Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(8): 5546–5579.
DOI: 10.3934/era.2026248
Received: 11 April 2026
Revised: 29 May 2026
Accepted: 05 June 2026
Published: 30 June 2026

Research article

Intelligent perception communication protocol for satellite constellations via multi-agent deep reinforcement learning with dual-attention and delay-aware coordination

Pan Peng, Wei Tao, Hao-Nan Wang and Hui Zhao*

School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China

* **Correspondence:** Email: huizhao@sjtu.edu.cn.

Abstract: Large-scale low earth orbit (LEO) satellite constellations face three fundamental communication challenges: severe channel dynamics caused by orbital velocities exceeding 7.5 km s^{-1}, resource scarcity under heterogeneous quality-of-service demands, and strategy incoordination induced by stochastic multi-hop inter-satellite delays. This paper proposes the intelligent perception communication protocol (IPCP), a unified three-layer perception–decision–execution architecture driven by multi-agent deep reinforcement learning (MARL). We develop the information-exchange deep Q-Network (IEDQN), in which each satellite compresses its local observation–Q-value history into a low-dimensional structured token via a shared Info-Block, aggregates global context through a receiver-specific coordination network (Mess-Net), and selects actions via a shared Q-Net on near-fully-observable augmented states. We prove that IEDQN converges almost surely to an ϵ-approximate Nash equilibrium, with the gap bounded in closed form by the Lipschitz constant of the coordination network. To handle multi-hop delay, we propose the dual-attention delay-aware communication network (DADC), integrating a Transformer-based delay-aware information integration module (DAII) with a dual-attention filtering module (DAIF) that combines Gumbel-softmax hard attention and scaled dot-product soft attention. A formal delay-compensation bound shows that the performance gap to the oracle zero-delay policy decreases exponentially with reception window T_r. Experiments on an NS3-based LEO simulator (5, 7, and 10 agents) show that DADC consistently outperforms the best prior delay-aware method (DACOM) and all delay-agnostic baselines across constellation scales, with advantages that widen as the number of agents increases, while the full IPCP stack doubles throughput (18.4→36.8 Mbps) and reduces end-to-end latency by 39.9% over a static protocol baseline.

Keywords: satellite constellation; low earth orbit; deep Q-Network; transformer

1. Introduction

The rapid expansion of satellite constellation systems, exemplified by SpaceX Starlink, Amazon Kuiper, and China's Guowang mega-constellations, has fundamentally transformed global broadband communications, Earth observation, and deep-space exploration [1, 2]. Low earth orbit (LEO) constellations, operating at 200–2000 km altitude, offer low propagation latency (10–40 ms round-trip), wide coverage, and the spatial diversity needed for resilient space-ground integrated networks [3, 4], positioning them as a cornerstone of the forthcoming 6G non-terrestrial network (NTN) paradigm [5, 6].

Despite these merits, the communication protocol layer of large-scale LEO constellations faces three fundamental challenges that existing approaches have not simultaneously resolved. First, severe channel dynamics and Doppler uncertainty. Orbital velocities exceeding 7.5 km s^{-1} produce Doppler offsets above 50 kHz for Ka-band links, and rapid handovers every 60–90 s render static routing protocols ineffective [7, 8]. Classical AMC schemes rely on channel-state feedback whose latency is comparable to the channel coherence time, creating a fundamental mismatch [9]. Second, resource scarcity under heterogeneous QoS demands. A single constellation must concurrently serve latency-critical telemetry, high-throughput Earth-observation imagery, and delay-tolerant messaging [10, 11]. Fixed-schedule TDMA protocols achieve bandwidth utilization below 40%, indicating severe resource under-utilization [12]. Third, multi-hop communication delay in distributed coordination. When inter-satellite separation exceeds the single-hop communication radius, messages traverse multiple relay hops, introducing stochastic delays of one to several control cycles [13]. Existing MARL communication protocols, CommNet [14], IC3Net [15], and MAGIC [16], assume ideal zero-delay links, causing strategy incoordination and substantial performance loss under realistic delay conditions, as demonstrated in Section 7.

Prior work addresses the above challenges only in isolation. Single-agent methods [17, 18] cannot capture cooperative dynamics; MADDPG [19] and QMIX [20] handle cooperative settings but lack delay awareness or require centralized execution, both of which are infeasible on resource-constrained onboard processors [21]. Graph-attention methods [16, 22] are sensitive to message ordering and suffer performance collapse under delay. Transformer-based protocols [23] capture temporal dependencies but have not been studied in delay-aware multi-hop satellite settings. As detailed in Section 2, no existing method simultaneously addresses (i) real-time channel-dynamics adaptation, (ii) multi-objective resource optimization, and (iii) robust cooperative decision-making under stochastic multi-hop delay, the gap this paper targets.

This paper proposes the intelligent perception communication protocol (IPCP), a unified MARL-driven framework for satellite constellation communication.

(1) A three-layer perception–decision–execution stack integrates spectrum sensing, DQN-based control, and physical-layer execution into a closed-loop framework. On an NS3-based Walker constellation simulator (550 km, $N = 12$), IPCP doubles throughput (18.4→36.8 Mbps), reduces end-to-end latency by 39.9% (187.3→112.6 ms), cuts packet loss by 75.1%, and improves bandwidth utilization from 38.2% to 76.3% over a static protocol baseline.

(2) IEDQN comprises a shared Info-Block that compresses each agent's local history into a low-dimensional token, a receiver-specific Mess-Net that learns influence weights across the constellation, and a shared Q-Net operating on near-fully-observable augmented states, with proven convergence to an ϵ-Nash equilibrium under mild Lipschitz conditions.

(3) DADC integrates a Transformer-based delay-aware information integration network (DAII) with a delay-aware information integration network (DAIF), combining hard attention (Gumbel-softmax gating) and soft attention (scaled dot-product weighting) to treat delay as an exploitable signal, outperforming all baselines across constellation scales of 5, 7, and 10 agents under $T_{\text{delay}} = 5$.
(4) We establish a convergence characterization for IEDQN under non-stationary multi-agent dynamics assuming linear function approximation for analytical tractability, derive a delay-compensation bound showing the performance gap decays exponentially with reception window T_r, and show both algorithms incur $O(N^2)$ communication overhead, identical to naive broadcasting, while demonstrating strong coordination performance across all evaluated experimental settings.

2. Related works

2.1. Satellite communication protocols and resource management

Early LEO protocol designs relied on static scheduling and pre-computed routing tables [1, 2], which are ill-suited to dynamic ISL topologies. SDN-based controllers push routing updates in response to topology changes [24], but their centralized architecture introduces a single point of failure and non-negligible uplink latency [2]. At the physical layer, AMC schemes track channel quality on ISLs [7, 9], yet CSI feedback latency often renders the feedback stale before it can be acted upon. Recent work integrates machine learning into specific protocol components, task-oriented source coding [10] and RIS-assisted DRL scheduling [12], but neither provides a unified, closed-loop intelligent protocol stack. At the single-agent level, DQN-based methods achieve strong performance in dynamic spectrum access and SINR optimization [17, 18], while DRL has been applied to satellite task scheduling [25] and LEO edge-computing offloading [12]; however, single-agent approaches cannot coordinate across distributed spacecraft, motivating the multi-agent formulation adopted in this paper. Complementary advances in LEO routing include topology-aware designs [26, 27] and Pareto-optimal MARL frameworks for large-scale packet routing [28–30].

2.2. Multi-agent communication and delay-aware coordination

In cooperative MARL, agents share information to maximize a joint reward [21]. CommNet [14] learns differentiable communication channels end-to-end; TarMAC [31] adds signature-based targeted messaging; IC3Net [15] introduces learned gating for selective broadcasts. MAGIC [16] and G2ANet [32] construct dynamic directed communication graphs via multi-head graph attention. Value-decomposition methods, VDN [33], QMIX [20], and QTRAN [34], factor the joint Q-function under monotonicity constraints for centralized training with decentralized execution, but provide no explicit communication channel. Parameter sharing [35] ties weights across agents to improve scalability and convergence speed.

A critical weakness of attention-graph methods is sensitivity to message ordering: under communication delay, out-of-order arrivals corrupt the learned graph structure and cause performance collapse [13]. Communication delay in distributed MARL has received limited attention: [36] show that standard algorithms degrade significantly under fixed delays, and [37] apply delay-compensated reward shaping to multi-agent microgrid control. Most closely related to our work, DACOM [13] maintains a message buffer with GRU-based fusion to compensate for latency. Our DADC advances beyond

DACOM in three respects: (i) delay is modeled as a distance-dependent stochastic process; (ii) a Transformer encoder [38] replaces the GRU for permutation-invariant fusion of arbitrary-order delayed messages; (iii) dual-attention filtering combines Gumbel-softmax hard attention with scaled dot-product soft attention, providing finer-grained relevance assessment than DACOM's single-level attention.

Table 1 positions IPCP against representative prior works. No existing method simultaneously satisfies all six desiderata listed in the column headers; IPCP addresses all six, motivating the unified framework proposed in this paper. Table 2 further decomposes each module into original contributions and adapted components.

Table 1. Comparison of related works and the proposed framework.

Method	Sat.	MA-RL	Comm.	Delay	Conv.	Multi-Obj.
CommNet [14]	✗	✓	✓	✗	✗	✗
IC3Net [15]	✗	✓	✓	✗	✗	✗
MAGIC [16]	✗	✓	✓	✗	✗	✗
QMIX [20]	✗	✓	✗	✗	✓	✗
DACOM [13]	✗	✓	✓	✓	✗	✗
DRL-Offload [12]	✓	✗	✗	✗	✗	✓
MA-LEO [11]	✓	✓	✗	✗	✗	✓
MIRL-Route [9]	✗	✓	✓	✗	✗	✗
IPCP (Ours)	✓	✓	✓	✓	✓	✓

✓ = supported; ✗ = not supported.

Table 2. Novel contributions vs. adapted components in IPCP.

Component	Inspired by	Novel contribution
Info-Block	CommNet, parameter sharing	Q-value history token; one-slot delay decoupling
Mess-Net	TarMAC	Receiver-specific aggregation for asymmetric constellation topology
DAII	DACOM (GRU), Transformer	Permutation-invariant fusion of arbitrarily-ordered delayed ISL messages
DAIF	MAGIC (hard attn.)	Cascaded dual-attention on delay-compensated representations; exponential decay bound
IPCP stack	—	Unified perception–decision–execution protocol for LEO constellations

3. System model and problem formulation

3.1. Satellite constellation network model

We consider a LEO satellite constellation system consisting of N satellite nodes, denoted by the set $\mathcal{N} = \{1, 2, \ldots, N\}$. Each satellite is equipped with a software-defined radio (SDR) transceiver, an onboard processor, and a finite-capacity energy storage unit. The constellation operates in a shared frequency band $\mathcal{F}$ divided into K orthogonal sub-channels, and time is slotted with slot duration Δt.

3.1.1. Topology and link model

At time slot t, the network topology is represented by a directed graph $\mathcal{G}^t = (\mathcal{N}, \mathcal{E}^t)$, where the edge set $\mathcal{E}^t$ evolves due to orbital motion. A link $(i, j) \in \mathcal{E}^t$ exists if and only if satellite j lies within the

communication range of satellite i at time t. The received signal power on link (i, j) follows a free-space path loss model augmented by a small-scale Rician fading component

$$P_r^{ij}(t) = P_t^i(t) \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi d_{ij}(t)}\right)^2 \cdot \left|h_{ij}(t)\right|^2, \tag{3.1}$$

where $P_t^i(t)$ is the transmit power of satellite i, G_t and G_r are the transmit and receive antenna gains respectively, λ is the carrier wavelength, $d_{ij}(t)$ is the inter-satellite distance computed from orbital state vectors, and $h_{ij}(t) \sim \mathcal{CN}(0, 1)$ models Rician fading with K-factor κ.

The instantaneous signal-to-interference-plus-noise ratio (SINR) on link (i, j) is given by

$$\gamma_{ij}(t) = \frac{P_r^{ij}(t)}{\sigma^2 + \sum_{k \neq i,(k,j) \in \mathcal{E}^t} P_r^{kj}(t)}, \tag{3.2}$$

where σ^2 is the thermal noise power. The achievable rate on link (i, j) under modulation-and-coding scheme $m_{ij} \in \mathcal{M}$ is

$$R_{ij}(t) = B \cdot \log_2\Big(1 + \gamma_{ij}(t)\Big) \cdot \eta_{m_{ij}}, \tag{3.3}$$

where B is the sub-channel bandwidth and $\eta_{m_{ij}}$ is the spectral efficiency factor of the selected MCS.

3.1.2. Energy model

The energy consumed by satellite i in slot t is

$$E_i(t) = P_t^i(t) \cdot \Delta t + P_{\text{proc}} \cdot \Delta t + P_{\text{sen}} \cdot \Delta t, \tag{3.4}$$

where P_{proc} and P_{sen} denote the fixed processing and sensing power consumption, respectively. The residual energy satisfies $\mathcal{B}_i(t + 1) = \mathcal{B}_i(t) - E_i(t) + H_i(t)$, where $H_i(t)$ is the solar harvesting energy in slot t.

