Review of Computer Engineering Research
2027 Vol. 14, No. 1, pp. 21-51
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/76.v14i1.5375
© 2027 Conscientia Beam. All Rights Reserved.

STG-DDPG: A deep reinforcement learning-based QoS-aware routing framework for high-mobility MANETs

Awadhesh Kumar Rai[1]
Akhilesh A. Waoo[2+]

[1]*Department of Computer Science and Engineering, AKS University, Satna, Madhya Pradesh, India.*
[1]*Email: awadheshrai.lko@gmail.com*
[2]*Department of Computer Applications, Information Technology and Sciences, AKS University, Satna, Madhya Pradesh, India.*
[2]*Email: akhilesh_e1735@aksuniversity.com*


(+ Corresponding author)

Article History
Received: 19 May 2026
Revised: 7 August 2026
Accepted: 3 September 2026
Published: 20 November 2026

Keywords
DDPG
Deep reinforcement learning
Graph convolutional networks
Intelligent routing
Mobile Ad Hoc Networks
QoS-aware routing
Temporal modeling.

ABSTRACT

Mobile Ad Hoc Networks (MANETs) in dense and high-mobility environments suffer from topology changes, route failures, and communication instability, which degrade Quality of Service (QoS). Traditional routing protocols struggle to maintain reliable communication under such conditions. Current Deep Reinforcement Learning (DRL) approaches are limited in scalability and prediction. Therefore, this study developed a predictive QoS-aware routing framework, which can adapt to the dynamic nature of networks. A Spatio-Temporal Graph-Embedded Deep Deterministic Policy Gradient (STG-DDPG) framework was proposed, representing the MANET topology as a time-evolving graph. Graph Convolutional Networks (GCNs) extracted spatial topology features, Long Short-Term Memory (LSTM) networks modeled temporal mobility patterns, and Deep Deterministic Policy Gradient (DDPG) optimized continuous-action routing. The framework was evaluated via NS-3 simulations for 50 to 150 mobile nodes, varying mobility speeds, and traffic loads, and compared with conventional routing protocols and DRL baselines. The results showed that the proposed framework improved the Packet Delivery Ratio (PDR) by 18.4%, reduced the end-to-end delay by 22.6%, increased throughput by 23.3%, and reduced energy consumption by 17.0% compared to the Ad hoc On-Demand Distance Vector (AODV) routing. Statistical analysis revealed these increases were significant at the 95% confidence level, and route volatility was reduced by as much as 27%. The results demonstrate that the proposed framework can offer a scalable and predictive routing solution for highly dynamic MANET environments, and it can enable reliable QoS-aware communication in applications such as Unmanned Aerial Vehicle (UAV) networks, emergency response systems, intelligent transportation systems, and other mission-critical mobile communication infrastructures.

Contribution/Originality: This work contributes to the existing literature on MANET routing by integrating graph-based spatial learning, temporal mobility modeling, and DRL in a unified framework. In this paper, we propose a novel STG-DDPG method for predictive, QoS-aware routing. This study is among the few to jointly exploit GCNs, Long Short-Term Memory (LSTMs), and continuous-action reinforcement learning to enable scalable and proactive routing in dense, high-mobility MANET environments.

1. INTRODUCTION

MANETs operating in high-mobility and dense deployment scenarios such as UAV swarms, vehicular ad-hoc networks, and cooperative robotic clusters exhibit highly dynamic and non-stationary topological behavior [1, 2]. In such environments, the underlying network can be modeled as a time-varying graph whose connectivity structure

© 2027 Conscientia Beam. All Rights Reserved.

evolves rapidly due to node mobility [3]. The resulting system is characterized by continuous graph evolution, high-dimensional state representations, frequent link breakages, and significant route volatility, which severely degrade routing stability and compromise QoS guarantees [4]. Topology-based and reactive approaches such as AODV and DSDV are examples of conventional routing protocols that rely on periodic updates or on-demand route-discovery mechanisms [5, 6]. In high-mobility dense MANETs, such mechanisms incur substantial route rediscovery latency, excessive control packet overhead, and oscillatory forwarding decisions [5]. As node velocity and density increase, the stale-route problem becomes dominant, leading to packet loss, delay inflation, and energy inefficiency [4].

Recent advances in DRL propose adaptive routing mechanisms that learn environment-aware policies [7, 8]. However, value-based methods such as Deep Q-Networks (DQN) suffer from scalability problems since the discrete action space grows exponentially in dense networks [9, 10]. Moreover, it has been observed that many existing DRL-based routing strategies ignore the inherent graph-based architecture of MANETs [9]. These approaches have reactive tendencies instead of truly predictive behavior, in that they do not explicitly consider temporal mobility dynamics [10]. The basic research problem is thus to design a routing policy that fulfils four critical requirements: (i) predictive, rather than reactive, decision-making; (ii) scalability with increasing node density; (iii) computational feasibility for real-time execution; and (iv) explicit QoS optimization under non-stationary mobility conditions. To address these challenges, an STG-DDPG framework is proposed by integrating GCNs for spatial topology learning and LSTMs for temporal mobility modeling within a continuous deterministic policy optimization framework, as shown in Figure 1, which enables proactive routing decisions that foresee link degradation before failure occurs. Extensive NS-3 simulations and reinforcement learning experiments demonstrate that the proposed framework improves routing stability, reduces route rediscovery frequency, and enhances QoS performance under high-mobility, dense MANET conditions.

Existing MANET routing protocols and many recent DRL-based routing approaches are unable to simultaneously address topology awareness, temporal mobility prediction, scalability, and QoS optimization in dense, high-mobility environments [11-13]. As network density and mobility increase, routing decisions become increasingly reactive, leading to route instability, excessive control overhead, packet loss, and degraded QoS performance. Therefore, an intelligent routing framework capable of predictive and scalable routing optimization is required.

Although recent DRL-based routing approaches have improved adaptability, most existing methods employ flat state representations and do not explicitly exploit the graph structure of MANETs. Furthermore, temporal mobility dynamics are rarely incorporated into routing optimization, limiting predictive routing capability. Existing solutions typically address topology awareness, temporal learning, or reinforcement learning independently rather than within a unified framework. Consequently, scalable predictive routing for dense and highly dynamic MANETs remains insufficiently explored.

The novelty of this work lies in the unified integration of graph-structured spatial learning, temporal mobility prediction, and continuous-action reinforcement learning within a single routing framework. The proposed STG-DDPG framework differs from existing approaches that do not rely on graph-based topology learning, temporal sequence modeling, or reinforcement learning individually but combine these capabilities to enable predictive QoS-aware routing. This integration allows consideration of topology evolution and link degradation in routing decisions while maintaining scalability in dense, high-mobility MANET environments.

The main contributions of this work are summarized as follows:

1. A new STG-DDPG framework is proposed for QoS-aware routing in high-mobility, dense MANET environments.
2. The spatiotemporal representation is unified by fusing GCNs for topology-aware spatial feature extraction and LSTM networks for temporal mobility modeling.

