN are lower than that of CGB-MATSC (18.70 std and 25.50 μ) and IHG-MA(15.10 std and 21.70 μ). Although the MC2 A curve goes upward til simulation time 2650, starts oscillating and goes downward afterwards till 3400, however, increases at the end of the simulation journey. While the Metalight curve slowly goes up til simulation time 2500, but, there is a sharp increase in til 3300, and a downward trend till the end. IG-RL curve sharply goes upward before simulation time 1950, then starts decreasing a little bit till simulation 3500, and then stays plateau till the end. Furthermore, CCGN, CGB-MATSC, and IHG-MA curves show that average waiting time peaks at different simulation times, but peaks of these models remain lower than those of IG-RL and Metalight. After achieving the peaks, curves of CGN-MATSC and IHG-MA remain plateau which shows that traffic flows have to stop at most intersections. As more and more traffic flows are added in the network with the increase in simulation time, the average waiting for time curve of CGB-MATSC, IHG-MA, IG-RL and Metalight goes upward with time and then plateaus very high as compared to CCGN. This concludes that traffic flows stop at the most intersection and chances of stopping for the green light increase with the simulation time. On the contrary, our CCGN model has a peak from 1750 to 2100 simulation time duration, but the model consolidates itself and the curve goes down sharply and remains very below as compared to other models which shows that traffic flows are getting the signal freeway or have to wait for green light rarely. During the whole simulation time, the curve becomes zero several times, which means traffic flows do not need to wait at that time. This exhibits that CCGN is communicating effectively, and achieving our objective of a signal-free corridor once traffic flow has waited on the initial intersection.

Average travelling time (TT), reported in Fig. 7(b) is shorter when using CCGN than the compared models. As the traffic flows get on the signal when approaching subsequent intersections, traffic flows commute fast and reach the destination in less time. CCGN is showing a peak value of 980 at the 2900 simulation time which is due to the presence of the maximum number of traffic flows in the network since the beginning of the simulation. The number of Traffic flows in the network becomes maximum when there is at least one traffic flow between each intersection pair. When the number of traffic flows becomes maximum for the first time, CCGN takes little time for recovery, and then again travel time sharply decreases. Moreover, the peak value of CGB-MATSC is around 1400 at the simulation time 2550, while the IHG-MA peak value is 1180 at the simulation time 2730. Furthermore, std 680 and 799 μ of CCGN is lower than that of CGB-MATSC (std 935 and 1975 μ) and IHG-MA (std 855 and 2030 μ). The average travelling time curve of all models has oscillations, the reason is the conclusion of traffic flows dynamically. When new traffic

Table 5
Performance comparison on different traffic scenarios.

Model	Statistic	Traffic Scenarios					
		SQ1		SQ2		SQ3	
		Avg.WT	Avg.TT	Avg.WT	Avg.TT	Avg.WT	Avg.TT
IG-RL Devailly et al. (2021)	Mean	28.90	782.25	31.20	820.32	28.50	2055.25
	Std dev.	9.84	538.20	12.00	722.12	35.63	650
Metalight Zang et al. (2020)	Mean	14.32	881.28	16.51	923.91	23.90	2120
	Std dev.	8.74	642.50	11.87	780.18	17.45	950
MA2C Chu et al. (2019)	Mean	25.31	1047.32	26.21	1153.52	33.51	3045
	Std dev.	12.84	400.12	18.35	465.32	21.86	1011.58
IHG-MA Yang et al. (2021)	Mean	18.25	795.71	19.25	821.52	22.77	2030
	Std dev.	17.25	641.23	16.55	470.50	15.10	855
CGB-MATSC Wang et al. (2021)	Mean	19.20	715.56	23.20	793.70	25.50	1975
	Std dev.	11.53	191.12	146.50	202.43	18.70	935
CCGN	Mean	5.30	320.81	5.81	390.50	10.20	799
	Std dev.	3.47	210.33	4.79	245.35	8.70	680

flow is added, models need to reassess the present situation of TIN and adjust the settings of all the intersections to make space for the new traffic flows. We can also observe that the values of CGB-MATSC, and IHG-MA decreased but not indeed at zero, which shows that their trips are not completed, while the value of CCGN is zero which indicates that all trips are concluded.