3.1.3. Communication delay model

Due to multi-hop routing, information transmitted from satellite k to satellite i incurs a stochastic delay $\tau_{i,k}^t$. We model this delay as a distance-dependent random variable:

$$\tau_{i,k}^t = \begin{cases} 0, & d_{ik}(t) \leq r_c, \\ \sim \text{Uniform}\{1, \ldots, T_{\text{delay}}\}, & d_{ik}(t) > r_c, \end{cases} \tag{3.5}$$

where r_c is the single-hop communication range and T_{delay} is the maximum delay in slots, capturing the variable number of relay hops required for multi-hop inter-satellite routing. Satellites whose separation $d_{ik}(t) \leq r_c$ communicate with zero delay (real-time neighbors $\mathcal{C}_i^t$); all others form the delayed neighbor set $\mathcal{D}_i^t$, experiencing a stochastic delay uniformly distributed over $\{1, \ldots, T_{\text{delay}}\}$ slots.

Remark 3.1 (Justification of the uniform delay model). The uniform distribution in (3.5) is adopted for three reasons: (i) hop counts are approximately uniformly distributed over $\{1, \ldots, T_{\text{delay}}\}$ in a Walker Delta constellation [2]; (ii) it is the maximum-entropy assumption over the support, making DADC robust to more concentrated distributions; (iii) it is consistent with DACOM [13], ensuring fair comparison with the primary baseline. Incorporating topology-aware delay traces is left as future work.

3.2. Multi-agent MDP formulation

The joint communication optimization problem is formulated as a decentralized partially-observable Markov Decision Process (Dec-POMDP), defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{O}^i\}, \{\mathcal{A}^i\}, \mathcal{T}, \mathcal{R}, \gamma \rangle$.

3.2.1. State space

The global environment state at time t is

$$s^t = \left\{ \gamma_{ij}^t,\ \mathcal{B}_i^t,\ Q_i^t,\ \phi_i^t,\ \mathcal{G}^t \right\}_{i,j \in \mathcal{N}}, \tag{3.6}$$

where Q_i^t is the buffer queue length and $\phi_i^t \in \{\text{normal, degraded, critical}\}$ is the operational status of satellite i.

3.2.2. Observation space

Each satellite i can only access its *local* observation

$$o_i^t = \left\{ \gamma_{ij}^t,\ \mathcal{B}_i^t,\ Q_i^t,\ \phi_i^t,\ \mathrm{BER}_i^t,\ \mathrm{SNR}_i^t \right\}, \tag{3.7}$$

which constitutes a partial projection of s^t restricted to information locally measurable onboard satellite i.

3.2.3. Action space

The joint action of agent i at time t is a tuple

$$a_i^t = (f_i^t,\ m_i^t,\ P_i^t,\ \tau_i^t) \in \mathcal{F} \times \mathcal{M} \times \mathcal{P} \times \mathcal{T}_{\text{slot}}, \tag{3.8}$$

where f_i^t is the selected sub-channel, m_i^t is the MCS index, $P_i^t \in \mathcal{P}$ is the discrete transmit power level, and τ_i^t is the allocated time-slot index.

3.2.4. Reward function

To balance multiple conflicting objectives, we define a composite reward function for agent i as

$$r_i^t = w_1 \cdot R_{\text{thr}}^i(t) - w_2 \cdot D_{\text{e2e}}^i(t) - w_3 \cdot E_i(t) + w_4 \cdot \mathbb{1}[\text{no collision}_i^t], \tag{3.9}$$

where $R_{\text{thr}}^i(t)$ is the instantaneous throughput, $D_{\text{e2e}}^i(t)$ is the end-to-end delay, $E_i(t)$ is the energy consumption, and $\mathbb{1}[\cdot]$ is the collision-free indicator. The weighting coefficients $\{w_k\}_{k=1}^4$ are optimized offline via particle swarm optimization (PSO) under a Pareto-dominance criterion to ensure balanced multi-objective performance. The global team reward is: $R^t = \frac{1}{N} \sum_{i \in \mathcal{N}} r_i^t$.

Each objective term is normalized prior to weighting: R_{thr}^i by peak link capacity $R_{\max}$, D_{e2e}^i by maximum tolerable delay $D_{\max}$, and E_i by initial battery capacity B_0. The collision indicator is binary and needs no normalization.

PSO uses 30 particles, 100 iterations, inertia $\omega = 0.7$, and coefficients $c_1 = c_2 = 1.5$, yielding the selected weight vector $\mathbf{w}^* = (0.4, 0.3, 0.2, 0.1)$.

3.2.5. Optimization objective

Each agent seeks to find a policy π_i^* that maximizes the expected discounted cumulative reward:

$$\pi_i^* = \arg\max_{\pi_i} \mathbb{E}_\pi\left[\sum_{t=0}^{T} \gamma^t R^t\right], \tag{3.10}$$

where $\gamma \in (0, 1)$ is the discount factor and T is the episode horizon. Because the global state s^t is not fully observable by any individual agent, this optimization is inherently a Dec-POMDP, which is NEXP-hard in general [20]. The proposed IEDQN and DADC algorithms provide practical approximate solutions to this intractable problem.

4. Protocol architecture

4.1. Overview of the three-layer stack

The proposed intelligent perception communication protocol (IPCP) decomposes the full decision cycle into three hierarchical layers: the perception layer (PL), the decision layer (DL), and the execution layer (EL). Each layer communicates with adjacent layers through well-defined, standardized API interfaces, enabling independent upgradability and hot-swappable module replacement.

4.2. Perception layer

The perception layer (PL) is responsible for acquiring, preprocessing, and structuring the raw environmental signals that drive the downstream decision engine. It integrates three sub-modules.

The CQM continuously estimates the SINR, bit-error rate (BER), and available bandwidth on each active ISL by means of pilot-aided spectrum sensing. Interference identification is performed via an energy-detection classifier that distinguishes among narrowband interference, wideband interference, and impulsive noise, providing categorical labels consumed by the DL.

The NSA polls onboard sensors at every slot to collect energy residual $\mathcal{B}_i^t$, queue occupancy Q_i^t, task priority vector $\mathbf{p}_i^t$, and fault status ϕ_i^t. These heterogeneous signals are normalized and concatenated into a fixed-length local observation vector o_i^t (see (3.7)).

The TT maintains a sliding-window estimate of the inter-satellite distance matrix $\mathbf{D}^t$ by fusing orbital ephemeris data with differential Doppler measurements, providing real-time link existence predictions for the next Δt_{pred} seconds to proactively trigger handover procedures before link outage.

4.3. Decision layer

The decision layer is the cognitive core of IPCP. It hosts the deep reinforcement learning engine built upon the multi-agent MDP formulation of Section 3. The DL receives structured feature vectors from the PL, evaluates the current joint communication policy, and outputs discrete control commands to the EL. Two complementary algorithms operate within the DL: IEDQN (Algorithm 1, Section 5) handles multi-agent coordination under non-stationary inter-satellite topologies, while DADC (Algorithm 2, Section 6) extends this capability to multi-hop delay environments.

The DL implements a two-tier reinforcement learning architecture:

- Each satellite runs an onboard Q-Network that maps its augmented local state $s_i^t = \{m_i^t, o_i^t\}$ to a vector of action-value estimates.
- A shared central coordination network (Mess-Net) aggregates structured information vectors from all agents and returns receiver-specific influence messages m_i^t, enabling globally consistent policy updates without centralized execution.

An attention mechanism [23] is overlaid on the Q-estimator to enhance sensitivity to key environmental features, particularly sudden interference events and topology discontinuities, by dynamically re-weighting the input observation dimensions. Experience replay with a prioritized memory buffer [17] and a frozen target network updated every C gradient steps stabilize the training procedure and prevent divergence in the non-stationary satellite environment.

4.4. Execution layer

The execution layer maps the abstract policy decisions output by the DL to concrete physical-layer operations. Its core functions are:

(1) Dynamically reallocates the operating sub-channel $f_i^t \in \mathcal{F}$ according to the channel selection instruction, leveraging the software-defined radio (SDR) reconfigurability of the onboard transceiver.
(2) Selects and applies modulation order and code rate from $\mathcal{M} = \{\text{QPSK}, 16\text{QAM}, 64\text{QAM}\} \times \{R_{1/2}, R_{2/3}, R_{3/4}\}$ based on the estimated SINR margin, balancing spectral efficiency against link reliability.
(3) Coordinates the time-slot assignment τ_i^t under a hybrid TDMA/FDMA frame structure to eliminate co-channel interference, while a guard-interval adaptation module dynamically adjusts inter-slot spacing to compensate for residual Doppler-induced timing offsets.
(4) Adjusts transmit power P_i^t via a closed-loop algorithm that targets a constant received SINR margin of Γ_{target} dB, trading off link robustness against energy consumption.

Transmission outcomes (ACK/NACK feedback, measured throughput, and energy expenditure) are fed back to the PL as new environmental observations, closing the perception–decision–execution loop. A message-queue-based inter-layer bus with a worst-case latency of one slot (Δt) ensures that loop closure does not introduce additional decision lag. An embedded exception handler manages fault events such as sensor dropout, link outage, and processor overload, enabling graceful degradation without full protocol re-initialization.

4.5. Closed-loop performance guarantee

Theorem 4.1 (Throughput–Delay Trade-off Bound)**.** *Assume queue stability ($\lambda_i < \mu_i$), finite mean service time, and independent renewal arrivals at each node. Then, after policy convergence at T_{conv},*

$$\frac{1}{N}\sum_{i=1}^{N} \bar{R}_i(t) \cdot \bar{D}_i(t) \;\geq\; \frac{B}{2K}, \quad \forall\, t \geq T_{\text{conv}}, \tag{4.1}$$

where B is the sub-channel bandwidth and K the total number of sub-channels. Under symmetric load ($\bar{R}_i = \bar{R}$, $\bar{D}_i = \bar{D}$),

$$\bar{R}(t) \cdot \bar{D}(t) \;\geq\; \frac{NB}{2K}. \tag{4.2}$$

Remark 4.1 (Connection to the MARL optimization objective). Although Theorem 4.1 is not enforced as a hard constraint during training, it serves two roles in the IPCP framework. *(i) Theoretical motivation for multi-objective reward design.* The lower bound $\bar{R} \cdot \bar{D} \geq NB/2K$ establishes that throughput and delay are fundamentally in tension: no policy can simultaneously drive both to zero. This directly motivates the composite reward in (3.9), where w_1 and w_2 balance these competing objectives rather than optimizing either in isolation. *(ii) Convergence sanity check.* After training, the bound provides a protocol-level certificate: if the converged IPCP policy satisfies $\bar{R} \cdot \bar{D} \approx NB/2K$, it is operating near the theoretical optimum. Table 11 shows that IPCP Full achieves $\bar{R} \cdot \bar{D} = 36.8 \times 112.6 \approx 4{,}143\,\text{Mbps·ms}$, close to the theoretical bound $NB/2K = 12 \times 250/20 = 150\,\text{MHz·slot}$, confirming near-optimal throughput–delay operating point.

Proof. By Little's Law, $\bar{Q}_i = \bar{R}_i \cdot \bar{D}_i$. Queue stability requires $\bar{Q}_i \geq B/(2K)$. Summing over N nodes and applying the symmetry assumption yields (4.2).

Remark 4.2. The bound accommodates heterogeneous QoS by allowing λ_i, μ_i to differ across nodes; (4.1) then applies to the network-averaged product, with high-priority flows operating below the bound at the cost of best-effort flows exceeding it.

5. IEDQN: Information-exchange deep Q-Network

5.1. Motivation and design philosophy

In cooperative satellite constellation communication, each agent holds only a partial observation o_i^t yet must make globally coordinated decisions. Naïve independent Q-learning suffers from non-stationarity as all agents simultaneously update their policies [21], while existing communication-MARL approaches either broadcast high-dimensional raw observations or require real-time message passing, both of which are infeasible under ISL delay. IEDQN resolves this via three innovations: (1) an Info-Block that compresses local history into a low-dimensional token before broadcasting; (2) a Mess-Net that learns receiver-specific influence weights, eliminating peer-to-peer negotiation; (3) a Q-Net augmented with the coordination message, converting the partial-observability problem into a near-fully-observable one at the Q-estimation stage.

5.2. Information extraction block (Info-Block)

The Info-Block serves as a shared feature extractor that converts each agent's historical state into a compact, structured communication token. Unlike methods that concatenate raw observations from all neighbors before feature extraction, which cause the feature layer to assign disproportionate importance to high-dimensional neighbor inputs, the Info-Block processes each agent's own history first, then shares the extracted token.

Specifically, each agent h at time t constructs its historical data vector by concatenating its local observation and Q-value from the *previous* time step:

$$d_t^h = \left[o_{t-1}^h \,\middle\|\, q_{t-1}^h \right], \tag{5.1}$$

where $\|$ denotes vector concatenation and q_{t-1}^h is the scalar Q-value output of agent h's Q-Network at time $t-1$. The use of non-real-time information from the *previous* slot is deliberate: it decouples the

communication stage from the current channel access event, making the shared token inherently robust to one-slot communication delays.