© 2027 Conscientia Beam. All Rights Reserved.

3. The routing optimization problem is formulated as a continuous-control task using DDPG, thereby avoiding the action-space explosion associated with discrete reinforcement learning approaches.
4. A predictive routing mechanism that can predict link degradation and route instability before failure. This permits proactive route adaptation.
5. Large-scale NS-3 simulations, statistical significance testing, confidence interval analysis, scalability analysis, and ablation studies are conducted to validate the effectiveness of the proposed framework under dense and highly dynamic MANET scenarios.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model and problem formulation. Section 4 describes the proposed STG-DDPG methodology. Details of the experimental setup and implementation are given in Section 5. In Section 6, experimental results and statistical analysis are presented. Section 7 summarizes the results and discusses implications. Section 8 deals with threats to validity. Section 9 concludes the paper and offers practical implications, limitations, and future research directions.

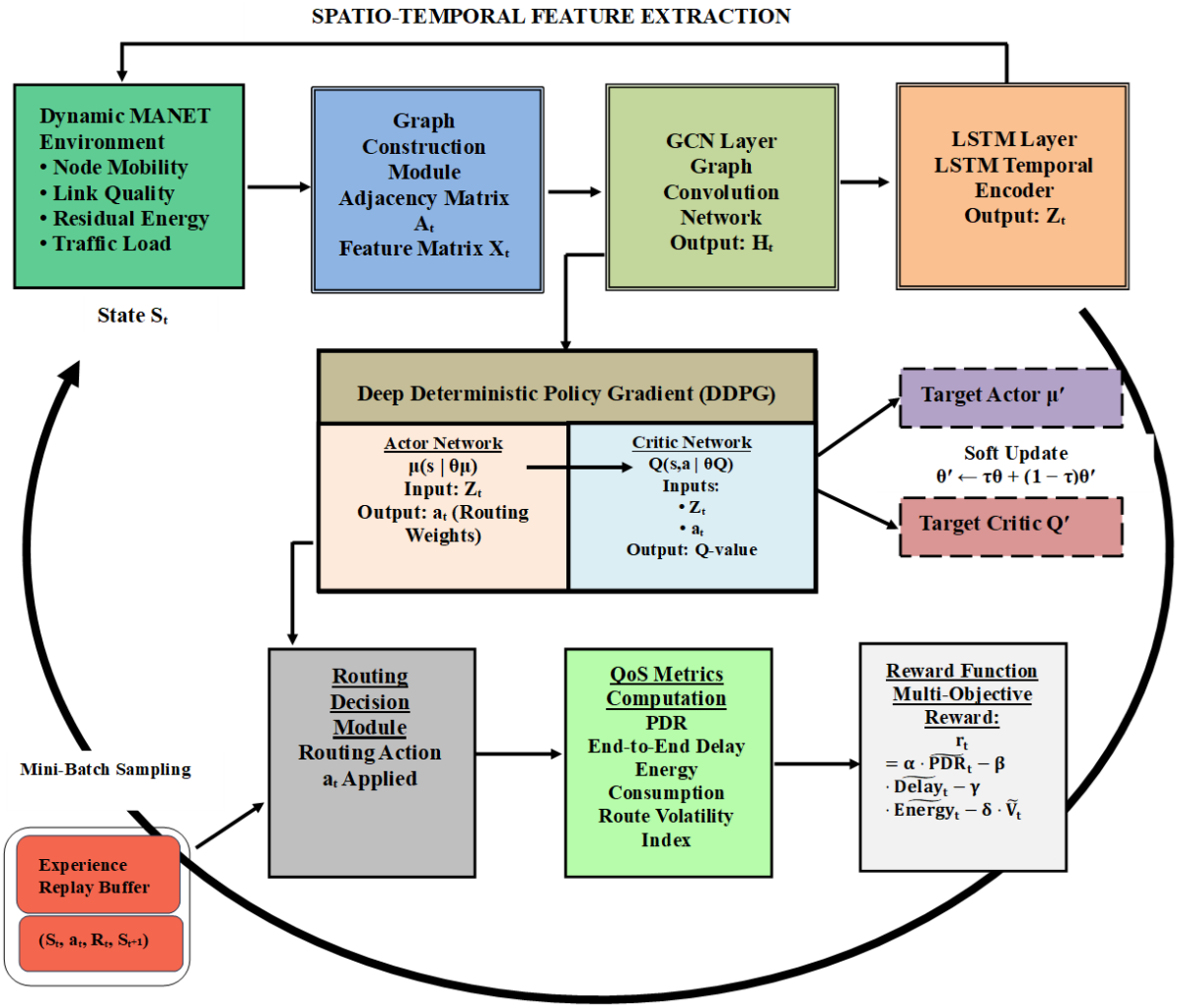


Figure 1. Comprehensive architecture of the proposed STG-DDPG framework.

2. RELATED WORK

Routing in MANETs has been thoroughly studied over the past two decades [1, 2], with approaches broadly grouped into topology-based, geographic, and learning-driven methods [6]. However, high-mobility, dense MANET scenarios introduce unique challenges that remain partly unaddressed by these methods. Over the past decade, routing research in MANETs has evolved from heuristic, topology-driven mechanisms to adaptive, learning-based strategies, where early approaches emphasized reactive and proactive table maintenance, followed by geographic forwarding

© 2027 Conscientia Beam. All Rights Reserved.

schemes for scalability [6]. More recently, machine learning and DRL techniques have been introduced to address dynamic topology and QoS optimization [14, 15]. Despite this progression, most solutions remain reactive and structurally unaware of graph evolution dynamics [16, 17].

2.1. Topology-Based Routing Protocols

AODV and DSDV, which are topology-driven protocols, rely either on on-demand route discovery or periodic routing table updates [5, 6]. These approaches perform well in reasonably dynamic environments [1, 2] but their effectiveness degrades significantly under high mobility [3, 18]. Repeated rediscovery processes are caused by route breakages, which occur quite often, and they introduce additional latency and increase control overhead. Periodic updates and broadcast-based mechanisms lead to excessive control packet exchanges, resulting in higher energy consumption. In dense network scenarios, the broadcast storm problem further augments congestion and packet collisions, thereby degrading overall network performance [2, 18]. These protocols suffer from huge performance degradation due to frequent topology changes and flooding of control messages. This can be proven by empirical observations when node speed is more than 15–20 m/sec [3]. This behavior is a reflection of the inherent reactiveness of traditional topology-based routing, as it cannot predict link failures. As a result, these limitations lead to meaningful degradation in QoS, reflected in reduced PDR, increased end-to-end delay, and elevated energy consumption in high-mobility dense MANET environments.

