Selective Inter-Intersection Communication for Partially Observable Corridor Traffic Control via Gated-Attention Graph MARL

Hamza Mukhtar
Computer Science Department, University of Engineering and Technology, Lahore, Pakistan
hamza.hm.mukhtar@gmail.com

Abstract

Urban traffic corridors exhibit strong upstream-downstream coupling (e.g., spillback, platoon dispersion, and downstream blocking), such that locally optimal signal actions can induce corridor-wide stop-and-go propagation and network breakdown. Recent reinforcement-learning (RL) and graph-based multi-agent traffic signal control (TSC) methods improve adaptability, but two deployment-critical gaps remain: (i) partial observability caused by sparse/noisy sensing and missing mid-link states, and (ii) scalable coordination in large corridors where dense communication and indiscriminate graph aggregation introduce redundancy, over-smoothing, and training instability. Moreover, existing approaches often address temporal inference and communication selectivity in isolation, or rely on static sparsification heuristics that cannot capture regime-dependent causal influences. To close these gaps, we propose Gated-Attention Multi-Agent Reinforcement Learning framework (GAMARL) for corridor-scale TSC that couples temporal reconstruction with communication-efficient coordination in a two-level Traffic Intersection Network (TIN). Each intersection agent uses a Transformer-based temporal encoder to infer short-horizon latent dynamics from partial observations. Coordination is achieved through a two-stage inter-agent module that performs hard link gating to prune weak or regime-irrelevant dependencies, followed by attention-weighted fusion to preserve directional dominance among retained upstream-downstream neighbors. At the corridor layer, a Central Cooperation Graph (CCG) applies degree-normalized masked graph fusion to support corridor-level credit assignment under centralized learning and decentralized execution. A progression-oriented reward penalizes downstream waiting after corridor entry to align local actuation with stop-minimizing corridor objectives. Evaluations in SUMO on three real-world corridors (SQ1-SQ3) demonstrate consistent improvements over six recent state-of-the-art baselines, including scalability to SQ3 with 121 intersections. GAMARL achieves 36.1±1.8 s AWT, 207.9±6.0 s ATT, and 1.48±0.07 stops on SQ1; 41.7±2.3 s, 225.4±7.0 s, and 1.61±0.08 on SQ2; and 48.5±2.8 s, 249.6±8.5 s, and 1.86±0.09 on SQ3, yielding up to ≈3-4% lower waiting time, ≈2% lower travel time, and ≈4-5% fewer stops compared to the strongest baseline.

Keywords: Deep Learning; Transformer; Partially Observable; Graph Neural Networks; Adaptive Traffic Signal Control; Multi-Agent Reinforcement Learning

1 Introduction

Traffic congestion imposes substantial societal, environmental, and economic burdens, including increased travel time, elevated greenhouse-gas emissions, and excessive fuel consumption [48], [68]. In response, automating signal operations through Adaptive Traffic Signal Control (TSC) has demonstrated considerable potential for mitigating recurrent and non-recurrent congestion by adjusting control decisions to time-varying demand [61]. Urban traffic corridors are characterized by strong upstream-downstream coupling: queue spillback, platoon dispersion, and downstream blocking can propagate across consecutive intersections and degrade network efficiency. A

broad spectrum of interdisciplinary optimization and control techniques has been explored to improve TSC performance, including fuzzy-rule systems [11] evolutionary search via genetic algorithms [47] immune-network-based optimization [15] and neural-network-driven controllers [31]. More recently, reinforcement learning (RL) has emerged as a compelling paradigm for TSC because it can continuously adapt to dynamic, real-time traffic conditions by learning control strategies from interaction data. In the conventional single-agent setting, an RL controller regulates an isolated intersection by interacting with the traffic environment under a Markov Decision Process (MDP) formulation. Within this framework, the controller maps observed traffic descriptors (e.g., cumulative delay, waiting time, and queue length) to control actions (e.g., phase switching, green extension/reduction, or cycle-length adjustment), thereby inducing a policy that is iteratively refined through trial-and-error. During learning, the agent repeatedly observes the environment, selects actions, and updates the policy controller network (PCN) to maximize the accumulated reward, balancing the exploitation of learned behavior with the exploration of alternative strategies [44], [81]. For example, model-free Q-learning has been applied to signal timing using queue length as the state representation and aggregated time delay as the reward signal [3]. Such state-action-reward formulations have proven effective for single-agent control at isolated intersections [23], [30].

Multi-intersection control is naturally a multi-agent problem, where each intersection operates with local sensing and limited visibility of corridor conditions. Partial observability is common in practice due to sparse sensing, limited penetration into the connected vehicle, and unreliable measurements. MAPOLight studies a partially observed V2I setting and shows that observation availability can affect stability and convergence of cooperative controllers [58]. Missing-data scenarios further complicate learning; DiffLight addresses offline TSC with missing observations and rewards by combining diffusion-based imputation with decision making and a spatiotemporal Transformer module [5]. In parallel, recent multi-intersection methods increasingly rely on graph-structured representations and spatiotemporal encoders to capture corridor dependencies, e.g., dynamic heterogeneous graph updates with memory-enhanced learning [79] and neighbor-level information modeling for large networks [70]. Representation quality has also been targeted explicitly; CLlight introduces contrastive learning to strengthen MARL representations for cooperative TSC [29]. These developments indicate that robust corridor control requires (i) structured spatiotemporal representations and (ii) mechanisms that can operate reliably when observations are incomplete.

To overcome these limitations, recent studies have advanced multi-intersection control from multiple complementary directions. [25] proposed a multi-objective optimization framework that jointly accounts for safety and efficiency through carefully designed reward formulations. [6] incorporated spatial context into heterogeneous-graph representations of traffic states, substantially strengthening perception and relational reasoning. [27] combined graph convolution with attention-based aggregation and temporal convolutional modeling to better capture spatiotemporal dynamics, while [28] emphasized that heterogeneous traffic-state characteristics can render indirect optimization insufficient for fully characterizing inter-agent interactions. From a game-theoretic perspective, [1] introduced a decentralized control scheme based on Nash bargaining, enabling flexible phase sequencing to accommodate fluctuating demand. More recently [40] presented a dynamic spatiotemporal graph-fusion architecture with parallel learning modules for richer feature extraction, and [2] developed a hybrid adaptive control algorithm tailored to heterogeneous urban traffic conditions.

A central challenge is that effective coordination often requires inter-intersection communication, but naive communication patterns (e.g., global pooling or dense exchange) can introduce redundant information, increase bandwidth and computation, and destabilize learning as the network scales. Communication learning in MARL has therefore focused on selectivity: when and how agents communicate and how messages are fused. Attention-based communication provides flexible weighting of peer information [38], while gating mechanisms can suppress unhelpful links

and reduce redundant exchange [51]. More recent work has highlighted that even after message-level sparsification, redundancy can persist in the integrated embeddings at the receiver side; DRMAC addresses this issue using dimensional analysis and redundancy-reduction mechanisms [53]. Contrastive objectives have also been applied to improve message representations; MAIL proposes information-preserving graph contrastive learning for multi-agent communication [21]. For corridor traffic control, these findings suggest that selectivity must be enforced both at the link level (which intersections influence each other) and at the fusion level (how retained messages are weighted and integrated).

To address the challenges of partial observability, communication redundancy, and scalable coordination in multi-intersection traffic corridors, we propose Gated-Attention Multi-Agent Reinforcement Learning (GAMARL) for the corridor-scale traffic signal control framework that explicitly couples partial-observation inference with selective inter-intersection coordination in a two-level graph architecture. The corridor is modeled as a Traffic Intersection Network (TIN) in which each junction operates as a decentralized agent on a Distributed Interaction Graph (DIG), encoding its local, noisy measurements via a Transformer-based temporal encoder to recover latent short-horizon dynamics (e.g., arrival waves and queue growth) under limited sensing. To prevent the instability and redundancy induced by dense message exchange, each agent augments its local embedding with a two-stage communication module that (i) applies hard gating to prune weak or regime-irrelevant neighbors and (ii) performs attention-weighted fusion over the retained links, yielding a sparse, decision-relevant context that aligns with upstream-downstream causality. At the corridor layer, a centralized coordinator operates over a Central Cooperation Graph (CCG) and performs degree-normalized graph fusion on masked node embeddings, producing a corridor representation that captures spillback, blocking, and progression dependencies while respecting the learned sparsity pattern. GAMARL is trained under a centralized-learning/decentralized-execution perspective using value-based RL with structured per-intersection phase-duration actions to avoid combinatorial joint-action explosion, and it incorporates curriculum growth to mitigate early-stage non-stationarity as network size increases. Finally, a two-tier reward design couples intersection-level congestion reduction with a corridor-level downstream waiting penalty (zeroed at the entry) to bias learning toward progression-consistent, stop-minimizing behavior rather than locally greedy discharge that propagates shockwaves across the corridor.

The following are the main contributions.

- We propose a Gated-Attention Graph Multi-Agent Reinforcement Learning (GAMARL) that jointly addresses partial observability and scalable inter-intersection coordination for multi-intersection traffic signal control.
- We model the corridor as a Traffic Intersection Network (TIN) with (i) a Distributed Interaction Graph (DIG) for decentralized intersection policies and (ii) a entral Cooperation Graph (CCG) for corridor-level.
- To address the critical challenge of communication redundancy and partial observability, , we introduce a two-stage inter-agent communication mechanism that performs hard link gating followed by attention-weighted fusion, while preserving causally dominant upstream-downstream dependencies.
- Each agent employs a Transformer-based temporal encoder to infer short-horizon latent dynamics (e.g., arrivals and queue growth) from limited and noisy local measurements.
- We adopt centralized learning with decentralized execution using value-based RL and structured per-intersection phase-duration actions to avoid joint-action explosion, and incorporate a curriculum growth strategy to mitigate early-stage non-stationarity as corridor size increases.

2 Related Work

2.1 Selective Inter-Agent Communication

Selective communication is fundamental to cooperative MARL under partial observability, where each agent's local observation is insufficient to resolve corridor-scale dependencies. Yet, indiscriminate exchange (e.g., all-to-all messaging or global pooling) introduces redundant or weakly relevant information, increases bandwidth and computational costs, and can destabilize learning as the agent population grows. A dominant mechanism is attention-based prioritization, which emphasizes informative peers while suppressing irrelevant senders. ATOC employs attentional communication to trigger coordination and integrate messages for cooperative decision-making [38], whereas TarMAC learns targeted addressee selection and aggregation, enabling agents to direct information to relevant recipients rather than broadcasting [16]. Since soft attention can still diffuse non-trivial mass over many candidates in dense interaction graphs, explicit sparsification is often introduced. IC3Net uses gating to modulate continuous communication and mute messages when unhelpful [51], and SchedNet schedules limited communication opportunities by selecting which agents should broadcast under bandwidth constraints [39].

Beyond structural sparsity, recent methods explicitly optimize message *informativeness* and reduce redundancy in the communicated representation. IMAC applies an information bottleneck objective to encourage compact, decision-relevant messages [60], while I2C investigates individually inferred communication to reduce reliance on explicit transmissions [20]. SMS formulates message selection via contribution estimation using Shapley-inspired utility reasoning [72], and DRMAC shows that redundancy can remain even after sparsification, motivating dimension-level redundancy analysis for improved efficiency [53]. In parallel, topology-learning approaches adapt the communication graph to task-dependent dependencies: CCT learns correlated communication topologies [22], and MAGIC combines learned communication decisions with graph-attention aggregation over a learned interaction graph [46]. Robustness under noise is increasingly emphasized: graph information bottleneck methods learn minimal sufficient representations to filter irrelevant variations [19], and contrastive objectives such as MAIL strengthen message embeddings by enforcing consistency across graph views [21].

Collectively, these works suggest that cooperation under partial observability benefits from *multi-level selectivity*: (i) structural sparsification (gating/scheduling) to control communication load [39, 51], (ii) fine-grained prioritization among retained peers (attention/targeting) [16, 38], and (iii) redundancy control and robustness in the fused message space (bottlenecks, utility estimation, representation regularization, and topology learning) [19, 21, 53]. For corridor traffic control, where decision-relevant dependencies are predominantly localized along upstream–downstream directions, these insights motivate a two-stage design that first prunes irrelevant inter-intersection links via gating and then applies attention to weight the remaining contributors, consistent with gated-attention communication mechanisms [46, 51].

2.2 Partial Observability for Traffic Networks

Urban traffic signal control is inherently partially observable because each intersection relies on localized, noisy sensing. This prevents direct observation of spillback, platoon dispersion, and latent demand, complicating multi-intersection coordination and making full-state assumptions brittle; surveys identify partial observability and sensing unreliability as key barriers to scalable RL-based deployment [43]. Corridor MARL is therefore commonly modeled as a Dec-POMDP/POSG, where agent i selects actions from its local observation ω_i^k (or short histories) rather than the latent global state, inducing learning challenges from unobserved traffic evolution and non-stationarity due to concurrently adapting neighbors, which intensify under limited CAV penetration and unreliable communication [58].

Mitigation approaches cluster into three complementary directions: (i) learned inter-agent

communication to enrich ω_i^k with decision-relevant neighborhood summaries, as in ALCORL and incentive-shaped messaging schemes [82, 83]; (ii) temporal inference from observation histories, where Transformer-based controllers recover latent dynamics (e.g., queue growth, platoon arrivals) via long-range dependency modeling [65]; and (iii) structure-aware spatial representation learning using graph encoders and sparse knowledge sharing to propagate corridor cues efficiently and stabilize training [24, 36]. In deployment, missing data from outages/occlusions further exacerbates partial observability; DiffLight couples diffusion-based generative modeling with reward conditioning to maintain robust control under missing observations [5]. Effective coordination typically requires at least one of communication enrichment [82, 83], temporal reconstruction [65], or structured spatial propagation with sparsity [24, 36], motivating selective inter-intersection messaging (e.g., gated/attention-weighted fusion) to reduce uncertainty without destabilizing learning [5, 58].

2.3 Graph-Based Intersection Networks

Urban traffic dynamics are strongly coupled and propagative such as spillback, platoon dispersion, and downstream blocking link the evolution of local states across connected junctions taht makes graph abstractions (intersections or lane/movement groups as nodes; road links as edges) a natural basis for scalable traffic signal control (TSC). Early graph-based MARL demonstrated that learned message passing can outperform handcrafted coordination by adapting inter-intersection influence to time-varying demand; CoLight operationalizes this with graph attention for index-free neighborhood aggregation and dynamic large-network coordination [67].

Recent methods replace static pooling with structured, expressive encoders to capture heterogeneous dependencies and improve stability. HG-M2I employs hierarchical graph representation learning with a mutual-information objective to tighten correspondence between observed traffic states and embeddings, improving performance and transferability [76]. Multi-level graph designs further disentangle within-intersection interactions (lane/movement competition) from cross-intersection coupling (corridor propagation); MGMQ instantiates an upper-level network graph and lower-level intersection graph using graph attention and GraphSAGE-style aggregation, augmented with action masking to support arbitrary phase sets and strong zero-shot synthetic-to-real transfer [62].

A key limitation of many GNN-based controllers is static adjacency, which can underfit time-varying correlations (e.g., shifting dominant upstream influences). DSMEL introduces adaptive updates for dynamic heterogeneous graphs and a dual-memory architecture that decouples spatial/temporal processing via multi-head attention [79], while FGLight models neighborhood heterogeneity through adaptive neighbor selection and weight-based attention to improve convergence and stability [70]. These results motivate selective, context-dependent graph fusion rather than uniform aggregation.

Robust corridor control further requires joint spatiotemporal modeling of queue growth, arrival waves, and discharge hysteresis. Spatiotemporal graph learning commonly fuses graph operators with temporal modules (TCN/GRU/ attention); STG4Traffic surveys these gated temporal-graph fusion patterns [42]. Heterogeneous formulations explicitly encode spatial and temporal edge types; HTSTGC constructs a heterogeneous spatio-temporal graph with gated fusion for joint feature integration [71]. In TSC, prediction-aware fusion improves proactivity: TG-MADDPG couples short-term forecasting with graph attention for anticipatory cooperative control [52]. Stability-oriented actor-critic integration is exemplified by G-DESAC, which combines GNN features with entropy-constrained SAC for single- and multi-intersection control [35]. Finally, auxiliary self-supervision can strengthen representations and generalization; CLlight applies contrastive learning to improve cooperative MARL embeddings, indicating that representation quality (not only topology) is central to coordination [29]. Collectively, these studies support graph-structured corridor controllers that employ expressive graph fusion, adaptively prioritize

relevant neighbors, and integrate spatiotemporal encoders to align local actuation with corridor-level objectives [29, 35, 52].

