This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

Spatial Load-Aware and Knowledge-Driven Routing Strategy for Dual-Layer LEO Satellite Networks

Ge Gao, Dengke Wang, *Member, IEEE*, Chen Chen, *Senior Member, IEEE*, Changhong Wang, and Xiaoheng Tan, *Member, IEEE*

***Abstract*—With the rapid proliferation of low Earth orbit (LEO) satellite networks, dynamic topologies and uneven traffic fluctuations pose significant routing challenges. Traditional single-layer LEO routing is highly susceptible to congestion, substantially increasing end-to-end latency. On the other hand, current deep reinforcement learning (DRL) methods frequently suffer from blind exploration in massive state-action spaces, severely hindering convergence. To address these challenges, we introduce spatial load-aware inter-layer collaboration for congestion alleviation by designing a heterogeneous dual-layer LEO satellite network architecture. Under this architecture, we integrate queuing delay models and minimum-cost path logic as a priori heuristic knowledge to guide agent pre-training, thereby effectively overcoming the early-stage exploration predicament of DRL methods under purely data-driven mechanisms. Extensive simulations show that, in the 53° dual-layer constellation, SLAK-R's online training reward stabilizes at approximately 1500 episodes versus 2500 for D3QN without pre-training. At $P_{\text{cong}} = 0.30$, SLAK-R achieves approximately 20% lower average end-to-end latency than D3QN without pre-training.**

***Index Terms*—LEO satellite networks, routing, DRL, dual-layer, spatial load-aware, knowledge-driven.**

I. Introduction

With the development of 6G Space-Air-Ground Integrated Networks (SAGINs) [1] [2], Low Earth Orbit (LEO) satellite constellations have emerged as critical infrastructures for global interconnectivity, owing to their low latency and high throughput [3]. However, the high density and mobility of mega-LEO constellations lead to severely time-varying network topologies, rendering them susceptible to frequent inter-satellite link handovers and localized congestion. Therefore, designing efficient network architectures and routing strategies is essential for guaranteeing reliable satellite transmission [4].

In the evolution of network architectures, traditional single-layer LEO networks rely on excessive inter-satellite hops for long-distance communications, leading to severe queuing delays and an increased risk of congestion. Reference [5] introduced MEO/GEO nodes to construct multi-layer networks that reduce the hop count. However, incorporating higher-orbit satellites incurs a prohibitive propagation delay penalty, failing to meet the ultra-low latency requirements of 6G. Thus, the dual-layer LEO satellite network architecture emerges as a promising paradigm balancing efficiency and latency. By introducing an upper LEO backbone tier at a higher altitude for traffic offloading, this architecture effectively breaks the congestion bottlenecks of single-layer networks while preserving the inherent low-latency attributes of LEO systems.

Regarding routing strategy design, traditional centralized approaches (e.g., SDN-based architectures in [6]) rely heavily on global state collection. This introduces critical limitations in mega-constellations, such as excessive signaling overhead and delayed updates. Conversely, the distributed routing strategies in [7], [8] avoid congestion and reduce modeling complexity without global network snapshots. Their low signaling overhead makes them more suitable for mega-constellations.

Recently, Deep Reinforcement Learning (DRL) has been extensively applied to distributed routing optimization due to its dynamic adaptability [9], [10]. Nevertheless, purely data-driven DRL algorithms face a severe cold-start problem: during the initial training phase, the agent's blind trial-and-error exploration within massive state-action spaces leads to slow convergence, routing loops, and early-stage congestion [11]. Consequently, injecting inherent physical network rules into the DRL framework as a priori knowledge to construct a "Knowledge-Driven" mechanism constitutes an effective approach to elevate routing performance, as it achieves both convergence efficiency and dynamic adaptability. Table I includes representative routing studies and highlights the architectural and methodological distinctions of SLAK-R.

To address the aforementioned challenges, this paper proposes a comprehensive Spatial Load-Aware routing scheme that introduces innovations at both the constellation architecture and algorithmic logic levels. The primary contributions of this paper are summarized as follows:

(1) We design a novel dual-layer LEO satellite network architecture, aimed at overcoming the severe congestion inherent in single-layer networks through spatial load-aware inter-layer collaboration, and avoiding the high propagation delays of hybrid LEO/MEO/GEO architectures.

(2) We propose a Spatial Load-Aware and Knowledge-Driven Routing Strategy, named SLAK-R, to proactively bypass congested satellite nodes and surmount the blind exploration predicament that purely data-driven DRL algorithms suffer during the initial training phase.

(3) We conduct extensive ablation studies and comparative simulations, validating the superiority of the proposed method in both convergence efficiency and end-to-end latency.

This work was supported by the National Natural Science Foundation of China under Grant 62471074. (*Corresponding author: Dengke Wang.*)

Ge Gao, Dengke Wang, Chen Chen, and Xiaoheng Tan are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: gaoge@stu.cqu.edu.cn; dengke.wang@cqu.edu.cn).