2.2. Geographic and Position-Based Routing

Geographic or position-based routing schemes aim to improve scalability by making use of node location information [6] to enable greedy forwarding decisions, and by eradicating the need for complete route discovery, these approaches reduce control overhead and improve efficiency in large-scale networks [1]. However, in high-mobility scenarios, their performance is constrained by rapidly changing topology conditions. Neighbor information can become outdated before updates are propagated [5], which can lead to link failures, routing voids, and increased packet loss. In addition to these drawbacks, purely location-based forwarding decisions fail to account for temporal mobility patterns, limiting the ability to predict future link stability. Even though geographic routing reduces signaling overhead compared to topology-based approaches, it remains highly sensitive to stale neighbor information and local minima problems under dynamic conditions. The lack of temporal modeling limits proactive adaptation and results in degraded QoS performance, e.g., lower PDR, higher delay, and higher energy consumption in fast-changing MANET environments.

2.3. Reinforcement Learning-Based Routing

Recent advancements in reinforcement learning have permitted adaptive routing strategies that learn policies from environmental interaction [14, 15]. Early works used Q-learning or discrete DRL approaches, such as DQN [15]. Though even these routing methods adapt better than traditional approaches, they suffer from the curse of dimensionality in dense MANETs, as the discrete action space grows exponentially with the number of neighboring nodes [16, 17]. To address continuous control challenges, algorithms such as DDPG have been introduced for networking optimization [19, 20], where continuous action spaces allow smoother routing weight adjustments and reduce oscillatory behavior [21]. But most existing DDPG-based routing schemes treat network states as flat feature vectors, neglecting the inherent graph structure of MANETs [11, 22]. Without topology-aware feature extraction, such approaches fail to generalize across varying network densities.

2.3.1. Categorization of DRL-Based Routing Approaches

Recent advances in DRL have introduced adaptive routing mechanisms capable of learning optimal forwarding policies under dynamic network conditions. Existing DRL-based routing methods can be broadly categorized into

© 2027 Conscientia Beam. All Rights Reserved.

three main classes. The first category consists of value-based discrete routing approaches, primarily built upon Deep Q-Network (DQN) architectures [15], in which the agent selects the next hop from a discrete action space. This discrete formulation is effective in small-scale scenarios, but it leads to rapid action-space expansion as node density increases, resulting in combinatorial complexity and slower convergence in dense MANET environments. The second category uses policy-gradient and actor–critic methods, which optimize routing decisions through direct policy learning [20, 23]. Compared with purely value-based models, these methods show improved stability. Nevertheless, many implementations still rely on flattened state representations, limiting their ability to capture the underlying graph topology of the network. The third category consists of Multi-Agent Reinforcement Learning (MARL) frameworks [12, 24], in which multiple nodes learn routing strategies collectively. Although promising for distributed optimization, MARL approaches often suffer from coordination overhead and non-stationarity challenges, particularly under high-mobility conditions. Despite these advancements, several structural limitations remain evident across the literature. First, most DRL-based routing schemes use fixed-length vectorized state inputs, resulting in limited graph awareness and insufficient exploitation of spatial topology information [11, 17]. Second, temporal mobility patterns are seldom explicitly modeled, limiting the estimation of predictive link stability [25, 26]. Then, discrete action formulations lead to scalability problems as network size increases [16]. Finally, empirical evaluations are often conducted on moderate-scale networks, i.e., fewer than 100 nodes, meaning that performance in large-scale, dense MANETs is under-validated [9, 16]. Such limitations affect routing performance, leading to instability, delay, and low packet-delivery reliability in dense and highly dynamic MANET environments. In contrast to the above approaches, the proposed STG-DDPG framework addresses these structural limitations through three key design principles. First, explicit graph-convolutional modeling is employed to preserve spatial topology information [27, 28], enabling topology-aware state embedding rather than flattened vector representations. Next, temporal mobility dynamics are captured using sequence modeling to facilitate predictive link-stability estimation [25, 29] under high-mobility conditions. Then, a deterministic continuous-action policy is adopted to eliminate discrete-action explosion in dense networks [19], thereby improving scalability and convergence stability. Collectively, these design choices establish a unified spatio-temporal continuous-control routing paradigm suitable for dense, high-speed MANET environments.

2.4. Graph Neural Networks (GNNs) in Networking

GNNs have recently gained significant attention for modeling relational dependencies in communication networks Wu et al. [27] and Dai et al. [30]. Wu et al. [27] presented a comprehensive survey of GNN architectures and identified their effectiveness in learning topology-aware representations that preserve structural relationships among network entities. Based on this, Almasan et al. [11] proved that combining GNNs with DRL can improve routing optimization by leveraging network topology in policy learning. These studies have demonstrated GNNs as a promising tool for intelligent networking applications.

In the first graph-based networking research, Rusek et al. [28] presented communication network topology modeling based on GNNs. Later improvements have made GNNs a powerful method for network optimization and smart routing. Recent studies have shown that incorporating graph representations with reinforcement learning improves routing optimization by retaining network topology information during policy learning [11, 28]. This evolution from graph-based network representation to graph-aware reinforcement learning lays the foundation for the proposed STG-DDPG framework.

GNN-based models can aggregate information from neighboring nodes using graph convolution operations. This can effectively capture spatial topology patterns that are difficult to represent using conventional feature-vector approaches [28, 31]. Thus, graph-based learning has been successfully applied to network optimization, traffic prediction, and routing-related tasks [11, 25, 30]. However, most existing methods focus mainly on spatial topology

© 2027 Conscientia Beam. All Rights Reserved.

representation and do not explicitly model the temporal mobility dynamics. Similarly, temporal sequence learning methods often neglect the underlying graph structure of evolving MANET topologies [25].

Therefore, existing studies are usually concerned with spatial and temporal network features, respectively, which makes them unable to proactively predict topology evolution and future route degradation. Moreover, the integration of graph-based spatial learning, temporal mobility modeling, and continuous-action reinforcement learning [11, 22] has not been sufficiently studied in dense and highly dynamic MANET environments. This limitation motivates the development of the proposed STG-DDPG framework, which integrates GCN-based spatial representation, LSTM-based temporal modeling, and DDPG-based continuous routing optimization in a unified architecture.

To allow a thorough comparison of recent DRL-based MANET routing approaches, Table 1 summarizes their main features, such as spatial modelling, temporal awareness, action formulation, predictive capacity, and scalability.

Table 1. Comparative Analysis of Existing Routing Approaches and the Proposed STG-DDPG Framework.