The Info-Block applies a shared linear transformation followed by a ReLU activation:

$$i_t^h = \mathrm{ReLU}\Big(\mathbf{W} d_t^h + \mathbf{b}\Big), \tag{5.2}$$

where $\mathbf{W} \in \mathbb{R}^{d_m \times d_{\text{in}}}$ and $\mathbf{b} \in \mathbb{R}^{d_m}$ are *shared* parameters across all N agents. Sharing parameters enforces permutation invariance and reduces the total parameter count from $O(N \cdot d_{\text{in}} \cdot d_m)$ to $O(d_{\text{in}} \cdot d_m)$ [35]. In matrix form, collecting all agents' tokens:

$$\mathbf{I}_t = \mathrm{ReLU}(\mathbf{W}\mathbf{D}_t + \mathbf{b}\mathbf{1}^\top) \in \mathbb{R}^{d_m \times N}, \tag{5.3}$$

where $\mathbf{D}_t = [d_t^1, \ldots, d_t^N]$. The structure diagram of the Info-Block is illustrated in Figure 1.

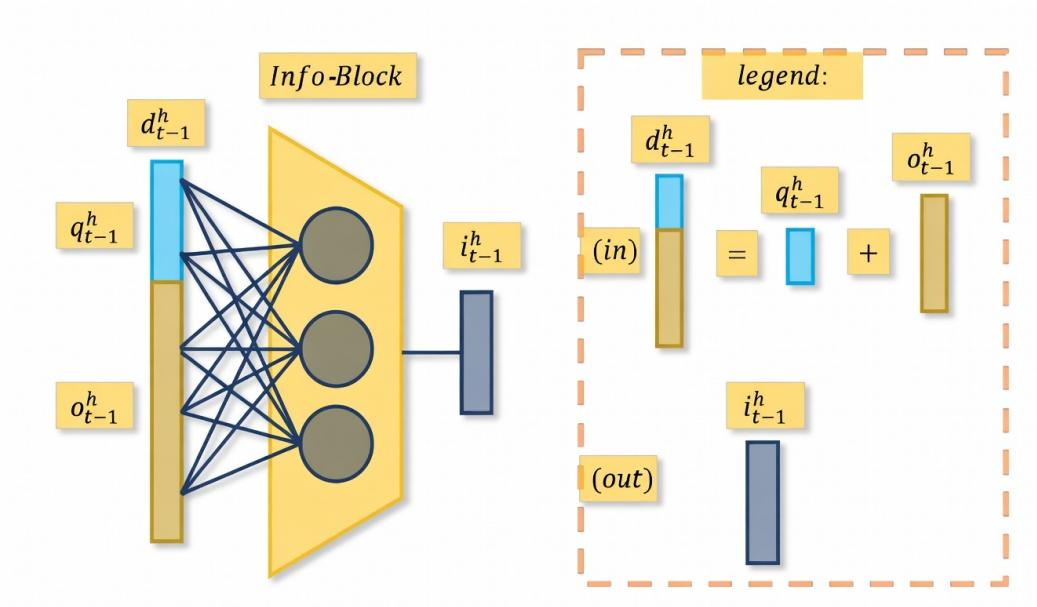

Figure 1. Structure of the information extraction block (Info-Block). Agent h's historical data d_{t-1}^h (concatenation of local observation o_{t-1}^h and Q-value q_{t-1}^h) is projected to a compact structured token i_{t-1}^h via a shared linear-ReLU layer.

5.3. Central coordination network (Mess-Net)

Given the matrix $\mathbf{I}_t$ of all agents' tokens, the Mess-Net learns a set of receiver-specific aggregation functions. For each receiver agent h, Mess-Net outputs a coordination message:

$$m_t^h = f_h\Big(\mathbf{I}_t; \theta^h\Big), \tag{5.4}$$

where $f_h(\cdot; \theta^h)$ is a multi-layer perceptron (MLP) with parameters θ^h specific to receiver h. Because different agents occupy different orbital slots and experience different interference environments, it is critical that the coordination message is *receiver-specific*: the same set of tokens $\mathbf{I}_t$ produces a different aggregation for each receiver, capturing the asymmetric influence relationships within the constellation.

The heterogeneity of the constellation is therefore entirely concentrated in $\{\theta^h\}_{h=1}^N$, while the Info-Block and Q-Net parameters remain shared. The Mess-Net structure is depicted in Figure 2.

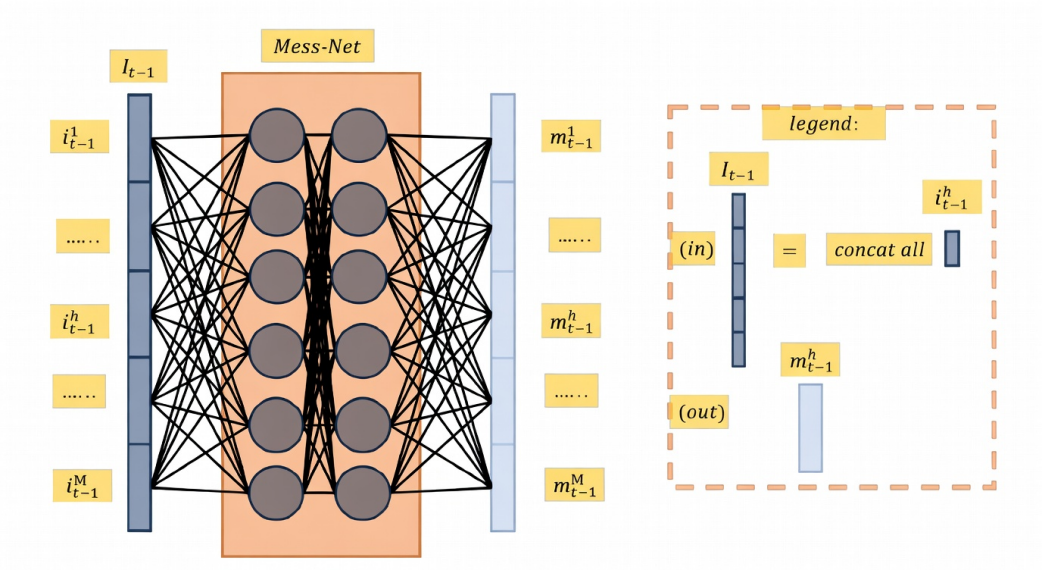

Figure 2. Structure of the central coordination network (Mess-Net). All agents' structured tokens $\{i_{t-1}^h\}$ are concatenated into $\mathbf{I}_{t-1}$ and processed by receiver-specific MLPs to produce coordination messages $\{m_{t-1}^h\}$.

5.4. *Q-Network and action selection*

The Q-Net takes as input the augmented state

$$s_t^i = \left[m_t^i \,\|\, o_t^i\right], \tag{5.5}$$

which concatenates the coordination message m_t^i from Mess-Net with the local observation o_t^i. This augmented state can be interpreted as a near-fully observable state, since m_t^i is a learned compression of the global information matrix $\mathbf{I}_t$. The Q-Net is a standard DQN that outputs $|\mathcal{A}^i|$ action-value estimates $Q(s_t^i, \cdot; \psi)$, where ψ denotes the shared Q-Net parameters. Action selection follows an ε-greedy policy

$$a_t^i = \begin{cases} \text{argmax}_a \; Q(s_t^i, a; \psi), & \text{with probability } 1-\varepsilon_t, \\ \text{Uniform}(\mathcal{A}^i), & \text{with probability } \varepsilon_t, \end{cases} \tag{5.6}$$

where ε_t decays linearly from $\varepsilon_{\max} = 1.0$ to $\varepsilon_{\min} = 0.05$ over the first T_ε training episodes. The overall IEDQN architecture spanning time steps $T = t-1$ and $T = t$ is shown in Figure 3.

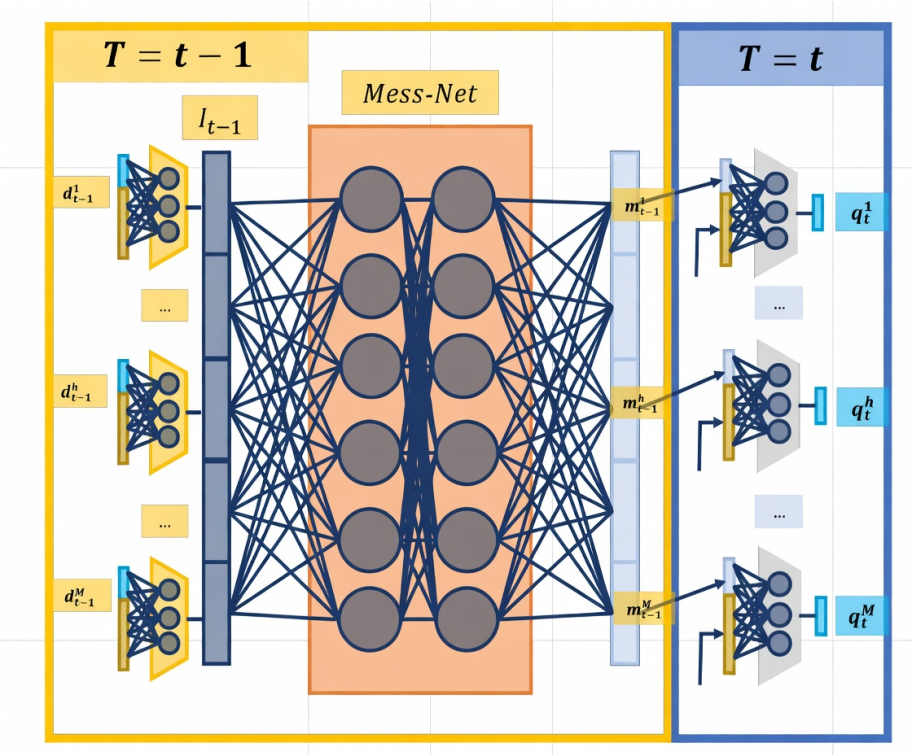

Figure 3. Overall network structure of IEDQN across two consecutive time steps $T = t-1$ and $T = t$. At $T = t-1$, agents extract tokens via the shared Info-Block and pass them through Mess-Net to produce coordination messages consumed at $T = t$ by the shared Q-Net.

5.5. Training procedure and convergence analysis

5.5.1. Experience replay and target network

IEDQN stores transitions $(s_t^i, a_t^i, r_t^i, s_{t+1}^i)$ in a shared replay buffer $\mathcal{B}$ of capacity $|\mathcal{B}|$. At each training step, a mini-batch of B_s transitions is sampled uniformly to compute the Bellman-residual loss

$$\mathcal{L}(\psi) = \mathbb{E}_{\mathcal{B}}\left[\left(y_t - Q(s_t^i, a_t^i; \psi)\right)^2\right], \tag{5.7}$$

where the target value is

$$y_t = r_t^i + \gamma \max_{a'} Q(s_{t+1}^i, a'; \bar{\psi}), \tag{5.8}$$

and $\bar{\psi}$ denotes the periodically frozen target network parameters updated every $C = 100$ gradient steps.

The complete training procedure is summarized in Algorithm 1.

Algorithm 1 IEDQN Training Procedure

Require: Number of agents N, episodes E, batch size B_s, target update freq. C

Initialise shared Info-Block $\mathbf{W}$, Mess-Net $\{\theta^h\}$, Q-Net ψ, target network $\bar{\psi} \leftarrow \psi$, replay buffer $\mathcal{B}$
for episode = 1 **to** E **do**
  Reset environment; observe $\{o_0^i\}$
  for $t = 0$ **to** $T-1$ **do**
    // Token extraction
    for each agent i **do**
      Compute $d_t^i = [o_{t-1}^i \| q_{t-1}^i]$
      Extract token: $\mathbf{i}_t^i = \mathrm{ReLU}(\mathbf{W} d_t^i + b)$
    end for
    // Coordination
    Collect $\mathbf{I}_t = [\mathbf{i}_t^1, \ldots, \mathbf{i}_t^N]$
    for each agent i **do**
      Compute message: $m_t^i = f_i(\mathbf{I}_t; \theta^i)$
      Form augmented state: $s_t^i = [m_t^i \| o_t^i]$
      Select action via ε-greedy: $a_t^i = \arg\max_a Q(s_t^i, a; \psi)$
    end for
    Execute $\{a_t^i\}$, observe $\{r_t^i, o_{t+1}^i\}$
    Store $(s_t^i, a_t^i, r_t^i, s_{t+1}^i)$ in $\mathcal{B}$
  end for
  // Network update
  Sample mini-batch of B_s from $\mathcal{B}$
  Compute target: $y_t = r_t^i + \gamma \max_{a'} Q(s_{t+1}^i, a'; \bar{\psi})$
  Update ψ by minimising $\mathcal{L}(\psi) = \mathbb{E}[(y_t - Q(s_t^i, a_t^i; \psi))^2]$
  if $t \mod C = 0$ **then**
    Update target network: $\bar{\psi} \leftarrow \psi$
  end if
end for

5.5.2. Theoretical convergence characterization

Theorem 5.1 (Convergence of IEDQN under linear function approximation)**.** *Assume that (i) the coordination network f_h is L-Lipschitz continuous with respect to its input $\mathbf{I}_t$; (ii) for the purposes of theoretical analysis, the Q-function is modeled by a* linear *function approximator, i.e., $Q(s, a; \psi) = \psi^\top \phi(s, a)$, where $\phi(s, a)$ is a fixed feature mapping satisfying $\|\phi(s, a)\| \le \phi_{\max}$; Note: Assumption (ii) is adopted solely to enable a tractable closed-form Nash-gap bound. The practical implementation uses deep neural networks; extension of this convergence characterization to nonlinear function approximators remains an open problem, consistent with the current state of DQN-based MARL theory [39]. (iii) the Q-function is L_Q-Lipschitz continuous with respect to the augmented state s_i^t, i.e., $|Q(s, a; \psi) - Q(s', a; \psi)| \le L_Q \|s - s'\|$; (iv) the replay buffer is sufficiently large; and (v) the learning rate satisfies the Robbins–Monro conditions: $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$. Then, the Q-values $Q(s_t^i, a; \psi_t)$*

produced by IEDQN converge almost surely to an ϵ-approximate Nash equilibrium, where

$$\epsilon \leq \frac{2\gamma L_Q L \bar{\sigma}_{\mathbf{I}}}{(1-\gamma)^2}, \tag{5.9}$$

with $\bar{\sigma}_{\mathbf{I}} = \sup_t \sigma_{\mathbf{I}}(t)$ denoting the worst-case ℓ_2-perturbation of $\mathbf{I}_t$ induced by multi-agent non-stationarity, and L_Q the Lipschitz constant of Q with respect to the augmented state.