2.4 State, Action, and Reward Design

Corridor-scale RL for traffic signal control is highly sensitive to the joint design of (i) state/observation, (ii) action space, and (iii) reward, since these determine information availability, controllable degrees of freedom, and alignment of learning signals with corridor objectives [69]. This coupling becomes more critical in multi-intersection settings due to partial observability, upstream-downstream interactions, and scalability limits in sensing and optimization.

State/observation. Typical encodings rely on lane-level features (queues, waiting time, occupancy, arrivals/discharges), but high-resolution states increase computation and can be brittle under noisy or missing sensing. Recent work therefore prioritizes compact and consistent representations that preserve coordination cues while stabilizing training. Bouktif et al. show that explicitly aligning state and reward definitions improves convergence in deep RL controllers [4]. Under limited observability, MAPOLight targets low connected-vehicle penetration and adopts aggregation to reduce dimensionality without discarding coordination-relevant information [58]. When measurements are missing (e.g., sensor failures), robust pipelines integrate missingness handling with control; DiffLight couples generative modeling with reward-conditioned learning, implying that state designs must tolerate missing data without corrupting gradients [5].

Action design. Controllers commonly choose the next phase/phase group or a phase-duration decision subject to feasibility constraints (min/max green, clearance). Duration-based control directly shapes discharge opportunities and progression timing but introduces discretization choices (interval length and granularity). Overly short intervals induce oscillatory switching ("green flicker"), whereas overly long intervals reduce responsiveness under rapid demand shifts. PPO-based studies treat interval selection as a first-order variable: Huang and Qu adopt longer phase decision intervals [32], and Wang et al. compare multiple intervals (e.g., $5\,\mathrm{s}/10\,\mathrm{s}/15\,\mathrm{s}$), showing that timing materially affects efficiency and should be tuned jointly with state and reward [59].

Reward design. Standard rewards penalize delay, queues, stops, or travel-time proxies, yet corridor operation requires that local gains translate into network-level benefits (progression, spillback prevention, reduced stop-go propagation). This is typically enforced via explicit coordination and multi-objective shaping. Fang et al. propose a network-wide agent-coordinated framework whose reward jointly encodes safety, efficiency, and coordination [26]. In signal-free corridor formulations, reward alignment is even more central: Mukhtar et al. promote progression-like behavior using downstream stop/wait penalties within a centralized collaborative architecture [45]. Reviews highlight the growing use of context-adaptive, multi-objective rewards to balance efficiency, safety, and sustainability, while warning that improper weighting and unreliable observations can destabilize learning [69].

These studies supports: (i) compact observations for robustness and scalability under limited sensing [5, 58], (ii) phase-duration actions with carefully chosen decision intervals to avoid oscillations while retaining responsiveness [32, 59], and (iii) reward shaping that couples local efficiency with progression-oriented or multi-objective corridor terms [26, 45]. These findings motivate treating $(\text{state}, \text{action}, \text{reward})$ as an integrated specification in corridor MARL [69].

3 Problem Formation

Motivated by prior studies on graph-structured multi-agent learning for networked control [9], [17], [74] we model the corridor traffic-intersection network using a two-level graph formulation. At the intersection level, each junction is treated as an autonomous decision-making agent embedded in a distributed interaction graph (DIG), where it optimizes an intersection-specific

objective while exchanging structured information with its directly connected neighbors. At the corridor level, these local agents are coordinated through a central cooperation graph (CCG) that aggregates cross-intersection context, ensuring that locally efficient actions collectively support a corridor-scale target (e.g., progression-oriented, signal-free operation).

Within the DIG, the multi-intersection corridor control task is formalized as a stochastic game (Markov game), consistent with recent graph-based MARL formulations [8], [41]. Intersections are represented as graph vertices and road connectivity defines the edges, enabling message passing and interaction-aware reasoning over the network. Under this abstraction, cooperative behavior is realized by allowing nodes to share compact summaries of their local traffic conditions, while the CCG acts as a centralized coordination layer that consolidates these distributed signals into a network-level representation used to guide joint control decisions, in a manner analogous to centralized graph controllers [9].

3.1 TIN as a Graph Network

In a signal-free corridor setting, achieving smooth progression cannot be accomplished by optimizing each intersection independently. Even if a junction minimizes its own delay (local objective), corridor performance depends on coordinated information flow across upstream and downstream intersections (global objective). Because each intersection observes only a partial view of the corridor, we model the Traffic Intersection Network (TIN) as a collaborative graph that supports structured knowledge exchange, consistent with the two-level formulation introduced earlier [8], [9]. We represent the corridor as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each intersection $i \in \mathcal{V}$ is a node and each physical connectivity or control-dependency relationship is an edge $(i, j) \in \mathcal{E}$. Local sensing signals (e.g., queue lengths, approach counts, discharge indicators) are used for intersection-level actuation, while the graph structure enables the propagation of corridor-relevant context such as stream direction, inter-intersection spacing, expected arrival times, and residual queues. For example, when a platoon is held at the entry of the corridor, information describing the stream's status can be relayed to the next intersections and continuously updated as the stream advances, enabling downstream agents to anticipate arrivals and allocate right-of-way accordingly. This graph abstraction therefore captures both (i) local actuation needs and (ii) corridor-level dependencies that arise from traffic propagation.

To aggregate and filter information over $\mathcal{G}$, we employ a graph encoder that maps node features to intersection embeddings. Let the node-feature matrix at decision step k be $F^k \in \mathbb{R}^{M \times d_f}$ (with $M = |\mathcal{V}|$), and let the corridor connectivity be described by an adjacency matrix $A \in \{0, 1\}^{M \times M}$. A graph fusion operator produces intersection embeddings $Z^k = \{z_i^k\}_{i \in \mathcal{V}}$ that summarize both local state and neighborhood context. Importantly, corridor coordination requires selective dependence: the most informative neighbor for a given intersection is typically not the entire network but a small set of causally connected upstream/downstream nodes. In particular, when a stream is discharging through intersection i, the immediately upstream intersection often provides the strongest predictive signal for near-future arrivals, whereas during queue formation or spillback, downstream conditions become more critical. Thus, rather than assuming uniform relevance across neighbors (a common drawback in dense aggregation), the controller must learn to emphasize only decision-relevant dependencies [9].

To address this limitation explicitly, our corridor controller incorporates selective inter-intersection communication: a gating stage prunes weak or irrelevant links, and an attention stage assigns continuous weights among the retained neighbors. This mitigates over-smoothing and redundancy that can arise from repeatedly aggregating over dense neighborhoods and is consistent with relational MARL practices where learned embeddings are fed to a policy module for control [37]. The resulting intersection embeddings serve as the primary inputs to the local decision rule, enabling each agent to optimize its intersection-level objective while contributing to the corridor-level goal of signal-free progression through coordinated, context-aware control.

3.2 TSC as Markov Game Abstraction

Multi-intersection traffic signal control is naturally a multi-agent sequential decision problem, and a common way to formalize it is through a stochastic (Markov) game, which generalizes the single-agent MDP to multiple interacting decision makers. This abstraction is widely adopted in MARL because it provides a structured description of how agents jointly influence traffic evolution, even though learning stable joint policies (and computing equilibria) remains challenging in general [12], [66], [75]. In our corridor setting, we model the problem as a partially observable stochastic game with one agent per intersection. Let the set of controlled intersections be $\mathcal{V}$ with $M = |\mathcal{V}|$. At decision step k, the corridor has a latent global traffic state $X^k \in \mathcal{X}$, but each intersection $i \in \mathcal{V}$ only receives a local observation $\omega_i^k \in \Omega_i$ derived from its sensors and (selectively) communicated neighbor context. Each agent chooses an action $u_i^k \in \mathcal{U}_i$, and the joint action is $U^k = \{u_i^k\}_{i\in\mathcal{V}}$. The corridor dynamics evolve according to a controlled transition model $P\big(X^{k+1} \mid X^k, U^k\big)$, capturing how queues, arrivals, and discharges propagate across connected intersections.

Each agent receives an instantaneous performance signal $\rho_i^k \in \mathbb{R}$ (e.g., based on delay, queue, or stop-related measures), and policies are represented as decentralized mappings $\mu_i : \Omega_i \to \Delta(\mathcal{U}_i)$, where $\Delta(\cdot)$ denotes a distribution over feasible actions. The long-term objective for agent i is to maximize its expected discounted return $J_i = \mathbb{E}\Big[\sum_{h=0}^{\infty} \beta^h \rho_i^{k+h}\Big]$, where $\beta \in [0, 1)$. This formulation explicitly reflects the partial observability and interdependence across intersections that arise in corridor control, and it provides a consistent basis for incorporating our selective inter-intersection communication mechanism i.e., enriching ω_i^k only with decision-relevant neighbor information while optimizing coordinated policies under corridor-level objectives [12], [66], [75].

3.3 Global Action Space

At each decision epoch k, every intersection agent selects a control action from a finite discrete set that specifies both (i) the movement group to be served and (ii) the associated green holding duration. This phase duration formulation is widely used in RL-based TSC because it provides direct control over service allocation while respecting operational constraints [75], [73]. Formally, for intersection i, the action is $u_i^k \in \mathcal{U}_i$, where the candidate set consists of four phase options paired with discretized green times:

$$\begin{aligned}\mathcal{U}_i = \Big\{&(\psi^{\mathrm{NS}},\, \gamma_i^{k,1}),\ (\psi^{\mathrm{NSL}},\, \gamma_i^{k,2}),\\ &(\psi^{\mathrm{EW}},\, \gamma_i^{k,3}),\ (\psi^{\mathrm{EWL}},\, \gamma_i^{k,4})\Big\}.\end{aligned} \tag{1}$$

Here, ψ^{NS} denotes the North-South through movement, ψ^{NSL} denotes the North-South left-turn movement, ψ^{EW} denotes the East-West through movement, and ψ^{EWL} denotes the East-West left-turn movement. The scalar $\gamma_i^{k,c}$ (with $c \in \{1, 2, 3, 4\}$) is the green duration associated with selecting the c-th movement option at epoch k. This representation avoids an implicit assumption sometimes made in earlier formulations, namely that all intersections share identical feasible action sets by explicitly defining $\mathcal{U}_i$ per intersection. To ensure feasible and non-oscillatory operation, the green duration is constrained as $\gamma_i^{k,c} \in [\gamma_{\min}, \gamma_{\max}]$. In addition, a fixed clearance interval is enforced between consecutive phases; we denote the yellow time by γ^{yel} and set it as $\gamma^{\mathrm{yel}} = \gamma_{\max}/\kappa_{\mathrm{dec}}$, where κ_{dec} is a design constant. The remaining control and environment hyperparameters are summarized in Table 1.

3.4 Global observation space Ω

Corridor traffic control is partially observable because an individual intersection can only measure local conditions (e.g., queues and arrivals on its approaches) and cannot directly sense intermediate link states or downstream blocking beyond its sensing range. Therefore, at decision

Table 1: Environment, control, and graph-interface hyperparameters for the corridor TIN Markov game.

Parameters	**SQ1, SQ2, SQ3**
Traffic / simulation settings	
Traffic flow unit (per lane)	360 veh/lane/h
Demand multiplier (scenario index)	$C \in \{SQ_1, SQ_2, SQ_3\}$
Vehicle length	$l_v = 4.0$ m
Grid cell length (discretization)	$L_{\text{cell}} = 4.0$ m
Maximum speed	$v_{\max} = 48$ km/h
Maximum acceleration	$a_{\max} = 1.2$ m/s^2
Maximum deceleration	$a_{\text{dec}} = 2.5$ m/s^2
Markov game control constraints	
Controlled intersections	$M = \lvert\mathcal{V}\rvert$ (scenario dependent)
Phase options per intersection	$\Psi = \{\psi^{\text{NS}}, \psi^{\text{NSL}}, \psi^{\text{EW}}, \psi^{\text{EWL}}\}$
Green duration bounds	$\gamma \in [\gamma_{\min}, \gamma_{\max}]$, $\gamma_{\min} = 10$ s, $\gamma_{\max} = 60$ s
Yellow clearance time	$\gamma^{\text{yel}} = \gamma_{\max}/\kappa_{\text{dec}}$ (fixed per setting)
Discount factor	β
Graph observation	
Node-feature dimension	$d_f = 11$
Adjacency / dependency matrix	$A \in \{0,1\}^{M \times M}$
Intersection participation mask	$m^k \in \{0,1\}^M$

step k, each intersection agent $i \in \mathcal{V}$ receives a local observation $\omega_i^k \in \Omega_i$ derived from on-site sensors and selectively communicated neighbor context. For centralized coordination and graph-based fusion, we organize the corridor-level input into a structured form with three components: node attributes, inter-intersection dependency, and an intersection participation mask.

We define the structured global input at step k as $y^k = \big(F^k, A, m^k\big)$, where F^k is the node-feature matrix, A is the adjacency/dependency matrix, and m^k is a binary mask. This representation is designed to support scalable graph fusion while explicitly acknowledging that observations may be incomplete and noisy (i.e., some quantities are proxies rather than direct measurements). We model the TIN using heterogeneous descriptors that summarize intersection, traffic stream, and road/lane characteristics (Table 2). For each intersection i, we define a feature vector $f_i^k = \big(f_i^{I,k}, f_i^{S,k}, f_i^{R,k}\big)$, where $f_i^{I,k}$ contains intersection-level signals (e.g., queue and service status), $f_i^{S,k}$ characterizes the dominant traffic stream/platoon context relevant to corridor progression, and $f_i^{R,k}$ encodes road/lane-related descriptors. A key stream-related subset is computed as follows.

- We encode the active approach or stream direction at intersection i using a one-hot vector d_i^k (e.g., East $[1,0,0,0]$, West $[0,1,0,0]$, North $[0,0,1,0]$, South $[0,0,0,1]$). This avoids treating the signal "position" as a scalar category without structure.

- Since instantaneous speed can be noisy, we use a baseline-plus-deviation form: $\hat{v}_i^k = \bar{v}_i^k + \Delta v_i^k$, where $\bar{v}_i^k$ is a historical/rolling estimate and Δv_i^k captures recent variation. This makes explicit that speed is an estimate, not a ground-truth measurement.

- Because mid-link sensing is often unavailable, we infer a proxy for stream position on the

Table 2: Heterogeneous feature groups used to construct the intersection node attributes for the TIN graph in the proposed gated-attention MARL framework.

Feature group	**Attributes**
Intersection state $f_i^{I,k}$ (2-d)	Residual queue length q_i^k on controlled approaches (veh / normalized occupancy), Aggregated waiting-time (delay) proxy w_i^k on influenced movements (s / normalized).
Stream / platoon context $f_i^{S,k}$ (7-d)	Active stream direction one-hot $d_i^k \in \{0,1\}^4$ (East/West/North/South), Rolling (historical) speed estimate $\bar{v}_i^k$, Short-term speed deviation Δv_i^k, Inbound stream position proxy $\ell_i^k = p_i^{\text{in}} - (\bar{v}_i^k + \Delta v_i^k)\Delta_i^k$, where p_i^{in} is an entry reference and Δ_i^k is an elapsed travel/progression term.
Road / lane descriptors f_i^R (2-d)	Inter-intersection spacing / inbound link length L_i^{in} (m), Inbound lane count n_i^{lane} (or capacity proxy, scenario-dependent).

inbound link:

$$\ell_i^k = p_i^{\text{in}} - (\hat{v}_i^k \, \Delta_i^k), \tag{2}$$

where p_i^{in} is a reference entry location and Δ_i^k denotes an elapsed travel/progression term. This is treated as an approximate feature; the model learns its utility through training rather than assuming perfect accuracy.

Stacking all intersection feature vectors yields:

$$F^k = \left[f_i^k\right]_{i \in \mathcal{V}} \in \mathbb{R}^{M \times d_f}, \qquad d_f = 11, \tag{3}$$

where $M = |\mathcal{V}|$ is the number of controlled intersections and d_f is the feature dimension reported in Table 2.