Changhong Wang is with Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd.., Jinan 250101, China (e-mail: wangchh01@inspur.com).

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

TABLE I
COMPARISON OF REPRESENTATIVE ROUTING STUDIES AND SLAK-R

Reference	Network/architecture	Routing/learning method	Information/knowledge use	Difference from SLAK-R
[5]	GEO/MEO/LEO with terrestrial gateways	Hierarchical DDQN: GEO budgeting and MEO routing	Geometry, blockage, hop budget, angle, and SNR information	Higher-orbit coordination rather than dual-layer LEO offloading
[6]	GEO-controlled software-defined LEO	Centralized FC–LSTM DQN with k-shortest-path actions	Global network states; DPBRS reward shaping and incremental training	Different control scope and knowledge-injection mechanism
[7]	Mega-LEO with ground relays	Distributed minimum-hop and queue-aware heuristic	Queue lengths, queueing delay, and queue-space thresholds	Model-driven satellite–terrestrial routing without learned adaptation
[8]	LEO direct-to-device	Path-informed distributed collaborative DRL	Link/queue states, geography, and path-level reward reallocation	Different D2D scenario and path-based reward reallocation
[9]	Fully distributed LEO	GAT-enhanced MADRL with RSPH guidance	One-hop neighbor states, queue states, and shortest-path-hop guidance	RSPH-based reward rather than expert-transition pre-training
[10]	Satellite–terrestrial	Constrained CTDE MARL	Local observations and global critics; energy and packet-loss constraints	Different architecture and constraint formulation
SLAK-R (this work)	**Dual-layer LEO supporting inter-layer traffic offloading**	**Shared D3QN with expert-transition pre-training followed by online fine-tuning**	**One-hop neighbor load ratios and relative geometry; no network-wide state observations during routing**	**Joint design of dual-layer LEO offloading, load awareness, and expert-guided training**

II. SYSTEM MODEL AND PROBLEM FORMULATION

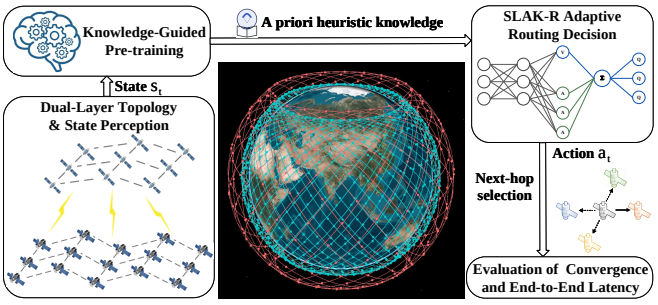


Fig. 1. Overall architecture and workflow of SLAK-R. The dense lower LEO layer L_1 supports low-latency access, while the sparse upper layer L_2 serves as a spatial backbone. Intra-layer and inter-layer links provide connectivity. Expert-guided D3QN pre-training is followed by online fine-tuning and hop-by-hop next-hop selection from active neighbors. Evaluation covers convergence, end-to-end latency, and congestion robustness.

A. System Architecture

In this section, we construct a dual-layer LEO satellite network architecture as shown in Fig. 1 to overcome the performance bottlenecks of traditional single-layer constellations under dynamic traffic loads through 3D topological stratification and spatial traffic offloading. Based on the virtual topology concept, we abstract this network architecture as a time-evolving undirected graph, denoted by $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V}$ represents the satellite nodes, and $\mathcal{E}(t)$ denotes the active inter-satellite links (ISLs) at time t.

The node set, $\mathcal{V} = \mathcal{V}_{L1} \cup \mathcal{V}_{L2}$, comprises two components deployed at varying orbital altitudes: a lower-layer LEO constellation (denoted by $L1$) at altitude H_{L1} and an upper-layer LEO constellation (denoted by $L2$) at altitude H_{L2}, where $H_{L1} < H_{L2}$. Let P_k and S_k denote the number of orbital planes and the number of satellites per plane in layer $k \in \{L1, L2\}$, respectively. Consequently, the total number of satellites in layer k is given by $N_k = P_k \times S_k$, yielding a macroscopic network scale of $N = N_{L1} + N_{L2}$ nodes. Specifically, the $L1$ nodes, $\mathcal{V}_{L1}$, are densely deployed at a lower orbital altitude and are primarily responsible for facilitating low-latency direct access and fine-grained coverage for massive terrestrial terminals. Conversely, the $L2$ nodes, $\mathcal{V}_{L2}$, are relatively sparsely distributed at a higher orbital altitude, endowing each satellite with a broader coverage footprint. For long-distance traffic, the $L2$ tier substantially reduces the end-to-end hop count, thereby mitigating cumulative processing and queuing delays. Furthermore, the set of ISLs, $\mathcal{E}(t) = \mathcal{E}_{intra}(t) \cup \mathcal{E}_{inter}(t)$, encompasses intra-layer and inter-layer links. Intra-layer links $\mathcal{E}_{intra}(t)$ connect adjacent satellites within and across orbital planes, establishing a highly structured grid topology within each tier for fundamental connectivity. Inter-layer links $\mathcal{E}_{inter}(t)$ are dynamically instantiated based on 3D geometrical visibility. Crucially, these inter-layer links provide the physical pathways for spatial load-aware collaboration, enabling congested $L1$ nodes to offload traffic to the $L2$ tier, requiring an $L2$ satellite to be within the line-of-sight and maximum communication range of an $L1$ satellite. L_2 provides longer-span paths around localized L_1 congestion without MEO/GEO relays, potentially reducing cumulative queueing and processing delays. These benefits require additional satellite and link resources and can incur inter-layer traversal delay; offloading is therefore optional.