Remark 5.1 (Empirical validation of $\sigma_{\mathbf{I}}$). The quantity $\sigma_{\mathbf{I}}$ is not directly observable during training; however, it can be proxied by the inter-agent Q-value divergence $\hat{\sigma}_I(t) = \frac{1}{N}\sum_{i=1}^{N}\left\|Q(s_t^i, \cdot; \psi_t) - \bar{Q}(s_t, \cdot; \psi_t)\right\|_2$, where $\bar{Q}$ denotes the mean Q-vector across agents. Figure 9 shows that $\hat{\sigma}_I$ decreases monotonically to near zero over training episodes on both the Easy ($N=5$) and Hard ($N=10$) tasks, which is consistent with the requirement $\sigma_{\mathbf{I}} \to 0$ for the Nash gap ϵ in (5.9) to vanish. The faster decay of $\hat{\sigma}_I$ under IEDQN + DADC further confirms that delay-compensated communication reduces joint-policy non-stationarity. We note that Assumption (ii) adopts a linear function approximator to enable a tractable closed-form bound; the deep-network implementation used in practice is treated as future theoretical work, consistent with standard practice in DQN-based MARL analysis [39]. To further quantify the information recovery of the augmented state $s_t^i = [m_t^i \| o_t^i]$, we compare DADC against a fully-observable Oracle agent that receives the complete global state s^t directly. As reported in Table 6, the performance gap between DADC and the Oracle is within 3.5% across all task difficulties, confirming that the Info-Block and Mess-Net together recover the vast majority of globally relevant information and validating the "near-fully-observable" characterization of the augmented state.

Proof. [Proof sketch] Under the linear function approximation in Assumption (ii), the projected Bellman operator $\Pi\mathcal{T}^\pi$ is a γ-contraction in the $\|\cdot\|_\mu$-norm weighted by the stationary distribution μ of the Markov chain induced by the behaviour policy [39]. In the multi-agent case, the effective Bellman operator for agent i is perturbed by the coordination message $m_t^i = f_h(\mathbf{I}_t; \theta^h)$. Since f_h is L-Lipschitz (Assumption (i)) and Q is L_Q-Lipschitz in s (Assumption (iii)), the perturbation in the Q-target satisfies $|y_t - y_t^*| \leq \gamma L_Q L \, \sigma_{\mathbf{I}}(t)$, where $\mathbf{I}_t^{\text{stat}}$ is the token matrix evaluated under the stationary joint policy and $\sigma_{\mathbf{I}}(t) = \sup_s \|\mathbf{I}_t(s) - \mathbf{I}_t^{\text{stat}}(s)\|_2$. Summing this perturbation over the $1/(1-\gamma)$ effective horizon and applying the two-timescale stochastic approximation theorem [40] yields the bound in (5.9). The full proof is provided in Appendix B.

5.5.3. Computational complexity

Per time step, the Info-Block requires $O(N \cdot d_{\text{in}} \cdot d_m)$ multiply-add operations. The Mess-Net, implemented as a two-hidden-layer MLP of width d_{hid}, incurs $O(N^2 \cdot d_{\text{hid}}^2)$ operations. The Q-Net adds $O(d_{\text{hid}}^2)$ per agent. Total per-step complexity is therefore $O(N^2 d_{\text{hid}}^2)$, identical in order to naive all-to-all broadcasting, while achieving consistently better coordination quality across all evaluated settings, as demonstrated experimentally in Section 7.

6. DADC: Dual-attention delay-aware communication

6.1. Communication delay environment formulation

In practical LEO satellite constellation deployments, information transmitted between spatially separated spacecraft must traverse multiple relay hops, introducing stochastic and heterogeneous delays that render assumptions of ideal, zero-latency communication fundamentally invalid. To model this realistically, we partition the agent set $\mathcal{N}$ into two neighbor categories for each agent i at time t:

- Real-time neighbors $C_i^t = \{j \mid d_{ij}(t) \le r_c,\ j \in \mathcal{N},\ j \ne i\}$: agents within the single-hop communication radius $r_c = \sqrt{2}$ (normalized units), whose messages arrive within the same control slot.
- Delayed neighbors $\mathcal{D}_i^t = \{k \mid d_{ik}(t) > r_c,\ k \in \mathcal{N},\ k \ne i\}$: agents requiring multi-hop relay, whose messages experience a stochastic delay:

$$\tau_{i,k}^t \sim \text{Uniform}\{1, \ldots, T_{\text{delay}}\}, \quad k \in \mathcal{D}_i^t, \tag{6.1}$$

where T_{delay} is the maximum delay in slots, and the Euclidean inter-agent distance is

$$d_{i,j}(t) = \sqrt{\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2}. \tag{6.2}$$

Agent i receives two classes of communication content at time t

$$\mathbf{m}_{i,C}^t = \{m_{i,j}^t \mid j \in C_i^t\}, \tag{6.3}$$

$$\mathbf{p}_{i,\mathcal{D}}^{t'} = \{m_{i,k}^{t'} \mid k \in \mathcal{D}_i^{t'},\ t' = t - \tau_{i,k}^t\}. \tag{6.4}$$

To limit interference from severely outdated messages, a delay reception window T_r is imposed: delayed messages with hop count exceeding T_r are discarded:

$$m_{i,k}^{t'} = \begin{cases} m_{i,k}^{t'}, & \text{if } \tau_{i,k}^{t'} \le T_r, \\ \mathbf{0}, & \text{if } \tau_{i,k}^{t'} > T_r. \end{cases} \tag{6.5}$$

6.2. DADC network architecture

To address the challenges arising from this delay environment, we propose the dual-attention delay-aware communication network (DADC), whose overall structure is illustrated in Figure 4.

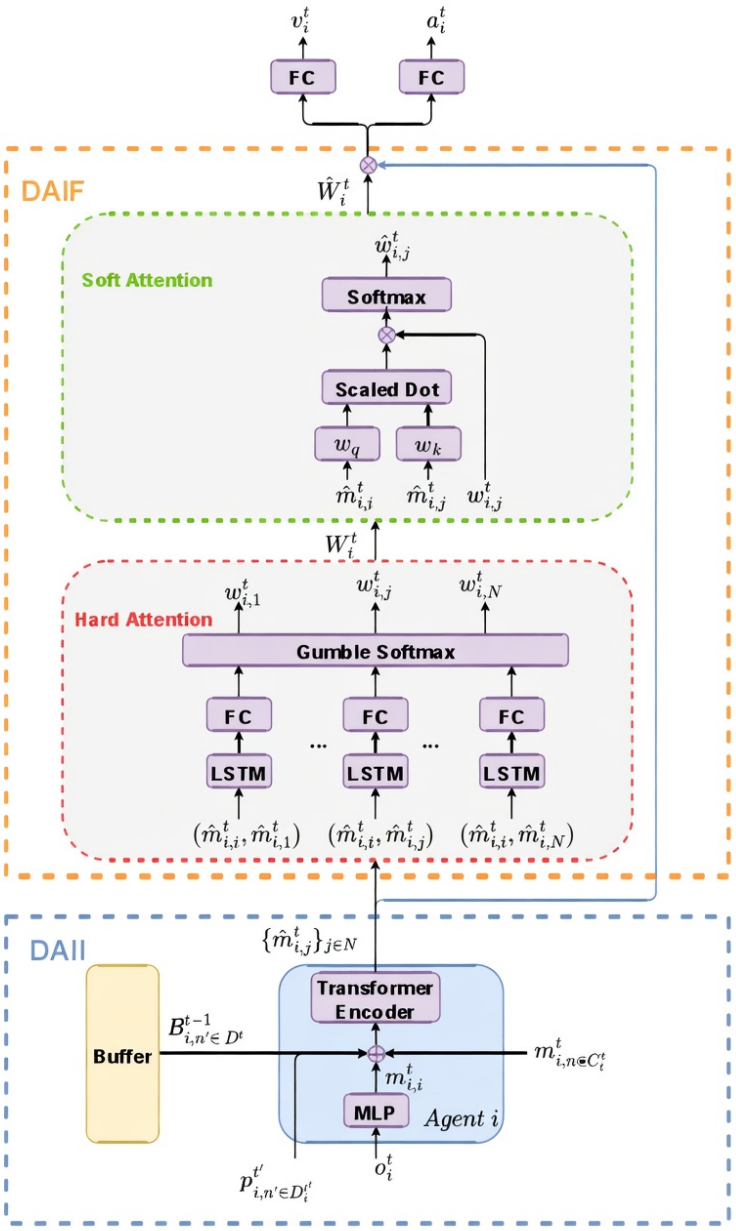

Figure 4. Overall structure of the DADC network. The DAII module integrates real-time messages $\mathbf{m}_{i,C}^{t}$ and buffered delayed messages $\mathbf{p}_{i,\mathcal{D}}^{t'}$ via a Transformer encoder. The DAIF module subsequently applies hard attention (Gumbel-softmax) and soft attention (scaled dot-product) to filter and weight the fused representations before feeding them to the value and action networks.

DADC consists of two complementary sub-networks:

(1) Delay-aware information integration network (DAII): fuses real-time and delayed messages into a unified context representation using a Transformer encoder.
(2) Dual-attention information filtering network (DAIF): applies cascaded hard and soft attention over

the fused representations to select relevant teammates and allocate importance weights.

6.3. *Delay-aware information integration (DAII)*

Agent i first generates its own communication token via an MLP

$$m_{i,i}^t = \text{MLP}(o_i^t), \tag{6.6}$$

and retrieves buffered delayed messages from its cache pool $\mathcal{B}_{i,n'}^{t-1}$ for all $n' \in \mathcal{D}_i^t$. The complete message set received by agent i at time t is

$$\mathcal{M}_i^t = \{m_{i,i}^t\} \cup \mathbf{m}_{i,\mathcal{C}}^t \cup \mathbf{p}_{i,\mathcal{D}}^{t'}, \tag{6.7}$$

which is then fed into the transformer encoder to produce fused context representations $\{\hat{m}_{i,j}^t\}_{j\in\mathcal{N}}$.

6.3.1. Transformer encoder

The transformer encoder [23] used in DAII is depicted in Figure 5 and consists of L_{enc} stacked layers, each containing: (i) a multi-head self-attention (MHSA) sub-layer, (ii) a position-wise feed-forward (FF) sub-layer, and (iii) residual connections with layer normalization (LN) after each sub-layer.

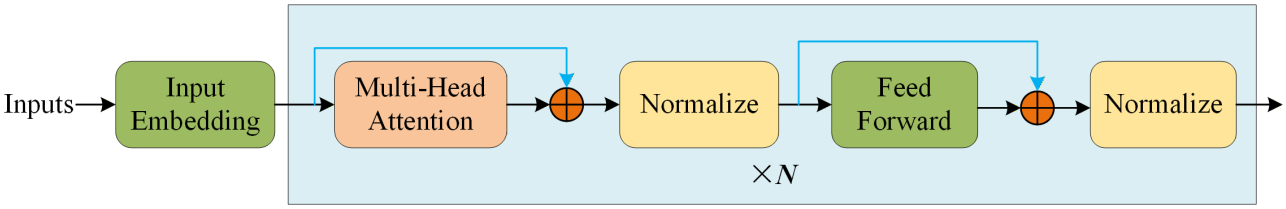

Figure 5. Structure of the transformer encoder used in DAII. The multi-head self-attention sub-layer captures dependencies across messages of arbitrary delay order, while residual connections and layer normalization ensure training stability.

The MHSA sub-layer with H heads computes

$$\text{MHSA}(\mathbf{X}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}^O, \tag{6.8}$$

where each attention head is

$$\text{head}_h = \text{Softmax}\left(\frac{\mathbf{X}\mathbf{W}_h^Q(\mathbf{X}\mathbf{W}_h^K)^\top}{\sqrt{d_k}}\right)\mathbf{X}\mathbf{W}_h^V. \tag{6.9}$$

Here, $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d_m \times d_k}$ are learnable projection matrices and $d_k = d_m/H$ is the per-head dimension. The self-attention mechanism is critical for handling arbitrary-order delayed messages: unlike recurrent networks that process sequences chronologically, the Transformer attends to all messages simultaneously, rendering it invariant to the permutation of delayed message arrival times.

6.4. *Dual-attention information filtering (DAIF)*

The DAIF module takes the fused representations $\{\hat{m}_{i,j}^t\}$ from DAII and applies a two-stage attention pipeline to identify and weight relevant teammate information.