The interaction structure among intersections is encoded by a binary matrix $A \in \{0,1\}^{M \times M}$, where $A_{ij} = 1$ indicates that information from intersection j is considered potentially relevant to intersection i. This matrix can represent physical connectivity and/or learned dependence structure; importantly, we do not assume that all connected neighbors are equally useful. Instead, the subsequent gated-attention communication module learns which edges should be emphasized or suppressed, addressing a common shortcoming of dense neighborhood aggregation (e.g., over-smoothing) in graph encoders. To handle variable corridor configurations (e.g., sub-corridors, inactive junctions, or partial deployment), we introduce a binary mask $m^k \in \{0,1\}^M$, which gates intersection embeddings after graph fusion. A value $m_i^k = 1$ includes intersection i in coordination, while $m_i^k = 0$ suppresses it. This avoids an implicit assumption often made in fixed-size formulations that every intersection is always active and fully observed. Overall, the global observation design $y^k = (F^k, A, m^k)$ provides a structured interface for corridor-level fusion, while each agent's local observation ω_i^k remains inherently partial. This motivates the selective communication mechanism in our model, where gated pruning and attention-weighted fusion are used to extract only decision-relevant inter-intersection context rather than performing indiscriminate aggregation [10], [8].

3.5 Reward function

To drive corridor-level progression while retaining intersection -level responsiveness, we employ a two-tier reward design consisting of (i) a local utility for each intersection and (ii) a corridor

utility that encourages signal-free movement after entry. This structure follows the principle of using frequently measurable and spatially decomposable feedback signals in multi-intersection RL, which helps stabilize learning and improves scalability [12], [75], [73]. Importantly, the corridor term is designed to encourage near-stop-free progression downstream; it does not assume that progression can be guaranteed under all demand regimes. For each intersection agent i at decision step k, we define a local reward that penalizes residual congestion and delay measured on its influenced approaches and neighbor-coupled movements. Let $\mathcal{N}_i$ denote the set of relevant interaction partners (e.g., adjacent intersections or coupled approaches) and let γ_i^k be the executed green holding duration associated with the selected phase. The local reward is

$$\rho_i^k \;=\; -\sum_{j\in\mathcal{N}_i}\Big(q_{i,j}^{\,k+\gamma_i^k} \;+\; d_{i,j}^{\,k+\gamma_i^k}\Big), \tag{4}$$

where $q_{i,j}^{\,k+\gamma_i^k}$ is the queue length (or queued vehicle count) associated with the movement/link influenced by neighbor j, evaluated after the action interval, and $d_{i,j}^{\,k+\gamma_i^k}$ is the corresponding aggregated delay. This definition emphasizes operationally observable quantities and avoids dependence on hard-to-measure corridor-wide latent variables, which is a common practical limitation in real deployments.

Local optimization alone can yield locally efficient but corridor-inconsistent behavior (e.g., pushing queues downstream or producing stop-and-go waves). To align local decisions with the corridor objective, we introduce a network reward that aggregates local utilities while penalizing downstream waiting after corridor entry. Let e be the designated entry intersection for a traffic stream, and let π_i^k denote the waiting-time penalty at intersection i for that stream at step k. We set $\pi_e^k = 0$ so that the model does not over-penalize the unavoidable entry hold, while penalizing waiting at downstream intersections to promote progression. The corridor reward is defined as

$$R_{\text{net}}^k \;=\; \frac{1}{M}\sum_{i\in\mathcal{V}}\Big(\rho_i^k \;-\; \pi_i^k\Big), \qquad \pi_e^k = 0, \tag{5}$$

where $M = |\mathcal{V}|$ is the number of controlled intersections. This construction encourages a regime in which the controller "pays" more for stops and waiting after vehicles have entered the corridor, thereby biasing coordination toward smoother upstream-downstream discharge patterns consistent with a signal-free progression goal. To maintain numerical stability and keep returns within a finite range during learning, we adopt a finite-horizon episodic setting. An episode begins when a traffic stream enters the controlled corridor and ends when it exits the TIN. This definition ties the effective horizon to corridor length and traversal dynamics, providing a consistent training protocol for corridor-scale coordination.

4 Proposed Model

We represent the Traffic Intersection Network (TIN) as a graph-structured corridor system in which intersections are modeled as graph nodes (Figure 1). The proposed approach separates control into two coupled layers to capture both intersection-level actuation and corridor-level coordination. At the local layer, each intersection is treated as an autonomous agent that controls its own movement groups (four-phase options) using locally observed traffic measurements. At the corridor layer, all intersections are connected through a central cooperation graph (CCG) that supports information fusion and enables the learning of coordinated actions that are consistent with corridor objectives (e.g., progression-oriented, signal-free operation). At each decision step k, the local controller at intersection i uses its local observation ω_i^k (queues, arrivals, discharge indicators, etc.) and augments it with selectively communicated neighbor context. To model within-intersection coupling among competing movements, we organize the movement groups of a junction using a mesh-style interaction pattern, allowing the controller to share information

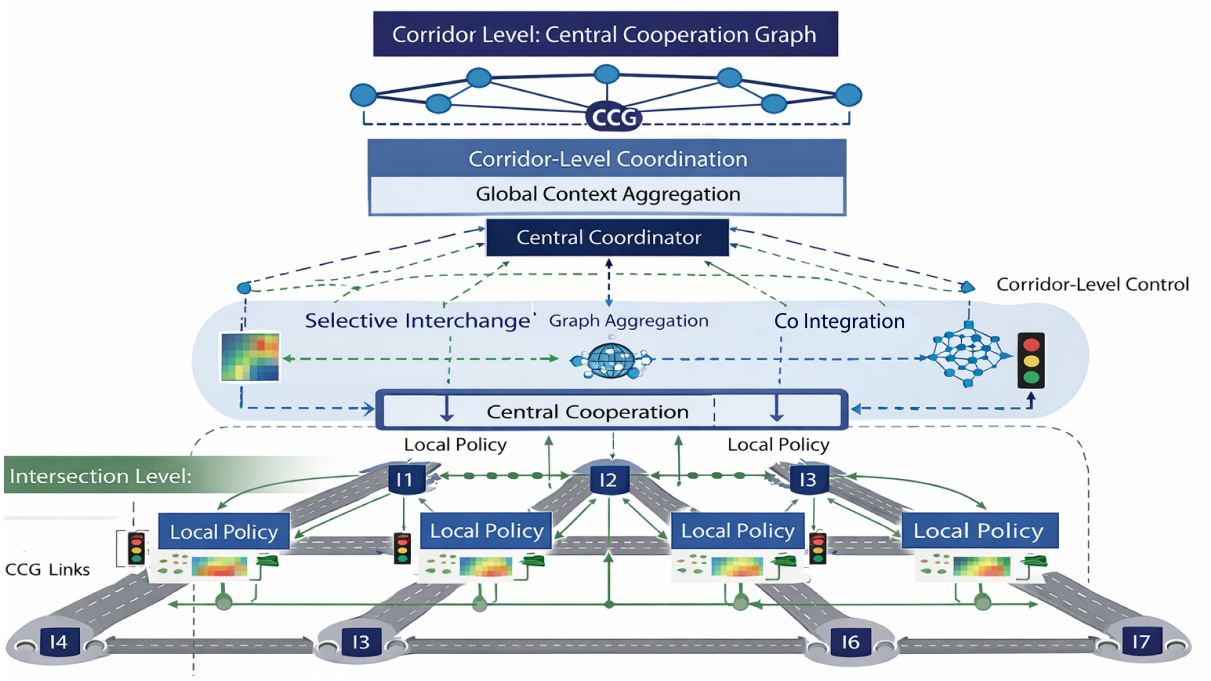

Figure 1: Illustration of Traffic Intersection Network (TIN) and the proposed two-level control network. The corridor is modeled as a Central Cooperation Graph (CCG) (top), where a central coordinator performs global context aggregation over intersection embeddings to support corridor-level coordination. At the intersection layer (bottom), each junction operates a local policy within a Distributed Interaction Graph (DIG), using locally observed traffic cues (e.g., queue-length/flow descriptors) to produce phase-duration actions. Selective inter-intersection communication is implemented via gated-attention message exchange: DIG links capture local neighbor interactions, while CCG links convey compact corridor context for coordination and integration.

across movements that compete for right-of-way. In parallel, corridor-level dependencies are captured through the CCG, where each intersection exchanges compact summaries with relevant upstream/downstream neighbors. Unlike dense message aggregation, which can introduce redundancy and dilute informative signals, our framework uses a gated-attention communication mechanism: a gating stage suppresses weak or irrelevant inter-intersection links, followed by an attention stage that assigns continuous importance weights to the retained neighbors. This design directly targets scalability and stability under partial observability.

For learning, we combine two complementary training principles. First, we adopt a multi-agent curriculum strategy that gradually increases the number of concurrently trained intersections, which helps stabilize optimization in large networks and reduces early-stage non-stationarity [41], [49]. Second, we follow a centralized learning perspective for corridor coordination, where global context is available to the coordination module during training while each intersection still executes decisions based on its own policy interface and receives messages [9]. To encode spatiotemporal traffic patterns efficiently, we use a Transformer-style encoder for representation extraction rather than recurrent models, enabling parallel processing and better handling of longer temporal dependencies [56]. Spatial interaction modeling is handled by the graph fusion operator over the CCG (e.g., graph convolution/graph attention style aggregation). Finally, the corridor-level decision module is implemented using a value-based RL controller (Deep Q-learning), where the learned corridor embedding guides action evaluation; to avoid the common drawback of combinatorial growth in joint action spaces, the decision structure is implemented in a factorized/structured manner consistent with per-intersection phase-duration actions.

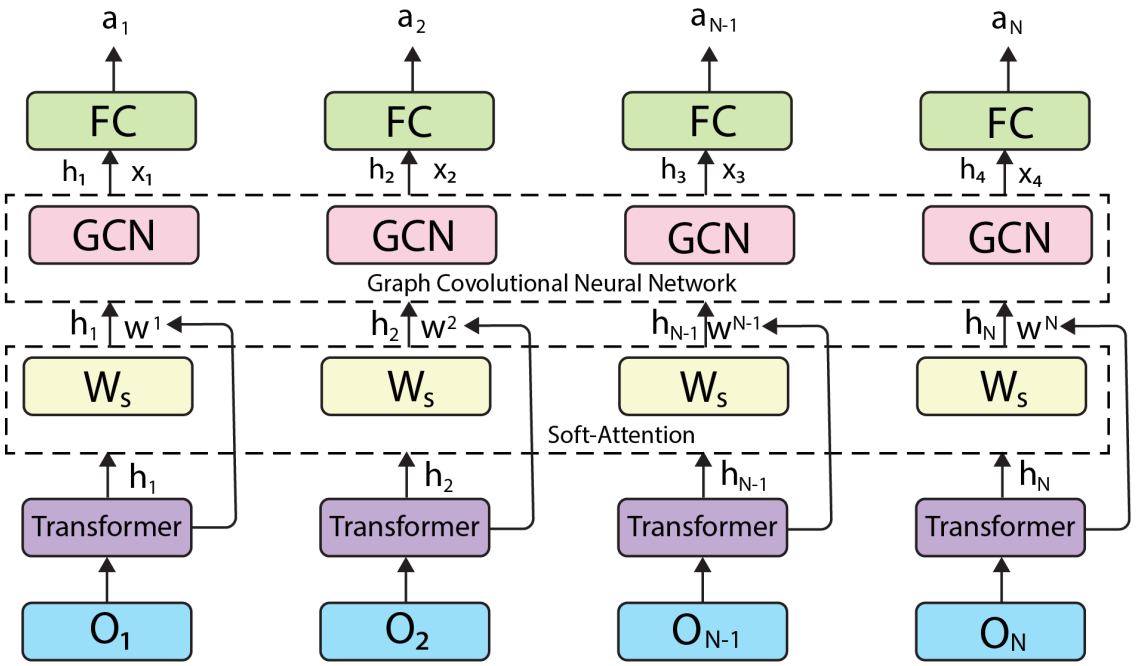

Figure 2: Local policy controller architecture for each intersection agent. At decision step k, the agent's partial observation O_i is encoded by a Transformer to produce a latent feature h_i. A soft-attention projection (W_s) computes relevance weights over communicated neighbor embeddings, enabling selective aggregation of inter-intersection information. The resulting interaction-aware representation is processed by a graph convolutional module (GCN) to fuse spatial dependencies and obtain an updated node embedding. Finally, a fully connected (FC) action head maps the fused features to the local control output a_i (phase-duration decision), supporting scalable signal control under partial observability through gated/attention-based communication and graph-based fusion.

4.1 Local policy network

Prior communication-based MARL methods often compress all agents' messages into a single pooled vector and broadcast this aggregate back to every agent [33], [37]. While computationally convenient, dense pooling can be problematic in corridor TSC because it encourages each agent to ingest information from many irrelevant peers. In practice, a softmax-based weighting may assign non-zero probability mass to a large number of agents, which (i) reduces interpretability of learned dependencies, (ii) weakens the influence of a few truly critical neighbors, and (iii) increases the risk of overfitting when the aggregated vector is mapped through deep networks' This is particularly undesirable in corridor control, where an intersection's decision is usually dominated by a small set of upstream/downstream intersections and not by the entire network.

In our framework as shown in Figure 2, at decision step k, each intersection agent i receives its local observation $\omega_i^k \in \Omega_i$. Because the environment is partially observable, we first encode ω_i^k into a compact representation and then construct a collaboration context from selectively chosen peers. Specifically, we define an observation encoder $\mathcal{E}(\cdot)$ and compute a feature embedding $h_i^k = \mathcal{E}(\omega_i^k)$. The local control decision is produced by a policy μ_i that conditions on both the encoded feature and a neighbor-derived context vector c_i^k:

$$u_i^k \sim \mu_i\left(h_i^k,\, c_i^k\right). \tag{6}$$

Here, u_i^k is the phase-duration action, h_i^k summarizes the local traffic state, and c_i^k captures

decision-relevant inter-intersection information. To improve representation capacity without the sequential overhead of recurrent models, we adopt a Transformer-style encoder to extract position/temporal-aware features [56], [34]. Inspired by Transformer-based encoders used in cooperative learning settings [7], we compute

$$\tilde{h}_i^k = \mathrm{TransEnc}\Big(h_i^k\Big) = \mathrm{TransEnc}\Big(\mathcal{E}(\omega_i^k)\Big). \tag{7}$$

This avoids a common drawback of LSTM-based communication models namely limited parallelism and increased parameter burden when scaling to many agents [41]. To prevent message overflow and to make cooperation scalable, we use a two-stage mechanism that (i) prunes irrelevant peers and (ii) weights retained peers according to relevance. For each candidate contributor j, we compute a hard gate and a soft score:

$$g_{i\leftarrow j}^k = \mathcal{M}_{\mathrm{gate}}\Big(\tilde{h}_i^k, \tilde{h}_j^k\Big), \qquad a_{i\leftarrow j}^k = \mathcal{M}_{\mathrm{att}}\Big(W_s\tilde{h}_i^k, \tilde{h}_j^k\Big), \tag{8}$$

where $g_{i\leftarrow j}^k \in \{0,1\}$ (or a near-binary relaxation) removes weak/irrelevant dependencies, $a_{i\leftarrow j}^k \in \mathbb{R}$ is an attention score, and W_s is a learnable projection. The effective contribution coefficient is $\alpha_{i\leftarrow j}^k = g_{i\leftarrow j}^k \cdot a_{i\leftarrow j}^k$. The collaboration context is then computed as a gated, weighted sum over neighbor embeddings:

$$c_i^k = \sum_{j\in\mathcal{V}} \alpha_{i\leftarrow j}^k \tilde{h}_j^k = \sum_{j\in\mathcal{V}} g_{i\leftarrow j}^k\, a_{i\leftarrow j}^k\, \tilde{h}_j^k. \tag{9}$$

This construction corrects a common limitation in fully soft attention: rather than distributing weight across many weak contributors, it explicitly enforces sparsity through gating, which better matches corridor causality (dominant upstream/downstream influences). The resulting policy uses c_i^k to incorporate only decision-relevant inter-intersection signals, improving robustness under partial observability and enhancing scalability to longer corridors.