Ref.	Study	Spatial Modeling	Temporal Modeling	Continuous Action	Predictive Routing	Scalability
Almasan, et al. [11]	Almasan, et al. [11]	✔ (GNN)	✘	✘ (DRL-based)	✘	Moderate (Routing optimization)
Alanazi and Zareei [12]	Alanazi and Zareei [12]	✔ (GNN)	✘	✘ (Discrete MARL)	✘	✔ (Distributed MANETs)
Zhang et al. [32]	Zhang et al. [32]	✔ (Graph-based DRL)	✘	✘ (Value-based DRL)	✘	Moderate (Ad hoc networks)
Jin, et al. [9]	Jin, et al. [9]	✘	✘	✘ (Value-based DRL)	Partial	Moderate (Urban networks)
Upadhyay et al. [33]	Upadhyay et al. [33]	✘	✘	✘ (DRL-based)	Partial	✔ (VANET-scale)
Li, et al. [22]	Li, et al. [22]	✔ (Graph Transformer + DRL)	✘	✔ (Policy-based RL)	Partial	✔ (Large-scale Software-Defined Networking (SDN))
Al-Mashhadani and Karoui [13]	Al-Mashhadani and Karoui [13]	✘	✘	✔ (Actor–Critic DRL)	Partial	✔ (SDN-integrated MANET)
—	Proposed STG-DDPG	✔ (Spatial GCN)	✔ (Temporal LSTM)	✔ (Continuous DDPG)	✔ (Predictive link stability)	✔ (Dense high-mobility MANETs)

Note: ✔ indicates presence of feature, ✘ indicates absence, and "Partial" indicates limited or indirect support in the referenced study. The comparison presented in Table 1 demonstrates that existing approaches generally address either topology-aware learning, temporal modeling, or reinforcement learning-based optimization independently. However, the proposed STG-DDPG framework combines all three capabilities into a single routing architecture, thus solving the main limitations pointed out in the literature.

2.4.1. Critical Analysis of Existing Studies

Table 1 provides a comparative analysis of some main disadvantages of existing MANET routing techniques. Traditional routing protocols are primarily built on reactive or proactive route-maintenance mechanisms and have limited adaptability to highly dynamic network conditions [2, 5, 6]. Recent reinforcement learning-based approaches improve the adaptability of routing. However, many of these approaches utilize discrete action spaces, which cause scalability issues when the network density increases [9, 13, 23]. Moreover, many graph-based routing strategies leverage topology-aware representations but do not explicitly model temporal mobility dynamics and are thus limited in their ability to predict future link failures [11, 12]. Temporal learning methods tend to ignore the underlying graph structure of MANETs, leading to an incomplete representation of network dynamics [25, 26]. Another

© 2027 Conscientia Beam. All Rights Reserved.

important limitation is that most of the existing studies are evaluated under moderate-scale network settings, leaving their effectiveness in dense and highly mobile MANET environments largely unexplored [13, 23]. These observations point to the necessity of a unified routing framework that jointly captures spatial topology information, temporal mobility behavior, and continuous routing optimization. To overcome these limitations, the proposed STG-DDPG framework integrates GCN-based spatial embedding, LSTM-based temporal modeling, and DDPG-based continuous-action routing optimization.

2.5. Research Gap and Motivation

Despite substantial progress, three critical limitations persist in the current literature, as follows:

Lack of joint spatial-temporal modeling of dynamic MANET topology [11, 25].

No continuous updating of routing weights using graph information [11, 32].

Limited ability to predict when links will break in high-mobility situations [3, 9].

Most RL-based routing methods remain reactive [12, 16], which adapt only after performance degradation is observed. Furthermore, scalability across varying node densities is rarely addressed explicitly [17, 30].

This work proposes a unified STG-DDPG framework that integrates graph-convolutional spatial aggregation, temporal mobility modeling, and continuous deterministic policy optimization to overcome these gaps. Dynamic topology evolution is embedded into a compact latent representation, and the method enables predictive QoS-aware routing decisions that anticipate link degradation before failure occurs. This integration represents a significant advancement over existing topology-based, geographic, and learning-driven routing mechanisms, as validated in Section 6.

2.6. Positioning of the Proposed STG-DDPG Framework

The preceding discussion highlights that existing MANET routing approaches are limited [2, 30] by the absence of integrated spatial–temporal modeling and reliance on discrete action formulations, which restrict scalability and adaptability in dense, high-mobility environments. While recent works have explored graph neural network–based representations [11, 12] and graph-aware reinforcement learning for resilient network control [27], these methods typically address spatial or learning aspects in isolation. In contrast to previous models, the STG-DDPG framework unifies three key components within a single architecture: first, topology-aware spatial representation through graph convolution; next, temporal mobility modeling via sequence learning; and, finally, deterministic continuous-action policy optimization. This combined design helps make smart routing decisions by predicting link problems in advance and adjusting to changing network conditions. Hence, it provides a scalable and reliable solution for routing that is efficient in dense and highly mobile MANETs and overcomes the major limitations of existing techniques.

3. SYSTEM MODEL AND PROBLEM FORMULATION

A high-mobility MANET is modeled as a time-varying graph in Equation 1 with dynamic node features. The objective is to learn a QoS-aware routing policy under continuously changing network conditions.

3.1. Network Model

We consider a high-mobility dense MANET [1, 2] consisting of N mobile nodes distributed within a bounded two-dimensional region. The network at this time is modeled as a dynamic undirected graph [17, 32].

$$G_t = (V, E_t) \quad (1)$$

Where G_t denotes the MANET graph at time t,

Where $V = \{v_1, v_2, \ldots, v_N\}$ represents the set of nodes, v_i denotes the i^{th} node, and $E_t \subseteq V \times V$ denotes the set of active wireless links at time t.

A link $(v_i, v_j) \in E_t$ exists if:

© 2027 Conscientia Beam. All Rights Reserved.

$$\| p_i(t) - p_j(t) \| \leq R \quad (2)$$

Where $p_i(t), p_j(t) \in \mathbb{R}^2$ denote the two-dimensional positions of nodes v_i and v_j at time t, respectively; denotes the Euclidean distance between the two nodes; and R represents the maximum wireless communication range. Equation 2 determines network connectivity by establishing a communication link whenever the distance between two nodes remains within the communication radius.

Each node v_i is characterized by a time-dependent feature vector, as defined in Equation (3).

$$x_i(t) = [p_i(t), v_i(t), e_i(t), q_i(t)] \quad (3)$$

Where $x_i(t)$ denotes the feature vector associated with node v_i at time $t, p_i(t)$ represents the two-dimensional position of node $v_i, v_i(t)$ denotes its velocity, $e_i(t)$ represents the residual energy level, and $q_i(t)$ denotes the queue length reflecting local congestion status. These features collectively characterize the mobility, energy availability, and traffic conditions in each node and serve as the input state representation for routing optimization.

Stacking all node features forms the node feature matrix given in Equation (4).

$$X_t \in \mathbb{R}^{N \times d} \quad (4)$$

Where X_t denotes the node feature matrix at time t, N as the total number of nodes in the network, and d as the dimensionality of each node feature vector. It explains that the matrix X_t is formed by concatenating all node feature vectors to provide a global representation of the network state for graph-based learning. The time-varying topology is represented by the adjacency matrix defined in Equation (5).

$$A_t \in \mathbb{R}^{N \times N} \quad (5)$$

Where A_t denotes the adjacency matrix representing network connectivity at time t, and each element a_{ij} indicates the existence of a communication link between nodes v_i and v_j. The adjacency matrix reflects the dynamic topology of the MANET and enables the GCN to collect information from neighboring nodes in the spatial feature extraction process. The combination of the node feature matrix X_t and adjacency matrix A_t provides a complete graph representation of the MANET at time t. While X_t captures node-specific attributes such as mobility, energy, and congestion status, A_t models the connectivity relationships among nodes. This combination of matrices is the input to the GCN component, which allows it to make topology-aware routing decisions.