B. Traffic and Queuing Model

To characterize the spatial traffic heterogeneity caused by non-uniform terrestrial users [12], this section constructs a geography-aware non-uniform traffic model and an M/M/1 queuing approximation.

1) Regional Hotspots and Load Modeling: Based on the current sub-satellite point coordinates, we partition the global network nodes into a high-load congestion zone $\mathcal{Z}_{high}$ and a low-load background zone $\mathcal{Z}_{low}$. Let $\mathcal{R}_{high}$ denote the regions with dense terrestrial traffic. Denote the latitude and longitude of node v by Lat_v and Lon_v, respectively. The set of satellites in the high-load congestion zone is defined as:

$$\mathcal{Z}_{high} = \{v \in \mathcal{V} \mid (Lat_v, Lon_v) \in \mathcal{R}_{high}\}. \tag{1}$$

Correspondingly, the low-load background zone is $\mathcal{Z}_{low} = \mathcal{V} \backslash \mathcal{Z}_{high}$. Furthermore, let $\rho_v \in [0, 1)$ represent the normalized

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

traffic load ratio of node v, which is defined as the ratio of the packet arrival rate λ_v to the physical service rate μ_v at the current time slot.

In the traffic generation model, satellite node loads are strictly governed by their geographical regions. Specifically, we define ρ_{cong} as the congestion load threshold. Nodes in $\mathcal{Z}_{high}$ are configured with higher base traffic and a greater bursty congestion probability P_{cong} to generate congested nodes whose ρ_v exceeds ρ_{cong}. Conversely, nodes in $\mathcal{Z}_{low}$ maintain low, stable base traffic with P_{cong} approaching 0. This geography-aware configuration dynamically generates time-varying localized hotspots within the network.

2) M/M/1 Queuing and Congestion Penalty: As traffic aggregates, ρ_v at localized hotspots approaches 1. To quantify its impact on end-to-end performance, we employ an M/M/1 queuing model to characterize the buffer waiting behavior of node v. Let $\rho_{\text{th}} \in (0,1)$ denote the load safety threshold, we formulate a piecewise queuing cost function T^v_{queue}:

$$T^v_{queue} = \begin{cases} T^{(u,v)}_{trans} \frac{\rho_v}{1-\rho_v}, & \text{if } 0 \le \rho_v < \rho_{th}, \\ \mathcal{P}_{drop}, & \text{if } \rho_v \ge \rho_{th}, \end{cases} \tag{2}$$

where $\rho_v = \lambda_v/\mu_v$ is the effective node-level utilization and $\mathcal{P}_{\text{drop}}$ is the constant packet-loss penalty. The link-scaled equivalent queueing approximation T^v_{queue} uses ρ_v for congestion amplification and $T^{(u,v)}_{\text{trans}}$, defined in Sec. II-C, for the hop-specific service-time scale. This model smoothly scales queuing costs under light loads while imposing a hard-cutoff penalty during severe congestion, compelling the routing algorithm to proactively avoid congestion.

C. Delay Model

Let $T_{u,v}$ denote the total latency incurred by a data packet traversing a single hop over link (u,v). This per-hop latency consists of four components and is formulated as:

$$T_{u,v} = T^{(u,v)}_{prop} + T^{(u,v)}_{trans} + T^v_{proc} + T^v_{queue}, \tag{3}$$

where the constituent delay components are defined as follows. Specifically, the propagation delay is $T^{(u,v)}_{prop} = D_{u,v}/c$, where $D_{u,v}$ is the Euclidean distance of link (u,v) and c is the speed of light in a vacuum. The transmission delay is $T^{(u,v)}_{trans} = L/B_{u,v}$, with L being the packet size and $B_{u,v}$ the link bandwidth. Additionally, the processing delay T^v_{proc} for packet handling at the node is modeled as a hardware-dependent constant t_{proc}. Finally, the queuing delay T^v_{queue}, which represents the buffer waiting time, is approximated by Eq. (2) based on the M/M/1 model.

