Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
ISSN 3107-2798 (online)
E-mail: jicn.office@ffspub.com

Article

Edge-Aware Hybrid-Action Reinforcement Learning for Latency-Sensitive Cooperative Bus Signal Priority in Vehicular Edge Networks

Hui Deng[1,2,†], Xiaofei Wang[1,†], Zipeng Wang[3], Zesong Tian[2], Wen Du[4]
[1] *College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*
[2] *China Telecom Digital City Technology Co., Ltd., Xiong'an New Area, Hebei 071000, China*
[3] *College of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China*
[4] *Xiong'an National Innovation Center Technology Co., Ltd., Xiong'an New Area, Hebei 071000, China*
[†] *E-mail: denghui_xa@189.cn (H.D.); xiaofeiwang@tju.edu.cn (X.W.)*

Received: June 30, 2026/ Revised: August 1, 2026 / Accepted: August 4, 2026 / Published online: August 19, 2026

Abstract: Dense short-block road networks require bus signal priority (BSP) decisions to be generated and delivered within short and reliable control windows. This paper presents Edge-HyAR-BSP, a deadline-aware cooperative BSP framework that supports edge-side execution in vehicular edge networks. Roadside edge nodes perform decentralized low-latency inference, while the cloud supports centralized training and model updates. The framework represents each priority decision as a coupled phase-duration action and checks its executability under bus ETA, signal-safety, compensation, and edge-side deadline constraints. Green extension, red truncation, and cross-cycle compensation are integrated to improve bus passage while limiting disturbance to general traffic. The project platform covers 102 signalized intersections in the Rongdong District of Xiong'an New Area. Detailed operational evaluation is conducted on a 15-intersection corridor served by Route 302, while robustness and ablation analyses are performed in simulation using a topology derived from the 102-intersection network. In the selected pre-/post-deployment periods, bus speed increased by 14.64%–30.09%, aggregate bus delay decreased by 38.28%–65.13%, and bus stops decreased by 42.01%–58.62%. Edge deployment reduced mean end-to-end decision latency from 89.7 ms to 24.6 ms. Simulation results show gradual degradation under increased delay, packet loss, and workload. The findings provide operational case-study evidence for the feasibility of edge-aware hybrid-action BSP, while broader multi-route and cross-city validation remains necessary.

Keywords: Vehicular Edge Computing; Edge Intelligence; Deadline-Aware Control; Bus Signal Priority; Hybrid-Action Reinforcement Learning; Multi-Agent Traffic Signal Control
https://doi.org/10.64509/jicn.23.133

1 Introduction

For urban bus systems, signalized intersections are one of the most direct sources of delay, with stops and waiting at junctions accounting for an estimated 15%–30% of total bus travel time [1]. In Bus Signal Priority (BSP), signal timing is dynamically adjusted (i.e., green phases are extended, red phases are shortened, or a special transit phase is inserted) so that buses are enabled to traverse intersections more rapidly. [2].

In recent years, a new urban spatial paradigm—characterized by narrow roads, dense blocks, and short intersection spacing (typically 150–300 m)—has emerged in cities such as Xiong'an New Area, posing practical challenges for traffic signal control. Dense short-block urban networks, such as the Rongdong District considered in this study, exhibit strong coupling between neighboring intersections: queue spillback from one intersection can propagate to upstream and downstream junctions within a few cycles, causing vehicles to experience frequent stops and starts across short intersection intervals. According to the project evaluation report, the

† Corresponding authors: Hui Deng, Xiaofei Wang
* Academic Editor: Mengwei Xu

narrow cross-sections (15–25 m) further compress motor vehicles, non-motorized vehicles, and pedestrians into compact spaces with elevated conflict point densities compared with conventional road networks.

Meanwhile, digital transportation systems are evolving from offline analysis to real-time sensing, online traffic situation inference, and closed-loop signal control [3–5]. In this transformation, the underlying computing and communication infrastructure has become important for the timeliness, safety, and regional coordination of traffic control. If all vehicle and roadside sensing data are transmitted to a remote cloud for centralized decision-making, the resulting communication delay, cloud queuing delay, and control feedback delay may reduce the available timing margin and action-delivery reliability under communication congestion or cloud workload fluctuations. Edge computing offers a promising paradigm: by deploying reinforcement learning (RL) inference models on roadside edge nodes, traffic states can be processed close to intersections, and signal control actions can be generated with lower latency [6–8]. For edge-assisted traffic control, however, low latency alone is not sufficient. The reliability of a signal control action also depends on whether sensing, inference, and action delivery can be completed before the valid control window expires. In BSP, a delayed decision may miss the bus arrival time and introduce unnecessary disturbance to general traffic phases. Communication delay, edge inference latency, edge workload, and packet loss should therefore be considered not only as system performance indicators, but also as contextual variables that influence the traffic control policy.

This study is motivated by and evaluated using data from the Xiong'an New Area Science and Technology Innovation Project (Grant No. 2022XAGG0126), which focused on multi-granularity traffic simulation, collaborative control, BSP, vehicle-road cooperation, and edge-cloud collaborative computing in the narrow-road dense-network environment of the Rongdong District.

Despite recent advances in RL-based traffic signal control and BSP research, several important directions remain insufficiently explored. At the method level, most existing RL-based signal control methods adopt purely discrete action spaces—IntelliLight [9], PressLight [10], and CoLight [11] select only signal phases without adjusting green duration—whereas BSP requires fine-grained control over both the phase choice and its duration. Bouktif *et al.* [12] took a first step toward hybrid action spaces for single-intersection control, but did not extend the approach to multi-intersection coordination or employ advanced action representation mechanisms. Published RL-based BSP works such as D^2TSP [13] and multi-objective DRL-TSP [14] also remain limited to isolated intersections or small networks. At the scenario level, evidence for coordinated BSP in dense short-block urban networks is scarce: existing BSP studies mainly consider isolated intersections or arterial corridors, while the interaction between short intersection spacing, queue spillback, and multi-intersection BSP has received little attention. Large-scale project-based operational evidence remains limited compared with the extensive simulation-based literature. At the system level, existing BSP methods typically lack systematic cycle recovery mechanisms after priority triggering, which can lead to phase disorder and green wave disruption. Furthermore, communication delay, edge inference latency, and edge workload are rarely treated as contextual factors in RL-based signal control formulations.

The Graph Attention Network (GAT), Centralized Training with Decentralized Execution (CTDE), Conditional Variational Autoencoder (CVAE) and Hybrid Action Representation (HyAR) adopted in this paper are mature general reinforcement learning technologies. Edge operational indicators including communication latency, computing load and packet loss are integrated into the reinforcement learning state of the BSP scheme to construct deadline-constrained models. Hybrid actions subject to bus-specific constraints are adopted to jointly optimize signal phases and green light durations. An edge-aware action feasibility projection embedded with multi-layer verification is designed to ensure executable decisions under degraded network and computing conditions. A cross-cycle compensation mechanism is developed to balance bus operation efficiency and traffic fairness for general traffic. Large-scale field validations are implemented based on a real-world road network consisting of 102 signalized intersections in Xiong'an New Area, and the research gap concerning field deployment implementations of BSP in dense short-block urban networks is filled.

The main contributions are summarized as follows. First, we formulate cooperative BSP as a deadline-constrained edge-executable control problem, in which edge-side communication delay, inference latency, workload, bandwidth, and packet loss are treated as execution-context variables that affect whether a priority decision can be safely delivered within the valid signal-control window. Second, a phase-duration coupled priority action mechanism is developed to jointly determine the bus-serving phase and its green-time adjustment under signal-safety, bus ETA, and compensation constraints. Third, a deadline-aware feasibility projection mechanism is introduced to filter or adjust candidate priority actions according to minimum green time, clearance time, phase order, and edge-side control-window constraints. Fourth, a cross-cycle compensation strategy and project-derived operational evaluation are provided to examine the feasibility of the framework in a dense short-block road network.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model and problem formulation, including the project data source. Section 4 describes the proposed Edge-HyAR-BSP method. Section 5 reports experimental results. Section 6 discusses limitations, validity threats, and future work. Section 7 concludes the paper.

2 Related Work

2.1 Edge Intelligence and Vehicular Edge Computing

Edge intelligence aims to push AI model training, inference, and decision-making from centralized cloud platforms toward network edge nodes, thereby reducing response latency and improving context awareness [6, 7]. In vehicular networks, this paradigm is particularly important because vehicles and roadside sensing devices continuously generate time-sensitive data, while many control actions must be executed within short decision windows. Vehicular edge computing (VEC) further integrates communication, computation, and storage resources near vehicles, RSUs, and roadside infrastructure, enabling low-latency perception, inference, and actuation [8]. General resource scheduling and service orchestration in edge and MEC systems have been extensively studied [15, 16]. Computing power networks represent a related line of work that treats distributed computing resources as a schedulable and rentable utility [5, 17]. MEC standards define a framework for providing cloud computing capabilities and IT services at the network edge, characterized by ultra-low latency, high bandwidth, and real-time access to network information [18].

2.2 Edge-Assisted Intelligent Transportation and Traffic Control

Existing edge-assisted Intelligent Transportation System (ITS) studies have mainly investigated perception offloading, traffic state estimation, vehicle trajectory prediction, digital twin platforms, and Vehicle-to-Everything (V2X) service scheduling [19, 20]. However, fewer studies integrate edge-side operating conditions, such as communication delay, inference latency, workload, and packet loss, directly into closed-loop traffic signal control. In contrast, this paper treats these edge-side factors as contextual state variables that influence the signal control policy, rather than as external system metrics. This makes the BSP framework aware of the edge computing conditions under which its decisions are executed.

2.3 Reinforcement Learning for BSP and Traffic Control

BSP methods are commonly built around green extension, red truncation, and phase insertion [1, 2]. These strategies are intuitive and can be implemented on conventional signal controllers. Their limitation lies not in the lack of priority logic, but in the difficulty of choosing suitable priority parameters when traffic conditions change rapidly. In dense short-block networks, a small extension at one intersection may immediately affect downstream queues, which makes single-intersection priority rules less reliable [21–23]. Multi-intersection coordination has been explored through fuzzy logic [24] and connected-vehicle-based approaches [25–27], but rule-based design limits adaptability.

RL-based traffic signal control has demonstrated adaptive phase selection using queue length, waiting time, and traffic pressure as inputs [9–11, 28, 29]. Recent DRL-based BSP methods such as D^{2}TSP [13] and multi-objective DRL-TSP [14] apply RL to priority decisions. However, most of these methods describe the signal action as a discrete phase choice—acceptable for general signal control but less suitable for BSP, where the duration of green extension or red truncation directly determines whether a bus can pass. Furthermore, they remain focused on isolated intersections or small networks, and do not consider edge-side computing latency as part of the control environment.

2.4 Hybrid-Action Multi-Agent Reinforcement Learning for Edge-Deployed Control

Multi-agent RL (MARL) enables distributed decision-making for multi-intersection control. Chu *et al.* [30] validated MARL on a 400-intersection grid. The CTDE paradigm allows agents to use global information during training while relying only on local observations during execution, making it suitable for edge deployment scenarios. Beyond RL, model predictive control has also been applied to connected vehicle platoons under dynamic communication topologies [31].

Hybrid action spaces have been studied in robotics and game AI. Li *et al.* [32] proposed Hybrid Action Representation (HyAR), which maps discrete actions and their associated continuous parameters into a compact latent action space using an embedding table and a CVAE. However, the original HyAR was not designed for multi-agent traffic control, BSP constraints, or edge-condition-aware execution. Bouktif *et al.* [12] introduced hybrid actions for single-intersection signal control, and Haarnoja *et al.* [33] proposed SAC for continuous action spaces. Directly applying these algorithms to BSP is not straightforward: phase order, minimum green time, bus ETA, and compensation constraints must all be respected within the RL decision cycle. These studies indicate that latency-sensitive BSP requires a decision process that can jointly handle phase-duration coupling, neighboring-intersection influence, bus-arrival feasibility, and edge-side execution deadlines. In this paper, existing hybrid-action and multi-agent coordination tools are adapted to this BSP-specific control setting rather than being treated as new generic learning architectures.