3.2. Mobility and Link Dynamics

Nodes follow high-mobility patterns, such as Gauss–Markov [5], which results in non-stationary topology evolution. The link duration between two nodes depends on their relative velocity and direction [9]. The Link Expiration Time (LET) between nodes i and j is given by:

$$LET_{ij} = \frac{-(ab+cd)+\sqrt{(a^2+c^2)R^2-(ad-bc)^2}}{a^2+c^2} \quad (6)$$

Where LET_{ij} denotes the Link Expiration Time between nodes i and j, R represents the wireless communication range, (v_i^x) and (v_i^y) denote the velocity components of node i along the x- and y-axes, respectively, (v_j^x) and (v_j^y) denote the velocity components of node j, $(p_i^x)$and (p_i^y) represent the coordinates of node i, and (p_j^x) and (p_j^y) represent the coordinates of node j. The intermediate variables are defined as $(a = v_i^x - v_j^x)$, $(b = p_i^x - p_j^x)$, $(c = v_i^y - v_j^y)$, and $(d = p_i^y - p_j^y)$. Equation 6 estimates the expected duration for which a communication link remains available before the distance between the two nodes exceeds the communication range. Higher (LET_{ij}) values indicate more stable links and are therefore preferred during route selection to reduce route failures and packet retransmissions. Frequent link breaks lead to route instability and retransmissions. Therefore, LET is incorporated as an important routing feature to enable proactive selection of stable communication paths in highly dynamic MANET environments.

© 2027 Conscientia Beam. All Rights Reserved.

3.3. QoS Metrics

The QoS metrics used to evaluate routing performance [23, 34] are Throughput (Th_t), Packet Delivery Ratio PDR, End-to-End Delay ($Delay_t$), Energy Consumption ($Energy_t$), Route Volatility Index (V_t). Among these metrics, the Route Volatility Index is introduced to quantify route stability in highly dynamic MANET environments and is formally defined as:

$$V_t = \frac{N_{\text{route changes}}}{T} \quad (7)$$

Where (V_t) denotes the Route Volatility Index at time (t), ($N_{\text{route changes}}$) represents the total number of route changes observed during the measurement interval, and T denotes the duration of the observation interval. Equation 7 quantifies the stability of routing paths by measuring the frequency of route changes over time. Smaller values of (V_t) correspond to more stable routes with fewer route rediscoveries, and larger values correspond to frequent topology disruptions and route instability. This study also considers the Route Volatility Index to assess the capability of the developed STG-DDPG framework to preserve stable communication routes in highly dynamic MANET environments.

3.4. Reinforcement Learning Formulation

The routing problem is formulated as a Markov decision process (MDP), as defined in Equation (8).

$$M = (S, A, P, R, \gamma) \quad (8)$$

Where M denotes the Markov Decision Process (MDP) used to model the routing problem. S represents the state space describing the current network condition, A denotes the action space corresponding to routing decisions, P represents the state transition probability, R denotes the reward function that evaluates routing performance, and $\gamma \in (0,1)$ is the discount factor that determines the relative importance of future rewards [14]. The MDP formulation allows the routing agent to learn adaptive routing policies that maximize the long-term QoS performance under dynamic MANET conditions.

3.4.1. State Definition

The state S_t corresponds to the spatio-temporal representation of the network [27].

$$S_t = Z_t \quad (9)$$

Where S_t denotes the network state observed by the routing agent at time t, and Z_t represents the spatio-temporal embedding generated from the graph-based spatial features and temporal mobility information. The state representation for the routing decision-making of the reinforcement learning agent is described in Equation 9. The state that successfully captures the dynamic behavior of the MANET environment includes the topology and mobility characteristics.

Spatial embedding [28].

$$H_t = \text{GCN}(G_t, X_t) \quad (10)$$

Where H_t is the spatial embedding generated by the Graph Convolutional Network (GCN), G_t represents the MANET graph at time t, and X_t is the node feature matrix containing mobility, energy, and congestion-related information. Topology-aware spatial features can be obtained by aggregating information from its neighboring nodes, as described in Equation 10. These spatial representations help the routing agent find reliable communication regions and connectivity patterns in the network. Temporal embedding:

$$Z_t = \text{LSTM}(H_t, Z_{t-1}) \quad (11)$$

Where Z_t is the spatio-temporal embedding at time t, H_t is the current spatial embedding provided by the GCN, and Z_{t-1} is the spatio-temporal embedding at the previous time step. The LSTM network can learn the temporal dependencies and the mobility evolution of the MANET by means of Equation 11. The LSTM component learns the

© 2027 Conscientia Beam. All Rights Reserved.

history of topology changes and mobility patterns to better predict future network states and facilitate proactive routing decisions.

3.4.2. Action Space

The action is defined in a fixed-dimensional continuous space [19], as formulated in Equation (12).

$$a_t \in R^N \tag{12}$$

Where a_t denotes the continuous action vector is generated by the DDPG actor network at time t, and N represents the total number of nodes in the network. Each element of a_t corresponds to the routing preference assigned to a candidate forwarding node. The continuous action representation makes it possible to perform fine-grained routing decisions and avoids the action-space explosion typical for discrete routing formulations. Only neighboring nodes are valid forwarding candidates, so a masking and normalization step is performed to obtain the final routing action. The normalized routing action used for next-hop selection is defined in Equation (13).

$$\hat{a}_t = \frac{M_t \odot a_t}{\sum_i (M_t \odot a_t)_i} \tag{13}$$

Where $\hat{a}_t$ denotes the normalized routing action used for next-hop selection $M_t \in \{0,1\}^N$ represents the binary neighbor mask at time t, a_t denotes the continuous action vector generated by the actor network, and $\odot$ denotes the element-wise multiplication. The neighbor mask avoids invalid forward candidates by assigning zero weight to non-neighbor nodes. The normalization operation converts the masked action vector into a valid probability distribution over the potential next-hop nodes. This mechanism guarantees that the routing decisions are topology-aware and consistent with the current network connectivity while allowing stable reinforcement learning.

3.4.3. Reward Function

To eliminate inconsistencies and ensure practical interpretability, the reward function is defined as Song et al. [34].

$$r_t = \alpha \widetilde{PDR}_t - \beta \widetilde{Delay}_t - \gamma \widetilde{Energy}_t - \delta \tilde{V}_t \tag{14}$$

Where r_t denotes the immediate reward received by the routing agent at time t, $\widetilde{PDR}_t$ represents the normalized Packet Delivery Ratio $\widetilde{Delay}_t$ denotes the normalized End-to-End Delay, $\widetilde{Energy}_t$ represents the normalized Energy Consumption, and $\tilde{V}_t$ denotes the normalized Route Volatility Index. The weighting coefficients $\alpha, \beta, \gamma, and \delta$ determine the relative importance of each QoS metric and satisfy $\alpha + \beta + \gamma + \delta = 1$. Equation 14 encourages the routing agent to maximize packet delivery while minimizing delay, energy consumption, and route instability, thereby achieving balanced QoS-aware routing performance.