D. Problem Formulation

This paper aims to determine optimal routing paths in dual-layer LEO satellite networks, minimizing end-to-end latency while maintaining low congestion and packet loss rates. Let the binary matrix $\mathbf{X} = \{x_{u,v}\}_{N\times N}$ denote the routing decision from source s to destination d, where $x_{u,v} \in \{0,1\}$ is a binary decision variable, and $x_{u,v} = 1$ indicates that link (u,v) is selected for the forwarding path. The objective function is formulated as:

$$\begin{aligned} &\min_{\mathbf{X}} \sum_{(u,v)\in\mathcal{E}} x_{u,v} T_{u,v} \\ \text{s.t.}\quad & C1: \sum_{\{v|(u,v)\in\mathcal{E}\}} x_{u,v} - \sum_{\{v|(v,u)\in\mathcal{E}\}} x_{v,u} \\ &\qquad = \begin{cases} 1, & u = s \\ -1, & u = d \\ 0, & \text{otherwise} \end{cases}, \quad \forall u \in \mathcal{V}, \\ & C2: \sum_{\{v|(u,v)\in\mathcal{E}\}} x_{u,v} \le 1, \quad \forall u \in \mathcal{V}\setminus\{d\}, \\ & C3: x_{u,v}\cdot\rho_v < \rho_{th}, \quad \forall (u,v)\in\mathcal{E}, \\ & C4: x_{u,v} \in \{0,1\}, \quad \forall (u,v)\in\mathcal{E}. \end{aligned} \tag{4}$$

$C1$ enforces flow conservation, while $C2$ alone does not exclude cycles. Removing a cycle preserves feasibility and, with strictly positive link costs, reduces the objective; thus an optimal route is loop-free. $C3$ restricts selected next-hop loads below ρ_{th}, and $C4$ imposes binary decisions. Eq. (4) assumes a quasi-static topology and fixed background loads within each packet-routing episode. Constraints $C1$–$C4$ describe within-snapshot feasibility, without modeling inter-slot queue evolution or the packet's effect on subsequent background loads.

III. PROPOSED SLAK-R ALGORITHM

A. MDP Formulation for Routing

Within the DRL framework, this section models the hop-by-hop routing process as a Markov Decision Process (MDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. Satellite nodes act as agents, executing forwarding actions based on local observations and receiving rewards from the dynamic environment. t indexes forwarding hops rather than physical time slots. For a successful nonterminal hop, $v_c^{(t+1)} = v_{\text{next}}^{(t)}$, making the next local state and candidate set $\mathcal{A}(s_{t+1})$ depend on a_t even within a fixed background snapshot.

State Space $\mathcal{S}$: When a packet arrives at forwarding node v_c at step t, state $s_t \in \mathcal{S}$ is defined as:

$$s_t = \big[\mathbf{F}(v_c), \mathbf{F}(v_d), \mathbf{L}\big(\mathcal{N}(v_c)\big)\big], \tag{5}$$

where $\mathbf{F}(v_c)$ and $\mathbf{F}(v_d)$ represent the 3D physical coordinates (or topological indices) of the current node v_c and destination v_d, respectively, providing the agent with a directional vector for routing. Furthermore, $\mathbf{L}\big(\mathcal{N}(v_c)\big)$ denotes the real-time load features of v_c's one-hop neighbors $\mathcal{N}(v_c)$, which is strictly represented by the normalized load ratio ρ_v in this work. During online routing, the spatial load-aware policy uses one-hop neighbor load ratios and relative geometry, without network-wide load observations.

Action Space $\mathcal{A}$: For the current node v_c, action $a_t \in \mathcal{A}$ is the selection of a next-hop relay from its candidate neighbors, given by:

$$\mathcal{A}(s_t) = \big\{ v_{next} \mid v_{next} \in \mathcal{N}_{intra}(v_c) \cup \mathcal{N}_{inter}(v_c) \big\}, \tag{6}$$

where $\mathcal{N}_{intra}$ and $\mathcal{N}_{inter}$ denote the neighbors connected via intra-layer and inter-layer ISLs, respectively. At each

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

hop, active links define $\mathcal{A}(s_t)$, masking unavailable actions during selection and TD-target maximization. All satellites share online and target D3QN parameters; each network outputs one scalar Q-value from four candidate features: normalized destination distance, directional alignment, layer indicator, and node load ratio. Neighbor changes alter only the evaluation count, not input/output dimensions. Inter-layer forwarding needs no separate layer-selection action. Scoring costs $O(d_t C_Q)$ per hop, where $d_t = |\mathcal{A}(s_t)|$ and C_Q is the fixed per-candidate cost.

Reward Function $\mathcal{R}$: At each decision step t, the satellite environment evaluates the action a_t taken in state s_t according to the reward function, and returns an immediate scalar reward $r_t = \mathcal{R}(s_t, a_t)$ to the routing agent. Based on the delay components defined in Eq. (3), the composite reward function $\mathcal{R}(s_t, a_t)$ is formulated as:

$$\mathcal{R}(s_t,a_t)=\begin{cases} R_{success}, & \text{if } v_{next}=v_d,\\ -\mathcal{P}_{drop}, & \text{if } \rho_{v_{next}}\geq\rho_{th},\\ -T^e_{prop}-T^e_{trans} & \\ \quad -T^{v_{next}}_{proc}-T^{v_{next}}_{queue}, & \text{otherwise,}\end{cases} \tag{7}$$

where $R_{success}$ is a constant positive reward incentivizing the agent to find the shortest converging path, and $\mathcal{P}_{drop}$ is a severe penalty for congestion-induced packet loss.