2.5 Summary and Research Gap

Prior studies have established useful foundations for edge-assisted ITS, RL-based traffic signal control, and hybrid-action reinforcement learning. For latency-sensitive BSP in dense short-block vehicular edge networks, three issues are still not well addressed. First, most RL-based signal control methods use discrete phase actions and do not jointly optimize phase selection and green duration. Second, existing BSP studies mainly focus on isolated intersections or small corridors, with limited consideration of queue propagation in dense networks. Third, edge-side communication delay, inference latency, workload, bandwidth, and packet loss are usually treated as external system metrics rather than contextual variables that affect signal-control decisions. Edge-HyAR-BSP is designed to address these issues from a BSP-control perspective. It represents each priority decision as a coupled phase-duration action, evaluates whether the candidate action can be executed within bus-arrival and edge-side deadline constraints, and applies compensation-aware adjustment to reduce disturbance to general traffic phases. These tools are used only to support the BSP control process, whose key requirements are phase-duration coupling, executable priority timing, neighbor-intersection influence, and compensation-aware recovery. Table 1 summarizes the functional coverage of different BSP-related method categories, with emphasis on whether hybrid actions, multi-intersection coordination, edge-side execution awareness, and bus-priority compensation are jointly considered.

Table 1: Comparison of method categories for BSP requirements.

Method Category	Hybrid Action	Multi-Int. Coord.	Edge-Aware	Bus Compens.
Rule-based BSP	–	–	–	Partial
Discrete RL control	–	Partial	–	–
Hybrid RL control	✓	Limited	–	Limited
Edge-HyAR-BSP	✓	✓	✓	✓

3 System Model and Problem Formulation

3.1 End-Edge-Cloud Collaborative Architecture

We consider a dense urban road network consisting of I signalized intersections, each equipped with roadside sensing devices (cameras, radars, detectors), signal controllers, and edge computing nodes. The architecture is inspired by general end–edge–cloud collaborative computing principles [15, 19, 34] and is adapted here for real-time control. Buses and connected vehicles upload their position, speed, ETA, and passenger occupancy information through V2X communication. The control architecture, illustrated in Figure 1, operates as follows:

- **Perception layer (end):** Vehicles and roadside sensors collect real-time traffic states including vehicle counts, queue lengths, waiting times, bus arrival flags, and ETA values.
- **Edge computing layer:** Each roadside edge node receives local lane-level states (vehicle count, queue length, waiting time), bus states (arrival flag, ETA, speed), and edge states (communication delay, inference latency, workload) as input. It aggregates relevant local and neighboring traffic observations, generates a candidate phase-duration priority action, checks its feasibility under signal-safety and edge-deadline constraints, and sends the adjusted action to the signal controller within the decision window.
- **Cloud platform layer:** The cloud is responsible for centralized training, historical data management, model validation, and periodic policy updates. Real-time multi-intersection coordination is conducted through neighboring-state aggregation and decentralized inference at roadside edge nodes.

This architecture, illustrated in Figure 1, forms a closed-loop process of "perception → edge inference → signal execution → feedback collection → cloud update." Compared with cloud-centric control, edge deployment reduces the state upload and control feedback paths, which is critical

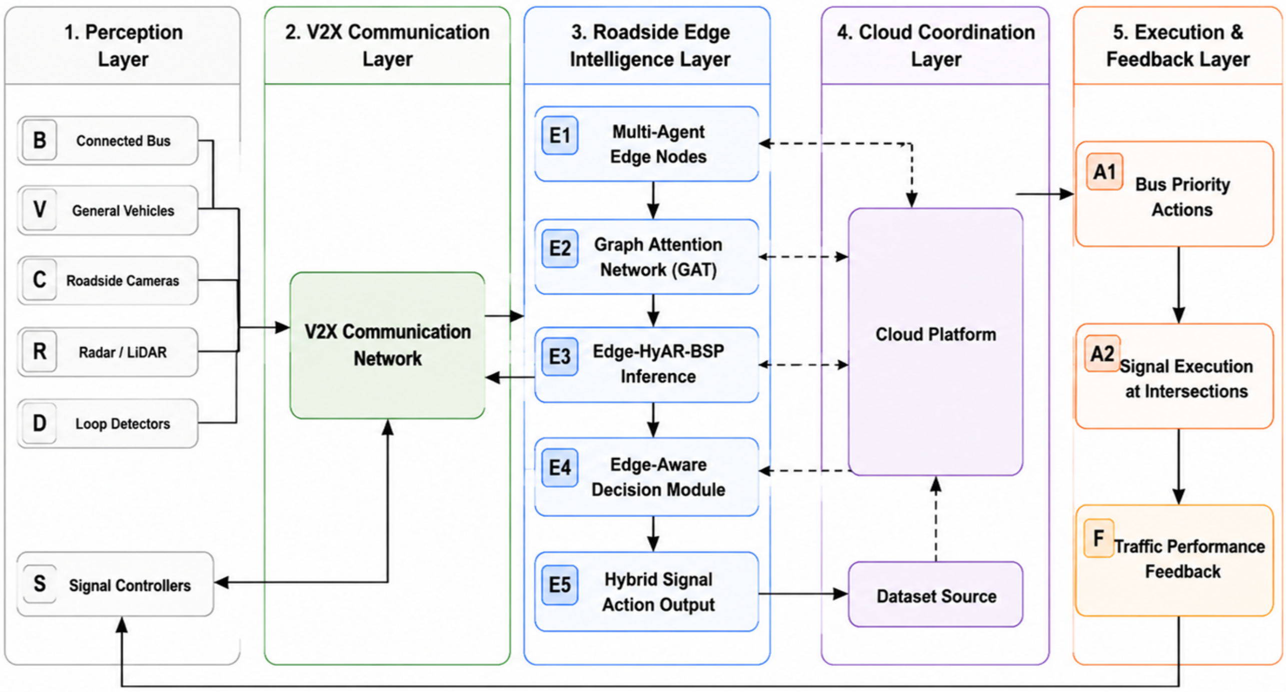

Figure 1: End-edge-cloud collaborative architecture. Connected vehicles and roadside sensors provide traffic, bus, and edge states to roadside edge nodes. Edge nodes generate candidate phase-duration priority actions, verify their feasibility under control-window constraints, and send executable actions to signal controllers. The cloud supports centralized training, model validation, and policy updates using project-based datasets from Rongdong.

for dense urban networks where BSP decision windows are short (typically 10–30 s).

3.2 Project-Based Data Source and Study Area

The study area is the Rongdong District of Xiong'an New Area, China, which features a narrow-road dense-network layout with short intersection spacing (150–300 m) and high signal coupling. The data used in this study are derived from the Xiong'an New Area Science and Technology Innovation Project "Research on Multi-Granularity Traffic System Simulation and Collaborative Control Technology for the Narrow-Road Dense-Network in the Xiong'an New Area" under Grant No. 2022XAGG0126.

The project developed a multi-granularity traffic simulation and collaborative control system covering large-scale traffic situation inference, multi-intersection signal coordination, BSP, vehicle-road cooperation, and edge-cloud collaborative computing. As illustrated in Figure 2 and summarized in Table 2, according to the project evaluation materials, the demonstration environment covered 102 signalized intersections, 5 bus routes with priority control, and 18,937 roadside sensing terminals in the Rongdong area, supporting 7 transportation modes including walking, non-motorized vehicles, buses, passenger cars, taxis, freight vehicles, and connected automated vehicles. The project has produced related publications in traffic flow modeling and cooperative control [35, 36].

The data used in this paper include bus GPS records, signal timing logs, BSP trigger logs, roadside detector records, and platform-level aggregated traffic performance indicators.

Table 2: Description of the project-based dataset

Item	Description
Project source	Xiong'an New Area S&T Innovation Project
Grant No.	2022XAGG0126
Study area	Rongdong District, Xiong'an New Area
Network scale	102 signalized intersections
Bus routes	5 routes; Route 302 for detailed BSP evaluation
Data sources	Bus GPS, signal timing logs, roadside detector records, platform statistics
Edge/computing data	Edge inference logs, communication delay records, task completion records
Data form	Aggregated and anonymized operational data
Excluded samples	Accidents, construction, extreme weather, equipment offline, incomplete signal cycles

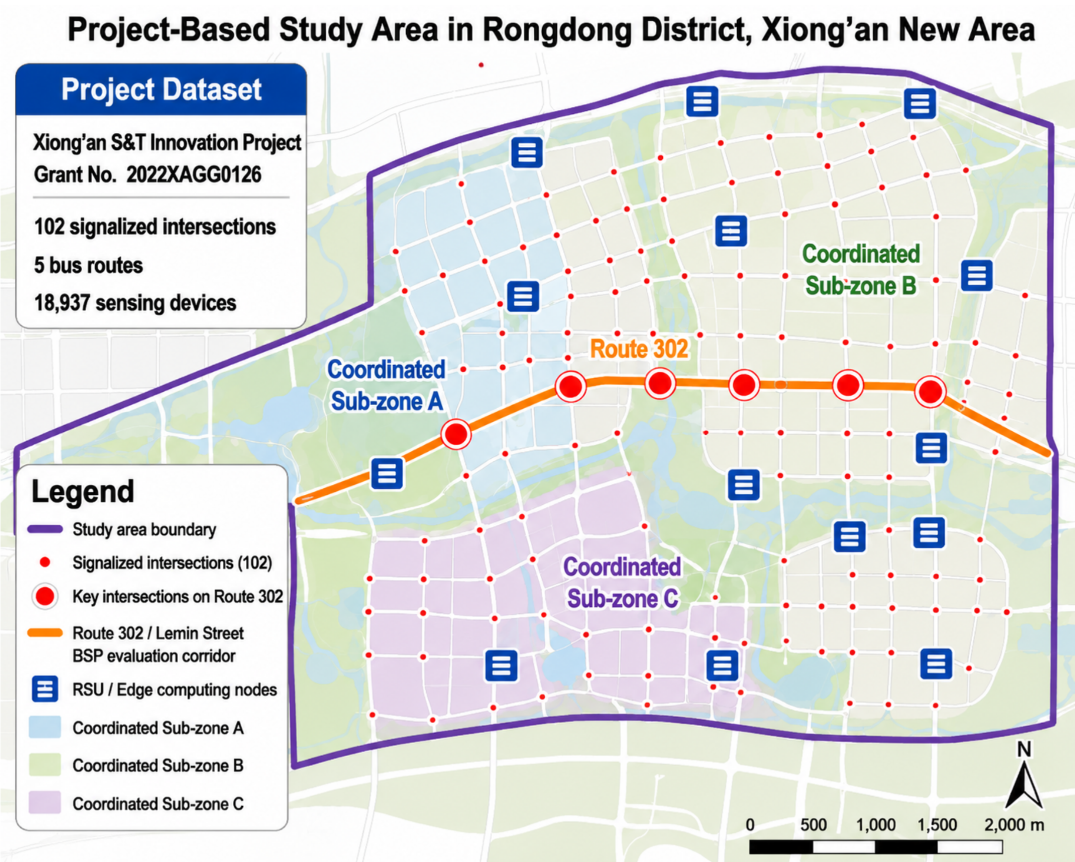

Figure 2: Project-based study area in the Rongdong District of Xiong'an New Area. The figure illustrates the real-world narrow-road dense-network environment used in this study, including 102 signalized intersections, five bus routes, and the Route 302/Lemin Street corridor selected for detailed BSP evaluation. The map also highlights the coordinated control subzones and the deployment locations of RSU/edge computing nodes, which support the collection of traffic states, BSP trigger logs, signal timing records, and edge computing indicators.

3.3 Traffic Network Model

Let the road network contain I signalized intersections, denoted by $\mathscr{I} = \{1, 2, \ldots, I\}$. Each intersection $i \in \mathscr{I}$ is controlled by an agent. The set of incoming lanes at intersection i is denoted by $\mathscr{L}_i$, and the set of available signal phases is denoted by $\mathscr{K}_i = \{1, 2, \ldots, K_i\}$.