4.2 Collaborative Global Policy Network

At each decision epoch k, the centralized coordinator interacts with the corridor environment by observing a structured corridor input, selecting a coordinated control decision, and receiving the resulting feedback as shown in Figure 3. We summarize this interaction using the transition tuple $\big(y^k,\, U^k,\, R_{\mathrm{net}}^k,\, y^{k+1}\big)$, where y^k is the global structured observation, U^k is the selected corridor control (a structured collection of per-intersection actions), and R_{net}^k is the corridor reward. Consistent with Section 3.2.2, we define the global input as $y^k = \big(F^k,\, A,\, m^k\big)$, where $F^k \in \mathbb{R}^{M\times d_f}$ is the node-feature matrix over the $M = |\mathcal{V}|$ intersections, $A \in \{0,1\}^{M\times M}$ is the dependency (adjacency) matrix, and $m^k \in \{0,1\}^M$ is an intersection participation mask. Unlike formulations that implicitly assume full participation and uniform neighbor relevance, this design allows partial deployment and supports selective fusion (via the mask and the gated-attention mechanism). We first map raw node features into a latent embedding space using a lightweight feed-forward encoder:

$$H^k = \Phi_{\mathrm{enc}}(F^k) \in \mathbb{R}^{M\times d}, \tag{10}$$

where d is the embedding dimension. The projected features are then fused over the corridor graph using a graph propagation operator. With a self-loop augmented adjacency $\tilde{A} = A + I$ and corresponding degree matrix $\tilde{D}$, the fused representation is computed as

$$G^k = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\,\tilde{A}\,\tilde{D}^{-\frac{1}{2}}\,H^k W_g + b_g\Big), \tag{11}$$

where $\sigma(\cdot)$ is a pointwise nonlinearity and W_g, b_g are learnable parameters. In practice, we avoid excessive stacking of graph layers because deep graph propagation can lead to over-smoothing,

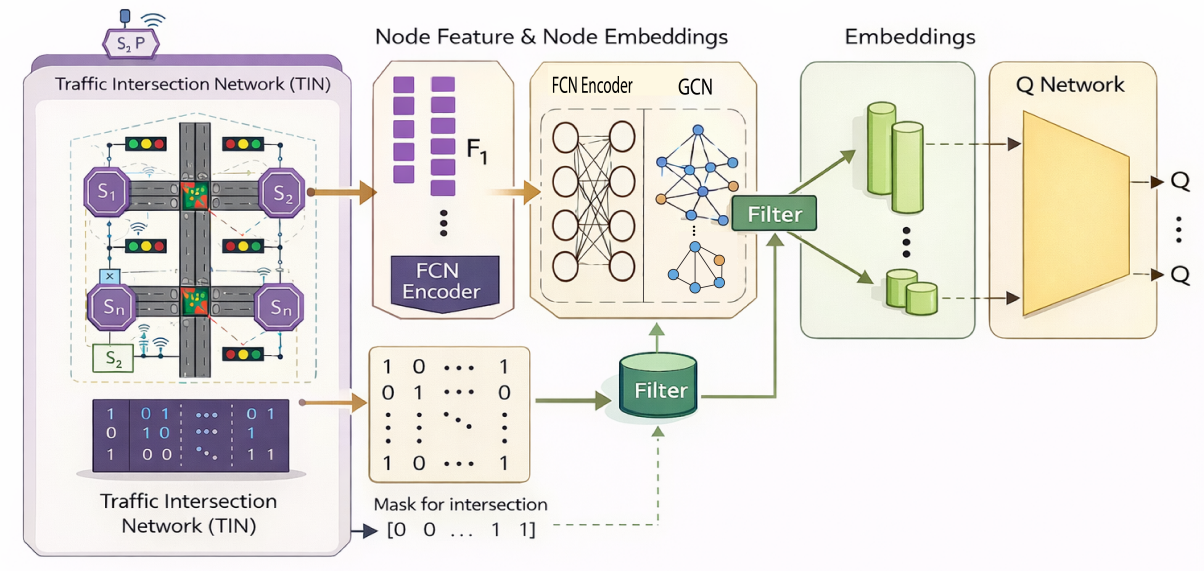

Figure 3: Local policy controller architecture for each intersection agent. At decision step k, the agent's partial observation O_i is encoded by a Transformer to produce a latent feature h_i. A soft-attention projection (W_s) computes relevance weights over communicated neighbor embeddings, enabling selective aggregation of inter-intersection information. The resulting interaction-aware representation is processed by a graph convolutional module (GCN) to fuse spatial dependencies and obtain an updated node embedding. Finally, a fully connected (FC) action head maps the fused features to the local control output a_i (phase-duration decision), supporting scalable signal control under partial observability through gated/attention-based communication and graph-based fusion.

making node embeddings less distinguishable. After graph fusion, we apply the intersection mask to retain only active nodes:

$$Z^k = \mathrm{diag}(m^k)\, G^k. \tag{12}$$

This explicitly corrects a common shortcoming of corridor models that assume a fixed set of always-active intersections. The corridor embedding Z^k is passed to a value function approximator that evaluates the quality of a candidate corridor action. Let Q_θ be a parameterized value model. We compute

$$\widehat{Q}_\theta\left(y^k, U^k\right) = Q_\theta\left(Z^k, U^k\right). \tag{13}$$

A practical concern is that a fully enumerated joint action over all intersections can grow combinatorially. To avoid this drawback, U^k is handled in a structured manner consistent with per-intersection phase-duration actions (Section 3.2.1), rather than treating corridor control as a single monolithic action with exponential size.

Training uses an experience replay buffer and mini-batch updates for stability. Following stable value-learning practices [54] we minimize the squared temporal-difference error over a mini-batch of size B:

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{b=1}^{B}\left(\eta_b - \widehat{Q}_\theta(y^k, U^k)\right)^2, \tag{14}$$

where η_b is the bootstrapped target value (computed using a target network and a max over next-step actions). This learning setup complements the gated-attention communication at the local policy level by providing a corridor-level critic that is sensitive to upstream-downstream dependencies while remaining numerically stable.

5 Experiment and Result

We evaluate the proposed gated-attention graph MARL framework in the SUMO microscopic traffic simulator [14]. Experiments are conducted under three real-traffic corridor scenarios (SQ1, SQ2, SQ3) adopted from prior benchmark studies [77, 78]. Within SUMO, the roadway geometry, traffic demand profiles, and intersection signal operations are explicitly modeled, enabling controlled replication of corridor dynamics. For training, the corridor control task is instantiated as a (partially observable) Markov game consistent with our formulation: each intersection agent operates over the defined observation interface and selects phase-duration actions to optimize the specified reward. Performance is assessed by comparing the proposed method against representative state-of-the-art baselines using standard intersection-level effectiveness metrics, including average waiting time, average travel time, and average number of stops, reported per traffic-flow setting.

5.1 Experimental setup

In this subsection, we describe the experimental configuration and key implementation details. The proposed corridor controller is trained on synthetic traffic environments to stabilize learning under controlled demand variations, and is subsequently evaluated and benchmarked on real-world corridor networks. All experiments are executed in the SUMO microscopic simulation platform, which provides a reproducible setting for modeling corridor geometry, traffic flows, and intersection-level control dynamics.

5.1.1 Training road network

For training, we construct a synthetic Traffic Intersection Network (TIN), denoted as $\text{TIN}^{a,b}$, where $a \in [1, 4]$ and $b \in [1, 6]$, following commonly used synthetic corridor configurations in prior work [18, 78]. The resulting network contains 25 intersections arranged using a mixture of two canonical junction motifs ("−" and "+"), enabling diverse corridor interaction patterns. Each road segment is modeled with three lanes; the total modeled corridor length is approximately 2100 m, and the lane width is set to 1.8 m.

To train the proposed corridor-level coordination module (CCG-based controller), we sample multiple network instantiations from this synthetic family and expose the agents to both unidirectional and bidirectional traffic streams. The traffic load is varied from 60 to 460 vehicles to improve robustness across demand regimes, while other simulator and control parameters are fixed as summarized in Table 1. To emulate progression-oriented operation within the corridor, the synthetic setup prioritizes the North-South movement with a demand of 420 veh/lane/hour, while the West-East movement is assigned a lower demand of 140 veh/lane/hour. Origin-destination endpoints are randomly assigned to boundary edges, and each run is initialized with different random seeds to encourage diverse state transitions and prevent overfitting to a narrow set of traffic realizations.

Training is executed using 20 parallel simulation instances, totaling approximately 1200 episodes (≈ 1.224 million interaction steps), with 1020 steps per episode. Experiments are run on a workstation equipped with an NVIDIA GTX 3080 GPU, 128 GB RAM, and a 6 GB frame buffer. On average, one episode requires 7.2 minutes, leading to an overall training time of 144.23 hours. To avoid excessively long rollouts caused by vehicles failing to clear the corridor within a reasonable time, we impose an episode timeout: any episode exceeding 12 minutes is terminated early to prevent simulation stagnation.

5.1.2 Evaluation road network

Following established evaluation protocols in traffic-signal control research [78, 80], we assess generalization of the proposed gated-attention graph MARL controller on three real-traffic

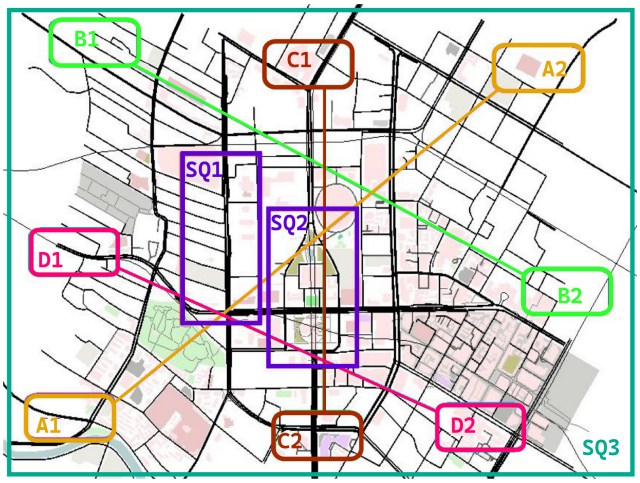

Figure 4: Overview of the three real-world corridor scenarios used for evaluation (SQ1-SQ3). The map illustrates the selected sub-networks (purple boxes) extracted from the urban road graph and the corresponding boundary inflow/outflow interfaces used to generate traffic demand. Colored labels (A1-A2, B1-B2, C1-C2, D1-D2) denote paired entry/exit endpoints that define directional OD streams traversing the corridor, while the thick black links indicate the primary arterial segments within each scenario.

network scenarios, denoted SQ1, SQ2, and SQ3 Figure 4. These scenarios are used to benchmark corridor-level coordination performance against representative state-of-the-art baselines. Scenarios SQ1 and SQ2 contain 15 and 20 intersections, respectively, whereas SQ3 is a larger composite network with 121 intersections that subsumes the junctions appearing in SQ1 and SQ2. Topologically, SQ1 is a representative branch-to-arterial configuration comprising multiple feeder branches connected to a main corridor, while SQ2 corresponds to a rarer mixed-layout network that combines triangular and rectangular substructures. Scenario SQ3 covers the full set of intersections and exposes the controller to a wider range of operational conditions and more complex interaction patterns, making it particularly suitable for stress-testing scalability and robustness. The overall configuration of these scenarios follows prior studies [63, 78], and the scenario descriptors are summarized in Table 3.

To ensure consistent and realistic microscopic dynamics across scenarios, the simulation parameters follow the settings in Table 1: the base per-lane flow unit is fixed at 360 veh/lane/hour, and traffic demand is scaled using a set of flow multipliers $\{1, 3, 5, 7, 9, 7, 5, 3, 1\}$ to generate time-varying demand profiles. Vehicle length and cell length are set to 4 m, the maximum speed is 48 km/h, and the acceleration/deceleration bounds are 1.2 m/s^2 and 2.5 m/s^2, respectively. Signal timing constraints use $\gamma_{\min} = 10$ s and $\gamma_{\max} = 60$ s, with a fixed yellow clearance interval of $\gamma^{\text{yel}} = 3.30$ s. These settings define feasible vehicle motion and phase actuation while preserving the realism of real-traffic corridor dynamics.

The considered networks include both "+" and "−" intersection geometries, leading to heterogeneous local conflict patterns and non-uniform corridor dependencies. All methods are evaluated under dynamic inflow conditions. Consistent with [78], we generate four traffic-flow sets (e.g., $(A1 \leftrightarrow A2)$, $(B1 \leftrightarrow B2)$, $(C1 \leftrightarrow C2)$, and $(D1 \leftrightarrow D2)$), where flows are injected at the designated entry junction with a stochastic inter-arrival interval and a time-varying vehicle count governed by the selected multiplier schedule (Table 1). Inflows for sets (A) and (B) are initiated every 500 s starting from $t = 0$ s, while sets (C) and (D) follow the same pattern but begin at 700 s, creating asynchronous stream interactions and more diverse upstream-downstream coupling. For statistical reliability, both training and evaluation are repeated over 10 independent random seeds. During evaluation, we run 10 parallel simulations per seed (totaling 100 simulation runs) and report aggregated performance across all runs. This protocol reduces sensitivity to

stochasticity in demand generation and microscopic vehicle interactions, and yields a more stable estimate of comparative performance.

5.2 Evaluation Metrics

We report performance using three standard effectiveness indicators commonly adopted in RL-based traffic signal control. **Average waiting time (AWT)** defined as the mean time spent by vehicles waiting/queuing at intersections over an episode, aggregated per traffic-flow setting [13, 78]. **Average travel time (ATT)** defined as the mean end-to-end trip duration for all vehicles from origin entry to destination exit [18, 80]. **Average number of stops** defined as the mean count of full stops experienced by vehicles across all intersections and flows in the network. This metric is particularly relevant to our progression-oriented objective, since reducing stop-and-go behavior is a key requirement for near signal-free corridor operation. In practice, lower stop counts typically co-occur with reduced waiting and travel times.

All algorithms are evaluated on the real-traffic networks SQ1-SQ3 that are not observed during training, under the dynamic inflow patterns described in Section 5.1.2. For computational consistency, each evaluation rollout is simulated for up to 3 h, after which the run is terminated to bound runtime on GPU-enabled execution. Each evaluation condition is repeated 10 times with distinct random seeds and stochastic demand realizations (randomized injection times and vehicle counts). Both unidirectional and bidirectional stream configurations are considered to assess robustness under asymmetric and opposing flows. Following the parallel evaluation protocol described earlier, experiments are executed using multiple concurrent SUMO instances and results are aggregated across all evaluation episodes; the same repetition strategy is applied during training.

5.3 Implementation Details

All experiments are conducted in the SUMO microscopic traffic simulator using the TraCI interface for step-wise control and logging [14]. The proposed controller is trained on a synthetic corridor family $\text{TIN}^{a,b}$ with 25 intersections ($a \in [1,4]$, $b \in [1,6]$), constructed by combining "+" and "−" junction motifs and three-lane road segments (total corridor length $\approx$ 2100 m; lane width 1.8 m). Evaluation is performed on the three real-traffic corridor scenarios SQ1-SQ3 (Figure 4), where SQ1 and SQ2 contain 15 and 20 intersections, respectively, and SQ3 contains 121 intersections and subsumes SQ1/SQ2. All scenarios use the same motion and signal constraints reported in Table 1.

At each decision step k, the centralized coordinator receives the structured input $y^k = (F^k, A, m^k)$ (Section 3.2.2). The node-feature matrix $F^k \in \mathbb{R}^{M \times d_f}$ stacks per-intersection descriptors with $d_f = 11$ (Table 2). The binary adjacency matrix $A \in \{0,1\}^{M \times M}$ is derived from the physical corridor topology (road connectivity), and the mask $m^k \in \{0,1\}^M$ enables filtering of inactive or excluded intersections (e.g., sub-corridor selection). The local observation ω_i^k for each intersection agent is formed from locally measurable signals (queues, delay proxy, discharge/arrival cues) and selectively communicated neighbor context.

The proposed method follows a two-level design (Figure 1): a local policy module operating on the distributed interaction graph (DIG) and a corridor-level module operating on the central cooperation graph (CCG).