Average stops (per traffic flow), reported in Fig. 7(c), are very lower when using CCGN than other models. The number of stops is compared at various traffic flow inclusion rates such as 0.1, 0.5,0.7 and 0.9. At the highest inflow rate, 0.5 (Traffic flow/min) Average stops of CCGN is 16.3, while CGB-MATSC, IHG-MA, and MA2C have 51.55, 59.5 and 92.70 stops respectively. The lower number of stops reduces the average waiting time and average travel time. Reducing the number of stops is the core objective which influences the average waiting and travelling time. Reducing the inflow rate, reducing the number of stops like when the inflow rate is 0.1 (Traffic flow/min), the number of stops becomes 9. A low inflow rate requires fewer communications between intersections, and traffic flows receive signals at most intersections. Metalight and IG-RL have the least ability to generalize their learning to unseen and dynamic traffic flows. The decentralized mechanism of these two models hinders the generalization and recovery mode when traffic flow is dynamic.

SQ3 is the most complex traffic network among all that include the SQ1 and SQ2 traffic networks, so we have analysed the performance in detail. Thus, the following parts about SQ1 and SQ2 analyse the diversion of trends from SQ3. This tells the behaviour of all models in different levels of complexity of traffic networks.

5.2.4. Evaluation on the SQ1 and SQ2 traffic scenario

CCGN has surpassed the other model in all the evaluation metrics plotted in Fig. 8 9. The performance of CCGN is relatively better on the SQ1 as compared to SQ2 and SQ3 because SQ1 is much simpler than others. As shown in Fig. 8(a)–9(a) it can be seen that the waiting time for SQ1 and SQ2 becomes zero most of the simulation time which shows that traffic flows have the signal-free corridor. Both. SQ1 and SQ2, also have the same behaviour in terms of average travel time which becomes zero at the end of simulation time which shows that CCGN is completing trips in both scenarios.

5.2.5. Impact of system scale

The proposed CCGN is impacted by the number of intersections in the traffic network. As the number of intersections increases, the performance of the CCGN system decreases and a negative correlation is observed between the network scale and performance.

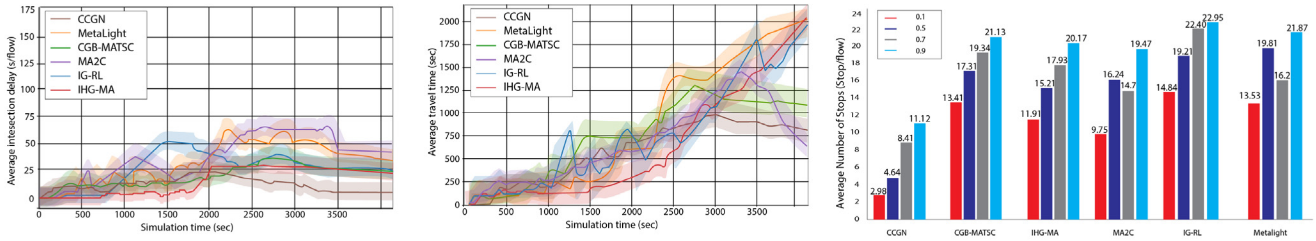
To demonstrate this relationship, three real-world traffic networks is analyzed: SQ1, SQ2, and SQ3. SQ1 has 15 intersections, SQ2 has 20 intersections, and SQ3 has 121 intersections. The analysis showed that as the number of intersections increases, the mean average waiting time and average travelling time also increase. For example, the mean average waiting time increased by 8.77% from 5.30 to 5.81 between SQ1 and SQ2, while a 43.03% increase was observed between SQ2 and SQ3. Similarly, the mean average travelling time increased by 17.84% and 51.12% from SQ1 to SQ2 and SQ2 to SQ3, respectively.