The traffic state of intersection i at time t is defined as:

$$s_i(t) = \left(t,\, p_{l_i},\, d_{l_i},\, q_{l_i},\, w_{l_i},\, o_{n_{\text{adj}}}\right) \tag{1}$$

where t is the current time, $p_{l_i} \in \{0, 1\}$ indicates whether lane l has the right-of-way, d_{l_i} is the vehicle count on lane l, q_{l_i} is the queue length, w_{l_i} is the average waiting time, and $o_{n_{\text{adj}}}$ represents aggregated information from all adjacent intersections.

3.4 Bus State Model

When a bus is detected approaching intersection i, the bus-related state is augmented as:

$$s_i^{\text{bus}}(t) = \left(B_{l_i},\, B_{\text{eta}},\, v_{\text{bus}},\, \omega_{\text{bus}}\right) \tag{2}$$

where $B_{l_i} \in \{0, 1\}$ indicates whether a bus is on lane l, B_{eta} is the estimated time of arrival at the stop line, v_{bus} is the current bus speed, and ω_{bus} denotes passenger occupancy or priority weight.

3.5 Edge Computing State Model

To capture edge computing constraints for edge-assisted BSP control, we explicitly define the edge state:

$$s_i^{\text{edge}}(t) = \left(\tau_{\text{comm}},\, \tau_{\text{infer}},\, \rho_{\text{load}},\, \beta_{\text{bw}},\, \eta_{\text{loss}}\right) \tag{3}$$

where τ_{comm} is the V2X communication delay, τ_{infer} is the edge inference latency, ρ_{load} is the edge node workload (CPU/memory utilization), β_{bw} is the available bandwidth, and η_{loss} is the packet loss or state loss ratio. These edge variables are observed contextual factors rather than controllable resource-allocation actions. They are treated as contextual observations that affect the reliability and timeliness of BSP execution: high communication delay reduces the alignment between bus ETA and signal timing, increased inference latency shrinks the feasible decision window, elevated workload degrades task completion rate, limited bandwidth constrains neighbor-state sharing, and packet loss undermines the accuracy of bus and queue state estimation. By including them in the augmented state, the policy can learn to select more conservative or robust signal actions under degraded edge conditions.

The complete augmented state is given by:

$$S_i(t) = \left(s_i(t),\, s_i^{\text{bus}}(t),\, s_i^{\text{edge}}(t)\right) \tag{4}$$

For each BSP decision task, the control action is useful only when the complete sensing–inference–delivery pipeline finishes before the valid control window expires. The effective control window is determined by the bus ETA, the current signal phase, and safety constraints:

$$\tau_{\text{comm},i}(t) + \tau_{\text{pre},i}(t) + \tau_{\text{infer},i}(t) + \tau_{\text{act},i}(t) \le W_{\text{ctrl},i}(t) \tag{5}$$

where $\tau_{\text{pre},i}$ is the preprocessing time, $\tau_{\text{act},i}$ is the action delivery and controller acknowledgement time, and $W_{\text{ctrl},i}(t)$ is the valid control window set by the bus ETA and phase constraints. When this condition is violated, the BSP action risks missing the bus arrival window and should be suppressed or projected to a safer general traffic action.

3.6 Optimization Objective

The objective is to minimize traffic, operational-risk, and decision-latency costs while accounting for time-varying edge conditions:

$$\begin{aligned} J(\pi) = \mathbb{E}_\pi \Big[\sum_{t=0}^{T} \xi^t \big(& w_b D_{\text{bus}}(t) + w_v D_{\text{veh}}(t) + w_q Q_{\text{len}}(t) + \\ & w_r R_{\text{risk}}(t) + w_d \cdot \mathbb{I}(T_{\text{e2e}} > W_{\text{ctrl}}) + w_s \cdot \text{AoI}(t) \big) \Big] \end{aligned} \tag{6}$$

where D_{bus} is bus delay, D_{veh} is general traffic delay, Q_{len} is queue length, R_{risk} is operational risk (spillback and congestion propagation), $\mathbb{I}(T_{\text{e2e}} > W_{\text{ctrl}})$ penalizes deadline violations, $\text{AoI}(t)$ penalizes state staleness, $w_b, w_v, w_q, w_r, w_d, w_s$ are non-negative weighting coefficients, and $\xi \in (0, 1)$ is the discount factor.

Table 3 summarizes how the major framework components relate to the edge-computing aspect of this work.

4 Proposed Method: Edge-HyAR-BSP

This section presents the control procedure of Edge-HyAR-BSP for deadline-aware cooperative BSP, including phase-duration action construction, latent duration decoding, neighbor-state aggregation, priority-action feasibility projection, compensation-aware execution, and edge-condition-sensitive reward design. The overall workflow is illustrated in Figure 3.

Table 3: Edge-computing relevance of major framework components.

Component	Edge-Computing Role	Control Effect
Roadside edge inference	Reduces communication and feedback delay	Enlarges BSP decision margin
Edge state variables	Represent delay, workload, bandwidth, and packet loss	Enable context-aware action selection
Deadline-aware feasibility projection	Filters actions that miss control windows	Improves execution reliability
Cloud-assisted CTDE	Uses global information for training	Supports decentralized edge execution
GAT neighbor aggregation	Shares spatial information	Improves corridor coordination

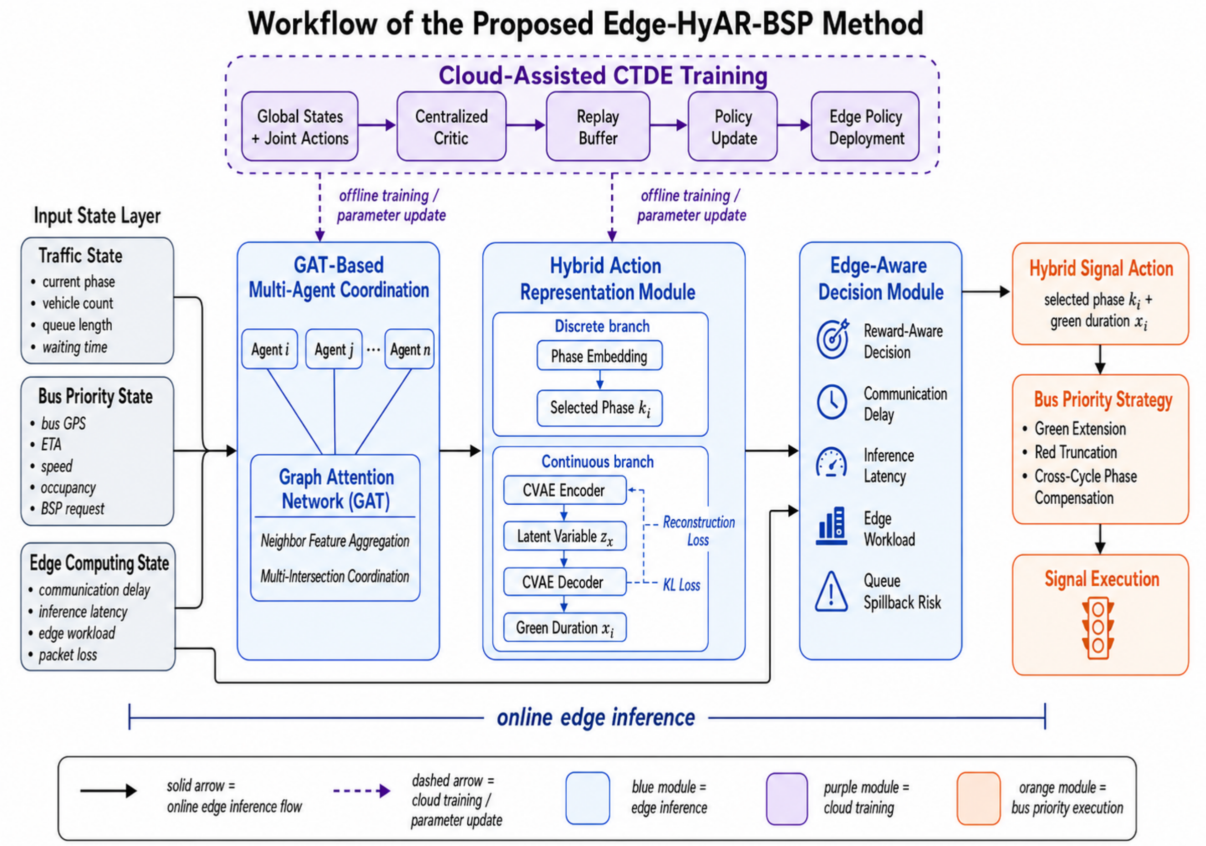

Figure 3: Workflow of the proposed Edge-HyAR-BSP method. The framework takes traffic, BSP, and edge computing states as inputs. Neighboring-intersection information is aggregated for corridor-level coordination. Phase-duration coupled actions are generated through latent representation and continuous decoding. Priority actions are executed through green extension, red truncation, and cross-cycle compensation, while cloud-assisted training periodically updates the edge policy.

4.1 Hybrid Action Space Definition

Unlike conventional discrete-action traffic signal control, we define a hybrid action for each intersection i:

$$\mathscr{A}_i = \{(k_i, x_{k_i}) \mid x_{k_i} \in \mathscr{X}_{k_i}, \forall k_i \in \mathscr{K}_i\} \tag{7}$$

where $k_i \in \mathscr{K}_i$ denotes the selected signal phase (discrete), and $x_{k_i} \in [x_{\min}, x_{\max}]$ denotes the corresponding green time duration (continuous). This formulation requires the agent to simultaneously output a discrete phase index and a continuous duration, providing finer-grained control than purely discrete action spaces.

4.2 Hybrid Action Representation Module

Direct optimization in the hybrid action space faces two challenges: (i) structural dependency between discrete phase selection and continuous duration (optimal green time distributions differ across phases); and (ii) conventional policy networks struggle to jointly output discrete and continuous distributions while maintaining their coordination. To represent the coupled discrete-continuous nature of BSP actions, this study adopts a latent action representation inspired by hybrid-action learning [32]. The representation is redesigned for signal control by conditioning the continuous duration decoder on bus ETA, phase feasibility, and edge-side execution constraints. Therefore, the objective of this module is not to introduce a new generic hybrid-action algorithm, but to adapt latent hybrid-action modeling to the feasibility requirements of BSP. The module uses a discrete action embedding table and a CVAE as implementation tools, while its design objective is to satisfy BSP-specific feasibility requirements, including phase order, green-time bounds, bus ETA, and edge-side execution deadlines. In this work, the latent representation is embedded into the BSP control process so that candidate phase-duration actions can be evaluated together with bus states, signal-safety rules, neighboring-intersection conditions, and edge-side execution constraints.

4.2.1 Discrete Action Embedding Table

A learnable embedding table $\mathbf{E}_{\zeta,K_i} \in \mathbb{R}^{K_i \times d_1}$ is constructed to map each discrete phase index k_i to a latent embedding $\mathbf{e}_{\zeta,k_i} \in \mathbb{R}^{d_1}$:

$$\mathbf{e}_{\zeta,k_i} = \text{Embedding}_\zeta(k_i) \tag{8}$$

where d_1 is the embedding dimension and ζ denotes learnable parameters. During inference, the nearest-neighbor search in the embedding space recovers the discrete phase $\hat{k}_i$ from the latent representation.