- Local policy per intersection. As shown in Figure 2, each agent encodes ω_i^k using a Transformer-style encoder to obtain $\tilde{h}_i^k$ [57]. Selective inter-intersection communication is implemented using a two-stage mechanism: a learned gate produces a sparse neighbor set, and a soft-attention projection W_s weights retained neighbor embeddings to form the context c_i^k (Eqs. (6)-(9)). The action head outputs a phase-duration decision $u_i^k \in \mathcal{U}_i$ (Section 3.2.1). To improve scalability across varying corridor sizes, we use parameter

sharing across intersections for the local encoder and policy head, while conditioning on node-specific features and neighbor context.

- Global policy central coordinator. As shown in Figure 3, the coordinator first projects F^k using a light weight FCN encoder Φ_{enc}, fuses information over the corridor graph using a single graph convolution layer), applies the mask m^k, and feeds the resulting corridor embedding Z^k to a Q-network to estimate action values).

Signal control follows the phase-duration action design in Section 3.2.1 with four movement groups per intersection, $\Psi = \{\psi^{\text{NS}}, \psi^{\text{NSL}}, \psi^{\text{EW}}, \psi^{\text{EWL}}\}$. Green durations satisfy $\gamma \in [\gamma_{\min}, \gamma_{\max}]$ with $\gamma_{\min} = 10$ s and $\gamma_{\max} = 60$ s, and a fixed clearance time $\gamma^{\text{yel}} = \gamma_{\max}/\kappa_{\text{dec}}$ is applied between phases (Table 1). The reward follows the two-tier design in Section 3.2.3, combining a local congestion/delay penalty with a downstream waiting penalty to encourage progression-oriented operation.

5.4 Training Parameters

The corridor-level decision module is trained using value-based deep RL with experience replay and a target network to improve stability [55]. The optimization minimizes the mini-batch temporal-difference loss. Training uses a replay buffer of size 10^5 and a mini-batch size $B = 256$. The discount factor is set to $\beta = 0.99$. The target network is updated every 1000 environment steps.

Action selection follows an ϵ-greedy strategy where ϵ is annealed from 1.0 to 0.05 over training. To reduce early-stage non-stationarity and stabilize multi-agent learning, we adopt a curriculum schedule that begins with $M_0 = 2$ controlled intersections and increases the active set by one intersection every 50 episodes until the full synthetic corridor is included [50, 64]. The same trained policy is then evaluated on SQ1-SQ3 without further fine-tuning to assess generalization.

Training is executed with 20 parallel SUMO instances, totaling approximately 1200 episodes (≈ 1.224 million interaction steps), with 1020 steps per episode. To prevent simulation stagnation in congested regimes, episodes are terminated if they exceed a fixed wall-clock timeout (12 minutes). Each training and evaluation configuration is repeated over 10 independent random seeds to reduce sensitivity to stochastic demand generation and microscopic vehicle interactions.

For SQ1-SQ3, the base per-lane flow unit is 360 veh/lane /hour (Table 1), and time-varying demand is generated using the multiplier sequence $\{1, 3, 5, 7, 9, 7, 5, 3, 1\}$. Four OD stream sets are constructed using the boundary pairs $(A1 \leftrightarrow A2)$, $(B1 \leftrightarrow B2)$, $(C1 \leftrightarrow C2)$, and $(D1 \leftrightarrow D2)$ (Figure 4). Streams (A) and (B) start at $t = 0$ s with injections every 500 s, while (C) and (D) follow the same pattern but start at $t = 700$ s, creating asynchronous interactions. During evaluation, each rollout is simulated for up to 3 h to bound runtime, and metrics are aggregated across seeds and runs as described in Section 5.

5.5 Comparative Models

We compare the proposed gated-attention graph MARL controller against six recent state-of-the-art methods on SQ1-SQ3 under identical SUMO settings, phase constraints (four movement groups with bounded green and fixed clearance), demand profiles, random-seed repetitions, and evaluation metrics (average waiting time, travel time, and stops). All baselines use the same measurements that form the structured interface $y^k = (F^k, A, m^k)$, and their outputs are mapped to the same phase-duration action space; the only exception is offline-only training when required by a method.

For centralized corridor coordination, we include CCGN, which couples local controllers with a centralized collaborative graph network and a value-based global decision module for signal-free corridor operation [45]. To evaluate robustness to missing or incomplete observations, we include DiffLight, an offline diffusion-based TSC framework with a spatiotemporal Transformer backbone

Table 3: Ablation study to evaluate the inter-intersection communication (local-only) on SQ3. Mean ± std over 10 seeds. Lower is better for all metrics.

Method	**SQ3 (All streams)**			**SQ3 (A/B streams)**			**SQ3 (C/D streams)**		
	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops
Local-only (No comm; c_i^k=**0**, CCG self-loops)	57.4±3.6	269.8±10.1	2.18±0.12	54.9±3.3	263.1±9.6	2.07±0.11	60.2±3.9	276.5±10.8	2.30±0.13
Ours (Gated-Attn Graph MARL)	48.5±2.8	249.6±8.5	1.86±0.09	46.9±2.5	244.2±8.0	1.78±0.08	50.2±3.1	255.0±8.9	1.94±0.10

Table 4: Ablation study on SQ3 traffic scenario on the removal of hard gating (attention-only communication). Mean ± std over 10 seeds. Lower is better for all metrics.

Method	**SQ3 (All streams)**			**SQ3 (A/B streams)**			**SQ3 (C/D streams)**		
	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops
Attention-only	49.3±3.6	252.2±10.4	1.92±0.13	47.4±3.1	245.7±9.7	1.82±0.11	51.3±4.1	258.7±11.2	2.02±0.14
Ours (Gated-Attn Graph MARL)	48.5±2.8	249.6±8.5	1.86±0.09	46.9±2.5	244.2±8.0	1.78±0.08	50.2±3.1	255.0±8.9	1.94±0.10

[5], and MAPOLight, which targets partially observed V2I settings and introduces multi-tier cooperative learning for stability under limited visibility [58]. For dynamic spatiotemporal graph modeling, we include DSMEL, which uses adaptive heterogeneous graph updates and memory-enhanced temporal processing [79]. As a selective-neighbor baseline, we include FGLight, which learns neighbor-level relevance via adaptive attention and stabilized value learning [70]. Finally, we include CLlight, which augments cooperative MARL with a contrastive representation objective to improve generalization across demand regimes [29].

5.6 Ablation Study

5.6.1 Impact of Inter-Intersection Communication

To quantify the role of explicit coordination, we construct a local-only ablation that disables message passing in both the distributed interaction graph (DIG) and the central cooperation graph (CCG) by setting $c_i^k = \mathbf{0}$ for all agents and restricting the CCG adjacency to self-loops. All remaining components (observation encoder, Transformer temporal encoder, phase-duration action space, reward, curriculum schedule, replay-buffer training, and evaluation protocol) are held fixed to isolate the effect of inter-intersection communication.

Table 3 shows that removing communication yields a systematic and substantial performance drop on SQ3. Aggregated over all OD streams, the local-only variant increases average waiting time from 48.5 ± 2.8 s to 57.4 ± 3.6 s ($+8.9$ s, $\approx$18%), average travel time from 249.6 ± 8.5 s to 269.8 ± 10.1 s ($+20.2$ s, $\approx$8%), and average stops from 1.86 ± 0.09 to 2.18 ± 0.12 ($+0.32$, $\approx$17%). The degradation is evident for both synchronous A/B streams and asynchronous C/D streams, but is markedly stronger under C/D, where partial observability is amplified by delayed stream activation and inter-stream interference. In particular, for C/D streams the local-only controller increases waiting time by $+10.0$ s ($50.2 \pm 3.1 \rightarrow 60.2 \pm 3.9$), travel time by $+21.5$ s ($255.0 \pm 8.9 \rightarrow 276.5 \pm 10.8$), and stops by $+0.36$ ($1.94 \pm 0.10 \rightarrow 2.30 \pm 0.13$), exceeding the corresponding A/B increases ($+8.0$ s wait, $+18.9$ s travel, $+0.29$ stops). These results indicate that, on the largest and most heterogeneous corridor (SQ3), inter-intersection communication is not merely an auxiliary enhancement but a primary mechanism for mitigating downstream blocking and spillback-induced stop waves, especially in asynchronous operating regimes where critical upstream/downstream cues are not locally observable.

5.6.2 Impact of Attention-Only Communication

We evaluate the necessity of link-level sparsification in the proposed gated-attention communication by removing the hard gating stage and retaining only soft attention for inter-intersection aggregation. In the full model, the effective contribution coefficient from neighbor j to receiver i at decision step k is $\alpha_{i\leftarrow j}^k = g_{i\leftarrow j}^k a_{i\leftarrow j}^k$, where the near-binary gate $g_{i\leftarrow j}^k$ suppresses weak or

Table 5: Evaluate the imoact of soft attention (gate-only communication) on SQ3 traffic scenario. Mean ± std over 10 seeds. Lower is better for all metrics.

Method	SQ3 (All streams)			SQ3 (A/B streams)			SQ3 (C/D streams)		
	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops
Gate-only	50.0±3.1	254.1±9.6	1.96±0.11	48.1±2.8	247.3±8.9	1.85±0.10	52.0±3.5	261.0±10.2	2.07±0.12
Ours (Gated-Attn Graph MARL)	48.5±2.8	249.6±8.5	1.86±0.09	46.9±2.5	244.2±8.0	1.78±0.08	50.2±3.1	255.0±8.9	1.94±0.10

irrelevant links and $a^k_{i\leftarrow j}$ assigns continuous relevance weights over candidates.

Table 4 shows that removing the gate causes a consistent degradation across all SQ3 subsets, with the largest deterioration appearing under the asynchronous C/D streams where directional dependencies and unobserved spillback effects amplify the cost of redundant message fusion. Over all streams, attention-only increases waiting time from 48.5 to 49.3 s and travel time from 249.6 to 252.2 s, while the average number of stops rises from 1.86 to 1.92. The degradation is modest for the synchronous A/B subset (e.g., stops 1.78 → 1.82 and travel time 244.2 → 245.7 s), but becomes more pronounced for C/D streams, where travel time increases by 3.7 s (255.0 → 258.7 s) and stops increase by 0.08 (1.94 → 2.02). Importantly, attention-only also exhibits systematically higher seed-level variance across metrics (e.g., stop std 0.09 → 0.13 for all streams and 0.10 → 0.14 for C/D), indicating reduced robustness and less stable coordination when neighborhoods remain dense. Overall, these results support the central claim that hard gating provides an essential structural prior for corridor-scale MARL on large networks: it curbs over-smoothing and redundancy in inter-intersection fusion, yielding both improved mean performance and tighter performance dispersion on the SQ3 scalability stress-test.

5.6.3 Ablation Study on Gate-Only Communication

To isolate the contribution of soft relevance weighting in the proposed gated-attention communication, we remove the attention module and retain only hard link selection. In the full model, the effective contribution coefficient from neighbor j to receiver i at decision step k is $\alpha^k_{i\leftarrow j} = g^k_{i\leftarrow j} a^k_{i\leftarrow j}$, where $g^k_{i\leftarrow j}$ is a learned near-binary gate and $a^k_{i\leftarrow j}$ is a soft attention score. In this ablation, we set

$$\alpha^k_{i\leftarrow j} = g^k_{i\leftarrow j}, \tag{15}$$

and aggregate uniformly over the retained neighbor set. Concretely, for $\mathcal{N}^k_i = \{j \mid g^k_{i\leftarrow j} = 1\}$, the context vector becomes

$$c^k_i = \frac{1}{|\mathcal{N}^k_i| + \epsilon} \sum_{j\in\mathcal{V}} g^k_{i\leftarrow j}\, \tilde{h}^k_j, \tag{16}$$

where $\tilde{h}^k_j$ is the encoded neighbor representation and ϵ avoids division by zero. All other components (observation interface $y^k = (F^k, A, m^k)$, Transformer encoder, CCG fusion, phas-duration action space, reward design, curriculum schedule, and value-based training) are kept unchanged to ensure that differences are attributable to removing soft weighting.

Table 5 indicates that gate-only communication consistently underperforms the full gated-attention model on the SQ3 scalability stress-test, with the gap widening as the interaction patterns become more heterogeneous. Aggregated over all streams, removing attention increases waiting time from 48.5 to 50.0 s and travel time from 249.6 to 254.1 s, while stops rise from 1.86 to 1.96. The degradation is smaller under the synchronous A/B streams (e.g., travel time 244.2 → 247.3 s and stops 1.78 → 1.85), but is notably larger under the asynchronous C/D streams, where corridor coordination requires prioritizing a dominant upstream/downstream influence at each step: travel time increases by 6.0 s (255.0 → 261.0 s) and stops increase by 0.13 (1.94 → 2.07), accompanied by higher seed-level variability (stop std 0.10 → 0.12). These trends support the interpretation that hard selection alone is insufficient when multiple candidate interactions coexist; uniform averaging over the retained set can dilute the most informative

Table 6: Ablation study on SQ3 traffic scenario for fixed Top-K neighbor selection vs. learned gating. Mean ± std over 10 seeds. Lower is better for all metrics. $\overline{|E_{\text{act}}|}$ reports the average number of active directed edges per decision step (communication budget).

Method (Selection rule)	**Wait (s)**	**Travel (s)**	**Stops**	$\overline{\lvert \boldsymbol{E}_{\text{act}} \rvert}$
Top-K by hop-distance (corridor heuristic), K=2	51.0±3.2	258.7±10.1	2.03±0.12	242±0
Top-K by attention score, K=1	51.7±3.4	260.8±10.6	2.06±0.12	121±0
Top-K by attention score, K=2	50.4±3.1	255.9±9.8	1.98±0.11	242±0
Top-K by attention score, K=4	49.7±3.0	253.8±9.4	1.94±0.10	484±0
Ours (Learned gating + attention)	48.5±2.8	249.6±8.5	1.86±0.09	312±18

neighbor signal, whereas the full model's attention weights preserve directional dominance and yield more stable progression-oriented control on large, heterogeneous corridors.

5.6.4 Impact of Neighbor Selection vs. Learned Gating

This ablation tests whether the performance gains attributed to the learned hard-gating module can be reproduced by simple deterministic sparsification heuristics. In the full model, neighbor contributions are modulated by a learned near-binary gate $g^k_{i\leftarrow j}$ and a soft attention score $a^k_{i\leftarrow j}$, yielding $\alpha^k_{i\leftarrow j} = g^k_{i\leftarrow j} a^k_{i\leftarrow j}$. We replace the learned gate with a fixed Top-K rule that selects exactly K neighbors per receiver at each decision step:

$$g^k_{i\leftarrow j} = \mathbb{I}\Big[j \in \text{Top-}K\big(\{a^k_{i\leftarrow \ell}\}_{\ell\in\mathcal{V}}\big)\Big], \qquad K \in \{1,2,4\}, \tag{17}$$

where Top-$K(\cdot)$ returns the K highest-scoring candidates under the same attention pre-score used in the full model. Soft attention fusion is retained on the selected set, so the ablation isolates link selection rather than weighting. As an additional corridor heuristic baseline, we consider Top-K selection by hop distance along the physical graph (ties resolved by downstream direction). All remaining components (observation interface $y^k = (F^k, A, m^k)$, Transformer encoder, CCG fusion, reward, training schedule, and phase-duration action space) are unchanged.

Table 6 shows that fixed Top-K selection yields a clear communication-performance trade-off, but none of the deterministic variants matches learned gating at comparable budgets. With a very tight budget (K=1, $\overline{|E_{\text{act}}|}$=121), the controller under-communicates and performance degrades markedly relative to the learned gate: travel time increases from 249.6 to 260.8 s and stops increase from 1.86 to 2.06, indicating insufficient context to anticipate upstream arrivals and downstream blocking in the large SQ3 network. Increasing K improves performance, but exhibits diminishing returns and still remains inferior to learned gating despite higher communication cost. For example, K=2 ($\overline{|E_{\text{act}}|}$=242) reduces travel time to 255.9 s and stops to 1.98, yet still trails the learned-gating baseline by 6.3 s in travel time and 0.12 stops while using a comparable edge budget (242 vs. 312 ± 18). Even at K=4 ($\overline{|E_{\text{act}}|}$=484), where communication cost exceeds the learned-gating model by $\approx 55\%$, performance remains worse (travel 253.8 s, stops 1.94), implying that additional messages mainly introduce redundancy rather than informative coordination.