3.5. Optimization Objective

The objective is to learn an optimal deterministic policy μ^* that maximizes the expected cumulative discounted reward [14].

$$\mu^* = \arg \max_{\mu} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{15}$$

Where μ^* denotes the optimal routing policy, μ represents a candidate routing policy, $\mathbb{E}[\cdot]$ denotes the expectation operator, r_t is the reward obtained at time t, T represents the decision horizon; and $\gamma \in (0,1)$ is the discount factor defined in Section 3.4. Equation 15 defines the optimization objective of the STG-DDPG framework, which seeks a routing policy that maximizes the expected cumulative discounted reward. Consequently, the routing agent learns forwarding decisions that provide long-term QoS benefits rather than optimizing only immediate network conditions. The policy gradient is given as Lillicrap et al. [19].

$$\nabla_{\theta^\mu} J \approx \mathbb{E}[\nabla_a Q(S_t, a \mid \theta^Q). \nabla_{\theta^\mu} \mu(S_t \mid \theta^\mu)] \tag{16}$$

© 2027 Conscientia Beam. All Rights Reserved.

Where J denotes the objective function, θ^{μ} represents the actor-network parameters, θ^{Q} denotes the critic-network parameters, $Q(S_t, a|\theta^{Q})$ is the critic-estimated action-value function, and $\mu(S_t|\theta^{\mu})$ represents the deterministic policy generated by the actor network. The actor network is updated during training by the policy gradient calculated by Equation 16. The deterministic policy gradient theorem enables the routing agent to efficiently optimize continuous routing actions, which allows adaptive next-hop selection without the combinatorial complexity of discrete routing decisions. The actor–critic learning mechanism enables the proposed STG-DDPG framework to continuously adapt routing decisions according to changing topology, congestion, and link-stability conditions. Consequently, the learned routing policy can proactively select communication paths that maximize long-term QoS performance in highly dynamic MANET environments.

3.6. Problem Statement

In a high-mobility, dense MANET modeled as a time-varying graph [1, 2], the objective is to learn a scalable routing policy that anticipates link expiration before disconnection, maximizes PDR, minimizes end-to-end delay and energy consumption, and reduces route volatility [34] to ensure stability and maintain computational efficiency for real-time deployment. These requirements call for one framework that can capture how nodes are connected, how the network changes over time, and how routing can be optimized, which motivates the STG-DDPG framework.

4. STG-DDPG METHODOLOGY

In this section, we introduce the STG-DDPG framework, which integrates graph-structured spatial learning, temporal mobility modeling, and continuous deterministic policy optimization for predictive QoS-aware routing in high-mobility, dense MANETs, as illustrated in Figure 2.

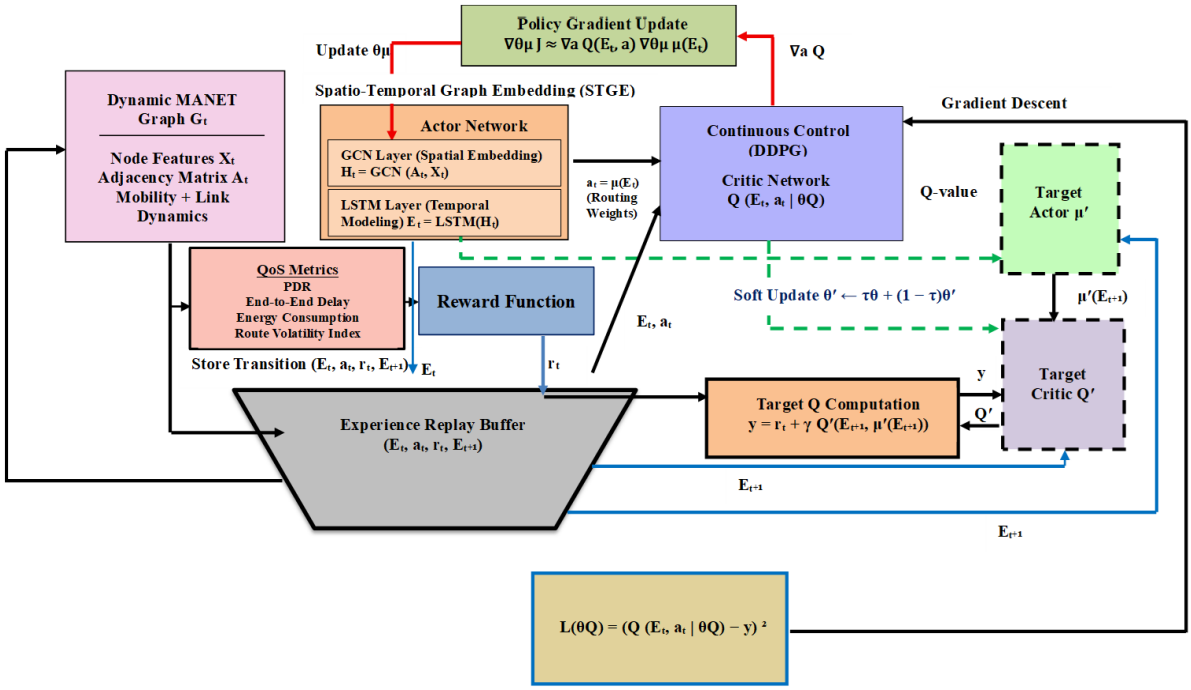


Figure 2. STG-DDPG Framework for Spatio-Temporal Graph-Based Routing in MANETs.

The framework contains three main components: 1) a Spatio-Temporal Graph Embedding (STGE) module, 2) a deterministic actor network, and 3) a critic network with target stabilization, which together form an integrated pipeline, as follows:

$$(G_t, X_t) \rightarrow \text{GCN} \rightarrow \text{LSTM} \rightarrow Z_t \rightarrow \text{Actor/Critic} \rightarrow a_t.$$

© 2027 Conscientia Beam. All Rights Reserved.

Where (G_t) denotes the dynamic MANET graph at time t, (X_t) represents the node feature matrix containing mobility, energy, and congestion-related information, and (Z_t) denotes the spatio-temporal embedding generated by the STGE module. The GCN part learns the topology-aware spatial features from the network graph, and the LSTM part learns the temporal mobility dynamics and the topology evolution. The resulting embedding (Z_t) is input to the actor–critic networks, which learn continuous routing policies and evaluate the routing actions to maximize the long-term QoS performance. The final action (a_t) is the routing decision for next-hop selection.

4.1. STGE

Routing in high-mobility MANETs requires joint modeling of spatial topology dependencies among nodes and temporal mobility patterns [25, 27] that influence link dynamics; to address this, a two-stage embedding pipeline is employed, consisting of spatial feature extraction using graph convolution followed by temporal modeling using an LSTM network.