4.2.2 Conditional Variational Auto-Encoder

In the BSP setting, the decoder is used to produce a green-time adjustment that is conditioned not only on the selected phase but also on traffic state and feasible bus-arrival windows. The continuous duration is therefore not generated independently of the phase; it is decoded from a compact latent variable so that different phases can correspond to different feasible duration distributions. The encoder (parameterized by ϕ) outputs the mean μ_x and standard deviation σ_x of the variational posterior:

$$q_\phi(z_x \mid x_{k_i}, s, \mathbf{e}_{\zeta,k_i}) = \mathcal{N}\left(\mu_x(s, \mathbf{e}_{\zeta,k_i}),\, \sigma_x^2(s, \mathbf{e}_{\zeta,k_i})\right) \quad (9)$$

The latent continuous action is sampled via the reparameterization trick:

$$\hat{z}_x = \mu_x + \sigma_x \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (10)$$

The decoder parameterized by θ_{dec} reconstructs the green time duration from the latent representation and conditions:

$$\hat{x}_{k_i} = f_{\theta_{\text{dec}}}(\hat{z}_x, s, \mathbf{e}_{\zeta,k_i}) \quad (11)$$

The CVAE is trained by minimizing:

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{z_x \sim q_\phi}\Big[\underbrace{\|x_{k_i} - \hat{x}_{k_i}\|_2^2}_{\mathcal{L}_{\text{recon}}} + \beta_{\text{KL}} \cdot \underbrace{D_{\text{KL}}(q_\phi \,\|\, p(z))}_{\mathcal{L}_{\text{KL}}}\Big] \quad (12)$$

where $\mathcal{L}_{\text{recon}} = \|x_{k_i} - \hat{x}_{k_i}\|_2^2$ is the reconstruction loss, β_{KL} is a weighting coefficient, $q_\phi = q_\phi(z_x \mid x_{k_i}, s, \mathbf{e}_{\zeta,k_i})$ is the variational posterior, and $p(z) = \mathcal{N}(0, \mathbf{I})$ is the standard normal prior. The KL divergence for diagonal Gaussian posterior is $\mathcal{L}_{\text{KL}} = \frac{1}{2}\sum_j(\mu_j^2 + \sigma_j^2 - 1 - \log\sigma_j^2)$.

The CVAE and embedding table are pretrained before the RL loop. An initial action dataset is collected using a rule-based BSP controller with randomly sampled feasible phase-duration pairs, and the CVAE is trained on this dataset to learn a compact latent representation of valid hybrid actions. After pretraining, the CVAE and embedding table parameters are frozen during the main RL training phase to avoid latent-space drift. The replay buffer stores the latent representation alongside each transition at collection time, so that critic evaluation uses a stable action encoding that does not change across training steps. The discrete phase is recovered from the embedding table via nearest-neighbor lookup during execution. During training, the actor's latent discrete output is used directly by the critic without nearest-neighbor discretization, so that gradients flow through the continuous latent representation. The nearest-neighbor search is applied only at execution time and when constructing the feasibility projection. The decoder generates the continuous green-time adjustment conditioned on the latent sample, state, and discrete phase embedding.

4.3 Multi-Agent Cooperative Learning Framework

4.3.1 CTDE Training Architecture

We adopt a centralized training with decentralized execution (CTDE) framework. During training, the critic network can access global traffic states, edge states, and joint actions of all agents to learn the hybrid action value function. During execution, each edge node performs inference based only on local observations and aggregated neighboring information, ensuring scalability and real-time performance.

The policy network follows an Actor-Critic architecture. The Actor network outputs the latent hybrid action $(\mathbf{e}, z_x)$, which is decoded via the HyAR module to the original action (k_i, x_{k_i}). The Critic network evaluates state-action pairs to guide policy updates. Training follows an off-policy deterministic policy gradient approach. The critic is updated by minimizing the temporal-difference loss:

$$\mathcal{L}_Q(\theta_i^Q) = \mathbb{E}_{(\mathbf{S},\mathbf{a},r,\mathbf{S}')\sim\mathcal{D}}\Big[\big(Q_{\theta_i^Q}(\mathbf{S}, \mathbf{a}) - y_i\big)^2\Big] \quad (13)$$

with the target value $y_i = r + \xi\, Q_{\bar{\theta}_i^Q}(\mathbf{S}', \mathbf{a}')$, where $\mathbf{S}$ is the global state, $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_I)$ is the joint latent action vector, and $\mathbf{a}'$ is computed from the target actor networks. The actor is updated via the deterministic policy gradient, where the critic evaluates the joint action with agent i's own action coming from its current policy while other agents' actions are taken from the replay buffer:

$$\mathcal{L}_\pi(\theta_i^\pi) = -\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\Big[Q_{\theta_i^Q}\big(\mathbf{S}, (\mathbf{a}_{-i}, \pi_{\theta_i^\pi}(\mathbf{S}_i))\big)\Big] \quad (14)$$

where $\mathbf{a}_{-i}$ denotes the latent actions of all agents except i from the replay buffer. Target networks are updated via soft Polyak averaging: $\bar{\theta}_i \leftarrow \tau_{\text{soft}}\,\theta_i + (1 - \tau_{\text{soft}})\,\bar{\theta}_i$ for both actor and critic. During centralized training, the critic receives the global state $\mathbf{S}$ and joint latent actions, whereas each actor uses only its local observation $\mathbf{S}_i$ and GAT-aggregated neighbor information. The replay buffer stores the candidate action $\tilde{\mathbf{a}}_t$, the executed feasible action $\mathbf{a}_t$, and the frozen latent representation $\mathbf{z}_t$ collected at the same decision step. The critic is trained on the executed feasible action and its corresponding stable latent code.

4.3.2 Graph Attention Network for Spatial Dependencies

In dense short-block corridors, neighboring intersections affect each other mainly through queue spillback and bus-arrival propagation. The GAT layer is therefore used to weight neighboring states according to their relevance to the current BSP decision, aggregating information from adjacent nodes with attention-weighted importance. For intersection i, the aggregated representation is:

$$\mathbf{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right) \quad (15)$$

where $\mathcal{N}(i)$ is the neighbor set of intersection i, α_{ij} is the attention coefficient between intersections i and j, $\mathbf{W}$ is a learnable weight matrix, $\mathbf{h}_j$ is the state representation of neighboring intersection j, and σ is a nonlinear activation function. The attention coefficient is computed as:

$$\alpha_{ij} = \frac{\exp\big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_j])\big)}{\sum_{k\in\mathcal{N}(i)}\exp\big(\text{LeakyReLU}(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_k])\big)} \quad (16)$$

where $\mathbf{a}$ is a learnable attention vector and $\|$ denotes vector concatenation. CTDE and GAT are used here as established coordination tools; the paper's focus is their integration with hybrid BSP actions, edge-side execution constraints, and feasibility projection rather than proposing a new graph learning architecture.

4.4 BSP Strategy

When a BSP request is triggered (i.e., a roadside detector detects an approaching bus), the edge node selects a priority strategy based on bus ETA, current phase, queue status, and neighboring traffic states. Two major priority strategies—green extension and red truncation—are employed, with cross-cycle compensation used to restore timing balance after priority actions.

4.4.1 Green Extension

Green extension is activated when the bus-serving phase is active, or is scheduled to become active before the bus arrives, and the estimated bus arrival time falls shortly after the nominal end of the bus-serving green interval, as illustrated in Figure 4(a). The green time of the bus-priority phase is extended by Δg_{bus}, determined as:

$$\Delta g_{\text{bus}} = \min\left\{\Delta T_{\text{bus}}, \sum_{p\in\mathcal{K}_i\setminus\{k_{\text{bus}}\}} \Delta g_{p,\max}\right\} \tag{17}$$

where ΔT_{bus} is the required extension to cover the bus arrival gap, $\Delta g_{p,\max}$ is the maximum compressible green time of conflicting phase p, and $\mathcal{K}_i\setminus\{k_{\text{bus}}\}$ excludes the bus-serving phase. The extension is bounded by minimum-green and maximum-extension constraints: $\Delta T_a \leq \Delta g_{\text{bus}} \leq \Delta T_b$.

4.4.2 Red Truncation

If a bus is detected during a non-bus-priority phase and is predicted to arrive before the next green phase starts, the red truncation strategy is activated, as illustrated in Figure 4(b). The conflicting phase is shortened, and the bus-priority phase is advanced by ΔT_{bus}, with the earliest green start time bounded by phase minimum constraints.

4.4.3 Cross-Cycle Phase Compensation

To maintain total cycle stability and prevent green wave disruption after BSP triggering, we design a cross-cycle phase compensation mechanism, as depicted in Figure 4(c). After BSP is triggered, the system compensates the affected phases in subsequent cycles. The compensation amount H satisfies:

$$H = -\Delta T_{\text{bus}} \tag{18}$$

The compensation is evenly distributed over N subsequent cycles, with each cycle compensating $-\Delta T_{\text{bus}}/N$. A compensation state variable $H_i(t)$ is introduced into the state space, representing the accumulated pending compensation at intersection i. A penalty term $r_{\text{comp}} = -\psi \cdot |H_i(t)|$ is added to the reward function to encourage cycle balance recovery. A maximum compensation threshold $H_{\max}$ is set: when

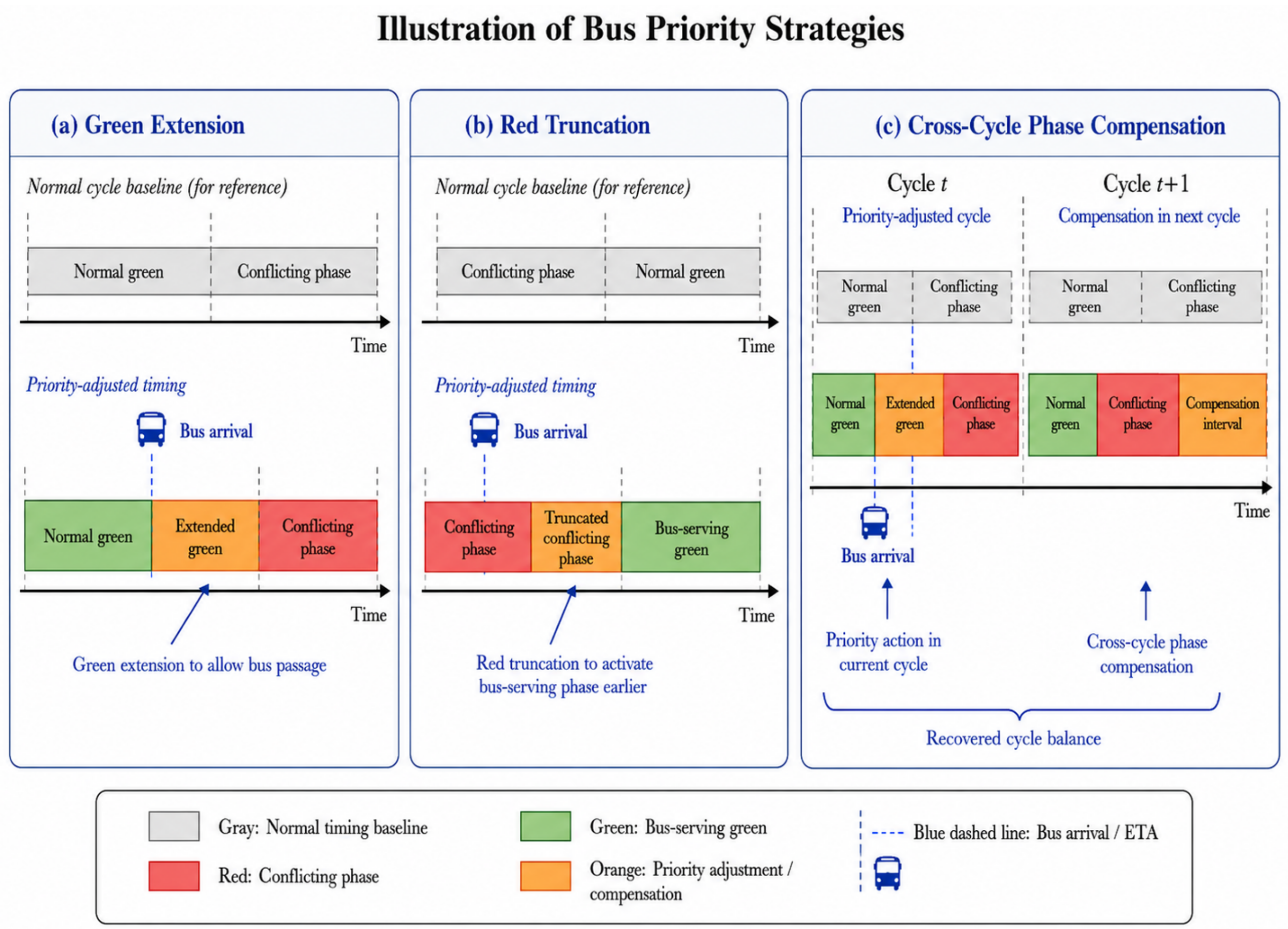

Figure 4: Illustration of BSP strategies in the proposed Edge-HyAR-BSP framework. (a) Green extension prolongs the current bus-serving green phase. (b) Red truncation shortens the conflicting phase to start the bus-serving phase earlier. (c) Cross-cycle phase compensation restores timing balance after priority actions.