B. Knowledge-Driven Pre-training

To mitigate blind exploration during early training, SLAK-R uses a separate pre-training stage based on expert routing transitions. Expert preferences enter through training data without adding this cost to the online reward, while subsequent fine-tuning permits adaptation beyond expert paths.

1) Spatial Load-Aware Generalized Cost Function: Based on the M/M/1 model, we formulate a cost metric integrating propagation distance and congestion risks. For link (u, v), this generalized cost is defined as:

$$W_{u,v}=\mathcal{H}(\rho_v)\alpha_i\left(D_{u,v}+\beta\frac{\rho_v}{1-\rho_v+\epsilon}\right), \tag{8}$$

where the term $\rho_v/(1-\rho_v+\epsilon)$ is the queuing congestion penalty, which grows asymptotically as ρ_v approaches 1. ϵ is a small constant preventing a zero denominator, and β is a scaling factor mapping this penalty to the physical distance scale. $\mathcal{H}(\rho_v) = +\infty$ for $\rho_v \geq \rho_{\text{th}}$ and 1 otherwise, excluding unsafe nodes from expert paths. α_i is a hierarchical bias factor that heuristically encourages long-distance traffic offloading onto $\mathcal{V}_{L2}$ for its hop-count advantage, which is set to ω when the next hop $v \in \mathcal{V}_{L2}$ and 1 otherwise.

2) Expert Demonstration and Network Pre-training: We compute the minimum-cost expert path $\mathcal{P}^* = \{n_1, n_2, \ldots, n_k\}$ using the generalized link cost $W_{u,v}$ in Eq. (8). The resulting expert transitions $(s_t, a_t^*, r_t, s_{t+1})$ are stored in $\mathcal{D}$ for TD/Q-value pre-training with loss:

$$\mathcal{L}(\theta)=\mathbb{E}_{(s_t,a_t^*,r_t,s_{t+1})\sim\mathcal{D}}\Big[\Big(r_t+\gamma\max_{a'}Q(s_{t+1},a';\theta^-) - Q(s_t,a_t^*;\theta)\Big)^2\Big]. \tag{9}$$

C. Online Learning and Adaptive Routing

After knowledge-guided offline pre-training, the agent enters online learning to fine-tune its routing policy through interaction with the satellite environment. At each hop, it uses an ϵ-greedy strategy with a decaying exploration rate to balance exploration and exploitation of prior knowledge. After executing a_t, the environment returns a reward r_t based on the M/M/1 queuing status, and the transition (s_t, a_t, r_t, s_{t+1}) is stored in the replay buffer $\mathcal{D}$.

To break temporal data correlations and smooth training oscillations, we sample random mini-batches from $\mathcal{D}$ for network updates. We adopt the Dueling Double Deep Q Network (D3QN) algorithm. By decoupling action selection and evaluation, Double Q-learning mitigates the overestimation bias of traditional DQN. Specifically, to compute the target Q-value Y_t^{online}, the online network θ selects the optimal next-hop action, while the target network θ^- evaluates it:

$$Y_t^{online}=r_t+\gamma Q\Big(s_{t+1},\arg\max_{a'}Q(s_{t+1},a';\theta,\alpha,\beta); \theta^-,\alpha^-,\beta^-\Big). \tag{10}$$

Subsequently, the online network θ is updated by minimizing the loss $\mathcal{L}(\theta) = \mathbb{E}[(Y_t^{online} - Q(s_t, a_t; \theta))^2]$. To improve state evaluation in congested environments, the Dueling architecture decouples the network into a state value stream $V(s_t; \theta, \alpha)$ and an advantage stream $A(s_t, a_t; \theta, \beta)$, aggregating them to compute the final Q-value:

$$Q(s_t,a_t;\theta,\alpha,\beta)=V(s_t;\theta,\alpha)+\Big(A(s_t,a_t;\theta,\beta) - \frac{1}{|\mathcal{A}|}\sum_{a'}A(s_t,a';\theta,\beta)\Big), \tag{11}$$

where $|\mathcal{A}|$ denotes the size of the action space. Benefiting from this decoupling mechanism, when traffic surges in a hotspot and the load ρ approaches the threshold ρ_{th}, the severe penalty $\mathcal{P}_{drop}$ swiftly degrades the state value $V(s_t)$ of the congested node. When evaluating the next hop, neighboring nodes detect the sharp drop in Q-values in that direction, autonomously redirecting traffic via ISLs to the lightly-loaded idle nodes.

The trained shared policy selects next hops using local geometry and neighbor loads. Orbital motion requires link maintenance and neighbor-state updates, whose costs depend on active neighbor count and update frequency and are not eliminated by parameter sharing. Algorithm 1 summarizes the complete SLAK-R execution flow.