At a traffic flow ratio of 0.9, SQ3 had 23.48 more stops than SQ2 and 27.58 more stops than SQ1. This comparison highlights that the performance of the CCGN system decreases as the number of traffic intersections increases. However, the increase in the number of intersections does not have a proportional effect on the mean waiting time, average travelling time, and number of stops. For instance, there was a 25% increase in the number of intersections between SQ1 and SQ2, but only a 8.7% increase in the mean waiting time and a 17.84% increase in the average travelling time. Similarly, there was an 83.47% increase in the number of intersections between SQ2 and SQ3, but only a 43.03% increase in the mean average waiting time and a 51.12% increase in the travelling time. This inverse relationship was reversed in terms of the number of stops, where there was a 36.87% increase between SQ1 and SQ2, but there is a 79.71% increase between SQ2 and SQ3.

5.2.6. Evaluation on the synthetic traffic scenario

The CCGN model has performed better on the synthetic traffic scenario compared to the other state-of-the-art model plotted in Fig. 10. Fig. 10(a) shows that the average waiting time of the CCGN model is much lower as compared to the other models. Moreover, the average travel time per trip is also short compared to all other models. CCGN has a lower average travel time and Waiting time because CCGN is trained enough that traffic flows do not require stopping at any intersections anymore, once traffic flows have waited at the first intersection of the TIN. Although the synthesis traffic scenario has complexities of the real-world traffic scenarios, it still does not represent and handle the real-world dynamicity of the traffic behaviours. Reasons for the better performance of CCGN can be inferred from the explanation of Fig. 10.

Whole analysis Fig. 7 8 9 10 summarizes that CCGN model performance has surpassed the other state-of-the-art models in the real world as well as on the synthetic traffic scenarios. Training on



(a) Average waiting time

(b) Average travel time

(c) Average Stops

Fig. 10. Comparison of CCGN performance with other state-of-the-art models on synthetic traffic scenario.

the synthetic traffic scenario, and evaluation performance on the real-world scenarios shows that learnt policies are generalizable to real-world traffic scenarios.

6. Conclusion

In this paper, we study the problem of information sharing in the multi-intersection to provide signal-free or minimum-step solutions for waiting for traffic flow. This works proposes a DRL-based model, a centralized collaborative graph network (CCGN), for traffic intersection control. CCGN combines a local policy network (LPN) and a collaborative network for global policy named CCGN. Decentralized LPN processes the local information of traffic flow through Transformer network to predict signals' next actions to manage the traffic at all four sides of each intersection and share the spatio-temporal information of traffic flow to a centralized collaborative network CCGN. Global policy network, CCGN, receives spatio-temporal traffic flows information as input where each node features are fed to a FCN and then GCN to get the connectivity dependencies between reference nodes and decision node in order to output the Q-value through a Q-Network. The CCGN model effectiveness is evaluated against state-of-the-art models in terms of average waiting time, travel time, and number of stops at intersections. Trained CCGN models on synthetic traffic scenarios can be generalized to real-world scenarios.

However, the model has limitations in scalability and performance, such as its sensitivity to uniform traffic flow and decreasing probability of a signal-free corridor with increasing traffic flows and intersections. Future research aims to overcome these limitations and build a more accurate and robust model, potentially focusing on network-wide information sharing through experimentation with different settings.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Adil Afzal reports was provided by University of Engineering and Technology. Adil Afzal reports financial support and administrative support were provided by University of Engineering and Technology.

Data availability

Data will be made available on request

Acknowledgement

The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number MoE-IF-UJ-R-22-04100709-1.

References

Abbracciavento, Francesco, Zinnari, Francesco, Formentin, Simone, Bianchessi, Andrea G, & Savaresi, Sergio M (2023). Multi-intersection traffic signal control: A decentralized MPC-based approach. *IFAC Journal of Systems and Control*, *23*, Article 100214.

Araghi, Sahar, Khosravi, Abbas, Johnstone, Michael, & Creighton, Doug (2013). Q-learning method for controlling traffic signal phase time in a single intersection. In *16th international IEEE conference on intelligent transportation systems* (pp. 1261–1265). IEEE.

Calvo, Jeancarlo Arguello, & Dusparic, Ivana (2018). Heterogeneous multi-agent deep reinforcement learning for traffic lights control.. In *AICS* (pp. 2–13).