$|H_i(t)| > H_{\max}$, further BSP triggering is prohibited until the accumulated compensation is reduced below the threshold.

4.5 Edge-Aware Reward Function

The reward function evaluates network-level traffic efficiency, BSP, fairness, and edge conditions:

$$r_t = \sum_{i\in\mathscr{I}}\Big[\sum_{l\in\mathscr{L}_i}\left(\alpha\Delta d_{l_i} - \beta q_{l_i} - \gamma w_{l_i}\right) + \kappa r_i^{\mathrm{bus}} - \mu r_i^{\mathrm{spill}} - \varphi r_i^{\mathrm{switch}} - \psi|H_i(t)| - \lambda_d D_i(t) - \lambda_s A_i(t) - \lambda_f F_i(t)\Big] \tag{19}$$

where Δd_{l_i}, q_{l_i}, w_{l_i} are lane-level throughput change, queue length, and waiting time. The BSP term $r_i^{\mathrm{bus}} = -\sum_{b\in B_i(t)}\omega_b D_b(t)$ penalizes bus delay, weighted by passenger occupancy ω_b. The terms r_i^{spill}, r_i^{switch}, and $|H_i(t)|$ denote spillback penalty, phase-switching penalty, and accumulated pending compensation (Section 4.4.3).

The three edge-related penalty terms penalize execution outcomes: $D_i(t) = \mathbf{1}_{\{T_{\mathrm{e2e},i} > W_{\mathrm{ctrl},i}\}}$ activates on deadline violation; $A_i(t) = \mathrm{AoI}_i(t) = t - u_i(t)$ measures state staleness in decision steps, normalized by the maximum tolerated age and computed as the maximum among local and neighboring state components (when a packet is lost, the previous valid state is retained and its AoI increases); $F_i(t) = \mathbf{1}_{\{\tilde{a}_i \notin \mathscr{A}_{\mathrm{feas},i}\}}$ penalizes candidate actions that fall outside the feasible set.

Communication delay, inference latency, and workload influence these outcomes through the state and feasibility constraints. Bandwidth and packet loss affect the policy through the augmented state and masked GAT aggregation. All Greek symbols denote non-negative weighting coefficients.

4.6 Edge-Aware Feasibility Projection

After the Actor and HyAR decoder generate a candidate hybrid action (k_i, x_{k_i}), the action is projected onto a feasible set before execution. The feasible set $\mathscr{A}_{\mathrm{feas},i}(t)$ is determined by signal-safety constraints, minimum green time, clearance time, phase order, bus ETA, compensation state, and the available edge-side control window:

$$\begin{aligned}\mathscr{A}_{\mathrm{feas},i}(t) = \big\{&(k_i, x_{k_i}) \mid x_{\min} \le x_{k_i} \le x_{\max},\ g_{\min} \le g_{k_i} \le g_{\max},\\ &T_{\mathrm{e2e},i}(t) \le W_{\mathrm{ctrl},i}(t),\ |H_i(t)| \le H_{\max},\\ &\text{phase-safety constraints satisfied}\big\}\end{aligned} \tag{20}$$

This projection prevents the policy from executing priority actions that are theoretically optimal in the traffic state but infeasible under current communication or computing delays. The projection procedure is summarized as follows:

1. Remove phases that violate the current phase order or clearance-time constraints.
2. Clip the candidate green duration to $[x_{\min}, x_{\max}]$ and $[g_{\min}, g_{\max}]$.
3. Verify that the bus ETA can be served within the remaining control window $W_{\mathrm{ctrl},i}$.
4. Check that $|H_i(t)| \le H_{\max}$ (compensation threshold not exceeded).
5. If multiple feasible actions remain, select the one closest to the candidate action under Euclidean distance in the latent space.
6. If no feasible priority action exists, execute the nominal general traffic action for the current signal plan.

When no feasible priority action exists, the controller reverts to a safe general traffic action within the current signal plan. This is distinct from a dedicated fault-tolerant fallback controller for severe edge-node failures, which remains future work (Section 6.6). This mechanism directly connects edge-side latency and reliability to the control decision space.

Because the feasibility projection includes discrete phase filtering and hard safety constraints, it is not fully differentiable. The actor is therefore optimized in the latent action space with an additional infeasibility penalty, while the critic is trained on executed feasible actions. This introduces an approximation between policy optimization and environment execution, which is partially mitigated by CVAE pretraining and the feasibility-aware penalty terms.

The training follows an off-policy actor-critic framework with deterministic policy gradients, operating on the latent hybrid actions produced by the HyAR module. During centralized training, the critic receives the global state and the joint latent action, where the latent action refers to the frozen encoding of the executed feasible hybrid actions (post-projection), not the pre-projection actor outputs. Each actor uses only local observations and GAT-aggregated neighbor information. During decentralized execution, only the actor and the HyAR decoder are deployed at each roadside edge node. The feasibility projection is applied during both online data collection and target-action generation. The replay buffer stores, alongside each transition, the executed feasible action in its original hybrid form (k_i, x_{k_i}) and its latent representation $(\mathbf{e}_{\zeta,k_i}, z_x)$ at collection time. Because the CVAE and embedding table are frozen after pretraining, the stored latent code remains consistent throughout RL training and no re-encoding is required during critic updates. This design ensures that the critic learns on actions the environment actually received, while the actor gradient flows through the stable latent space. The feasibility projection is integrated into the training loop as step 5–6 of Algorithm 1.

5 Experiments

5.1 Experimental Setup

5.1.1 Project-Based Study Area and Data Collection

The experimental data and case-study scenario are derived from the Xiong'an New Area Science and Technology Innovation Project (Grant No. 2022XAGG0126). The study area is the Rongdong District, which contains 102 signalized intersections divided into 3 coordinated control sub-zones (15–30 intersections each) and multiple bus routes. The road network features the typical narrow-road dense-block layout: average intersection spacing of approximately 200 m, road widths of 15–25 m, and 2–4 lanes per approach. Intersection characteristics are summarized in Table 4.

Algorithm 1 Edge-HyAR-BSP Training Procedure

Require: Road network $\mathscr{I}$, episodes M, discount factor ξ, learning rate η
Ensure: Trained decentralized actors $\{\pi_i\}_{i=1}^{I}$

Pretrain and freeze CVAE $(\zeta, \phi, \theta_{\text{dec}})$ and embedding table $\mathbf{E}_{\zeta,K_i}$; initialize Actor $\{\theta_i^{\pi}\}$, Critic $\{\theta_i^{Q}\}$, and target copies
for episode $= 1$ to M **do**
  Reset environment
  for timestep $t = 0$ to T **do**
    for each intersection $i \in \mathscr{I}$ in parallel **do**
      Assemble state $S_i(t)$: local traffic, bus, and edge states; aggregate neighbors via GAT
      Generate latent hybrid action, decode to (k_i, x_{k_i}) via HyAR module
      Project onto feasible set (min-green, clearance, phase order, bus ETA, compensation, control-window)
      Execute as GE, RT, or no-priority; observe reward r_t and next state $S_i(t+1)$
    end for
    Store $(\mathbf{S}_t, \tilde{\mathbf{a}}_t, \mathbf{a}_t, \mathbf{z}_t, r_t, \mathbf{S}_{t+1})$ in replay buffer $\mathscr{D}$, where $\tilde{\mathbf{a}}_t$ is the joint candidate action, $\mathbf{a}_t$ is the executed feasible action, and $\mathbf{z}_t$ is the frozen latent representation
    Sample mini-batch from $\mathscr{D}$; update Critic and Actor via loss functions (CVAE and embedding table remain frozen)
    Soft-update critic and actor target networks
  end for
end forreturn trained decentralized actors $\{\pi_i\}_{i=1}^{I}$

Table 4: Characteristics of test intersections

Intersection Range	Road Type	Lanes	Cycle (s)
1–25	Arterial roads	4	120
26–50	Secondary roads	3	100
51–75	Branch/connecting	2–3	90–100
76–102	BSP priority	3–4	90–120

A comparative dataset is constructed based on two distinct signal control schemes. Conventional fixed signal timing is adopted during the baseline phase, from which 14 sets of matched bidirectional bus operation records of Route 302 are gathered. For the optimized phase, the Edge-HyAR-BSP collaborative control framework is deployed on the identical corridor, and another 14 bidirectional bus trips under consistent departure timetables are recorded. The abnormal data entries are removed, leaving a total of 28 fully paired bus samples for controlled comparison.

The dataset is partitioned into three fixed time windows: morning peak (7:00–9:00), off-peak period (10:00–16:00), and evening peak (17:00–19:00). Bus performance metrics are extracted from GPS trajectories, arrival and departure logs, as well as BSP trigger data collected from Route 302 running between Mingzhuyuan Station and Yumeiyuan Station in both directions.

Because the dataset originates from an urban transportation project, raw trajectory-level and device-level records cannot be publicly released due to project management requirements and traffic operation data restrictions. This paper therefore reports anonymized and aggregated indicators. The results should be interpreted as project-based operational evidence rather than a randomized controlled trial.

Table 5 clarifies the scope and evidence type of each evaluation component to avoid conflation between project platform coverage and algorithmic evaluation scale.

Table 5: Evidence scope and evaluation scale of the study

Evidence Type	Network Scope	Main Metrics
Project platform coverage	102 intersections	Aggregated regional indicators
BSP corridor evaluation	15 intersections	Bus speed, delay, stops
Sub-zone coordination	15–30 per zone	Balance index, travel time
Simulation/ablation	102-intersection topology	Robustness, component analysis

The project-based evaluation covers comparable workdays across three time periods: morning peak (7:00–9:00), off-peak (10:00–16:00), and evening peak (17:00–19:00). Samples affected by accidents, construction, extreme weather, equipment offline status, or incomplete signal cycle records are excluded. Bus-related metrics are derived from bus GPS records, BSP trigger logs, and signal timing records of Route 302 (bidirectional) during the evaluation period. Edge decision records are obtained from the platform's edge-node operation logs. Because raw trajectory-level data are not releasable due to project data governance policies, all reported indicators are aggregated and anonymized at the corridor or platform level.

The project defined three demonstration scenarios: (1) regional coordinated signal control in the narrow-road dense-network area, (2) intelligent connected BSP with V2X cooperation, and (3) collaborative control and emergency response under traffic incidents. This paper focuses on the first two scenarios, covering multi-intersection coordination (Section 5.2.3) and BSP (Sections 5.2.1 and 5.2.2).

5.1.2 Baseline Methods

The proposed Edge-HyAR-BSP is constructed based on four mature algorithms (HyAR, CTDE, GAT, vanilla actor–critic RL), which are adjusted to satisfy BSP constraints. To resolve policy–execution mismatch, a non-differentiable feasibility projection operator is designed, and latent-space actor optimization paired with feasible-action critic training is adopted; edge operational metrics are also embedded into state and reward functions. The effectiveness of all customized modules is verified through the following baseline and ablation comparisons:

- **Fixed-Time:** Fixed signal timing optimized offline via the Webster method. Regional coordination and general-traffic comparisons are available from the project platform records.
- **Rule-based BSP:** Rule-based BSP using GE/RT with analytical thresholds [21]. Comparison is provided via

the project before-period baseline (pre-optimization stage without Edge-HyAR-BSP).