4.1.1. Spatial Feature Extraction via Graph Convolution

At times t, the network is represented by an adjacency matrix A_t and node feature matrix, as follows:

$$X_t \in \mathbb{R}^{N\times d}.$$

We compute the normalized adjacency matrix as follows [28].

$$\tilde{A}_t = A_t + I \tag{17}$$

Where $\tilde{A}_t$ denotes the adjacency matrix with self-connections, A_t represents the original adjacency matrix at time *t*, and *I* is the identity matrix. Equation 17 adds self-loops to the graph, enabling each node to maintain its own feature information during graph convolution while aggregating information from its neighbors simultaneously.

$$\tilde{\mathrm{D}}_t(\mathrm{i},\mathrm{i}) = \sum_j \tilde{\mathrm{A}}_t(\mathrm{i},\mathrm{j}) \tag{18}$$

Where $\tilde{D}_t$ denotes the diagonal degree matrix associated with the augmented adjacency matrix $\tilde{A}_t$, and $\tilde{D}_t(i,i)$ represents the degree of node *i*. Equation 18 computes the number of effective connections associated with each node and is used for adjacency normalization during graph convolution.

The graph convolution operation at the layer l is defined as Wu et al. [27].

$$\mathrm{H}_t^{(l+1)} = \sigma\left(\tilde{\mathrm{D}}_t^{-1/2}\tilde{\mathrm{A}}_t\tilde{\mathrm{D}}_t^{-1/2}\mathrm{H}_t^{(l)}\mathrm{W}^{(l)}\right) \tag{19}$$

Where $H_t^{(l)}$ denotes the node feature representation at graph convolution layer *l*, $H_t^{(l+1)}$ denotes the updated feature representation at layer *l+1*, $W^{(l)}$ represents the learnable weight matrix of the *l-th* GCN layer, and $\sigma(\cdot)$ denotes the Rectified Linear Unit (ReLU) activation function. The normalized adjacency operator $\tilde{D}_t^{-1/2}\tilde{A}_t\tilde{D}_t^{-1/2}$ enables each node to aggregate information from its neighboring nodes while preventing feature scaling issues caused by varying node degrees. Equation 19 combines node attributes and local network connectivity information to extract topology-aware spatial features, which allows the routing agent to identify reliable communication regions and connectivity patterns in the MANET.

Final embedding:

$$\mathrm{H}_t = \mathrm{H}_t^{(\mathrm{L})} \tag{20}$$

Where H_t is the final spatial embedding after L graph convolution layers, and $H_t^{(L)}$ represents the output of the last GCN layer. The representation of MANET generated by Equation 20 is topology-aware; that is, it encodes the node relationships, connectivity patterns, and local structural information. This spatial embedding is then fed to an LSTM module for temporal mobility modeling. The GCN component in the STGE module transforms the raw node features and network topology information into compact spatial representations. The embedding generated by aggregating information from neighboring nodes captures local and global topology characteristics, forming a robust basis for subsequent temporal modeling and routing optimization.

© 2027 Conscientia Beam. All Rights Reserved.

4.1.2. Temporal Mobility Modeling Via LSTM

To capture temporal dependencies, spatial embeddings are processed using an LSTM over a sliding window of length T, denoted as $\{H_{t-T+1}, \ldots, H_t\}$. The LSTM updates are defined as:

$$f_t = \sigma(W_f H_t + U_f h_{t-1} + b_f) \quad (21)$$

Where f_t denotes the forget gate at time t, H_t represents the spatial embedding generated by the GCN, h_{t-1} denotes the hidden state from the previous time step, and W_f, U_f, b_f are learnable parameters. Equation 21 quantifies the information history retained from the previous states of the network, allowing the model to discard outdated topology information and keep relevant mobility patterns.

$$i_t = \sigma(W_i H_t + U_i h_{t-1} + b_i) \quad (22)$$

Where i_t is the input gate at time t, and W_i, U_i, b_i are trainable parameters. Equation 22 controls the amount of new spatial information that is incorporated in the memory state and allows the model to adapt to recent topology and mobility changes in the MANET.

$$o_t = \sigma(W_o H_t + U_o h_{t-1} + b_o) \quad (23)$$

Where o_t is the output gate at time t, and W_o, U_o, b_o are Learnable parameters. The amount of information disclosed from the internal memory state to the hidden representation is given by Equation 23, which affects the final spatiotemporal embedding used for routing decisions.

$$\tilde{c}_t = \tanh(W_c H_t + U_c h_{t-1} + b_c) \quad (24)$$

Where $\tilde{c}_t$ is the candidate cell state, and W_c, U_c, b_c are learnable parameters. Equation 24 generates new candidate information based on the current spatial embedding and historical network context, giving an updated representation of evolving mobility and topology conditions.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (25)$$

Where c_t is the cell state at time t, c_{t-1} denotes the previous cell state, and $\odot$ represents element-wise multiplication. Equation 25 updates the memory state by combining historically retained information with new information from the network, thus allowing long-term modeling of the mobility trend and topology evolution.

$$h_t = o_t \odot \tanh(c_t) \quad (26)$$

Where h_t is the hidden state at time t. The temporal representation of the network is given by Equation 26, which combines the updated memory state and output gate. The hidden state contains information on historical mobility patterns, link dynamics, and temporal dependencies that are useful for routing optimization. The final spatio-temporal embedding is defined as:

$$Z_t = h_t \quad (27)$$

Where Z_t is the final spatio-temporal embedding used as the state representation for the DDPG routing agent. The output of the LSTM module is Equation 27, which integrates the historical mobility information and the topology-aware spatial features learned by the GCN. This representation enables the capture of mobility trends, velocity correlations, and link stability patterns and provides proactive routing decisions in highly dynamic MANET environments. The LSTM part allows the proposed framework to model the temporal dependencies that cannot be modeled by spatial graph learning alone. By recalling previous network states, LSTM is able to predict changes in topology, mobility trends, and future link stability, which helps in proactive route selection and reduces route failures in highly dynamic MANET environments.

4.2. Continuous Routing Optimization Via DDPG

Routing is formulated as a continuous control problem to avoid combinatorial explosion in next-hop selection [19].

4.2.1. Actor Network

The actor implements a deterministic policy.

© 2027 Conscientia Beam. All Rights Reserved.

$$a_t = \mu(Z_t \mid \theta^{\mu}) \quad (28)$$

Where a_t denotes the continuous action vector generated by the actor network at time *t*, Z_t represents the spatio-temporal state embedding obtained from the STGE module, $\mu(\cdot)$ denotes the deterministic policy function, and represents the learnable parameters of the actor network. Equation 28 maps the current network state to a routing action by assigning routing preferences to candidate forwarding nodes. The actor network enables continuous-action routing optimization, allowing fine-grained route selection according to topology conditions, mobility dynamics, and network congestion levels.