Chen, Sikai, Dong, Jiqian, Ha, Paul, Li, Yujie, & Labi, Samuel (2021). Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *36*, 838–857.

Chen, Deli, Lin, Yankai, Li, Wei, Li, Peng, Zhou, Jie, & Sun, Xu (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 34*, (pp. 3438–3445).

Chen, Miaojiang, Liu, Wei, Wang, Tian, Zhang, Shaobo, & Liu, Anfeng (2022). A game-based deep reinforcement learning approach for energy-efficient computation in MEC systems. *Knowledge-Based Systems*, *235*, Article 107660.

Chen, Lili, Lu, Kevin, Rajeswaran, Aravind, Lee, Kimin, Grover, Aditya, Laskin, Misha, et al. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, *34*, 15084–15097.

Cheng, Jialang, Wu, Weigang, Cao, Jiannong, & Li, Keqin (2016). Fuzzy group-based intersection control via vehicular networks for smart transportations. *IEEE Transactions on Industrial Informatics*, *13*, 751–758.

Chu, Tianshu, Wang, Jie, Codecà, Lara, & Li, Zhaojian (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *21*, 1086–1095.

Codeca, Lara, & Härri, Jérôme (2018). Monaco sumo traffic (most) scenario: A 3d mobility scenario for cooperative its. *EPiC Series in Engineering*, *2*, 43–55.

Darmoul, Saber, Elkosantini, Sabeur, Louati, Ali, & Said, Lamjed Ben (2017). Multi-agent immune networks to control interrupted flow at signalized intersections. *Transportation Research Part C (Emerging Technologies)*, *82*, 290–313.

Devailly, François-Xavier, Larocque, Denis, & Charlin, Laurent (2021). IG-RL: Inductive graph reinforcement learning for massive-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 7496–7507.

Ding, Chuan, Dai, Rongjian, Fan, Yue, Zhang, Zhao, & Wu, Xinkai (2022). Collaborative control of traffic signal and variable guiding lane for isolated intersection under connected and automated vehicle environment. *Computer-Aided Civil and Infrastructure Engineering*, *37*, 2052–2069.

El-Tantawy, Samah, Abdulhai, Baher, & Abdelgawad, Hossam (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, *18*, 227–245.

Foerster, Jakob, Assael, Ioannis Alexandros, De Freitas, Nando, & Whiteson, Shimon (2016). Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, *29*.

Genders, Wade, & Razavi, Saiedeh (2019). Asynchronous n-step Q-learning adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, *23*, 319–331.

Ghanim, Mohammad S., & Abu-Lebdeh, Ghassan (2015). Real-time dynamic transit signal priority optimization for coordinated traffic networks using genetic algorithms and artificial neural networks. *Journal of Intelligent Transportation Systems*, *19*, 327–338.

Graves, Russell T., Nelson, Zachariah E., & Chakraborty, Subhadeep (2021). A decentralized intersection management system through collaborative negotiation between smart signals. *Journal of Intelligent Transportation Systems*, 1–23.

Gu, Haotian, Guo, Xin, Wei, Xiaoli, & Xu, Renyuan (2021). Mean-field multi-agent reinforcement learning: A decentralized network approach. arXiv preprint arXiv:2108.02731.

Iqbal, Shariq, & Sha, Fei (2019). Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning* (pp. 2961–2970). PMLR.

Janner, Michael, Li, Qiyang, & Levine, Sergey (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, *34*, 1273–1286.

Jiang, Jiechuan, Dun, Chen, Huang, Tiejun, & Lu, Zongqing (2018). Graph convolutional reinforcement learning. arXiv preprint arXiv:1810.09202.

Kim, Gyeongjun, & Sohn, Keemin (2022). Area-wide traffic signal control based on a deep graph Q-network (DGQN) trained in an asynchronous manner. *Applied Soft Computing*, *119*, Article 108497.

Li, Zhenning, Yu, Hao, Zhang, Guohui, Dong, Shangjia, & Xu, Cheng-Zhong (2021). Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transportation Research Part C (Emerging Technologies)*, *125*, Article 103059.