- **Cloud-HyAR-BSP:** The cloud-based counterpart of Edge-HyAR-BSP, where all inference is performed in the cloud. Compared in terms of end-to-end latency and task completion rate.
- **HyAR-BSP w/o Edge:** The edge-awareness-removed variant, without edge state variables and edge-execution penalties. Used in the ablation study (Section 5.5) to isolate the contribution of edge-condition awareness.
- **Discrete-MARL-BSP:** A discrete-action variant sharing the same coordination structure, state, reward, and constraints as Edge-HyAR-BSP, but with green duration discretized into 5 fixed bins from $x_{\min}$ to $x_{\max}$.
- **GAT-Rule-BSP:** An attention-free variant using neighbor aggregation combined with fixed-rule GE/RT decisions (analytical thresholds), isolating the contribution of learned policy from graph-based coordination.

Generic edge offloading and resource scheduling algorithms are excluded from comparison, and the influence of edge-side operating conditions on real-time traffic control decisions is defined as the core research objective of this work. Targeted baselines including Cloud-HyAR-BSP and HyAR-BSP w/o Edge are established to quantify the performance gains contributed by edge deployment and edge-condition awareness.

The Discrete-MARL-BSP baseline is constructed by discretizing green time intervals into 5 fixed bins and embedding bus states and bus delay rewards into the reward function, serving as a discrete-action counterpart to our hybrid-action framework. A complete set of comparative baselines together with controlled ablation experiments are implemented to independently measure the effectiveness of hybrid action representation, GAT-based intersection coordination and edge-state perception modules. Reliable attribution of performance improvement to the proposed technical components is therefore achieved, and the interference brought by favorable problem formulation is eliminated. Detailed hyperparameter settings and simulation configurations for Edge-HyAR-BSP and all baselines are summarized in Table 6.

5.1.3 Evaluation Metrics

Traffic efficiency metrics: (1) Average bus delay (s); (2) Average bus speed (km/h); (3) Number of bus stops; (4) Average general traffic delay (s); (5) Average general traffic speed (km/h); (6) Average queue length (veh); (7) Queue spillback rate (%).

Edge computing metrics: (8) End-to-end decision latency (ms); (9) Edge inference latency (ms); (10) Communication delay (ms); (11) Edge node workload (%); (12) Task completion rate (%).

The improvement rate for each metric is calculated as:

$$\eta_+ = \frac{I_{\text{after}} - I_{\text{before}}}{I_{\text{before}}} \times 100\%, \quad \eta_- = \frac{I_{\text{before}} - I_{\text{after}}}{I_{\text{before}}} \times 100\% \tag{21}$$

where η_+ applies to positive metrics (speed, throughput) and η_- applies to negative metrics (delay, stops, queue length).

Table 6: Hyperparameter and Simulation Configurations of Edge-HyAR-BSP.

Category	Parameter	Value
SUMO	Road network	102 intersections, Rongdong
	Peak demand	18500 veh/h
	V2X protocol	IEEE 802.11p, 100m
	Sim step	1 s
	Episode length	3600 s
	Total episodes	8000
	Random seed	2026
	Comm. delay	0–150 ms
Env. Params	Edge workload	30%–95%
	Packet loss	0%–20%
	Max compensation	20 s
	Adjust window	10–30 s
	Replay buffer	100000
CVAE	Latent dim d_1, d_2	6, 6
	Pre-train lr	0.0001
	Encoder layers	256, (128,128)
	Decoder layers	(256,256), 128
	Pre-train samples	50000
	Random seed	2026
MADDPG	Actor lr	0.0003
	Critic lr	0.001
	Discount γ	0.99
	Actor layers	(256,128,128)
	Critic layers	(256,128,64)
	Batch size	256
	Soft update τ	0.005
	Replay buffer size	100000
Baselines	Fixed-Time	Webster timing scheme
	Rule-based BSP	GE/RT BSP
	Cloud-HyAR-BSP	Cloud inference version
	HyAR-BSP w/o Edge	Edge state and penalty removed
	Discrete-MARL-BSP	5-bin discrete green time
	GAT-Rule-BSP	GAT replaced by GE/RT rules

All baselines are grid-searched on the same SUMO platform; CVAE is frozen in RL training. Seed 2026 ensures reproducibility.

5.2 Main Results

5.2.1 BSP Performance

Table 7 presents the BSP performance of Route 302 (bidirectional) during the morning peak period.

In the Route 302 corridor, the operational data indicate measurable improvements in bus efficiency during the selected evaluation periods. The downlink direction benefited more than the uplink direction, likely because its signal offsets and queue conditions were more favorable. Aggregated corridor-level indicators show improved operational performance during the selected after period. Because per-vehicle trajectory records are unavailable for publication, the results should be interpreted as project-based aggregate evidence.

Table 7: BSP performance (Route 302, morning peak)

Metric	Direction	Before	After	Change (%)
Avg. speed (km/h)	Uplink	18.94	21.71	+14.64%
	Downlink	18.18	23.65	+30.09%
Travel time (s)	Uplink	630.93	551.93	-12.52%
	Downlink	671.64	524.79	-21.86%
Avg. delay (s)	Uplink	191.16	117.98	-38.28%
	Downlink	224.78	77.96	**-65.13%**
Avg. stops	Uplink	7.64	4.43	-42.01%
	Downlink	8.29	3.43	**-58.62%**

BSP trigger rate: 39.32% (6.37 triggers/hour on average per intersection).

5.2.2 General Traffic Performance

Table 8 summarizes the general traffic performance before and after deployment.

The operational data from the project platform indicate no aggregate degradation in general-traffic performance during the selected after period. The simulation-based ablation study further suggests that cross-cycle phase compensation contributes to this fairness outcome.

Table 8: General Traffic performance (morning peak)

Metric	Before	After	Change (%)
Average speed (km/h)	32.60	36.28	+11.29%
Average delay (s)	55.32	49.18	-11.10%
Average stops	1.65	1.46	-11.52%

5.2.3 Multi-Intersection Coordination

Table 9 compares regional coordination performance across control schemes.

The project platform's after-period multi-zone coordinated control, which incorporated the proposed BSP framework as one component, showed improved regional indicators compared with the no-control and fixed-time schemes. The sub-zone balance index improved by 62.1% (from 0.58 to 0.94), reduced travel time by 32.3% (from 18.6 to 12.6 min), and decreased congested links by 65.1% (from 32.4% to 11.3%) compared with no control. These regional indicators reflect the integrated project platform, in which Edge-HyAR-BSP served as the BSP-related control component.

Table 9: Regional coordination performance

Control Scheme	Balance Index	Travel Time (min)	Congested Links (%)
No regional coord.	0.58	18.6	32.4%
Fixed-time	0.65	16.8	27.1%
Single sub-zone	0.81	14.2	18.5%
Multi-zone coordinated	**0.94**	**12.6**	**11.3%**

5.2.4 Bus-Vehicle Impact Comparison

Table 10 directly compares bus and general traffic impacts, addressing the fairness concern of whether BSP degrades general traffic performance.

Table 10: Comparative impact of BSP on buses and general traffics

Metric	Bus	General traffic
Avg. speed change	+14.64%∼+30.09%	+11.29%
Avg. delay change	-38.28%∼-65.13%	-11.10%
Avg. stops change	-42.01%∼-58.62%	-11.52%

The bus speed, delay, and stop improvements are visualized in Figure 5(a)–(d).

Within the coverage of the test bus route (Route 302) and the selected sample period, no systematic spillback events were recorded at neighboring intersections. The operational data from the project platform indicate no aggregate degradation in general-traffic metrics during the after period. The simulation-based ablation study further suggests that cross-cycle phase compensation contributes to maintaining general-traffic fairness when BSP is active.

The bus and general traffic performance metrics in Tables 7, 8 and 9 are corridor-level field test results exclusively brought by the Edge-HyAR-BSP deployed on Route 302, which directly reflect the independent optimization effect of our proposed BSP framework. However, the regional coordination indicators in Table 9 originate from the integrated traffic operation platform of Rongdong District covering 102 signalized intersections, which integrates multiple functional modules such as regional signal coordination, vehicle-road cooperative perception and the proposed Edge-HyAR-BSP. Accordingly, the overall network-wide performance improvement cannot be fully attributed solely to Edge-HyAR-BSP.

To quantitatively evaluate the effectiveness of the proposed optimization strategy, paired operational data of Route 302 before and after optimization were statistically compared. The baseline and optimized mean values, relative improvement rates, and statistical significance results obtained via paired-sample t-tests are summarized in Table 11.

Table 11: Statistical results of Route 302 bus indicators before and after optimization

Evaluation Metric	Baseline Mean	Optimized Mean	Relative Change	p-value
Travel duration (s)	651.32	538.36	-17.34%	< 0.001
Waiting time (s)	121.82	32.86	-73.03%	< 0.001
Average stop count	7.96	3.93	-50.67%	< 0.001
Time loss (s)	210.97	97.97	-53.56%	< 0.001
Average speed (m/s)	10.98	11.03	$+0.47\%$	0.634

As illustrated by the statistical results, significant improvements in core operational indicators are achieved by the proposed optimization strategy. Bus travel duration, passenger waiting time, stop frequency and operation time loss are substantially reduced with high statistical significance ($p < 0.001$). In particular, bus bunching and irregular dwelling problems are effectively mitigated by the drastic decreases in passenger waiting time and redundant stopping behaviors. In contrast, no statistically significant change is observed for the marginal fluctuation of average running speed, which reveals that delay elimination rather than driving speed enhancement is primarily targeted by the optimization scheme.

5.3 Edge Deployment Evaluation

5.3.1 Edge vs. Cloud Latency and Deadline Completion

Table 12 compares the end-to-end decision latency and task completion rate between the edge-deployed and cloud-deployed implementations of the proposed framework.

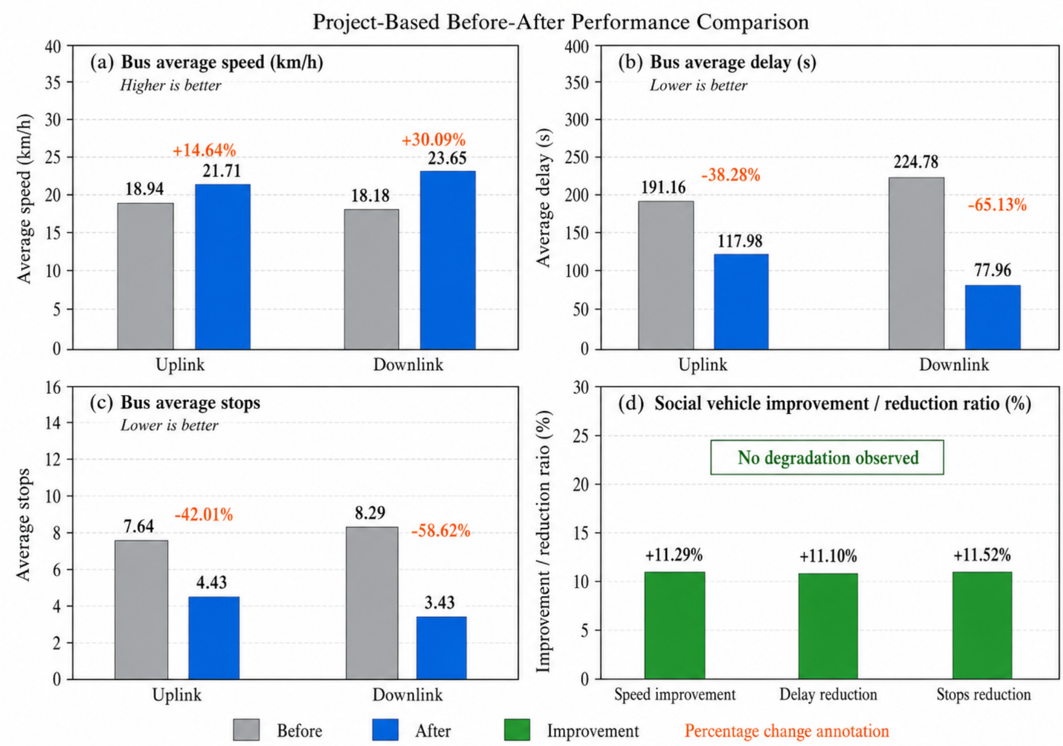

Figure 5: Project-based before-after performance comparison of bus and general traffic operations. (a) Average bus speed before and after optimization. (b) Average bus delay before and after optimization. (c) Average bus stops before and after optimization. (d) General traffic performance improvement and reduction ratios. Negative percentages in (b) and (c) denote reductions in bus delay and stops. The selected before–after records show improved bus indicators without observed aggregate deterioration in general-traffic indicators.