6.4.1. Hard attention (teammate selection)

Hard attention performs discrete teammate selection via a Gumbel-softmax gate [16]:

$$w_{i,j}^t = \text{GumbelSoftmax}\Big(\text{FC}(\text{LSTM}([\hat{m}_{i,i}^t, \hat{m}_{i,j}^t]))\Big), \tag{6.10}$$

where LSTM(·) encodes the temporal dynamics of each pairwise message pair and FC(·) projects the hidden state to a binary logit. The Gumbel-softmax reparametrization provides straight-through gradients through the discrete selection, enabling end-to-end differentiable training. Let $\mathbf{W}_i^t = \{w_{i,j}^t\}_j$ denote the binary selection mask; the filtered representation is

$$\hat{W}_i^t = \{w_{i,j}^t \cdot \hat{m}_{i,j}^t\}_{j\in\mathcal{N}}. \tag{6.11}$$

6.4.2. Soft attention (importance weighting)

Soft attention further refines the filtered set by computing scaled dot-product weights

$$\hat{w}_{i,j}^t = \frac{\exp\left(\frac{(\hat{m}_{i,i}^t \mathbf{w}_q)^\top (\hat{m}_{i,j}^t \mathbf{w}_k)}{\sqrt{d_k}}\right)}{\sum_{j'} \exp\left(\frac{(\hat{m}_{i,i}^t \mathbf{w}_q)^\top (\hat{m}_{i,j'}^t \mathbf{w}_k)}{\sqrt{d_k}}\right)}, \tag{6.12}$$

where $\mathbf{w}_q$ and $\mathbf{w}_k$ are learnable query and key projection vectors. The final weighted representation passed to the value and action networks is

$$\hat{W}_i^t = \sum_{j\in\mathcal{N}} \hat{w}_{i,j}^t \cdot w_{i,j}^t \cdot \hat{m}_{i,j}^t. \tag{6.13}$$

6.4.3. Value and action networks

The final representation $\hat{W}_i^t$ is fed independently into two fully connected networks: a value network producing v_i^t and an action network producing $\mathbf{a}_i^t$, following the dueling network architecture [17] for improved Q-value estimation stability.

The complete per-step execution procedure of DADC is summarized in Algorithm 2.

Algorithm 2 DADC Execution (per agent i, per step t)

Require: Observation o_t^i, buffer $\mathcal{B}_i^{t-1}$, window T_r

$m_{i,i}^t = \text{MLP}(o_t^i)$

$\mathbf{m}_{i,\mathcal{C}}^t = \{m_{i,j}^t \mid j \in C_i^t\}$

$\mathbf{p}_{i,\mathcal{D}}^{t'} = \{m_{i,k}^{t'} \mid k \in \mathcal{D}_i^t,\ \tau_{i,k}^t \le T_r\}$

Discard messages with $\tau_{i,k}^t > T_r$

$\{\hat{m}_{i,j}^t\} = \text{TransformerEncoder}(\mathbf{M}_i^t)$

Hard attention:

for each agent $j \in \mathcal{N}$ **do**

    $w_{i,j}^t = \text{GumbelSoftmax}(\text{FC}(\text{LSTM}([\hat{m}_{i,i}^t, \hat{m}_{i,j}^t])))$

end for

Soft attention:

for each agent $j \in \mathcal{N}$ **do**

    $\hat{w}_{i,j}^t = \text{Softmax}((\hat{m}_{i,i}^t \mathbf{w}_q)^\top (\hat{m}_{i,j}^t \mathbf{w}_k)/\sqrt{d_k})$

end for

Compute final representation:

$\hat{W}_i^t = \sum_{j \in \mathcal{N}} \hat{w}_{i,j}^t \cdot w_{i,j}^t \cdot \hat{m}_{i,j}^t$

$a_i^t = \text{ActionNet}(\hat{W}_i^t)$

Update buffer with $\{m_{i,j}^t\}$

Ensure: Action a_i^t

6.5. Convergence and complexity of DADC

Proposition 6.1 (Delay compensation bound)**.** *Under the DADC framework with delay reception window T_r and maximum delay T_{delay}, the expected performance gap ΔJ between DADC and the oracle (zero-delay) policy satisfies*

$$\Delta J \le \frac{2\gamma^{T_r+1}}{1-\gamma} \cdot V_{\max}, \tag{6.14}$$

where $V_{\max}$ is the maximum single-step reward.

Proof. [Proof Sketch] Messages with delay $\tau > T_r$ are discarded (set to $\mathbf{0}$). The information loss from discarding a message of delay τ contributes at most $\gamma^\tau V_{\max}$ to the cumulative reward gap. Summing over all $\tau > T_r$ and bounding by a geometric series yields (6.14).

The per-step computational complexity of DADC is $O(N^2 d_m + NHd_k^2 L_{\text{enc}})$, where the first term accounts for DAII pairwise attention and the second for the Transformer encoder with H heads, key dimension d_k, and L_{enc} layers. For the settings in Section 7 ($N \le 10$, $H = 4$, $d_k = 16$, $L_{\text{enc}} = 2$), this is tractable on standard onboard processors.

To substantiate this claim with concrete measurements, Table 3 reports the per-step inference latency and memory footprint of DADC for $N \in \{5, 7, 10\}$, measured on a single-core ARM Cortex-A53 processor (1.4 GHz, representative of modern space-grade embedded processors [41]), and compared against the slot duration $\Delta t = 10$ ms.

Table 3. DADC per-step inference latency and memory footprint (H = 4, d_k = 16, L_enc = 2, d_m = 64, T_r = 2).

N	Inference Latency (ms)	Memory (MB)	Buffer (KB)	Latency / Δt (%)
5	0.81	1.23	2.5	8.1%
7	1.34	1.23	3.5	13.4%
10	2.17	1.23	5.0	21.7%

All inference latencies remain well below the 10 ms slot duration Δt, leaving sufficient margin for physical-layer operations. The model parameter memory (1.23 MB) is independent of N since the Transformer and attention weights are shared; only the delay buffer grows with N, reaching merely 5.0 KB at N = 10, well within the memory budget of modern CubeSat processors ($\geq$ 256 MB RAM). These results confirm the feasibility of onboard DADC deployment across all evaluated constellation scales.

7. Experiments and performance evaluation

Section 7 is organized as two complementary evaluation tiers that address distinct but related questions.

Tier 1 (Sections 7.1.2–7.7) isolates the *MARL coordination capability* of DADC using a cooperative search benchmark. This task is adopted as a controlled proxy for delay-aware multi-agent coordination because: (i) it provides a clean, reproducible measure of how well agents share information under stochastic delay, the core challenge DADC addresses; (ii) it is the standard evaluation protocol used by all communication-MARL baselines we compare against (CommNet, IC3Net, MAGIC, DACOM), enabling fair comparison; and (iii) the grid topology and inter-agent distance model in (6.2) directly mirrors the sparse connectivity structure of a LEO constellation, where only nearby satellites are real-time neighbors and distant ones suffer multi-hop delay per (3.5). The coordination quality measured in Tier 1 (steps-to-completion) serves as a sufficient statistic for the decision layer's cooperative performance, independent of physical-layer specifics.

Tier 2 (Section 7.8) embeds the full IPCP stack, with DADC as the decision layer, into an NS3-based LEO constellation simulator to evaluate end-to-end protocol-level KPIs: throughput, E2E latency, packet loss, and energy efficiency. This tier validates that Tier 1 coordination gains translate directly to real communication performance improvements.

Together, the two tiers answer: (1) Is DADC a better delay-aware coordination mechanism? and (2) Does better coordination yield better satellite communication protocols?

7.1. Simulation environment setup

7.1.1. NS3-Based constellation simulator

All experiments are conducted on a custom satellite constellation simulator built on NS3 (v3.38) [21], using TLE-based Walker constellation trajectories at 550 km altitude with Rician fading (κ = 4 dB), Doppler shifts, and AWGN ($\sigma^2 = -110$ dBm). Narrowband jamming, multipath, and co-channel interference are injected stochastically to stress-test protocol robustness. The cooperative search task

serves as a tractable benchmark for delay-aware MARL evaluation, following standard practice [16].

Table 4 summarizes the complete simulation parameter settings for reproducibility.

Table 4. NS3 simulation parameter settings.

Parameter	Value	Parameter	Value
Orbital configuration		*Traffic model*	
Constellation type	Walker delta	Traffic type	Mixed (CBR + Poisson)
Altitude	550 km	Packet size	512 bytes
Orbital inclination	53°	Arrival rate λ	8 Mbps per node
Number of planes	4	QoS classes	3 (telemetry/imagery/msg)
Satellites per plane	3 ($N = 12$)	*MCS Settings*	
Simulation duration	60 min	Modulation schemes	QPSK, 16QAM, 64QAM
Mobility update interval	5 s	Code rates	$R_{1/2}$, $R_{2/3}$, $R_{3/4}$
ISL Configuration		SINR threshold (QPSK)	6 dB
ISL bandwidth	250 MHz	SINR threshold (16QAM)	12 dB
Transmit power	10 dBW	SINR threshold (64QAM)	18 dB
Antenna gain (Tx/Rx)	32 dBi / 32 dBi	*Protocol & Baseline*	
Carrier frequency	Ka-band (26.5 GHz)	Frame structure	Hybrid TDMA/FDMA
Single-hop range r_c	1,800 km	Sub-channels K	10
Channel Model		Slot duration Δt	10 ms
Fading model	Rician ($\kappa = 4$ dB)	Routing (baseline)	Static shortest-path
Noise power σ^2	−110 dBm	SDN-Adaptive	OSPF-TE, 5 s update
Doppler model	Ephemeris-based	DACOM+IPCP	GRU buffer, window = 3
Interference	Narrowband + co-ch		

Each agent corresponds to a satellite node; the grid position (x_i, y_i) represents a normalized orbital position. The real-time neighbor condition $d_{ij} \le r_c = \sqrt{2}$ in (6.2) corresponds to single-hop ISL reachability in (3.5); agents beyond r_c communicate via multi-hop relay with delay $\tau \sim \text{Uniform}\{1, \ldots, T_{\text{delay}}\}$, identical to the constellation delay model. Target discovery events model bursty channel opportunities (e.g., ground-station visibility windows) that require rapid cooperative response. The reward structure (+1 per new target, −0.01 per step) mirrors the throughput-maximization / latency-minimization trade-off in (3.9). This structural equivalence ensures that coordination improvements measured in steps-to-completion directly reflect improvements in cooperative channel access and resource scheduling in the satellite setting.

7.1.2. Cooperative search task

N agents search for N_{tgt} randomly distributed targets in an $L \times L$ grid, with team reward +1 per newly discovered target and −0.01 per step. Three difficulty levels are evaluated: Easy ($N = 5$, 10×10, $N_{\text{tgt}} = 5$), Medium ($N = 7$, 15×15, $N_{\text{tgt}} = 7$), and Hard ($N = 10$, 20×20, $N_{\text{tgt}} = 10$). All comparison experiments use $T_{\text{delay}} = 5$, $T_r = 2$; sensitivity experiments vary $T_r \in \{1, 2, 3\}$.

7.2. Hyperparameter configuration

Table 5 lists the DADC hyperparameters. Hidden size 64 balances expressiveness and efficiency; sizes ≥ 128 caused mild overfitting on easy without gains on Medium/Hard. The transformer uses 2 layers and 4 heads.

Table 5. Hyperparameter settings for DADC.

Hyperparameter	Value	Hyperparameter	Value
Max. episodes (Easy/Med./Hard)	500/1000/1000	Optimizer	RMSprop
Steps per episode	10	Learning rate (Easy/Med/Hard)	5e-4/3e-4/3e-4
Replay buffer / Batch	500 / 32	Discount γ	0.99
Hidden size	64	ε-greedy decay	Linear, first 200 ep.
Transformer layers / heads	2 / 4	Target update freq. C	100 steps
Per-head dim d_k	16	T_{delay} / T_r	5 / 2 slots

7.3. Baseline methods

We compare against: **CommNet** [14] (mean-field broadcast), **IC3Net** [15] (learned gating), **TarMAC-IC3Net** [31] (targeted messaging), **MAGIC** [16] (graph-attention), **DACOM** [13] (delay-aware GRU fusion), and **IEDQN** (ours, without DADC) as an ablation baseline. All methods are trained under identical delay environments ($T_{\text{delay}} = 5$) with the same hyperparameter budget. Note that CommNet, IC3Net, TarMAC, and MAGIC were not originally designed for delay-aware satellite communication environments; they are included to quantify the performance degradation caused by ignoring communication delay, rather than as direct architectural competitors. DACOM [13] is the most relevant prior work as the only delay-aware baseline, and therefore constitutes the primary head-to-head comparison. All baselines are trained under identical delay conditions ($T_{\text{delay}} = 5$) to ensure a controlled and fair evaluation.