Liu, Yong, Hu, Yujing, Gao, Yang, Chen, Yingfeng, & Fan, Changjie (2019). Value function transfer for deep multi-agent reinforcement learning based on N-step returns. In *IJCAI* (pp. 457–463).

Liu, Yong, Wang, Weixun, Hu, Yujing, Hao, Jianye, Chen, Xingguo, & Gao, Yang (2020). Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 34*, (pp. 7211–7218).

Lowe, Ryan, Wu, Yi I, Tamar, Aviv, Harb, Jean, Pieter Abbeel, OpenAI, & Mordatch, Igor (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, *30*.

Ma, Chengyuan, Yu, Chunhui, Zhang, Cheng, & Yang, Xiaoguang (2022). Signal timing at an isolated intersection under mixed traffic environment with self-organizing connected and automated vehicles. *Computer-Aided Civil and Infrastructure Engineering*.

Mousavi, Seyed Sajad, Schukat, Michael, & Howley, Enda (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, *11*, 417–423.

Niroumand, Ramin, Tajalli, Mehrdad, Hajibabai, Leila, & Hajbabaie, Ali (2020). Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. *Transportation Research Part C: Emerging Technologies*, *116*, Article 102659.

Odeh, Suhail M., Mora, A. M., Moreno, María N., & Merelo, J. J. (2015). A hybrid fuzzy genetic algorithm for an adaptive traffic signal system. *Advances in Fuzzy Systems*, *2015*, 11.

Van der Pol, Elise, & Oliehoek, Frans A. (2016). Coordinated deep reinforcement learners for traffic light control. In *Proceedings of learning, inference and control of multi-agent systems*: *Vol. 8*, (pp. 21–38).

Prashanth, L. A., & Bhatnagar, Shalabh (2011). Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In *2011 14th international IEEE conference on intelligent transportation systems* (pp. 1640–1645). IEEE.

Rahman, Md Mokhlesur, Najaf, Pooya, Fields, Milton Gregory, & Thill, Jean-Claude (2022). Traffic congestion and its urban scale factors: Empirical evidence from American urban areas. *International Journal of Sustainable Transportation*, *16*, 406–421.

Schlichtkrull, Michael, Kipf, Thomas N, Bloem, Peter, Berg, Rianne van den, Titov, Ivan, & Welling, Max (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference* (pp. 593–607). Springer.

Shen, Tao, Zhou, Tianyi, Long, Guodong, Jiang, Jing, Wang, Sen, & Zhang, Chengqi (2018). Reinforced self-attention network: A hybrid of hard and soft attention for sequence modeling. arXiv preprint arXiv:1801.10296.

Shou, Zhenyu, Di, Xuan, Ye, Jieping, Zhu, Hongtu, Zhang, Hua, & Hampshire, Robert (2020). Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning. *Transportation Research Part C (Emerging Technologies)*, *111*, 91–113.

Singh, Amanpreet, Jain, Tushar, & Sukhbaatar, Sainbayar (2018). Learning when to communicate at scale in multiagent cooperative and competitive tasks. arXiv preprint arXiv:1812.09755.

Van Hasselt, Hado, Guez, Arthur, & Silver, David (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 30*.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, Tong, Cao, Jiahua, & Hussain, Azhar (2021). Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *125*, Article 103046.

Wang, Yuanda, Liu, Wenzhang, Liu, Jian, & Sun, Changyin (2023). Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning-based control. *ISA Transactions*.

Wang, Yansen, Shen, Ying, Liu, Zhun, Liang, Paul Pu, Zadeh, Amir, & Morency, Louis-Philippe (2019). Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 33*, (pp. 7216–7223).

Wang, Min, Wu, Libing, Li, Man, Wu, Dan, Shi, Xiaochuan, & Ma, Chao (2022). Meta-learning based spatial-temporal graph attention network for traffic signal control. *Knowledge-Based Systems*, *250*, Article 109166.