Regarding network scale, orbital adjacency gives $|\mathcal{E}_{\text{intra}}(t)| = O(N)$, while $|\mathcal{E}_{\text{inter}}(t)| \leq N_{L1}N_{L2}$. Visibility and maximum communication-range constraints do not ensure size-independent neighbor counts, so denser constellations can increase candidate-scoring and link-maintenance costs. With fixed neural-network dimensions, per-satellite model size remains unchanged, while an H-hop route requires $O(C_Q \sum_{t=0}^{H-1} d_t)$ scoring operations, excluding topology construction, neighbor discovery, and expert-path generation. Additional L_2 satellites can provide more offloading paths, but latency gains depend on connectivity, capacity, and traffic.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

Algorithm 1 SLAK-R Algorithm

Input: Topology graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, pre-training episodes E_{pre}, online episodes E_{online}, safety threshold ρ_{th}.
Output: Learned routing policy and network parameters θ.

Initialize $\mathcal{D}$, θ, and $\theta^- \leftarrow \theta$;
for $episode = 1$ to E_{pre} **do**
  Compute $\mathcal{P}^*$ using $W_{u,v}$ in Eq. (8);
  Store expert transitions $(s_t, a_t^*, r_t, s_{t+1})$ into $\mathcal{D}$;
  Sample a mini-batch from $\mathcal{D}$ and update θ by minimizing $\mathcal{L}(\theta)$ via Eq. (9);
end for
for $episode = 1$ to E_{online} **do**
  Initialize state s_t;
  while s_t is not terminal **do**
    Select action a_t via ϵ-greedy strategy based on $Q(s_t, a; \theta, \alpha, \beta)$;
    Execute a_t; observe new state s_{t+1} and next-hop load $\rho_{v_{next}}$; compute the reward r_t via Eq. (7);
    Store transition (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}$;
    Sample a mini-batch from $\mathcal{D}$ and minimize $\mathcal{L}(\theta)$ using TD targets from Eq. (10);
    Update state: $s_t \leftarrow s_{t+1}$;
  end while
  Soft-synchronize $\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-$; and decay ϵ;
end for
return θ;

IV. SIMULATION RESULTS

The simulation constructs a dual-layer LEO satellite network comprising 1440 satellites. Specifically, $\mathcal{V}_{L1}$ contains 1152 satellites distributed across 48 orbital planes with 24 satellites per plane at an altitude of 550 km, while $\mathcal{V}_{L2}$ contains 288 satellites distributed across 24 orbital planes with 12 satellites per plane at 1150 km, with both sharing a uniform orbital inclination of 53°. Nodal processing delay is set to 0.1 ms, data packet size to 1 MB. The congestion load threshold and load safety threshold are set to $\rho_{\text{cong}} = 0.85$ and $\rho_{\text{th}} = 0.95$, respectively. These thresholds distinguish elevated queueing from near-saturation: $\rho_v/(1-\rho_v)$ is approximately 5.67 at $\rho_v = 0.85$ and approaches 19 as $\rho_v \to 0.95^-$. The safety threshold leaves a nominal 5% utilization margin. High-load congestion zones covering North America and Eurasia are bounded by latitudes $[30°, 60°]$ and longitudes $[-130°, -60°] \cup [-10°, 145°]$. Crucially, both the source and destination satellite nodes are randomly generated exclusively from within these high-load congestion zones. The constellation and hotspot settings define a controlled dense-access/sparse-backbone scenario with uneven traffic; fixed packet size and processing delay standardize per-hop delay calculations. For each inclination, evaluation fixes the orbital topology and independently resamples background loads between episodes, representing traffic snapshots rather than consecutive physical time slots.

The proposed SLAK-R agent is implemented using PyTorch. The experience replay buffer capacity is set to 10^5, with a mini-batch size of 128 and a learning rate of 5×10^{-4}. The discount factor is $\gamma = 0.98$, the soft update coefficient for the target network is $\tau = 0.005$, and ω is set to 0.9. For feasible L_2 candidates, $\omega = 0.9$ applies a 10% discount to the generalized expert cost, providing a soft backbone preference without relaxing the load-safety condition. Pre-training uses $E_{\text{pre}} = 200$ episodes, generating 1,201 expert transitions in approximately 2.97 s on an Intel Core Ultra 7 265K CPU, including expert-path generation and network updates. Finally, the exploration rate ϵ dynamically decays from 1.0 to 0.01, reaching approximately 0.0244 at online episode 1500.

We include a dual-layer adaptation of GraphPR in [9] as an external learned baseline. It uses GAT-based neighbor information exchange, independent routing agents, and RSPH guidance. Queue observations derive from node load ratios; actions include active intra-layer and inter-layer links.