The hop-distance heuristic further highlights the limitation of static sparsification: despite using the same budget as Top-K by attention with K=2 ($\overline{|E_{\text{act}}|}$=242), distance-based selection yields substantially worse outcomes (travel 258.7 s and stops 2.03), consistent with the fact that decision-relevant dependencies in corridors are not purely geometric but regime dependent (e.g., spillback emphasizes downstream constraints whereas discharge emphasizes upstream arrival structure). Overall, these results support the claim that the learned gating module provides context-dependent sparsification that is more efficient than fixed Top-K rules: it achieves the best performance while maintaining a moderate and non-trivial communication budget (312 ± 18 active directed edges/step), and it avoids both under-communication (K=1) and redundancy-induced diminishing returns (K=4).

Table 7: Ablation study on SQ3 traffic scenario to evaluate the impact of CCG graph fusion in the global policy. Mean ± std over 10 seeds. Lower is better for all metrics.

Method	**Wait (s)**	**Travel (s)**	**Stops**
No-fusion (G^k=H^k)	54.8±3.9	271.4±12.3	2.18±0.15
Mean aggregation (uniform neighbor mean)	51.9±3.3	260.7±10.4	2.03±0.12
Ours (Normalized GCN fusion)	48.5±2.8	249.6±8.5	1.86±0.09

Table 8: Evaluate the temporal encoder in the local policy on SQ2 Mean ± std over 10 seeds. Lower is better for all metrics.

Method	**Wait (s)**	**Travel (s)**	**Stops**
GRU encoder (same hidden size)	44.6±2.8	234.9±8.6	1.74±0.10
MLP + stacked recent observations (budget-matched)	47.2±3.4	242.8±10.2	1.86±0.12
Ours (Transformer encoder)	41.7±2.3	225.4±7.0	1.61±0.08

5.6.5 Impact of Graph Fusion in the Global Policy

This ablation isolates the role of corridor-level graph fusion in the centralized coordinator (CCG) by comparing the full normalized GCN propagation against two weakened variants (no-fusion and uniform mean aggregation), while keeping the local policy, gated-attention communication, action constraints, reward, and training protocol unchanged. As reported in Table 7, removing CCG fusion produces the most severe degradation on SQ3: the no-fusion variant increases average waiting time from 48.5 to 54.8 s (+6.3 s), average travel time from 249.6 to 271.4 s (+21.8 s), and stops from 1.86 to 2.18 (+0.32). Notably, it also exhibits substantially higher seed-level variability (e.g., travel-time std rising from 8.5 to 12.3), indicating that the coordinator becomes less stable when it cannot explicitly propagate cross-intersection context. Replacing normalized fusion with a simple mean aggregation partially recovers performance but remains consistently inferior to the proposed fusion: mean aggregation still increases waiting time to 51.9 s (+3.4 s), travel time to 260.7 s (+11.1 s), and stops to 2.03 (+0.17), with larger variance than the full model. Overall, these results confirm that structured, degree-aware CCG propagation is not merely an architectural convenience but a critical component for scalable corridor coordination on the largest and most heterogeneous network (SQ3), yielding both lower average delay-related metrics and improved robustness across seeds compared to either removing fusion entirely or applying uniform smoothing.

5.6.6 Impact of Temporal Encoder

This ablation examines whether the temporal encoder in the local policy materially improves robustness under partial observability and time-varying inflow, and whether the Transformer specifically provides an advantage over simpler temporal models. We replace the Transformer encoder used to produce $\tilde{h}_i^k$ with either (i) a GRU of the same hidden size or (ii) a budget-matched MLP operating on a stacked observation history, while keeping the gated-attention communication, CCG fusion, action space, reward, curriculum, and value-learning setup unchanged.We replace the Transformer with two alternatives while keeping the rest of the pipeline unchanged (gated-attention communication, CCG fusion, action space, reward, curriculum, and value learning). **(i) GRU encoder.** We replace TransEnc$(\cdot)$ with a single-layer GRU over a fixed-length observation history of L steps, using the same hidden size as the Transformer embedding dimension. The GRU final hidden state is used as $\tilde{h}_i^k$. **(ii) MLP with stacked history.** We remove recurrence/attention and instead concatenate the most recent L encoded observations into a vector and map it through an MLP to produce $\tilde{h}_i^k$. The MLP is configured to match the Transformer encoder's parameter budget as closely as possible.

Table 9: Ablation study on SQ3 traffic scenario to evaluate the corridor progression penalty π_i^k.

Method	**Wait (s)**	**Travel (s)**	**Stops**
w/o progression penalty	47.9±3.0	262.7±9.6	2.15±0.11
Ours (full reward with progression penalty)	48.5±2.8	249.6±8.5	1.86±0.09

As shown in Table 8, the Transformer yields the best performance across all metrics on SQ2. Relative to the proposed Transformer encoder (Wait 41.7 ± 2.3 s, Travel 225.4 ± 7.0 s, Stops 1.61 ± 0.08), substituting a GRU increases waiting time by +2.9 s (to 44.6 ± 2.8 s), travel time by +9.5 s (to 234.9 ± 8.6 s), and stops by +0.13 (to 1.74 ± 0.10). The degradation is more pronounced for the MLP baseline, which further raises waiting time by +5.5 s (to 47.2 ± 3.4 s), travel time by +17.4 s (to 242.8 ± 10.2 s), and stops by +0.25 (to 1.86 ± 0.12), while also exhibiting the highest variance across seeds (e.g., travel-time std increasing from 7.0 to 10.2). These results indicate that temporal modeling is not incidental: replacing the Transformer systematically worsens efficiency and increases instability, particularly under SQ2's heterogeneous topology and time-varying multipliers with asynchronous stream starts. The consistent margin over GRU suggests that self-attention-based temporal integration better captures short-horizon arrival/queue evolution and stabilizes coordination-relevant representations under non-stationary demand.

5.6.7 Ablation Study on Reward Shaping

This ablation tests whether the intended progression-oriented (near signal-free) behavior is explicitly induced by the corridor-level reward design, rather than emerging implicitly from local delay/queue minimization. In the full objective, the network reward combines per-intersection utilities with a downstream waiting penalty π_i^k (with $\pi_e^k = 0$ at the entry) to discourage stop-and-go propagation after vehicles enter the corridor. We remove this progression penalty by setting $\pi_i^k = 0$ for all intersections, while keeping the remaining components unchanged (gated-attention communication, CCG fusion, phase-duration constraints, curriculum schedule, and value-based training). The ablation is evaluated on SQ3 (121 intersections), the most challenging setting where locally greedy actions can readily induce downstream spillback and corridor-wide shockwaves.

Table 9 shows that eliminating π_i^k substantially degrades progression-related outcomes despite a near-neutral effect on local queuing. Specifically, waiting time slightly decreases from 48.5±2.8 s to 47.9 ± 3.0 s (-0.6 s), indicating that local congestion reduction can remain competitive when the controller is optimized only for ρ_i^k. However, this apparent local gain comes at a clear corridor-level cost: travel time increases by +13.1 s (from 249.6 ± 8.5 s to 262.7 ± 9.6 s), and the average number of stops rises markedly by +0.29 (from 1.86 ± 0.09 to 2.15 ± 0.11), corresponding to an approximate 15% increase in stop events. The larger deterioration in stops than in waiting time is consistent with a behavioral shift toward locally opportunistic phase allocations that reduce immediate queues but fail to preserve downstream progression, that amplifies stop-and-go waves and increasing end-to-end travel time. These results confirm that the corridor progression penalty is a principal mechanism by which the proposed framework aligns local actuation with corridor-consistent, stop-minimizing control on large networks.

5.6.8 Ablation Study on Sensitivity

Figure 5 studies the sensitivity of the proposed corridor MARL controller to the curriculum growth schedule used during training on the synthetic TIN. The comparison includes no curriculum (train on full M from the start), faster curriculum (increase the number of controlled intersections more aggressively), the proposed curriculum (start from M_0=2 and add one intersection every 50 episodes), and a Slower curriculum (add one intersection every 100 episodes). Solid line

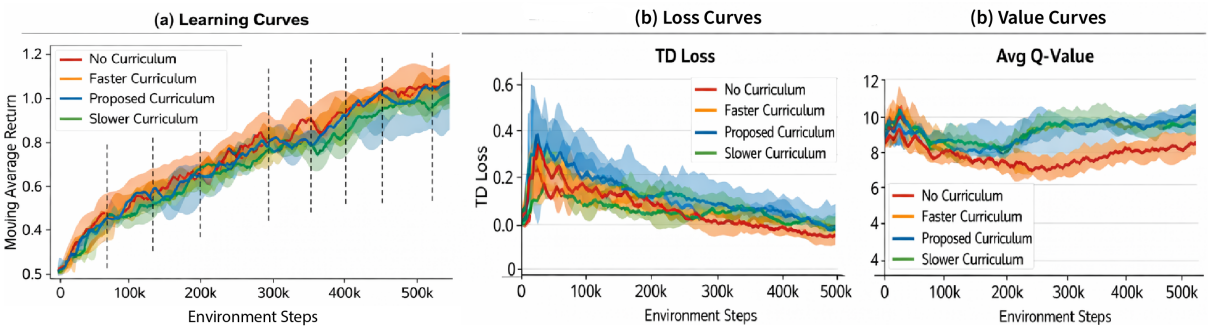

Figure 5: Curriculum schedule sensitivity during training on the synthetic TIN. Performance of No Curriculum, Faster Curriculum, Proposed Curriculum (M_0=2, +1 intersection every 50 episodes), and Slower Curriculum (+1 every 100 episodes). (a) Moving-average return vs. environment steps; vertical dashed lines mark curriculum expansion points. (b) TD loss vs. steps. (c) Average predicted Q-value vs. steps. Solid lines show mean over seeds and shaded bands indicate seed-level variability (e.g., ± std).

denotes the mean over seeds and the shaded band indicates seed-level variability (e.g., ± std / interquartile range). The vertical dashed lines in Figure 5(a) mark the curriculum expansion points where the active intersection set is increased, i.e., discrete rises in task complexity.

All schedules improve the moving-average return with increasing environment steps, but they differ in sample efficiency and stability under complexity jumps. The faster curriculum rises more quickly early in training, reflecting higher short-term sample efficiency, but exhibits larger oscillations around expansion points (dashed lines), consistent with transient non-stationarity when the environment complexity increases too abruptly. The proposed curriculum achieves a consistently strong return trajectory while smoothing the transition effects at the expansion points, indicating a better trade-off between fast learning and stable adaptation. The slower curriculum is the most conservative: it yields comparatively smoother progress but lags in return for a large portion of training, consistent with under-exposure to harder multi-intersection coordination early on. Training without curriculum converges more slowly and shows weaker robustness around the mid-to-late training regime, suggesting that learning directly on the full M-intersection game increases early-stage non-stationarity and slows policy improvement.

TD-loss trends provide a direct diagnostic of value-learning stability under the different schedules. Across all methods, TD loss decreases with training, but curricula differ in early variance and decay rate. The proposed curriculum shows a controlled decline in TD loss, indicating stable Bellman updates as the task grows. The faster curriculum initially reduces TD loss quickly but shows more fluctuation during the periods where complexity increases, consistent with abrupt distribution shifts in experience replay. The slower curriculum exhibits a smoother but more gradual reduction. In contrast, No curriculum tends to produce less stable early learning dynamics (spikes and slower settling), which aligns with the harder credit-assignment problem when many intersections learn concurrently from the beginning.

The evolution of the average predicted Q-value reflects how confidently the critic evaluates actions as training proceeds. The proposed curriculum maintains higher and more consistent Q-value estimates after the initial transient, suggesting more reliable value calibration under the gradually increasing coordination burden. The slower curriculum reaches comparable Q-value levels but typically later, consistent with delayed exposure to complex interactions. The Faster curriculum and no curriculum remain at lower Q-value plateaus for a longer period, indicating either conservative value estimates or less effective consolidation of long-horizon coordination benefits when the task difficulty changes too quickly (faster) or is maximal from the start (no curriculum).

Table 10: Performance comparison with SoTA models on SQ1-SQ3 (mean ± std over 10 seeds). Lower is better for all metrics. Best and second-best are highlighted.

Method	**SQ1**			**SQ2**			**SQ3**		
	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops	Wait (s)	Travel (s)	Stops
CCGN [45]	38.7±2.1	214.5±6.8	1.62±0.08	44.9±2.7	233.1±7.9	1.74±0.09	52.8±3.4	261.6±9.8	2.08±0.11
DiffLight [5]	40.2±2.4	219.1±7.2	1.68±0.10	46.7±2.9	238.8±8.3	1.81±0.10	51.6±3.1	258.9±9.1	2.02±0.10
MAPOLight [58]	39.1±2.3	216.7±7.0	1.65±0.09	45.5±2.8	236.4±8.1	1.79±0.10	53.9±3.6	265.2±10.2	2.12±0.12
DSMEL [79]	37.9±2.0	212.8±6.5	1.59±0.08	43.8±2.6	231.0±7.6	1.72±0.09	50.7±3.0	256.4±9.0	1.98±0.10
FGLight [70]	37.5±1.9	211.6±6.3	1.56±0.08	43.2±2.5	229.8±7.4	1.70±0.09	49.8±2.9	254.7±8.7	1.94±0.10
CLlight [29]	38.1±2.1	213.4±6.6	1.60±0.09	44.1±2.7	232.3±7.7	1.73±0.09	50.9±3.1	257.1±9.2	1.99±0.10
Ours (Gated-Attn Graph MARL)	**36.1±1.8**	**207.9±6.0**	**1.48±0.07**	**41.7±2.3**	**225.4±7.0**	**1.61±0.08**	**48.5±2.8**	**249.6±8.5**	**1.86±0.09**

5.7 Performance Comparison with SoTA Models

We compare the proposed gated-attention graph MARL controller against six recent SoTA models such as CCGN [45], DiffLight [5], MAPOLight /[58], DSMEL [79], FGLight [70], and CLlight on the three real-world corridor scenarios SQ1-SQ3 under the same SUMO configuration, phase constraints, and demand schedules described in Section 5.1.2. Specifically, we evaluate four OD stream sets per scenario with time-varying inflow multipliers $\{1, 3, 5, 7, 9, 7, 5, 3, 1\}$ and asynchronous stream start times (t=0 s for sets A/B and t=700 s for sets C/D). Each method is tested over 10 random seeds with 10 parallel simulations per seed; rollouts are capped at 3 h to bound runtime. Performance is reported using the standard effectiveness metrics: average waiting time, average travel time, and average number of stops [13, 18, 78, 80]. For each metric, we report mean ± standard deviation across seeds, and we compute the relative improvement of our method over the strongest baseline in the same setting.

Table 10 summarizes the results on SQ1-SQ3. Overall, the comparison suite spans complementary strengths: CCGN provides a strong centralized signal-free corridor baseline [45]; DiffLight is tailored to missing-observation regimes via diffusion-based modeling [5]; MAPOLight targets partially observed V2I coordination [58]; DSMEL emphasizes dynamic heterogeneous graph updates with memory-enhanced spatiotemporal processing [79]; FGLight focuses on neighbor-level selective fusion for large networks [70]; and CLlight improves MARL representations using contrastive learning [29]. To isolate the effect of selective communication, all baselines are mapped to the same phase-duration action interface and use the same observation sources used to construct $y^k = (F^k, A, m^k)$.

We additionally report results under partial-observation stress by randomly masking a fixed fraction of node attributes in F^k at test time (while keeping A and m^k unchanged), and evaluating the same trained policies without fine-tuning. Under this protocol, methods that rely on dense aggregation typically exhibit higher variance due to redundant or noisy message fusion, whereas selective mechanisms remain more stable. We report statistical significance using paired tests over seeds (two-sided p<0.05) when comparing our method to the best-performing baseline per scenario.