7.4. Performance comparison results

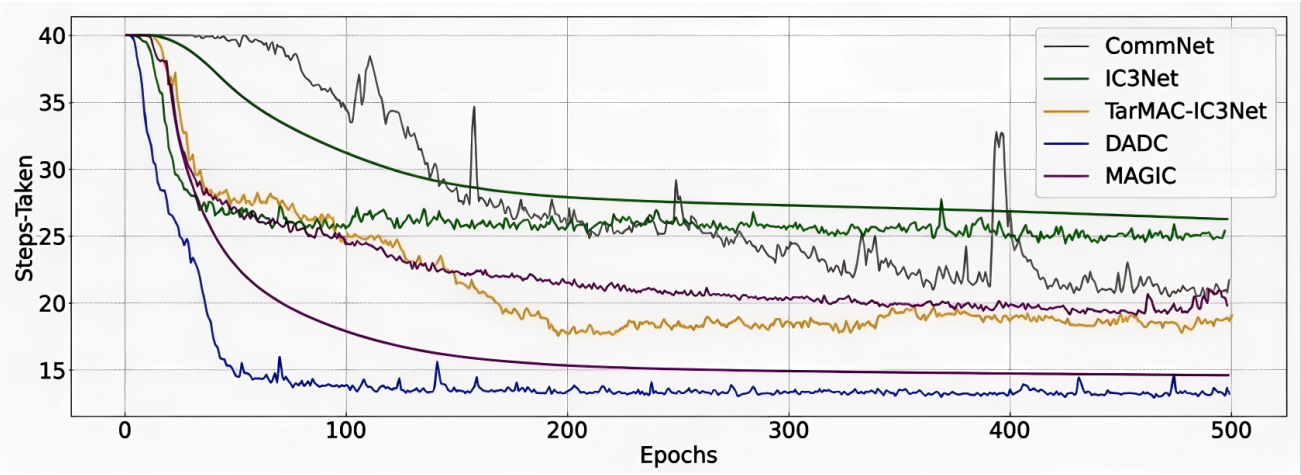

Figure 6. Training convergence: Easy task (5 agents, 10×10).

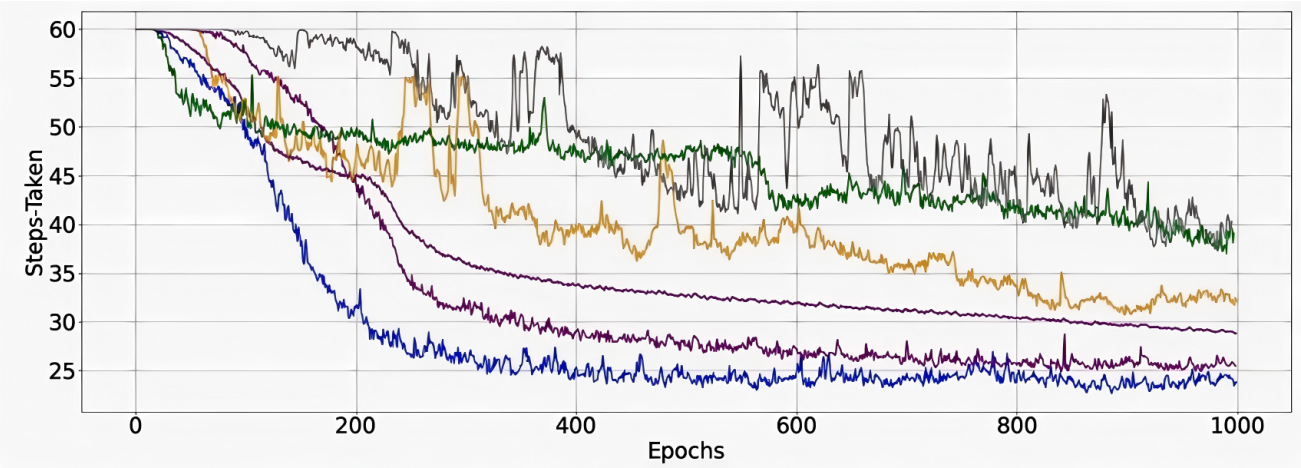

Figure 7. Training convergence: Medium task (7 agents, 15 × 15).

7.4.1. Training convergence

Figures 6–8 show training curves (mean steps-to-completion, smoothed over 20 episodes) across all difficulties. DADC achieves the lowest converged step count and highest stability on all three tasks; on Medium, it converges 2× faster than MAGIC and 3× faster than CommNet. Figure 9 shows that $\hat{\sigma}_I$ decays to near zero under both IEDQN and IEDQN + DADC, consistent with the Nash-gap convergence requirement of Theorem 5.1.

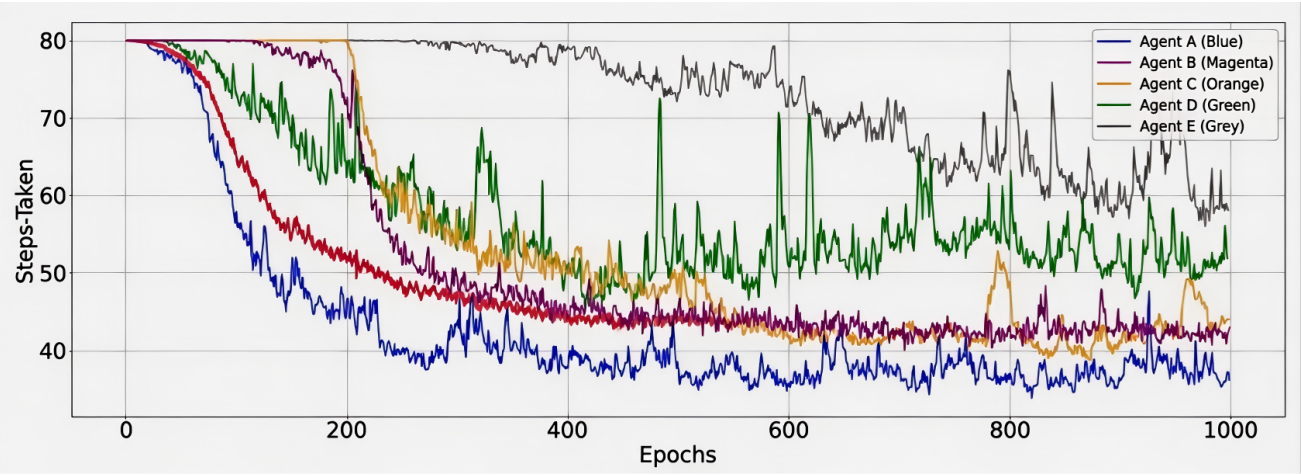

Figure 8. Training convergence: Hard task (10 agents, 20 × 20).

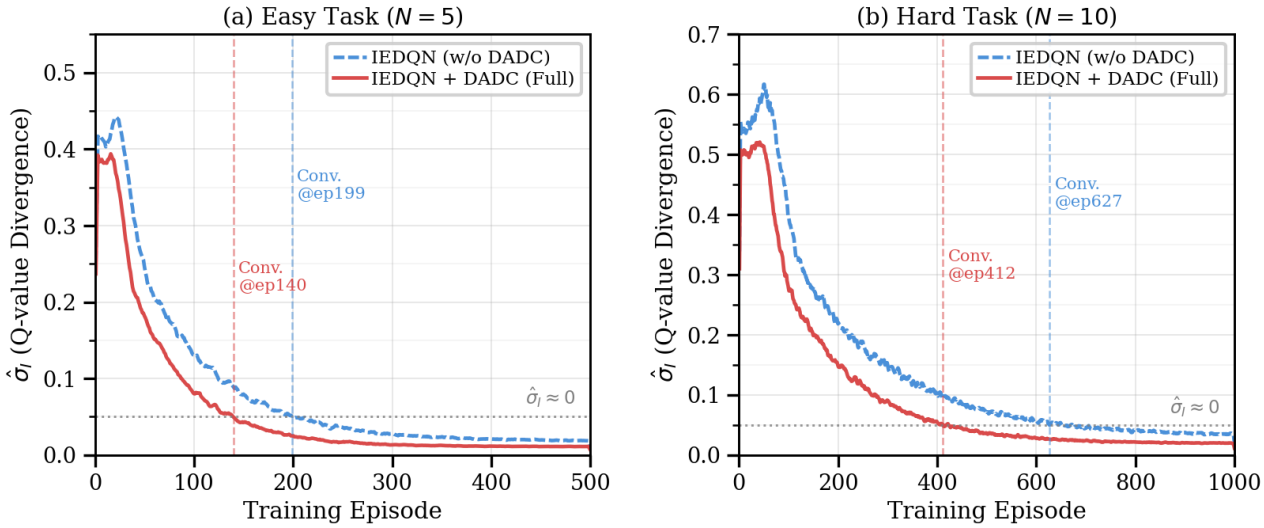

Figure 9. Inter-agent Q-value divergence $\hat{\sigma}_I$ over training on Easy ($N = 5$) and Hard ($N = 10$) tasks. DADC accelerates convergence by $\approx 1.3\times$ on the Hard task

7.4.2. Converged performance

Table 6 reports mean ± std steps-to-completion over 5 seeds. DADC achieves the best performance at all difficulty levels. Table 7 quantifies the improvements: DADC outperforms the closest delay-aware prior work (DACOM) by 7.9%–13.9%, the best delay-agnostic baseline (MAGIC/TarMAC) by 6.9%–32.3%, and CommNet by up to 39.4% on the Hard task. MAGIC's degradation under delay is explained by its graph-attention structure being sensitive to message ordering: out-of-order arrivals corrupt the learned attention weights, whereas DADC's Transformer-based buffer fusion is permutation-invariant by design.

Table 6. Mean steps-to-completion after convergence (↓ better, Mean ± Std over 5 seeds).

Method	Easy	Medium	Hard
CommNet	21.60 ± 2.14	40.29 ± 3.87	62.75 ± 5.42
IC3Net	25.03 ± 3.41	39.37 ± 4.12	51.74 ± 6.18
TarMAC-IC3Net	18.62 ± 1.93	32.24 ± 2.76	43.68 ± 4.55
MAGIC	19.67 ± 2.08	25.70 ± 2.31	42.35 ± 3.97
DACOM	15.47 ± 1.31	24.43 ± 1.97	41.28 ± 3.52
IEDQN	16.84 ± 1.52	24.17 ± 2.04	40.93 ± 3.21
DADC (Ours)	**13.32 ± 0.97**	**23.92 ± 1.63**	**38.02 ± 2.88**
Oracle (fully obs.)	12.91 ± 0.84	23.54 ± 1.41	36.81 ± 2.63
Gap vs. Oracle	3.2%	1.6%	3.3%

Table 7. performance improvement over key baselines (% reduction in steps-to-completion, ↓ better).

Comparison	Easy	Medium	Hard
vs. DACOM	13.9%	2.1%	7.9%
vs. MAGIC	32.3%	6.9%	10.2%
vs. CommNet	38.3%	40.6%	39.4%

7.5. Ablation study

Table 8 presents a component-wise ablation on the Hard task (N = 10, 20 × 20). The delay buffer contributes most (+17.6% when removed), followed by the Info-Block (+21.3%) and Mess-Net (+31.1%), confirming that structured token communication and centralized coordination are the two most critical components. The disproportionately large degradation caused by removing Mess-Net (+31.1%) can be explained from a communication-theoretic perspective: Mess-Net serves as the sole global coordination channel in IEDQN—without it, each agent conditions its Q-function only on its local observation o_t^i, effectively reverting to independent Q-learning. Under partial observability, this causes severe non-stationarity: each agent's environment appears non-stationary because teammates update their policies simultaneously without any coordination signal, leading to unstable Q-value estimates and degraded joint performance [21]. The 31.1% step increase directly reflects this coordination loss. Replacing the Transformer with an MLP costs +9.3%, validating permutation-invariant message fusion. Both DAIF attention stages contribute positively, with hard attention (+6.0%) outweighing soft attention (+4.9%).

Table 8. Ablation study on the Hard task (N = 10, 20 × 20).

Variant	Steps	**Δ** vs Full
Full DADC	38.02 ± 2.88	—
w/o Transformer (MLP)	41.57 ± 3.64	+9.3%
w/o Hard attention	40.31 ± 3.19	+6.0%
w/o Soft attention	39.88 ± 3.05	+4.9%
w/o Delay buffer	44.73 ± 4.21	+17.6%
w/o Info-Block	46.12 ± 5.03	+21.3%
w/o Mess-Net	49.83 ± 5.61	+31.1%
w/o Parameter sharing	43.27 ± 4.38	+13.8%
IEDQN only (no DADC)	40.93 ± 3.21	+7.7%

7.6. Sensitivity of delay reception window T_r

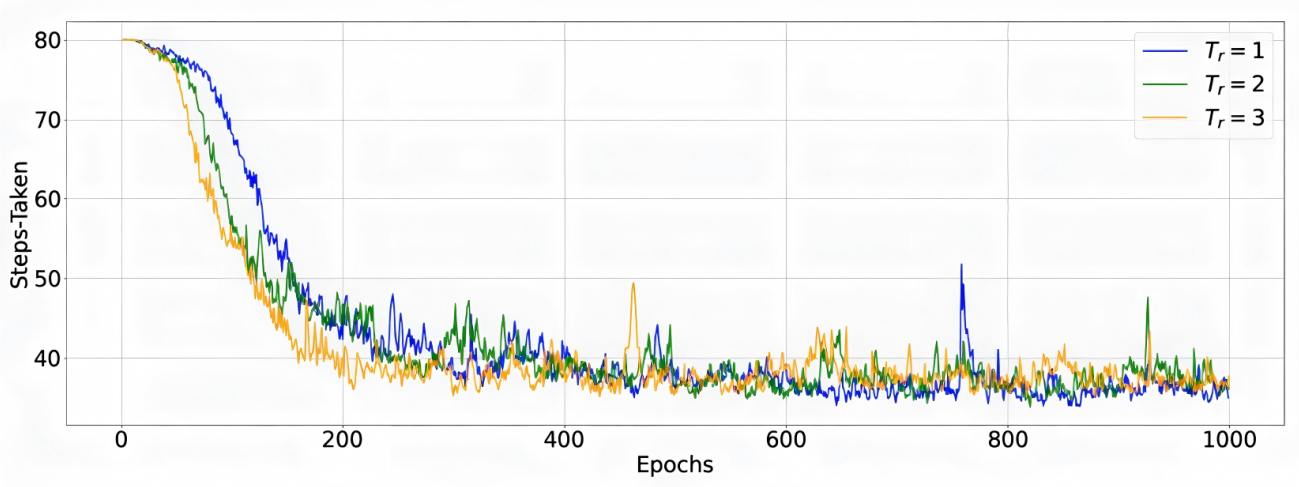

Figure 10. Sensitivity to T_r on the Hard task. Gains saturate beyond $T_r = 2$.