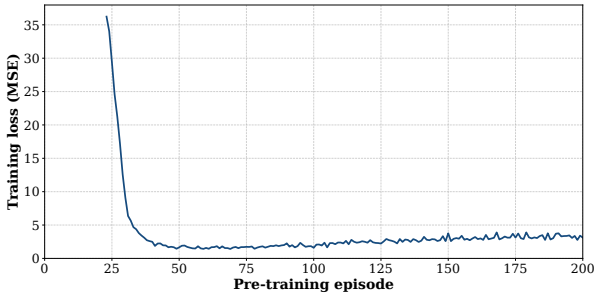


Fig. 2. Pre-training loss over 200 episodes.

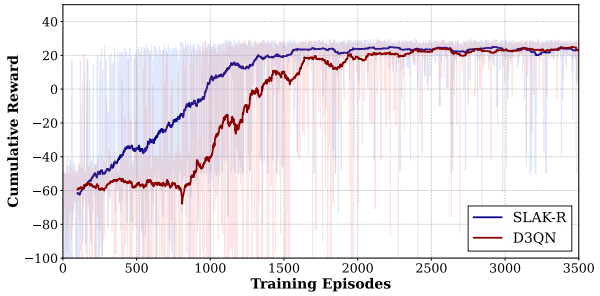


Fig. 3. Convergence of cumulative reward.

Fig. 2 shows the early-stage pre-training loss. Loss recording begins once the replay buffer contains more than 128 transitions. The loss decreases sharply and then fluctuates as replay samples and bootstrap targets evolve. Online fine-tuning uses the parameters obtained after all 200 pre-training episodes. Fig. 3 illustrates the agents' average cumulative reward curves during online training. To evaluate the knowledge-driven mechanism, we introduce D3QN without pre-training as an ablation baseline. Knowledge-guided pre-training reduces SLAK-R's initial blind exploration, exhibiting a smooth and rapid upward trajectory. Conversely, without heuristic guidance, D3QN experiences a cold-start problem, struggling with low-reward trial-and-error exploration during the first 1000 episodes. Ultimately, SLAK-R's online training reward stabilizes at approximately 1500 episodes, whereas the baseline stabilizes around 2500 episodes with pronounced fluctuations. Reward stabilization near online episode 1500 coincides with $\epsilon \approx 0.0244$; higher earlier exploration can delay the manifestation of pre-training benefits in the reward curve. This demonstrates that the knowledge-driven mechanism effectively optimizes agent's initial policy and convergence performance.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2026.3742413

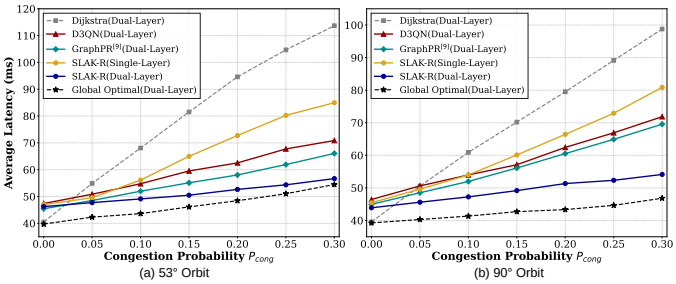


Fig. 4. Comparison of latency performance.

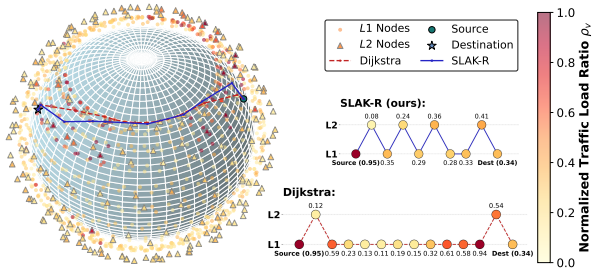


Fig. 5. Visualization of routing paths.

Fig. 4(a) depicts the variation of end-to-end average latency as the bursty congestion probability P_{cong} increases within the high-load zones. Notably, SLAK-R maintains a flat latency curve, closely tracking the Global Optimal baseline, which uses global state information to compute optimal paths, providing a lower-bound reference under the considered model. Even under severe congestion, i.e., $P_{cong} = 0.30$, SLAK-R achieves an average latency of approximately 56.7 ms, compared with 66.1 ms for GraphPR, corresponding to a 14.2% reduction. This result is consistent with its ability to detect increases in local ρ_v and proactively offload traffic to peripheral idle nodes or the wide-coverage L2 tier. In stark contrast, other baseline methods exhibit significant performance degradation as network loads intensify. Under ideal light-load conditions, i.e., as P_{cong} approaches 0, the spatial routing problem fundamentally degenerates into a shortest-path problem, allowing Dijkstra to achieve near-optimal latency. However, at $P_{cong} = 0.30$, Dijkstra's reliance on shortest paths raises delays to nearly 115 ms. Furthermore, to isolate the architecture's impact, we conduct an ablation analysis in a single-layer LEO network comprising only $\mathcal{V}_{L1}$. The single-layer SLAK-R exhibits a latency spike to around 85 ms, exposing the inherent bottleneck of topologies lacking a vertical routing dimension. While the dual-layer architecture provides such paths, exploiting them requires effective routing. Although D3QN shows basic congestion avoidance, lacking pre-training guidance entails additional trial-and-error learning. It tends to adopt suboptimal "conservative detouring" policies, with SLAK-R achieving approximately 20% lower latency at $P_{cong} = 0.30$.