The figure 6 illustrates the training dynamics on the synthetic traffic scenario on the reward the episodic cumulative reward (left) and the corresponding optimization loss (right) as a function of training episodes. Reward curves (left). All methods begin in a highly congested regime (large negative reward), reflecting the difficulty of early exploration under multi-intersection coupling and partial observability. After the initial exploration phase (roughly the first 250–300 episodes), the baselines start to improve as coordination emerges, but they typically plateau at substantially lower reward levels and exhibit persistent oscillations, indicating limited corridor-level credit assignment and sensitivity to non-stationary multi-agent interactions. In contrast, GAMARL shows the largest and most sustained reward improvement, transitioning earlier into a high-performing regime and continuing to increase toward the end of training. This behavior is consistent with the intended effect of selective inter-intersection communication and corridor-level coordination, which reduce redundant information mixing and help preserve progression-oriented

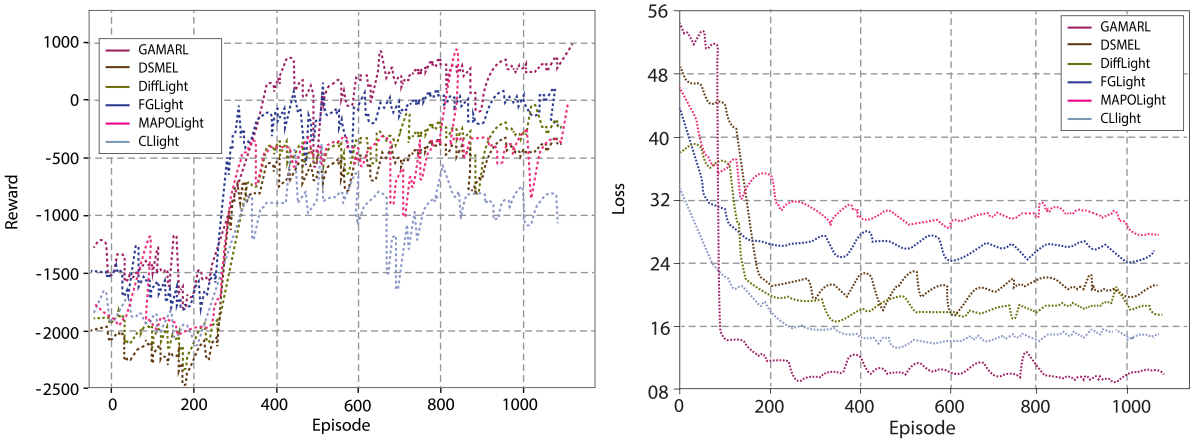

Figure 6: Training dynamics on the synthetic TIN. Episodic cumulative reward (left) and training loss (right) versus episode for the proposed GAMARL and SoTA models.

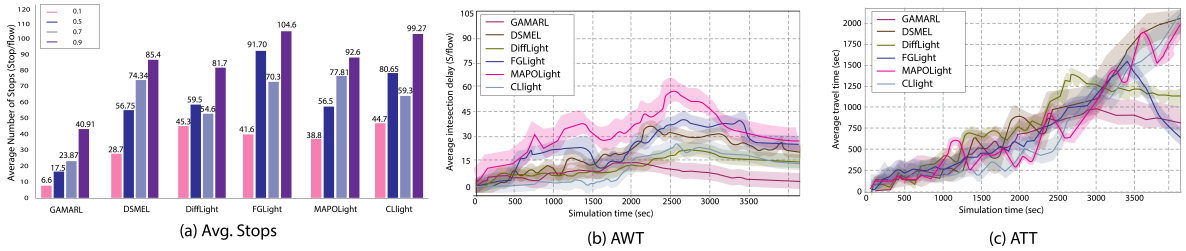

Figure 7: Performance comparison on the SQ3 corridor network. Performance of the proposed GAMARL against SoTA models under increasing traffic-intensity settings $(0.1, 0.5, 0.7, 0.9)$. (a) Average number of full stops per flow (lower is better). (b) Average intersection delay / waiting time (AWT, s/flow) over simulation time. (c) Average travel time (ATT, sec) over simulation time.

decisions as training proceeds.

Loss curves (right). The loss trajectories provide a direct proxy for training stability (e.g., Bellman/TD-style error for value-based learning or the algorithm's objective under the shared evaluation protocol). GAMARL exhibits a rapid loss decay early in training and maintains the lowest steady-state loss thereafter, with comparatively small fluctuations. This indicates more stable value estimation and faster consolidation of coordinated policies. By comparison, the baselines converge to higher loss plateaus and often retain noticeable oscillations throughout training, suggesting less stable optimization and weaker alignment between learned value estimates and long-horizon corridor outcomes.

5.7.1 Performance on SQ3 Traffic Network

Figure 7 compares the training-time behavior of the proposed GAMARL controller against SoTA model on the large-scale SQ3 corridor network. The results summarize both (i) aggregate progression quality via the stop metric and (ii) time-resolved efficiency indicators via delay and travel-time trajectories. For the time-series plots, solid curves denote the mean trend over runs (or seeds) and shaded regions indicate variability, highlighting stability under stochastic demand and microscopic traffic dynamics. Figure 7(a) reports the average number of full stops per flow under four increasing traffic-intensity settings (legend: $0.1, 0.5, 0.7, 0.9$). GAMARL consistently yields the fewest stops across all regimes, indicating stronger progression preservation and reduced stop-and-go propagation. The advantage becomes more pronounced at higher

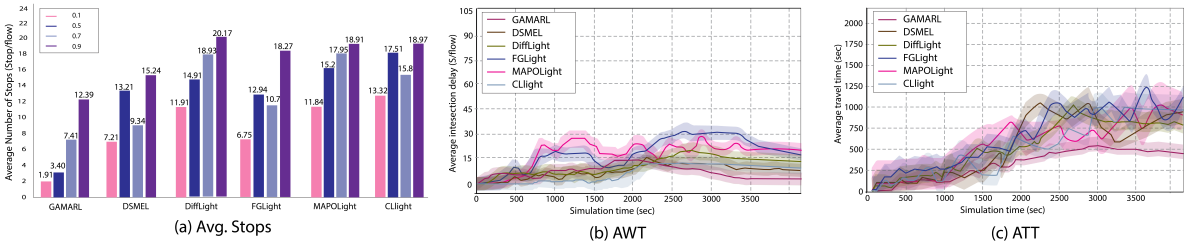

Figure 8: Performance comparison comparison on the SQ2 corridor network. Performance of the proposed GAMARL versus state-of-the-art models under increasing traffic-intensity settings $(0.1, 0.5, 0.7, 0.9)$.

load, where coordination failures typically amplify spillback and platoon fragmentation; for example, under the highest setting (0.9), GAMARL records substantially fewer stops (40.91) than the competing baselines (e.g., 81.7-104.6), demonstrating markedly better robustness in the congested regime.

Figure 7(b) plots the evolution of average intersection delay (s/flow) over simulation time. GAMARL maintains the lowest delay trajectory throughout the rollout and exhibits a smoother response during mid-horizon congestion build-up (around the central portion of the horizon), suggesting that the learned coordination policy mitigates transient queue amplification and recovers more rapidly after demand surges. In contrast, several baselines show elevated peaks and wider uncertainty bands, consistent with less stable credit assignment and weaker anticipation of upstream-downstream interactions under large-scale coupling. Figure 7(c) reports end-to-end average travel time (sec) over the same simulation horizon. GAMARL achieves consistently lower travel times and avoids the steep late-horizon growth observed in multiple baselines, which typically reflects corridor-level breakdown (spillback, blocking, and repeated stopping). The reduced growth rate and tighter variability envelope indicate that GAMARL better preserves discharge progression and prevents congestion waves from propagating across the long corridor.

5.7.2 Performance on SQ2 Traffic Network

Figure 9 compares GAMARL with DSMEL, DiffLight, FGLight, MAPOLight, and CLlight on the SQ2 corridor using (a) average stops per flow under four demand regimes (0.1/0.5/0.7/0.9), and (b-c) time-resolved average intersection delay (AWT) and average travel time (ATT), where solid curves denote mean performance and shaded bands reflect run/seed variability. GAMARL consistently yields the most progression-consistent behavior: it achieves the lowest stop counts across all regimes and maintains a markedly smaller increase as demand intensifies (e.g., 1.91 $\rightarrow$ 12.39 stops/flow from 0.1 $\rightarrow$ 0.9), whereas baselines rise more sharply at high demand (e.g., DSMEL 15.24, FGLight 18.27, DiffLight 20.17, MAPOLight 18.91, CLlight 18.97 at 0.9). In the temporal profiles, GAMARL sustains lower AWT and ATT trajectories with reduced mid-horizon surges and generally tighter variability envelopes, indicating improved stability under stochastic inflow and strong upstream-downstream coupling. Overall, the SQ2 results suggest that GAMARL more effectively suppresses stop-and-go propagation and spillback-induced degradation, translating into lower delay and travel-time escalation as corridor complexity and demand increase.

5.7.3 Performance on SQ1 Traffic Network

Figure 9 compares the performance on SQ1 using stop frequency and time-resolved delay metrics. In Figure 9(a), GAMARL consistently attains the lowest stop rate across all traffic-load settings $\{0.1, 0.5, 0.7, 0.9\}$ (1.27/3.90/6.13/8.02 stops per flow), while all baselines exhibit higher stop counts that grow more sharply with demand (e.g., DSMEL: 5.15/8.03/11.01/12.20; FGLight:

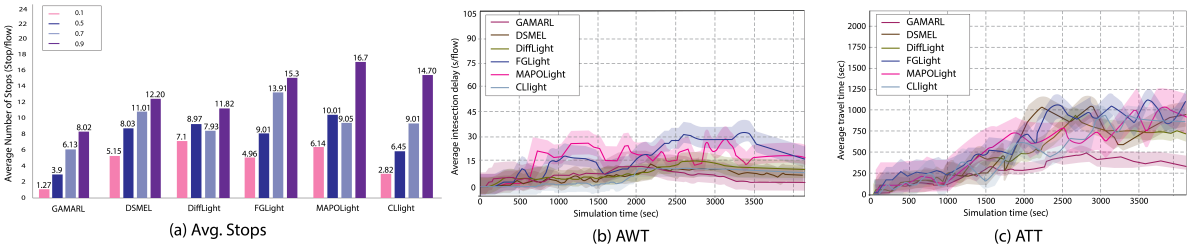

Figure 9: Performance comparison on the SQ1 corridor network.

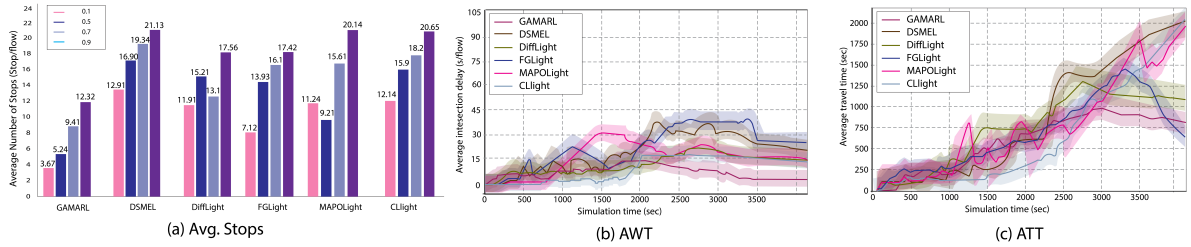

Figure 10: Performance comparison on the synthetic traffic scenario corridor network.

4.96/9.01/13.91/ 15.30; MAPOLight: 6.14/10.01/9.05/16.70), indicating more pronounced stop-and-go propagation under congestion. Figure 9(b)-(c) further show that GAMARL maintains a persistently lower envelope for AWT and ATT over simulation time, with reduced fluctuation and narrower variability bands compared to competitors, whereas several baselines experience larger mid-to-late-horizon surges and higher plateaus. Collectively, the results indicate that GAMARL yields more stable corridor coordination on SQ1, translating into fewer stops and lower intersection delay and end-to-end travel time, particularly as network load increases.

5.7.4 Performance Synthetic Traffic Network

The figure 10 compares training-time performance on the synthetic TIN. In Figure 10(a), the average number of stops increases monotonically with demand for all methods; however, GAMARL consistently achieves the lowest stop counts across all loads (e.g., 3.67/5.24/9.41/12.32 from low to high demand), indicating more effective suppression of stop-and-go propagation as the corridor becomes saturated. Figure 10(b) reports the evolution of AWT, where GAMARL maintains the lowest and most stable delay trajectory after an initial transient, while competing methods exhibit pronounced mid-horizon surges (approximately 2-3.5 ks), consistent with weaker anticipation of spillback and upstream-downstream coupling. Figure 10c) shows ATT, which grows with congestion for all methods; GAMARL mitigates this growth with smaller oscillations and tighter variability bands, reflecting more consistent progression under rising demand. Shaded regions denote run-to-run variability, and GAMARL's narrower bands particularly in Figure 10(b)-(c) suggest improved optimization stability and stronger robustness in congested regimes on the synthetic corridor.

5.8 Discussion

This work addresses corridor-scale TSC under partial observability and coordination scalability constraints. GAMARL combines Transformer-based temporal inference, gated-attention selective communication, and degree-normalized CCG fusion with masking. Results on SQ1-SQ3 and targeted ablations yield the following findings. Disabling coordination (no DIG messaging; self-loop-only CCG) substantially degrades performance on SQ3, with larger losses under asynchronous C/D streams than synchronous A/B streams (Table 3). This indicates that local

delay/queue minimization is insufficient in large corridors because spillback and downstream blocking are not directly observable at decision time, requiring explicit inter-intersection coupling to suppress stop-wave propagation.

Communication structure is also decisive. Attention-only aggregation increases mean cost and seed-level variance, particularly for C/D streams (Table 4), while gate-only aggregation yields larger degradations (Table 5). These results support a complementary role: hard gating limits redundancy and over-smoothing in dense neighborhoods, whereas soft attention preserves directional dominance among retained neighbors when multiple upstream/downstream constraints are simultaneously active. Fixed sparsification does not replicate learned selectivity. Top-K heuristics based on hop distance or attention prescores underperform learned gating across budgets, and increasing K exhibits diminishing returns despite higher communication cost (Table 6). This suggests that redundancy is not only inefficient but can be harmful by amplifying representation mixing at the receiver, while learned gating allocates communication capacity in a regime-dependent manner.

Corridor-level fusion in the CCG is a major contributor to both efficiency and stability. Removing fusion produces the largest deterioration on SQ3, and uniform mean aggregation only partially recovers performance (Table 7). Degree-normalized propagation appears necessary to maintain calibrated corridor representations under heterogeneous node degrees and to support corridor-scale credit assignment. Temporal modeling is similarly non-trivial under partially observed, time-varying inflow. Replacing the Transformer encoder with a GRU or an MLP over stacked histories degrades all metrics on SQ2 (Table 8), consistent with the need to reconstruct short-horizon latent dynamics (arrivals, dispersion, queue growth) from noisy proxies.

Reward shaping materially affects corridor behavior. Removing the downstream progression penalty slightly improves waiting time but substantially worsens travel time and stop frequency on SQ3 (Table 9), indicating that locally competitive policies can still induce corridor-level degradation via downstream interference. Finally, curriculum growth stabilizes value learning: gradual expansion reduces non-stationarity and improves Q-value calibration relative to no-curriculum or overly aggressive schedules (Figure 5).

5.9 Limitations

Although the proposed GAMARL framework demonstrates strong performance across synthetic and real-world corridor scenarios, several limitations should be acknowledged. The evaluation relies exclusively on the SUMO microscopic traffic simulator and therefore does not capture real-world uncertainties such as sensing failures, communication delays or packet loss, actuator malfunctions, and human driver non-compliance. In particular, the gated-attention communication mechanism assumes timely and reliable message exchange; its robustness under asynchronous or unreliable communication was not evaluated. Partial observability is addressed under bounded uncertainty. The observation space is constructed from engineered proxy features (e.g., inferred stream position and rolling speed estimates) that assume reasonably accurate local sensing and stable historical statistics. Severe sensing degradation scenarios, including prolonged detector outages or systematic bias, are not explicitly modeled.

The considered traffic networks are restricted to arterial-style corridors with homogeneous vehicle classes, fixed lane configurations, and standard four-phase signal logic. More complex urban settings such as irregular topologies, multimodal interactions, mixed autonomy traffic, or pedestrian and transit-priority operations are outside the scope of the current study, limiting direct transferability. Scalability is demonstrated empirically but not theoretically, While GAMARL scales to corridors with up to 121 intersections, no formal analysis is provided regarding convergence, computational complexity, or worst-case communication cost as network size increases. The interaction between curriculum learning, replay-buffer non-stationarity, and value-function approximation lacks theoretical guarantees.