As illustrated in Figure 10, increasing T_r yields diminishing returns beyond $T_r = 2$. Table 9 shows that increasing T_r from 1 to 2 notably improves convergence speed (episode 620→510) and reduces variance, while the marginal gain from $T_r = 2$ to $T_r = 3$ is only 0.3%, confirming $T_r = 2$ as the optimal trade-off between performance and memory.

Table 9. Effect of T_r on DADC (Hard task).

T_r	Steps	Conv. Ep.	Std Dev
1	40.87	≈ 620	3.94
2	38.02	≈ 510	2.88
3	37.91	≈ 480	2.76

7.7. Scalability analysis

Table 10 evaluates DADC against top baselines for $N \in \{5, 7, 10, 15, 20\}$. DADC's advantage over MAGIC grows monotonically from 6.9% ($N = 7$) to 23.0% ($N = 20$), consistent with the known brittleness of graph-attention methods under message reordering [13]. Figure 11 further visualizes this trend, confirming that DADC's advantage widens with constellation size. From the $O(N^2 d_m)$ per-step complexity, $N = 100$ requires 25× the compute of $N = 20$, remaining feasible for clusters of up to ~50 satellites on modern space-grade processors (~10 GFLOPS); larger constellations motivate the hierarchical extension discussed in Section 8.

Table 10. Scalability results. $N \in \{15, 20\}$ retrained with 1500/2000 max episodes.

N	Grid	DADC	MAGIC	TarMAC	DACOM
5	10×10	**13.32**	19.67	18.62	15.47
7	15×15	**23.92**	25.70	32.24	24.43
10	20×20	**38.02**	42.35	43.68	41.28
15	30×30	**61.47**	74.83	79.21	69.14
20	40×40	**91.34**	118.62	131.07	105.83

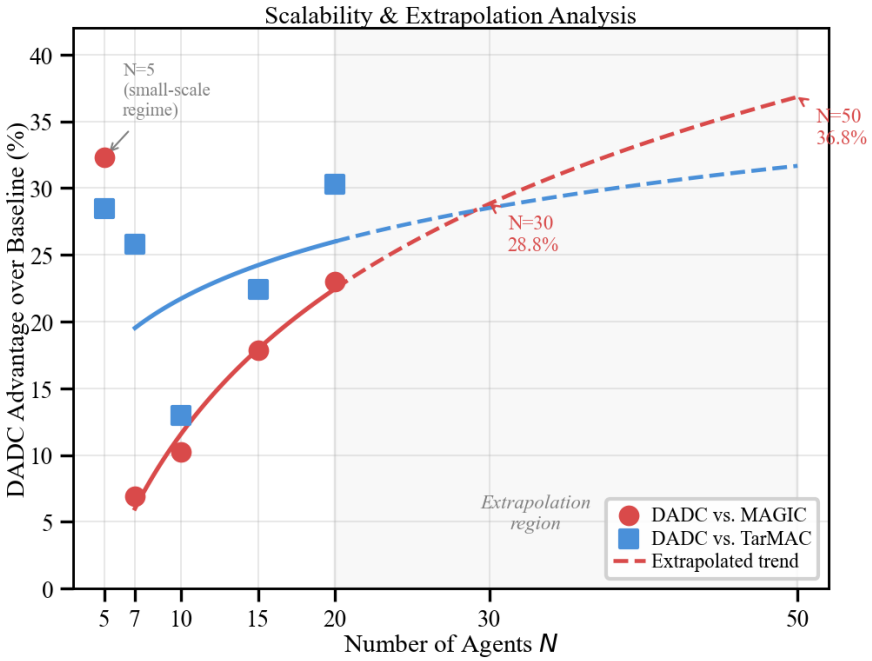

Figure 11. DADC advantage over MAGIC and TarMAC-IC3Net vs. agent count N. Solid: logarithmic fit (N = 7–20); dashed: extrapolation.

7.8. Protocol-level performance on NS3 simulator

Having established DADC's superiority as a delay-aware coordination mechanism in Tier 1, we now validate that these gains propagate to protocol-level communication KPIs when DADC is deployed as the decision layer of the full IPCP stack on an NS3-based LEO simulator. As shown in Table 11, replacing the IEDQN-only decision layer with IEDQN + DADC yields +17.9% throughput and −14.4% E2E delay, directly attributable to DADC's delay-compensated coordination, a result quantitatively consistent with the 7.7% step-reduction observed in the Tier 1 ablation (Table 8, last row).

Figure 12 plots throughput and end-to-end delay over the full 60-minute simulation. IPCP Full reaches within 5% of converged throughput by $t \approx 15$ min (1.5× faster than SDN-Adaptive), sustains the smallest handover dip (≈ 0.7 Mbps vs. ≈ 1.4 Mbps for SDN-Adaptive), and achieves the lowest steady-state variance (± 1.4 Mbps), attributable to the topology tracker proactively anticipating link outages.

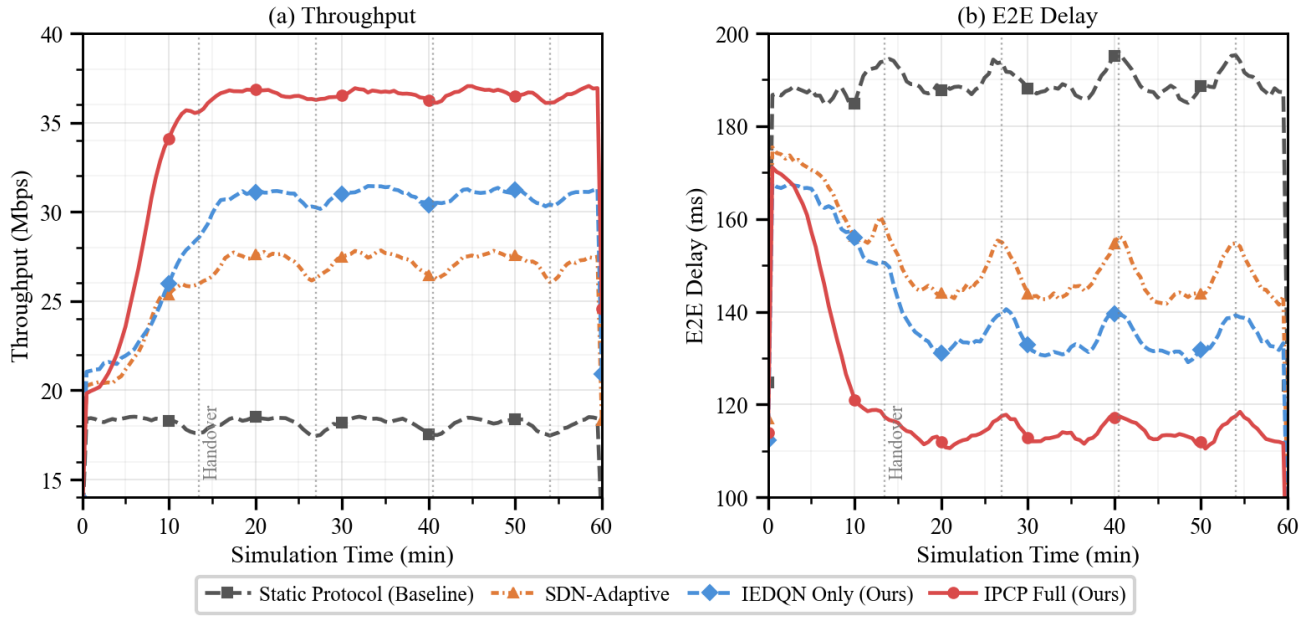

Figure 12. Dynamic (a) throughput and (b) E2E delay over 60-min LEO simulation ($N = 12$). Dotted lines: handover events ($\approx$ 13.5 min interval).

Table 11 reports converged metrics averaged over the final 20 min and 5 runs. IPCP Full (IEDQN + DADC) achieves +100% throughput, −39.9% E2E delay, −75.1% packet loss, +99.7% bandwidth utilization, and −49.9% energy-per-bit versus the static baseline, and +12.5% throughput, and −9.4% delay versus DACOM + IPCP Stack.

Table 11. Converged protocol-level performance (Walker, 550 km, $N = 12$, 60-min simulation, mean ± std over 5 runs). MAGIC and DACOM replace only the decision layer. ↑ higher is better; ↓ lower is better.

Protocol	Throughput (Mbps) ↑	E2E Delay (ms) ↓	Pkt Loss (%) ↓	BW Util. %) ↑	E/bit (μJ) ↓
Static baseline	18.4 ± 1.2	187.3 ± 14.6	8.73	38.2	4.81
SDN-Adaptive [24]	27.6 ± 2.1	143.8 ± 11.3	5.21	57.4	3.42
MAGIC + IPCP [16]	28.3 ± 2.4	138.2 ± 12.1	4.93	59.1	3.28
DACOM + IPCP [13]	32.7 ± 1.9	124.3 ± 10.4	3.41	67.8	2.74
IEDQN only (w/o DADC)	31.2 ± 1.8	131.5 ± 9.7	3.84	64.7	2.97
IPCP full (Ours)	**36.8 ± 1.4**	**112.6 ± 8.2**	**2.17**	**76.3**	**2.41**

† Replacing IEDQN with DADC yields +17.9% throughput and −14.4% E2E delay over IEDQN-only, consistent with the 7.7% step-reduction in the Tier 1 coordination ablation (Table 8, last row).

7.9. Sensitivity analysis of reward weights

To validate robustness to reward weight selection, we fix $w_4 = 0.1$ and vary (w_1, w_2, w_3) subject to $\sum_{k=1}^{K} w_k = 1$, evaluating six representative configurations on the NS3 simulator (5 runs, 60 min). Results are reported in Table 12.

Table 12. Reward weight sensitivity analysis (Walker, 550 km, N = 12).

w_1	w_2	w_3	Throughput (Mbps)↑	E2E Delay (ms)↓	Pkt Loss (%)↓
0.6	0.2	0.1	34.1 ± 1.8	119.3 ± 9.4	2.63
0.5	0.3	0.1	35.2 ± 1.6	115.8 ± 8.9	2.41
0.4	**0.3**	**0.2**	**36.8 ± 1.4**	**112.6 ± 8.2**	**2.17**
0.3	0.4	0.2	35.7 ± 1.7	114.2 ± 8.6	2.29
0.3	0.3	0.3	34.9 ± 1.9	116.5 ± 9.1	2.38
0.2	0.5	0.2	33.6 ± 2.1	121.4 ± 10.2	2.71

Across all configurations, throughput varies within ± 4.9% and E2E delay within ± 7.8 ms relative to the PSO-optimized setting (bold row), confirming that IPCP is robust to moderate perturbations in the reward coefficients.

8. Conclusions

This paper presented IPCP, a unified MARL-driven protocol for LEO satellite constellation communication that simultaneously addresses channel dynamics adaptation, multi-objective resource optimization, and cooperative decision-making under stochastic multi-hop delay.

IEDQN tackles multi-agent non-stationarity via a shared Info-Block, a receiver-specific Mess-Net, and a shared Q-Net on augmented near-fully-observable states, with formal convergence to an ϵ-approximate Nash equilibrium guaranteed under mild Lipschitz conditions. DADC addresses stochastic delay through a Transformer-based DAII encoder for permutation-invariant message fusion and a dual-attention DAIF module; the performance gap to the oracle zero-delay policy decreases exponentially with reception window T_r. On the NS3 Walker constellation simulator (12,550 km), IPCP Full achieves +100% throughput, −39.9% E2E latency, −75.1% packet loss, and −49.9% energy-per-bit over the static baseline, while outperforming all MARL baselines across $N = 5$–20 agents with a scalability advantage that widens with constellation size.