To evaluate cross-inclination transfer, both SLAK-R and GraphPR are trained on the 53° constellation and directly evaluated on the 90° polar constellation without further fine-tuning. The curves in Fig. 4(b) indicate that even facing unobserved topological variations, SLAK-R sustains a low and stable end-to-end latency. Fig. 5 intuitively visualizes the load-aware intelligent routing process, clearly demonstrating the proposed strategy's ability to bypass congested nodes. Furthermore, to evaluate the real-time execution efficiency, we test the per-hop inference time on an Intel Core Ultra 7 265K CPU. Remarkably, the SLAK-R agent executes routing decisions via a single neural network forward pass, requiring an average of only 0.067 ms per inference, whereas the Dijkstra algorithm demands 2.469 ms for global shortest-path recalculation per topology update. This approximately 37-fold acceleration mitigates online computational bottlenecks, proving SLAK-R highly feasible for resource-constrained satellite on-board computers.

V. CONCLUSION

In this paper, we propose SLAK-R, a spatial load-aware and knowledge-driven routing strategy for dual-layer LEO satellite networks. By integrating M/M/1 queuing theory and minimum-cost path logic to pre-train the agent, SLAK-R effectively overcomes the blind exploration predicament in the early stages of training. Under the 53° dual-layer constellation, SLAK-R's online training reward stabilizes at approximately 1500 episodes versus 2500 for D3QN without pre-training. At $P_{\text{cong}} = 0.30$, SLAK-R achieves approximately 20% lower average end-to-end latency than D3QN without pre-training.

REFERENCES

[1] L. Qu, G. Xu, Z. Zeng, N. Zhang, and Q. Zhang, "UAV-assisted RF/FSO relay system for space-air-ground integrated network: A performance analysis," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 6211–6225, 2022.

[2] G. Xu, N. Zhang, M. Xu, Z. Xu, Q. Zhang, and Z. Song, "Outage probability and average ber of UAV-assisted dual-hop FSO communication with amplify-and-forward relaying," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 8287–8302, 2023.

[3] N. Heydarishahreza, T. Han, and N. Ansari, "Spectrum sharing and interference management for 6G LEO satellite-terrestrial network integration," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 5, pp. 2794–2825, 2025.

[4] Z. Zhao, G. Xu, N. Zhang, and Q. Zhang, "Performance analysis of the hybrid satellite-terrestrial relay network with opportunistic scheduling over generalized fading channels," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2914–2924, 2022.

[5] J. Jang, I.-S. Cho, M. Shin, J. Kim, and S. Jung, "Gateway-assisted neural angular routing for hierarchical satellite networks," *IEEE Internet Things J.*, vol. 13, no. 6, pp. 12 322–12 339, 2026.

[6] W. Wei, L. Fu, H. Gu, X. Lu, L. Liu, S. Mumtaz, and M. Guizani, "Iris: Toward intelligent reliable routing for software-defined satellite networks," *IEEE Trans. Commun.*, vol. 73, no. 1, pp. 454–468, 2025.

[7] X. Feng, Y. Sun, and M. Peng, "Distributed satellite-terrestrial cooperative routing strategy based on minimum hop-count analysis in mega LEO satellite constellation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10 678–10 693, 2024.

[8] K. Yu, Q. Cui, J. Xie, B. Chen, P. Wang, Y. Hou, and X. Tao, "A path-informed distributed routing algorithm for LEO satellite direct-to-device communication exploiting collaborative DRL," *IEEE Trans. Veh. Technol.*, pp. 1–14, 2026.

[9] Y. Ran, Y. Ding, S. Chen, J. Lei, and J. Luo, "Fully-distributed dynamic packet routing for LEO satellite networks: A GNN-enhanced multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 74, no. 3, pp. 5229–5234, 2025.

[10] Y. Lyu, H. Hu, R. Fan, Z. Liu, J. An, and S. Mao, "Dynamic routing for integrated satellite-terrestrial networks: A constrained multi-agent reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1204–1218, 2024.

[11] R. Sun, N. Cheng, C. Li, W. Quan, H. Zhou, Y. Wang, W. Zhang, and X. Shen, "A comprehensive survey of knowledge-driven deep learning for intelligent wireless network optimization in 6G," *IEEE Commun. Surveys Tuts.*, vol. 28, pp. 1099–1135, 2026.

[12] C. Zhang, H. Zhang, J. Qiao, Z. Li, and M.-S. Alouini, "Tides: Traffic intelligence with deepseek-enhanced spatial–temporal prediction," *IEEE J. Sel. Areas Commun.*, vol. 44, pp. 2544–2558, 2026.

Authorized licensed use limited to: CHONGQING UNIVERSITY. Downloaded on October 13,2026 at 04:44:41 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.