Table 12: Edge vs. cloud deployment latency comparison

Deployment	Comm. (ms)	Inference/service latency (ms)	E2E (ms)	Compl. (%)
Cloud-HyAR-BSP	45.2	32.1	89.7	94.3%
Edge-HyAR-BSP	8.3	12.5	**24.6**	**99.1%**

For cloud deployment, the reported inference/service latency includes cloud-side queueing and container scheduling overhead before model execution. E2E latency includes communication, preprocessing, queueing, model inference, action serialization, and controller-delivery overhead; therefore, it is not equal to the sum of communication and inference latency alone. Task completion rate is defined as the proportion of BSP decision tasks completed within a predefined system deadline of 100 ms, which is the response-time threshold adopted by the project platform for completing state reception, inference, action transmission, and controller acknowledgement. Due to project logging constraints, only average latency and deadline completion rate were available from the edge-node operation logs; tail-latency analysis (e.g., p95, p99) will be included in future platform-level evaluation. Edge deployment reduced end-to-end decision latency by 72.6% (from 89.7 ms to 24.6 ms) and improved task completion rate from 94.3% to 99.1%. Although the mean latency of both deployment modes remains below the nominal BSP decision window, edge deployment was associated with a higher observed deadline-completion rate in the available platform logs, particularly under communication degradation and workload fluctuations.

5.4 Simulation-Based Edge Robustness Analysis

The 100 ms platform deadline is a fixed unified threshold preset by the Xiong'an traffic operation platform, which requires the full pipeline of state collection, edge inference and action delivery to be finished within 100 ms. The task completion rate is calculated as the proportion of all BSP decision tasks that satisfy this fixed 100-ms time limit, which is measured based on real edge node operation logs in field tests. Differently, the variable control window $W_{\text{ctrl},i}$ is dynamically updated for each bus arrival event, jointly determined by the bus ETA, current signal phase sequence and minimum signal safety intervals, representing the actual available time margin to trigger effective BSP. The valid-control rate counts the share of BSP requests where the entire sensing-decision-execution pipeline is completed before the dynamically changing $W_{\text{ctrl},i}$ expires, which is only evaluated in simulation with adjustable communication and computing conditions. The valid BSP action rate further imposes signal safety, phase order and cycle compensation constraints on top of the valid-control condition, calculated as the percentage of triggered BSP requests that generate feasible, constraint-compliant priority actions and can be successfully executed by signal controllers.

Because the project field data do not provide controlled variations in delay, packet loss, or workload, we use simulation to examine Edge-HyAR-BSP under varying edge computing conditions. The simulation environment mirrors the Rongdong road network topology and traffic demand patterns, with communication delay, packet loss, and edge node workload being artificially varied. All simulation-based improvements are calculated relative to an independently simulated rule-based BSP baseline with fixed GE/RT thresholds, rather than the project-based before period. The traffic demand is calibrated using aggregated project data, but the reported simulation outcomes are generated independently. Each configuration is evaluated over five random seeds, and results are reported as mean values.

5.4.1 Communication Delay Sensitivity

Table 13 reports the performance under different V2X communication delay levels.

The method maintains considerable BSP benefits under moderate delays (≤50 ms), and performance degrades gradually as communication delay increases. At 200 ms communication delay, the controller still achieves a 51.33% reduction in bus delay relative to the simulated rule-based BSP baseline, although the valid-control rate decreases to 87.6%.

Table 13: Sensitivity to communication delay (simulation-based)

Comm. (ms)	Bus Delay Change	Social Delay Change	Valid Ctl. (%)
10	-64.85%	-10.92%	99.5%
50	-63.21%	-10.14%	98.2%
100	-59.47%	-8.76%	95.1%
200	-51.33%	-5.42%	87.6%

Results obtained from SUMO co-simulation with Python-based edge emulation. "Valid Ctl." denotes the proportion of BSP requests for which the complete sensing–decision–execution process finishes before the traffic-control execution window expires. This metric differs from the platform deadline completion rate (100 ms) reported in Table 10; under simulated communication degradation, the effective control window varies with delay conditions.

5.4.2 Packet Loss Robustness

Table 14 examines robustness under packet loss (state-missing) scenarios.

Table 14: Robustness under packet loss (simulation-based)

Loss Rate	Bus Delay Change	Spillback Rate	Valid BSP Action (%)
0%	-64.88%	11.3%	99.1%
5%	-58.74%	14.2%	96.8%
10%	-49.85%	18.7%	92.4%
20%	-35.21%	26.5%	81.3%

Valid BSP action rate is defined as the proportion of triggered BSP requests for which a feasible priority action is generated and executed without violating phase-safety constraints. The method maintains effective control up to 10% packet loss, with valid BSP action rate remaining above 90%. At 20% loss, performance degrades notably, suggesting that a dedicated fallback controller, such as local fixed-time control, may be needed under severe state-missing conditions.

5.4.3 Edge Node Workload Impact

Table 15 shows performance at varying edge node workload levels.

Table 15: Impact of edge node workload (simulation-based)

Workload	Inference latency (ms)	Compl. (%)	Bus Delay Change
Low (30%)	8.7	99.5%	-65.02%
Medium (60%)	12.5	99.1%	-64.95%
High (85%)	28.3	94.7%	-58.67%
Overload (95%)	67.1	76.2%	-32.45%

Edge-HyAR-BSP maintains relatively stable traffic performance under low-to-high workload conditions. As workload increases, longer inference latency and lower task completion rates reduce the timeliness of BSP action delivery. The model uses workload as contextual information but does not directly allocate computing resources. Under overload conditions, task completion rate drops sharply, indicating that edge resource provisioning must account for peak-hour computation demands.

5.5 Simulation-Based Ablation Study

To quantify the contribution of each method component, we conduct simulation-based ablation experiments as shown in Table 16.

Table 16: Simulation-based ablation study results

Variant	Bus Delay Change	Social Delay Change
Full Edge-HyAR-BSP	**-64.90%**	-11.10%
w/o CVAE representation	-48.21%	-7.83%
w/o GAT coordination	-39.45%	-2.16%
w/o Phase compensation	-58.76%	+4.32%
w/o Edge state	-62.15%	-10.05%
w/o Edge-execution penalties	-63.88%	-10.47%
Purely discrete actions	-42.30%	-5.61%

The ablation results show several clear trends: (1) GAT-based multi-intersection coordination produces the largest reduction in bus-delay improvement, decreasing the benefit by 25.45 percentage points (from 64.90% to 39.45%) and nearly eliminating general-traffic delay improvement; (2) Replacing hybrid actions with purely discrete actions causes the second-largest reduction of 22.60 percentage points (from 64.90% to 42.30%), indicating the importance of continuous duration modeling for jointly handling phase selection and green-time adjustment; (3) Phase compensation is critical for fairness—without it, general traffic delay increases by 4.32%; (4) Edge state and edge latency penalty components provide moderate improvements in adaptation to edge conditions, although their individual traffic-only contributions are smaller than the coordination and action-space components.

6 Discussion

6.1 Edge-Computing Perspective

From an edge-computing perspective, the main contribution of Edge-HyAR-BSP is not limited to deploying an RL model at roadside nodes. The framework couples edge-side inference with traffic-control decision-making by incorporating communication delay, inference latency, workload, bandwidth, and packet loss into the control state. This enables the BSP policy to adapt to degraded edge conditions while preserving the timing constraints of signal control. Compared with existing vehicular edge computing studies, which mainly focus on task offloading, service scheduling, and resource orchestration, Edge-HyAR-BSP investigates how edge-side operating conditions affect real-time signal-control decisions. This difference shifts the focus from resource management to edge-aware control policy learning. Therefore, the proposed method can be viewed as an edge-intelligent closed-loop control framework for vehicular networks. General-purpose edge resource scheduling, computation offloading, and bandwidth allocation are complementary to this work. The focus here is edge-aware control policy learning, where edge-side operating conditions are observed and used to improve the feasibility of latency-sensitive BSP decisions.

6.2 Key Findings

The results highlight several practical points. First, the ablation tests suggest that treating green duration as a continuous decision is not a minor refinement, but a key element of BSP control. Coordination is also important: once GAT-based neighbor aggregation is removed, the benefit to general-traffic delay nearly vanishes. Edge deployment does not change the traffic logic itself, but it provides the controller with substantially more timing margin under short decision windows. The fairness results further indicate that cross-cycle phase compensation helps contain the negative impact of priority actions on other traffic. It is worth noting that GAT coordination and hybrid actions produce the largest traffic-efficiency gains, whereas edge-state awareness mainly improves deadline feasibility and robustness under degraded conditions rather than substantially increasing normal-condition traffic performance. This is consistent with the role of edge variables as execution-context modifiers rather than direct traffic optimizers.

6.3 Comparison with Existing Work

To place this work in context, we compare against three closely related lines of research: (1) Bouktif *et al*. [12] demonstrated hybrid actions for a single intersection—we extend hybrid-action BSP to a multi-agent setting, evaluate detailed operational BSP performance on a 15-intersection corridor, and conduct simulation-based robustness and ablation experiments on a topology derived from the 102-intersection project network; (2) CoLight [11] achieved GAT-based multi-intersection coordination but used only discrete actions without BSP—we integrate hybrid actions, BSP strategies, and edge awareness; (3) D^2TSP [13] applied DRL for BSP but was limited to isolated intersections with discrete actions—we achieve coordinated multi-intersection BSP with hybrid actions.

6.4 Limitations

Several limitations should be noted. One issue is that the regional coordination indicators were generated by the integrated project platform rather than by an isolated Edge-HyAR-BSP module; therefore the regional improvement cannot be attributed entirely to the proposed control framework. The before-after comparison on Route 302 covers a single bus corridor during specific time periods, and extending the evaluation to additional routes, seasonal conditions, and demand patterns would strengthen the evidence. The project data are available only as intersection-level aggregated statistics; individual vehicle trajectories were not retained in the platform records, which precludes formal paired significance testing at the per-intersection level. The field evaluation reflects normal operating conditions, whereas the controlled communication delay, packet-loss, and workload experiments are simulation-based. Correlated communication failures, complete edge-node outages, and cyber-security threats are not modeled and remain open questions for deployment at scale.

6.5 Project-Based Evidence and Validity Threats

The experimental data used in this paper are derived from a government-supported science and technology project in Xiong'an New Area, which provides a realistic application background and large-scale operational data. This strengthens the practical relevance of the proposed framework compared with purely simulation-based studies. However, the project-based nature of the data also introduces validity threats. First, the before-after comparison is not a randomized controlled experiment; it compares operational periods before and after system optimization. Second, the project-based operational data are available only as aggregated period-level indicators; exact sample counts (e.g., total workdays, bus trips, and BSP requests per period) are not retained in the platform-level records used for this paper, which precludes formal statistical power analysis and limits the granularity of the before-after comparison. Third, the edge computing robustness experiments (Section 5.4) are conducted in simulation, as project field data do not provide controlled variation of communication delay, packet loss, or edge workload. Therefore, the reported results should be interpreted as project-based case-study evidence supporting practical feasibility, rather than universally generalizable causal conclusions.