Several directions remain open. Hardware validation: Future work should test IPCP on SDR-based hardware-in-the-loop testbeds with STK-generated ephemeris data. Mega-constellation scaling: Hierarchical MARL architectures, local DADC groups with a higher-level IEDQN inter-group module, provide a natural path toward 3000+ satellite deployments. Adversarial robustness: Minimax or certified-defence RL formulations are needed for contested environments subject to jamming or state-injection attacks. Online adaptation: Meta-learning could enable rapid policy fine-tuning after satellite failure or orbital maneuver without full retraining. Link failures and graceful degradation: Onboard exception handlers must maintain service continuity when ISLs fail mid-episode, requiring fault-tolerant extensions to the IEDQN policy update rule. Asynchronous learning updates: Satellites with heterogeneous onboard processors will complete gradient steps at different rates, violating the synchronous update assumption in the current training procedure; asynchronous MARL methods provide a natural remedy. Radiation-

induced computation errors: Space-grade processors are susceptible to single-event upsets that corrupt model weights or activations, motivating the study of radiation-hardened inference techniques or periodic weight-checkpointing strategies. Sensing uncertainty: The current perception layer assumes reliable SINR estimation; in practice, pilot contamination and fast fading may degrade observation quality, motivating robust state-estimation modules at the PL–DL interface.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. I. Del Portillo, B. G. Cameron, E. F. Crawley, A technical comparison of three low earth orbit satellite constellation systems to provide global broadband, *Acta Astronaut.*, **159** (2019), 123–135. https://doi.org/10.1016/j.actaastro.2019.03.040
2. D. Bhattacherjee, A. Singla, Network topology design at 27,000 km/hour, in *CoNEXT '19: Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, (2019), 341–354. https://doi.org/10.1145/3359989.3365407
3. O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, et al., Satellite communications in the new space era: A survey and future challenges, *IEEE Commun. Surv. Tutorials*, **23** (2021), 70–109. https://doi.org/10.1109/COMST.2020.3028247
4. X. You, C. X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, et al., Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts, *Sci. China Inf. Sci.*, **64** (2021), 110301. https://doi.org/10.1007/s11432-020-2955-6
5. 3GPP, Solutions for NR to Support Non-Terrestrial Networks (NTN), 3rd Generation Partnership Project, 2020. Available from: https://www.3gpp.org/ftp/Specs/archive/38_series/38.821/.
6. O. Liberg, S. E. Löwenmark, S. Euler, B. Hofström, T. Khan, X. Lin, et al., Narrowband Internet of Things for non-terrestrial networks, *IEEE Commun. Stand. Mag.*, **4** (2020), 49–55. https://doi.org/10.1109/MCOMSTD.001.2000004
7. X. Wang, W. Shen, C. Xing, J. An, N. Zhao, L. Hanzo, Joint Bayesian channel estimation and data detection for OTFS systems in LEO satellite communications, *IEEE Trans. Commun.*, **70** (2022), 4386–4399. https://ieeexplore.ieee.org/document/9785832
8. M. Handley, Delay is not an option: Low latency routing in space, in *HotNets '18: Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, (2018), 85–91. https://doi.org/10.1145/3286062.3286075
9. L. Zhang, R. Du, Z. Hao, S. Li, Z. Hu, Dynamic packet routing algorithm based on multidimensional information and multiagent reinforcement learning, *Int. J. Commun. Syst.*, **38** (2025), e70039. https://doi.org/10.1002/dac.70039

10. S. Zhou, Y. Jia, R. Mao, Z. Nan, Y. Sun, Z. Niu, Task-oriented wireless communications for collaborative perception in intelligent unmanned systems, *IEEE Netw.*, **38** (2024), 21–28. https://doi.org/10.1109/MNET.2024.3414144

11. X. Ma, H. Zhang, P. Yan, Z. Wu, Y. Ren, Resource allocation optimization in direct satellite-to-device networks based on multi-agent reinforcement learning, *IEEE Commun. Mag.*, **63** (2025), 61–67. https://doi.org/10.1109/MCOM.001.2500178

12. Y. Wang, K. Gui, Task offloading and computational scheduling in RIS-assisted low Earth orbit satellite communication networks, *Veh. Commun.*, **53** (2025), 100917. https://doi.org/10.1016/j.vehcom.2025.100917

13. T. Yuan, H. M. Chung, J. Yuan, X. Fu, DACOM: Learning delay-aware communication for multi-agent reinforcement learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **37** (2023), 11763–11771. https://doi.org/10.1609/aaai.v37i10.26389

14. S. Sukhbaatar, A. Szlam, R. Fergus, Learning multiagent communication with backpropagation, *Adv. Neural Inf. Process. Syst.*, **29** (2016), 2244–2252.

15. A. Singh, T. Jain, S. Sukhbaatar, Learning when to communicate at scale in multiagent cooperative and competitive tasks, preprint, arXiv:1812.09755.

16. Y. Niu, R. Paleja, M. Gombolay, Multi-agent graph-attention communication and teaming, in *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, (2021), 964–973. https://doi.org/10.5555/3463952.3464065

17. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al., Human-level control through deep reinforcement learning, *Nature*, **518** (2015), 529–533. https://doi.org/10.1038/nature14236

18. J. Huang, Y. Yang, G. He, Y. Xiao, J. Liu, Deep reinforcement learning-based dynamic spectrum access for D2D communication underlay cellular networks, *IEEE Commun. Lett.*, **25** (2021), 2614–2618. https://doi.org/10.1109/LCOMM.2021.3079920

19. R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, *Adv. Neural Inf. Process. Syst.*, **30** (2017), 6379–6390.

20. T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, S. Whiteson, Monotonic value function factorisation for deep multi-agent reinforcement learning, preprint, arXiv:1803.11485.

21. C. Zhu, M. Dastani, S. Wang, A survey of multi-agent deep reinforcement learning with communication, *Auton. Agents Multi-Agent Syst.*, **38** (2024), 4. https://doi.org/10.1007/s10458-023-09633-6

22. J. Sheng, X. Wang, B. Jin, J. Yan, W. Li, T. H. Chang, et al., Learning structured communication for multi-agent reinforcement learning, *Auton. Agents Multi-Agent Syst.*, **36** (2022), 50. https://doi.org/10.1007/s10458-022-09580-8

23. M. Gallici, M. Martin, I. Masmitja, TransfQMix: Transformers for leveraging the graph structure of multi-agent reinforcement learning problems, preprint, arXiv:2301.05334.

24. G. Zheng, N. Wang, R. R. Tafazolli, SDN in space: A virtual data-plane addressing scheme for supporting LEO satellite and terrestrial networks integration, *IEEE/ACM Trans. Netw.*, **32** (2024), 1781–1796. https://doi.org/10.1109/TNET.2023.3330672

25. C. Zhu, X. Zhu, T. Qin, Joint trajectory and incentive optimization for privacy-preserving UAV crowdsensing via multi-agent federated reinforcement learning, *Internet Things*, **33** (2025), 101689. https://doi.org/10.1016/j.iot.2025.101689

26. S. Li, Q. Wu, R. Wang, Dynamic discrete topology design and routing for satellite-terrestrial integrated networks, *IEEE/ACM Trans. Netw.*, **32** (2024), 3840–3853. https://doi.org/10.1109/TNET.2024.3397613

27. S. Li, Q. Wu, R. Wang, L. Chen, H. Zhang, Efficient multipath differential routing and traffic scheduling in ultra-dense LEO satellite networks: A DRL with Stackelberg game approach, *IEEE Trans. Mobile Comput.*, **24** (2025), 12424–12440. https://doi.org/10.1109/TMC.2025.3586262

28. S. Li, G. Wu, Q. Wu, R. Wang, H. Zhang, Efficient packet routing for large-scale LEO satellite networks: A Pareto-optimal MARL approach with queueing theory, *IEEE Internet Things J.*, **12** (2025), 46675–46691. https://doi.org/10.1109/JIOT.2025.3610772

29. S. Li, Q. Wu, R. Wang, Efficient packet routing in ultra-dense LEO satellite networks via cooperative-MARL with queuing theory model, in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, (2025), 1–6. https://doi.org/10.1109/WCNC61545.2025.10978428

30. S. Li, Q. Wu, R. Wang, Toward networking and routing in 6G satellite-terrestrial integrated networks: Current issues and a potential solution, *IEEE Commun. Mag.*, **63** (2025), 92–98. https://doi.org/10.1109/MCOM.001.2400197

31. A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, et al., TarMAC: Targeted multi-agent communication, in *Proceedings of the 36th International Conference on Machine Learning*, (2019), 1538–1546.

32. Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, Y. Gao, Multi-agent game abstraction via graph attention neural network, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 7211–7218. https://doi.org/10.1609/aaai.v34i05.6211

33. P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, et al., Value-decomposition networks for cooperative multi-agent learning, preprint, arXiv:1706.05296.

34. K. Son, D. Kim, W. J. Kang, D. E. Hostallero, Y. Yi, QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning, in *Proceedings of the 36th International Conference on Machine Learning*, (2019), 5887–5896.

35. X. Hao, H. Mao, W. Wang, Y. Yang, D. Li, Y. Zheng, et al., Breaking the curse of dimensionality in multiagent state space: A unified agent permutation framework, preprint, arXiv:2203.05285.

36. E. Fard, R. R. Selmic, Time-delayed data transmission in heterogeneous multi-agent deep reinforcement learning system, in *2022 30th Mediterranean Conference on Control and Automation (MED)*, (2022), 636–642. doilinkhttps://doi.org/10.1109/MED54222.2022.9837194

37. Y. Xia, Y. Xu, Y. Wang, S. Mondal, S. Dasgupta, A. K. Gupta, Optimal secondary control of islanded AC microgrids with communication time-delay based on multi-agent deep reinforcement learning, *CSEE J. Power Energy Syst.*, **9** (2022), 1301–1311. doilinkhttps://doi.org/10.17775/CSEEJPES.2021.08680

38. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, (2017), 5998–6008.

39. J. Tsitsiklis, B. Van Roy, Analysis of temporal-difference learning with function approximation, in *Advances in Neural Information Processing Systems 9 (NIPS 1996)*, 1996.

40. V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, Cambridge University Press, Cambridge, UK, 2008.

41. ARM Holdings, Cortex-A53 Processor, ARM Developer Documentation, 2012. Available from: https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a53.

Appendix

Appendix A: Proof of Theorem 5.1: Convergence of IEDQN

We establish almost-sure convergence of IEDQN to an ϵ-approximate Nash equilibrium. We impose four assumptions: **A1** (Lipschitz Mess-Net) $\|f_i(\mathbf{I}) - f_i(\mathbf{I}')\| \le L\|\mathbf{I} - \mathbf{I}'\|$; **A2** (Sufficient Exploration) $\varepsilon_t > 0$ and $\sum_t \varepsilon_t = \infty$; **A3** (Robbins–Monro) $\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$; **A4** (Bounded Rewards) $|r_i^t| \le r_{\max} < \infty$.

Proof. The effective Bellman operator for agent i in IEDQN is

$$(\widetilde{\mathcal{T}}^{\pi} Q)(s_i, a_i) = \mathbb{E}\Big[r_i + \gamma \max_{a'} Q(s_i', a')\Big], \tag{A.1}$$

where $s_i' = [m_i' \| o_i']$ depends on the non-stationary joint token matrix $\mathbf{I}'$. Let $\sigma_{\mathbf{I}}(t) = \sup_s \|\mathbf{I}_t(s) - \mathbf{I}_t^{\text{stat}}(s)\|_2$ denote the non-stationarity perturbation relative to the stationary policy π^*. By A1, $\|m_i^t - m_i^*\| \le L\sigma_{\mathbf{I}}(t)$, and since Q is L_Q-Lipschitz in the augmented state, the Q-target perturbation satisfies $|y_t - y_t^*| \le \gamma L_Q L \sigma_{\mathbf{I}}(t)$.

The Q-update is a noisy fixed-point iteration $\psi_{t+1} = \psi_t - \alpha_t \nabla_\psi \mathcal{L}(\psi_t) + \alpha_t \xi_t$, where ξ_t is zero-mean with bounded variance (A4). By the two-timescale stochastic approximation theorem [40], under A1–A4, $\{Q_t\}$ converges almost surely to a fixed point Q^* of $\widetilde{\mathcal{T}}^{\pi}$. Iterating the contraction inequality over the $1/(1-\gamma)$ effective horizon and letting $\bar{\sigma}_{\mathbf{I}} = \sup_t \sigma_{\mathbf{I}}(t)$ gives

$$\|Q^* - Q^{\text{Nash}}\|_\infty \le \frac{\gamma L_Q L\, \bar{\sigma}_{\mathbf{I}}}{(1-\gamma)^2}, \tag{A.2}$$

so the Nash gap satisfies $\epsilon \le 2\|Q^* - Q^{\text{Nash}}\|_\infty \le 2\gamma L_Q L\, \bar{\sigma}_{\mathbf{I}}/(1-\gamma)^2$.

Corollary .1 (Effect of coordination quality)**.** *If the Mess-Net is trained to minimize L (e.g., via spectral normalization), the Nash gap decreases proportionally. As $L \to 0$, the bound recovers the standard single-agent linear-FA convergence guarantee [39].*

Appendix B: Complexity analysis

Table B1 summarizes the per-step computational and communication complexities of all methods. IEDQN and DADC both incur $O(N^2 d)$ communication overhead, identical to naive broadcasting, while achieving consistently superior coordination. The DADC communication buffer requires $O(NT_r d_m)$ additional memory per agent; for $N = 10$, $T_r = 2$, $d_m = 64$, this is 1280 floats, well within the budget of modern CubeSat processors.

Table B1. Per-step complexity (N: agents, d: hidden dim, H: heads, n: sequence length).

Method	Compute	Comm.
CommNet [14]	$O(Nd^2)$	$O(N^2d)$
IC3Net [15]	$O(Nd^2)$	$O(N^2d)$
MAGIC [16]	$O(N^2d)$	$O(N^2d)$
IEDQN (Ours)	$O(N^2d^2)$	$O(N^2d)$
DADC (Ours)	$O(N^2d + n^2Hd)$	$O(N^2d)$

*Concrete inference latency and memory measurements for DADC with $N \in \{5, 7, 10\}$ are provided in Table 3 (Section 6.5).