Wang, Jiawen, You, Lan, Hang, Jiayu, & Zhao, Jing (2023). Pre-trip reservation enabled route guidance and signal control cooperative method for improving network throughput. *Physica A. Statistical Mechanics and its Applications*, *609*, Article 128405.

Wei, Hua, Xu, Nan, Zhang, Huichu, Zheng, Guanjie, Zang, Xinshi, Chen, Chacha, et al. (2019). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1913–1922).

Welling, Max, & Kipf, Thomas N. (2016). Semi-supervised classification with graph convolutional networks. In *J. international conference on learning representations*.

Wen, Yifan, Zhang, Shaojun, Zhang, Jingran, Bao, Shuanghui, Wu, Xiaomeng, Yang, Daoyuan, et al. (2020). Mapping dynamic road emissions for a megacity by using open-access traffic congestion index data. *Applied Energy*, *260*, Article 114357.

Wu, Qiang, Wu, Jianqing, Shen, Jun, Du, Bo, Telikani, Akbar, Fahmideh, Mahdi, et al. (2022). Distributed agent-based deep reinforcement learning for large scale traffic signal control. *Knowledge-Based Systems*, *241*, Article 108304.

Xie, Donghan, Wang, Zhi, Chen, Chunlin, & Dong, Daoyi (2020). Iedqn: Information exchange dqn with a centralized coordinator for traffic signal control. In *2020 international joint conference on neural networks* (pp. 1–8). IEEE.

Xu, Biao, Ban, Xuegang Jeff, Bian, Yougang, Li, Wan, Wang, Jianqiang, Li, Shengbo Eben, et al. (2018). Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, *20*, 1390–1403.

Xu, Keyulu, Hu, Weihua, Leskovec, Jure, & Jegelka, Stefanie (2018). How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Yan, Liping, Zhu, Lulong, Song, Kai, Yuan, Zhaohui, Yan, Yunjuan, Tang, Yue, et al. (2023). Graph cooperation deep reinforcement learning for ecological urban traffic signal control. *Applied Intelligence*, *53*, 6248–6265.

Yang, Shantian (2023). Hierarchical graph multi-agent reinforcement learning for traffic signal control. *Information Sciences*, *634*, 55–72.

Yang, Shantian, & Pang, Bo (2022). An inductive heterogeneous graph attention-based multi-agent deep graph infomax algorithm for adaptive traffic signal control. *Information Fusion*, *88*, 249–262.

Yang, Shantian, & Yang, Bo (2021). A semi-decentralized feudal multi-agent learned-goal algorithm for multi-intersection traffic signal control. *Knowledge-Based Systems*, *213*, Article 106708.

Yang, Shantian, Yang, Bo, Kang, Zhongfeng, & Deng, Lihui (2021). IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. *Neural Networks*, *139*, 265–277.

Yang, Shantian, Yang, Bo, Wong, Hau-San, & Kang, Zhongfeng (2019). Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm. *Knowledge-Based Systems*, *183*, Article 104855.

Yang, Shantian, Yang, Bo, Zeng, Zheng, & Kang, Zhongfeng (2023). Causal inference multi-agent reinforcement learning for traffic signal control. *Information Fusion*, *94*, 243–256.

Zang, Xinshi, Yao, Huaxiu, Zheng, Guanjie, Xu, Nan, Xu, Kai, & Li, Zhenhui (2020). Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 34*, (pp. 1153–1160).

Zhang, Rusheng, Ishikawa, Akihiro, Wang, Wenli, Striner, Benjamin, & Tonguz, Ozan K. (2020). Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *22*, 404–415.

Zhang, Rusheng, Leteurtre, Romain, Striner, Benjamin, Alanazi, Ammar, Alghafis, Abdullah, & Tonguz, Ozan K. (2019). Partially detected intelligent traffic signal control: Environmental adaptation. In *2019 18th IEEE international conference on machine learning and applications* (pp. 1956–1960). IEEE.

Zhu, Ruijie, Li, Lulu, Wu, Shuning, Lv, Pei, Li, Yafei, & Xu, Mingliang (2023). Multi-agent broad reinforcement learning for intelligent traffic light control. *Information Sciences*, *619*, 509–525.