6.6 Future Work

A natural next step is extending the framework to emergency vehicles, autonomous buses, and multi-modal public transport, where similar latency-sensitive priority logic applies. Cross-region and cross-period validation with a proper randomized or quasi-experimental design would strengthen the evidence beyond the current single-corridor case study. On the algorithmic side, visualizing GAT attention weights could improve the interpretability of inter-intersection coordination. Safe switching between Edge-HyAR-BSP and local rule-based or fixed-time fallback control under severe edge-node failures or communication outages is an important engineering problem—the present implementation does not include an explicit fallback controller. Federated learning across multiple edge domains offers a potential path toward privacy-preserving model updates, which is particularly relevant for systems operating under urban data governance constraints.

7 Conclusion

This paper investigated latency-sensitive cooperative BSP in dense short-block road networks from an edge-executable control perspective. The proposed framework couples phase-duration priority decisions with neighbor-state aggregation, deadline-aware feasibility projection, and cross-cycle compensation. The project-derived Route 302 evaluation indicates that the framework can improve aggregate bus operating indicators while avoiding observed aggregate deterioration in general-traffic performance during the selected periods. The edge deployment logs further show that roadside inference provides a larger timing margin for BSP execution than cloud-centric deployment. Simulation results under controlled communication delay, packet loss, and workload variations suggest that the framework degrades gradually under moderate edge-side disturbance, although severe packet loss or edge overload still requires fallback control. These findings support the feasibility of treating edge-side operating conditions as part of the control context in latency-sensitive cooperative BSP. Future work should validate the framework across more routes, longer periods, and different cities, and should integrate explicit fault-tolerant switching mechanisms.

Funding

This work was supported by the Xiong'an New Area Science and Technology Innovation Project "Research on Multi-Granularity Traffic System Simulation and Collaborative Control Technology for the Narrow-Road Dense-Network in the Xiong'an New Area" under Grant No. 2022XAGG0126.

Author Contributions

Conceptualization, H.D. and X.W.; methodology, H.D., X.W., and Z.W.; software, H.D. and Z.T.; validation, H.D., Z.W., Z.T., and W.D.; formal analysis, H.D. and X.W.; data curation, H.D. and Z.T.; writing—original draft preparation, H.D.; writing—review and editing, H.D., X.W., Z.W., and Z.T.; supervision, X.W.; project administration, H.D. and X.W. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

All the authors declare that they have no conflict of interest.

Declaration of AI-Assisted Technology Use

During manuscript preparation, the authors used an AI-assisted language tool for linguistic polishing and structural feedback. All scientific content, data, analyses, references, and conclusions were independently reviewed and verified by the authors, who take full responsibility for the manuscript.

Data Availability

The data used in this study were collected from the Xiong'an New Area Science and Technology Innovation Project (Grant No. 2022XAGG0126). Due to project management requirements, data privacy, and restrictions on urban traffic operation data, raw trajectory-level and device-level records cannot be publicly released. Aggregated and anonymized data supporting the findings of this study may be available from the corresponding authors upon reasonable request and subject to approval by the project data management authority.

Acknowledgements

The authors thank the project partners of the Xiong'an New Area Science and Technology Innovation Project (Grant No. 2022XAGG0126) for supporting data collection, platform testing, and the demonstration environment.

References

[1] Lin, Y., Yang, X., Zou, N., Franz, M.: Transit signal priority control at signalized intersections: A comprehensive review. Transportation Letters, **7**(3), 168–180 (2015). https://doi.org/10.1179/1942787514Y.0000000044

[2] Skabardonis, A., Geroliminis, N.: Real-time monitoring and control on signalized arterials. Journal of Intelligent Transportation Systems, **12**(2), 64–74 (2008). https://doi.org/10.1080/15472450802023337

[3] Wang, Z., Gupta, R., Han, K., Wang, H., Ganlath, A., Ammar, N.: Mobility digital twin: Concept, architecture, case study, and future challenges. IEEE Internet of Things Journal, **9**(18), 17452–17467 (2022). https://doi.org/10.1109/JIOT.2022.3156028

[4] Sun, Y., Lei, B., Liu, J., Huang, H., Zhang, X., Peng, J.: Computing power network: A survey. China Communications, **21**(9), 109–145 (2024). https://doi.org/10.23919/JCC.ja.2021-0776

[5] Tang, X., Cao, C., Wang, Y., Zhang, S., Liu, Y., Li, M.: Computing power network: The architecture of convergence of computing and networking towards 6G requirement. China Communications, **18**(2), 175–185 (2021). https://doi.org/10.23919/JCC.2021.02.011

[6] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., Zhang, J.: Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, **107**(8), 1738–1762 (2019). https://doi.org/10.1109/JPROC.2019.2918951

[7] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., Zomaya, A.Y.: Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, **7**(8), 7457–7469 (2020). https://doi.org/10.1109/JIOT.2020.2984887

[8] Liu, L., Chen, C., Pei, Q., Maharjan, S., Zhang, Y.: Vehicular edge computing and networking: A survey. Mobile Networks and Applications, **26**(3), 1145–1168 (2021). https://doi.org/10.1007/s11036-020-01624-1

[9] Wei, H., Zheng, G., Yao, H., Li, Z.: IntelliLight: A reinforcement learning approach for intelligent traffic light control. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2496–2505 (2018). https://doi.org/10.1145/3219819.3220096

[10] Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K.: PressLight: Learning Max Pressure control to coordinate traffic signals in arterial network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1290–1298 (2019). https://doi.org/10.1145/3292500.3330949

[11] Wei, H., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., Zhang, W., Zhu, Y., Xu, K., Li, Z.: CoLight: Learning network-level cooperation for traffic signal control. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), pp. 1913–1922 (2019). https://doi.org/10.1145/3357384.3357902

[12] Bouktif, S., Cheniki, A., Ouni, A.: Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning. Sensors, **21**(7), 2302 (2021). https://doi.org/10.3390/s21072302

[13] Hu, W.X., Ishihara, H., Chen, C.: Deep reinforcement learning two-way transit signal priority algorithm for optimizing headway adherence and speed. IEEE Transactions on Intelligent Transportation Systems, **24**(8), 7920–7931 (2023). https://doi.org/10.1109/TITS.2023.3266461

[14] Dong, Y., Huang, H., Zhang, G., Jin, J.: Adaptive transit signal priority control for traffic safety and efficiency optimization: A multi-objective deep reinforcement learning framework. Mathematics, **12**(24), 3994 (2024). https://doi.org/10.3390/math12243994

[15] Luo, Q., Hu, S., Li, C., Li, G., Shi, W.: Resource scheduling in edge computing: A survey. IEEE Communications Surveys & Tutorials **23**(4), 2131–2165 (2021). https://doi.org/10.1109/COMST.2021.3106401

[16] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials **19**(3), 1657–1681 (2017). https://doi.org/10.1109/COMST.2017.2705720

[17] Wang, X., Deng, H., Qiu, C., Chen, Z., Luo, T., Ming, Z.: Computing power networks for unmanned aerial vehicles: A hierarchical resources trading market. Digital Communications and Networks **12**(4), 584–593 (2026). https://doi.org/10.1016/j.dcan.2024.12.002

[18] ETSI: Multi-access edge computing (MEC): Framework and reference architecture. ETSI GS MEC 003, V4.1.1 (2025). https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/04.01.01_60/gs_MEC003v040101p.pdf Accessed 30 June 2026.

[19] Gu, H., Zhao, L., Han, Z., Zheng, G., Song, S.: AI-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions. IEEE Communications Surveys & Tutorials, **26**(2), 1322–1385 (2024). https://doi.org/10.1109/COMST.2023.3338153

[20] Ge, C., Qin, S.: Digital twin intelligent transportation system (DT-ITS): A systematic review. IET Intelligent Transport Systems **18**(12), 2325–2358 (2024). https://doi.org/10.1049/itr2.12539

[21] Thodi, B.T., Chilukuri, B.R., Vanajakshi, L.: An analytical approach to real-time bus signal priority system for isolated intersections. Journal of Intelligent Transportation Systems **25**(1), 145–167 (2021). https://doi.org/10.1080/15472450.2020.1797504

[22] Girijan, A., Vanajakshi, L., Chilukuri, B.R.: Dynamic Thresholds Identification for Green Extension and Red Truncation Strategies for Bus Priority. IEEE Access, **9**, 64291–64305 (2021). https://doi.org/10.1109/ACCESS.2021.3074361

[23] Liu, H., Teng, K., Rai, L., Xing, J.: Transit Signal Priority Controlling Method Considering Non-Transit Traffic Benefits and Coordinated Phase States for Multi-Rings Timing Plan at Isolated Intersections. IEEE Transactions on Intelligent Transportation Systems, **22**(2), 913–936 (2020). https://doi.org/10.1109/TITS.2019.2961420

[24] Xu, M., Chai, J., Yan, Y., Qu, X.: Multi-agent fuzzy-based transit signal priority control for traffic network considering conflicting priority requests. IEEE Transactions on Intelligent Transportation Systems, **23**(2), 1554–1564 (2021). https://doi.org/10.1109/TITS.2020.3045122

[25] Zeng, X., Zhang, Y., Jiao, J., Yin, K.: Route-based transit signal priority using connected vehicle technology to promote bus schedule adherence. IEEE Transactions on Intelligent Transportation Systems, **22**(2), 1174–1184 (2021). https://doi.org/10.1109/TITS.2020.2963839

[26] Abdelhalim, A., Abbas, M.: A value proposition of cooperative bus-holding transit signal priority strategy in connected and automated vehicles environment. IEEE Transactions on Intelligent Transportation Systems, **23**(7), 9301–9306 (2021). https://doi.org/10.1109/TITS.2021.3086776

[27] Cvijovic, Z., Zlatkovic, M., Stevanovic, A., Song, Y.: Conditional transit signal priority for connected transit vehicles. Transportation Research Record: Journal of the Transportation Research Board **2676**(2), 490–503 (2021). https://doi.org/10.1177/03611981211044459

[28] Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., Li, Y., Xu, K., Li, Z.: Learning phase competition for traffic signal control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1936–1972 (2019). https://doi.org/10.1145/3357384.3357900

[29] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z.S.H., Bazzan, A.L.C., Far, B.: Reinforcement learning in urban network traffic signal control: A systematic literature review. Expert Systems with Applications, **199**, 116830 (2022). https://doi.org/10.1016/j.eswa.2022.116830

[30] Chu, T., Wang, J., Codeca, L., Li, Z.: Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems, **21**(3), 1086–1096 (2020). https://doi.org/10.1109/TITS.2019.2901791

[31] Wang, P., Deng, H., Zhang, J., Wang, L., Zhang, M., Li, Y.: Model predictive control for connected vehicle platoon under switching communication topology. IEEE Transactions on Intelligent Transportation Systems, **23**(7), 7817–7830 (2022). https://doi.org/10.1109/TITS.2021.3073012

[32] Li, B., Tang, H., Zheng, Y., Hao, J., Li, P., Meng, Z., Wang, L.: HyAR: Addressing discrete-continuous action reinforcement learning via hybrid action representation. In International Conference on Learning Representations(ICLR), pp. 1–19 (2022). https://doi.org/10.48550/arXiv.2109.05490

[33] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 1861–1870 (2018). https://doi.org/10.48550/arXiv.1801.01290

[34] Tang, X., Cao, C., Wang, Y., Zhang, S., Liu, Y., Li, M., He, T.: Computing power network: The architecture of convergence of computing and networking towards 6G requirement. China Communications **18** (2), 175–185 (2021). https://doi.org/10.23919/JCC.2021.02.011

[35] Yu, W., Ngoduy, D., Hua, X., Wang, W.: On the stability of a heterogeneous platoon-based traffic system with multiple anticipations in the presence of connected and automated vehicles. Transportation Research Part C: Emerging Technologies **157**, 104389 (2023). https://doi.org/10.1016/j.trc.2023.104389

[36] Liu, M., Zhu, M., Yao, M., Li, P., Tang, R., Deng, H.: Cooperative lane-changing control method for autonomous vehicles based on dynamic trajectory planning. Journal of Transportation Systems Engineering and Information Technology **24**(5), 65–78 (2024). https://doi.org/10.16097/j.cnki.1009-6744.2024.05.007