The reward formulation encodes an explicit preference for progression-oriented operation

via a downstream waiting penalty. Although effective for reducing stop-and-go behavior, this design may conflict with other objectives such as equity across approaches, pedestrian safety, or emissions reduction. Sensitivity to alternative multi-objective formulations was not explored.

5.10 Future Research Directions

Several research directions emerge from the identified limitations. Deployment-oriented evaluation remains a critical next step. Future studies should assess GAMARL under delayed, lossy, and asynchronous communication, heterogeneous sensor availability, and actuator noise, ideally through hardware-in-the-loop simulation or field trials. Robustness under severe observation failures warrants further investigation. Explicit integration of uncertainty-aware or state-reconstruction mechanisms (e.g., generative or Bayesian inference models) could improve control performance under missing, biased, or highly non-stationary sensing. Extending the framework to heterogeneous traffic participants is essential for broader applicability. Incorporating pedestrians, transit vehicles, emergency traffic, and mixed autonomy will require expanded state, action, and reward representations, along with safety-aware coordination mechanisms.

Scaling beyond corridor-level control motivates hierarchical and multi-resolution graph formulations. Region-level abstraction and inter-region coordination may be necessary for city-scale deployment, together with principled analysis of communication efficiency and representation stability. Theoretical characterization of gated-attention learning remains an open problem. Developing analytical tools to understand when learned sparsification improves stability, sample efficiency, and convergence in graph-based Markov games particularly under value-based learning would strengthen the theoretical foundations of the approach.

6 Conclusion

This work studied corridor-scale traffic signal control under partial observability and scalable coordination constraints. We proposed GAMARL framework that integrates temporal inference, selective inter-intersection communication, and graph-based corridor coordination. The corridor is modeled as a two-level Traffic Intersection Network. At the intersection level, a Transformer encoder infers short-horizon latent dynamics from local, noisy observations. Coordination is achieved via a two-stage communication module that performs hard link gating followed by attention-weighted fusion, enforcing sparse yet decision-relevant message exchange consistent with dominant upstream-downstream causality. CCG applies degree-normalized fusion over masked intersection embeddings to support corridor-level credit assignment under centralized learning and decentralized execution. A progression-oriented reward further aligns local actuation with corridor objectives by penalizing downstream waiting after corridor entry, discouraging stop-and-go propagation. Experiments in SUMO across three real-world scenarios (SQ1-SQ3) show that GAMARL consistently improves average waiting time, end-to-end travel time, and stop frequency relative to recent state-of-the-art baselines, with the largest gains under high-demand and asynchronous inflow regimes and on the 121-intersection SQ3 network. Ablations confirm that communication, learned gating, attention-based fusion, CCG graph propagation, temporal encoding, curriculum growth, and progression-oriented reward shaping each contribute materially to performance and stability; static sparsification or soft attention alone is insufficient.

References

[1] H. M. Abdelghaffar and H. A. Rakha. Development and testing of a novel game theoretic de-centralized traffic signal controller. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):231–242, dec 2019.

[2] A. Agarwal, D. Sahu, R. Mohata, K. Jeengar, A. Nautiyal, and D. K. Saxena. Dynamic traffic signal control for heterogeneous traffic conditions using max pressure and reinforcement learning. *Expert Systems with Applications*, 254:124416, nov 2024.

[3] S. Araghi, A. Khosravi, M. Johnstone, and D. Creighton. Q-learning method for controlling traffic signal phase time in a single intersection. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1261–1265. IEEE, oct 2013.

[4] Salah Bouktif, Abderraouf Cheniki, Ali Ouni, and Hesham El-Sayed. Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowledge-Based Systems*, 267:110440, 2023.

[5] Hanyang Chen, Yang Jiang, Shengnan Guo, Xiaowei Mao, Youfang Lin, and Huaiyu Wan. Difflight: a partial rewards conditioned diffusion model for traffic signal control with missing data. *Advances in Neural Information Processing Systems*, 37:123353–123378, 2024.

[6] J. Chen, L. Yang, C. Qin, Y. Yang, L. Peng, and X. Ge. Heterogeneous graph traffic prediction considering spatial information around roads. *International Journal of Applied Earth Observation and Geoinformation*, 128:103709, apr 2024.

[7] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 15084–15097, dec 2021.

[8] M. Chen, W. Liu, T. Wang, S. Zhang, and A. Liu. A game-based deep reinforcement learning approach for energy-efficient computation in mec systems. *Knowledge-Based Systems*, 235: 107660, jan 2022.

[9] S. Chen, J. Dong, P. Ha, Y. Li, and S. Labi. Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, 36(7):838–857, jul 2021.

[10] S. Chen, J. Dong, P. Ha, Y. Li, and S. Labi. Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, 36(7):838–857, jul 2021.

[11] J. Cheng, W. Wu, J. Cao, and K. Li. Fuzzy group-based intersection control via vehicular networks for smart transportations. *IEEE Transactions on Industrial Informatics*, 13(2): 751–758, 2017.

[12] 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*, 21(3):1086–1095, mar 2019.

[13] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE transactions on intelligent transportation systems*, 21(3):1086–1095, 2019.

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

[15] S. Darmoul, S. Elkosantini, A. Louati, and L. B. Said. Multi-agent immune networks to control interrupted flow at signalized intersections. *Transportation Research Part C: Emerging Technologies*, 82:290–313, sep 2017.

[16] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on machine learning*, pages 1538–1546. PMLR, 2019.

[17] F. X. Devailly, D. Larocque, and L. Charlin. Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7496–7507, apr 2021.

[18] François-Xavier Devailly, Denis Larocque, and Laurent Charlin. Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7496–7507, 2021.

[19] Shifei Ding, Wei Du, Ling Ding, Jian Zhang, Lili Guo, and Bo An. Robust multi-agent communication with graph information bottleneck optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3096–3107, 2023.

[20] Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. *Advances in neural information processing systems*, 33:22069–22079, 2020.

[21] Wei Du, Shifei Ding, Wei Guo, Yuqing Sun, Guoxian Yu, and Lizhen Cui. Multi-agent communication with information preserving graph contrastive learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 64–71, 2025.

[22] Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 456–464, 2021.

[23] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad. Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 18(3):227–245, jul 2014.

[24] Lingling Fan, Yusong Yang, Honghai Ji, and Shuangshuang Xiong. Optimization of traffic signal cooperative control with sparse deep reinforcement learning based on knowledge sharing. *Electronics*, 14(1):156, 2025.

[25] J. Fang, Y. You, M. Xu, J. Wang, and S. Cai. Multi-objective traffic signal control using network-wide agent coordinated reinforcement learning. *Expert Systems with Applications*, 229:120535, nov 2023.

[26] Jie Fang, Ya You, Mengyun Xu, Juanmeizi Wang, and Sibin Cai. Multi-objective traffic signal control using network-wide agent coordinated reinforcement learning. *Expert Systems with Applications*, 229:120535, 2023.

[27] T. Fu, L. Wang, S. Garg, M. S. Hossain, Q. Yu, and H. Hu. Adaptive signal light timing for regional traffic optimization based on graph convolutional network empowered traffic forecasting. *Information Fusion*, 103:102072, mar 2024.

[28] X. Fu, Y. Ren, H. Jiang, J. Lv, Z. Cui, and H. Yu. Cllight: Enhancing representation of multi-agent reinforcement learning with contrastive learning for cooperative traffic signal control. *Expert Systems with Applications*, 262:125578, mar 2025.

[29] Xiang Fu, Yilong Ren, Han Jiang, Jiancheng Lv, Zhiyong Cui, and Haiyang Yu. Cllight: Enhancing representation of multi-agent reinforcement learning with contrastive learning for cooperative traffic signal control. *Expert Systems with Applications*, 262:125578, 2025.

[30] W. Genders and S. Razavi. Asynchronous n-step q-learning adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 23(4):319–331, jul 2019.

[31] M. S. Ghanim and G. Abu-Lebdeh. Real-time dynamic transit signal priority optimization for coordinated traffic networks using genetic algorithms and artificial neural networks. *Journal of Intelligent Transportation Systems*, 19(4):327–338, oct 2015.

[32] Liben Huang and Xiaohui Qu. Improving traffic signal control operations using proximal policy optimization. *IET Intelligent Transport Systems*, 17(3):592–605, 2023.

[33] S. Iqbal and F. Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970. PMLR, may 2019.

[34] M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, volume 34, pages 1273–1286, dec 2021.

[35] Xianguang Jia, Mengyi Guo, Yingying Lyu, Jie Qu, Dong Li, and Fengxiang Guo. Adaptive traffic signal control based on graph neural networks and dynamic entropy-constrained soft actor–critic. *Electronics*, 13(23):4794, 2024.

[36] Xianguang Jia, Mengyi Guo, Yingying Lyu, Jie Qu, Dong Li, and Fengxiang Guo. Adaptive traffic signal control based on graph neural networks and dynamic entropy-constrained soft actor–critic. *Electronics*, 13(23):4794, 2024.

[37] J. Jiang, C. Dun, T. Huang, and Z. Lu. Graph convolutional reinforcement learning. *arXiv preprint arXiv:1810.09202*, oct 2018.

[38] Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. *Advances in neural information processing systems*, 31, 2018.

[39] Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1902.01554*, 2019.

[40] W. Liu, H. Li, H. Zhang, J. Xue, and S. Sun. Dynamic spatio-temporal graph fusion network modeling for urban metro ridership prediction. *Information Fusion*, 117:102845, may 2025.

[41] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao. Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7211–7218. AAAI Press, apr 2020.

[42] Xunlian Luo, Chunjiang Zhu, Detian Zhang, and Qing Li. Stg4traffic: A survey and benchmark of spatial-temporal graph neural networks for traffic prediction. *arXiv preprint arXiv:2307.00495*, 2023.

[43] Panagiotis Michailidis, Iakovos Michailidis, Charalampos Rafail Lazaridis, and Elias Kosmatopoulos. Traffic signal control via reinforcement learning: A review on applications and innovations. *Infrastructures*, 10(5):114, 2025.

[44] S. S. Mousavi, M. Schukat, and E. Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7): 417–423, sep 2017.

[45] Hamza Mukhtar, Adil Afzal, Sultan Alahmari, and Saud Yonbawi. Ccgn: Centralized collaborative graphical transformer multi-agent reinforcement learning for multi-intersection signal free-corridor. *Neural networks*, 166:396–409, 2023.

[46] Yaru Niu, Rohan R Paleja, and Matthew C Gombolay. Multi-agent graph-attention communication and teaming. In *AAMAS*, volume 21, page 20th, 2021.

[47] S. M. Odeh, A. M. Mora, M. N. Moreno, and J. J. Merelo. A hybrid fuzzy genetic algorithm for an adaptive traffic signal system. *Advances in Fuzzy Systems*, 2015(1):378156, 2015.

[48] M. M. Rahman, P. Najaf, M. G. Fields, and J. C. Thill. Traffic congestion and its urban scale factors: Empirical evidence from american urban areas. *International Journal of Sustainable Transportation*, 16(5):406–421, may 2022.

[49] A. Singh, T. Jain, and S. Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, dec 2018.

[50] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018.

[51] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018.

[52] Chao Sun, Yuhao Yang, Jiacheng Li, Weiyi Fang, and Peng Zhang. A multi-agent regional traffic signal control system integrating traffic flow prediction and graph attention networks. *Systems*, 14(1):47, 2025.

[53] Chuxiong Sun, Peng He, Rui Wang, and Changwen Zheng. Revisiting communication efficiency in multi-agent reinforcement learning from the dimensional analysis perspective. *arXiv preprint arXiv:2501.02888*, 2025.

[54] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 2094–2100. AAAI Press, mar 2016.

[55] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] Chong Wang, Yueqi Li, Jiale Chen, Jian Zhang, and Yu Xue. Cooperative traffic signal control for a partially observed vehicular network using multi-agent reinforcement learning. *Engineering Applications of Artificial Intelligence*, 160:111813, 2025.

[59] Lijuan Wang, Guoshan Zhang, Qiaoli Yang, and Tianyang Han. An adaptive traffic signal control scheme with proximal policy optimization based on deep reinforcement learning for a single intersection. *Engineering Applications of Artificial Intelligence*, 149:110440, 2025.

[60] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International conference on machine learning*, pages 9908–9918. PMLR, 2020.

[61] T. Wang, J. Cao, and A. Hussain. Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 125:103046, apr 2021.

[62] Tao Wang, Zhipeng Zhu, Jing Zhang, Junfang Tian, and Wenyi Zhang. A large-scale traffic signal control algorithm based on multi-layer graph deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 162:104582, 2024.

[63] Tong Wang, Jiahua Cao, and Azhar Hussain. Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. *Transportation research part C: emerging technologies*, 125:103046, 2021.

[64] Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. From few to more: Large-scale dynamic multiagent curriculum learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7293–7300, 2020.

[65] Xiaoyu Wang, Ayal Taitler, Scott Sanner, and Baher Abdulhai. Mitigating partial observability in adaptive traffic signal control with transformers. *arXiv preprint arXiv:2409.10693*, 2024.

[66] Y. Wang, Y. Shen, Z. Liu, P. P. Liang, A. Zadeh, and L. P. Morency. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7216–7223. AAAI Press, jul 2019.

[67] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1913–1922, 2019.

[68] Y. Wen, S. Zhang, J. Zhang, S. Bao, X. Wu, D. Yang, and Y. Wu. Mapping dynamic road emissions for a megacity by using open-access traffic congestion index data. *Applied Energy*, 260:114357, feb 2020.

[69] Feng Xiao, Jiaming Lu, Lu Li, Wenwen Tu, and Chaojing Li. Advances in reinforcement learning for traffic signal control: a review of recent progress. *Intelligent Transportation Infrastructure*, page liaf009, 2025.

[70] Hang Xiao, Huale Li, Shuhan Qi, Jiajia Zhang, and DingZhong Cai. Fglight: Learning neighbor-level information for traffic signal control. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 2181–2189, 2025.

[71] Jinhua Xu, Yuran Li, Wenbo Lu, Shuai Wu, and Yan Li. A heterogeneous traffic spatio-temporal graph convolution model for traffic prediction. *Physica A: Statistical Mechanics and its Applications*, 641:129746, 2024.

[72] Di Xue, Lei Yuan, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via shapley message value. In *IJCAI*, pages 578–584, 2022.

[73] S. Yang, B. Yang, H. S. Wong, and Z. Kang. Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm. *Knowledge-Based Systems*, 183:104855, nov 2019.

[74] S. Yang, B. Yang, Z. Kang, and L. Deng. Ihg-ma: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. *Neural Networks*, 139: 265–277, jul 2021.

[75] S. Yang, B. Yang, Z. Kang, and L. Deng. Ihg-ma: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. *Neural Networks*, 139: 265–277, jul 2021.

[76] Shantian Yang. Hierarchical graph multi-agent reinforcement learning for traffic signal control. *Information Sciences*, 634:55–72, 2023.

[77] Shantian Yang and Bo Yang. A semi-decentralized feudal multi-agent learned-goal algorithm for multi-intersection traffic signal control. *Knowledge-Based Systems*, 213:106708, 2021.

[78] Shantian Yang, Bo Yang, Zhongfeng Kang, and Lihui Deng. Ihg-ma: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. *Neural networks*, 139:265–277, 2021.

[79] Bao-Lin Ye, Peng Wu, Lingxi Li, Weimin Wu, Bo Song, and Xianchao Zhang. Multi-intersection traffic signal control based on dynamic spatiotemporal memory enhanced learning. *Control Engineering Practice*, 165:106606, 2025.

[80] Xinshi Zang, Huaxiu Yao, Guanjie Zheng, Nan Xu, Kai Xu, and Zhenhui Li. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1153–1160, 2020.

[81] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz. Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):404–415, mar 2020.

[82] Bin Zhou, Qishen Zhou, Simon Hu, Dongfang Ma, Sheng Jin, and Der-Horng Lee. Cooperative traffic signal control using a distributed agent-based deep reinforcement learning with incentive communication. *IEEE Transactions on Intelligent Transportation Systems*, 25(8): 10147–10160, 2024.

[83] Ruijie Zhu, Wenting Ding, Shuning Wu, Lulu Li, Ping Lv, and Mingliang Xu. Auto-learning communication reinforcement learning for multi-intersection traffic light control. *Knowledge-Based Systems*, 275:110696, 2023.