Masking and normalization are used to verify the validity of routing decisions.

$$\hat{a}_t = \frac{M_t \odot a_t}{\sum_i (M_t \odot a_t)_i} \quad (29)$$

Where $\hat{a}_t$ is the normalized routing action used for next-hop selection, $M_t \in \{0,1\}^N$ represents the binary neighbor mask at time *t*, a_t is the raw action vector generated by the actor network, and $\odot$ is element-wise multiplication. The neighbor mask discards invalid forwarding candidates by assigning zero weight to nodes that are not directly reachable. The normalization operation transforms the masked action vector into a valid probability distribution over feasible next-hop nodes. Equation 29 ensures that routing decisions remain topology-aware, satisfy network connectivity constraints, stabilize the training process, and support interpretable routing decisions. During training, exploration is facilitated using Ornstein–Uhlenbeck noise [19].

4.2.2. Critic Network

The critic estimates the action-value function.

$$Q(Z_t, a_t \mid \theta^{Q}) \quad (30)$$

Where $Q(Z_t, a_t \mid \theta^{Q})$ denotes the action-value function estimated by the critic network, Z_t represents the spatio-temporal state embedding at time *t*, a_t denotes the routing action generated by the actor network, and θ^{Q} represents the learnable parameters of the critic network. Equation 30 evaluates the quality of a routing decision by estimating the expected cumulative reward associated with a given state–action pair. The critic network gives feedback information to the actor network, which guides the learning process towards routing policies that maximize QoS performance in the long term.

The loss function is defined as:

$$L(\theta^{Q}) = \frac{1}{N} \sum_i (y_i - Q(Z_i, a_i \mid \theta^{Q}))^2 \quad (31)$$

Where $L(\theta^{Q})$ is the critic loss function, *N* represents the mini-batch size, *y_i* is the target value associated with the *i-th* training sample, and $Q(Z_i, a_i \mid \theta^{Q})$ represents the value predicted by the critic network. Equation 31 minimizes the mean squared error between the predicted and target action values, which allows the critic network to evaluate the routing decisions accurately during training.

The target value is:

$$y_i = r_i + \gamma Q'(Z_{i+1}, \mu'(Z_{i+1}) \mid \theta^{Q'}) \quad (32)$$

Where y_i is the target value used for critic training, r_i represents the immediate reward obtained after executing an action $a_i, \gamma \in (0,1)$ is the discount factor, Q' represents the target critic network, μ' is the target actor network, and $\theta^{Q'}$ represents the parameters of the target critic network. Equation 32 approximates the expected cumulative reward by summing the immediate routing reward and the discounted future reward. Target networks help stabilize training and reduce oscillations in policy learning. The critic network enables the STG-DDPG framework to assess the long-term impact of routing decisions rather than immediate network conditions. The critic forecasts future rewards of candidate routes to help the actor choose communication paths that give lasting improvements in packet delivery, delay, energy efficiency, and route stability. Target networks are used to improve convergence stability during training.

© 2027 Conscientia Beam. All Rights Reserved.

4.2.3. Deterministic Policy Gradient

$$\nabla_{\theta^\mu} J = \mathbb{E}[\nabla_a Q(Z, a \mid \theta^Q) \nabla_{\theta^\mu} \mu(Z \mid \theta^\mu)] \quad (33)$$

Where J denotes the expected cumulative reward, Z represents the spatio-temporal state embedding, θ^μ denotes the parameters of the actor network, θ^Q represents the parameters of the critic network, $Q(Z, a|\theta^Q)$ denotes the critic-estimated action-value function, and $\mu(Z|\theta^\mu)$ represents the deterministic policy generated by the actor network. The policy gradient for updating the actor network in training is calculated by Equation 33.

The routing agent can effectively optimize continuous routing actions by leveraging the deterministic policy gradient [19] theorem and accordingly choose the next hop based on network topology, mobility patterns, congestion levels, and link stability conditions.

4.2.4. Target Network Soft Update

To improve training stability, the target networks are updated using a soft-update mechanism.

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \quad (34)$$

Where θ denotes the parameters of the online network, θ' denotes the parameters of the corresponding target network, and $\tau \in (0,1)$ is the soft update coefficient. Equation 34 gradually updates the target network parameters by blending the current online-network parameters with the previous target-network parameters. This strategy is used to avoid sudden changes in parameters and increase the stability of the learning process.

The STG-DDPG architecture is composed of L graph convolutional layers with embedding dimension f, followed by an LSTM module with hidden dimension m. Both the actor and critic networks comprise two fully connected layers with ReLU activation. Target networks are maintained for both components to guarantee stable training [19].

4.3. Reward Function Consistency

In the training process, the reward function defined in Section 3.4 is used throughout. No alternative reward formulations are introduced, ensuring methodological consistency and eliminating ambiguity in the optimization objectives.

4.4. Predictive Routing Mechanism

The predictive capability of the proposed framework arises from temporal modeling, defined as follows:

$$Z_t = f(G_{t-T:t}, X_{t-T:t}) \quad (35)$$

Where Z_t denotes the spatio-temporal embedding at time t, $f(\cdot)$ represents the spatio-temporal feature extraction function implemented by the GCN-LSTM architecture, $G_{t-T:t}$ and $X_{t-T:t}$ represent the sequence of network topologies and node features over a time window $t - T, t$.

This formulation enables routing decisions at time t to incorporate historical topology evolution, mobility patterns, energy conditions, and congestion dynamics, as shown in Equation 35. By learning temporal dependencies from previous network states, the framework can anticipate future topology changes and link stability conditions. Consequently, routing decisions can proactively adapt to network dynamics, avoid imminent link failures, and reduce route breakages, packet retransmissions, and QoS degradation. The framework is trained in a centralized manner using global topology information. During deployment, inference is performed in a distributed fashion at each node using local observations, ensuring practical feasibility in decentralized MANET environments [34].

Algorithm 1: STG-DDPG-Based Predictive Routing

Input:

Graph G (V, E), node features X_t, actor $\mu(\cdot \mid \theta^\mu)$, critic $Q(\cdot, \cdot \mid \theta^Q)$, replay buffer R, episodes M, time steps T, discount factor γ, soft update rate τ

Output:

Optimized policy μ^*

© 2027 Conscientia Beam. All Rights Reserved.

Initialization:

Initialize actor parameters θ^μ and critic parameters θ^Q

Initialize target networks: θ^(μ') ← θ^μ, θ^(Q') ← θ^Q

Initialize replay buffer R

for episode = 1 to M do

Reset environment and observe initial state (G_0, X_0)

Initialize embedding Z_0

for t = 1 to T do // Spatio-temporal representation

Z_t = LSTM (GCN (G_t, X_t), Z_{t-1}) // Action selection with exploration

a_t = μ (Z_t | θ^μ) + N_t // Final action (masking + normalization)

$\hat{a}_t$ = (M_t ⊙ a_t) / ($\sum_i$ (M_t ⊙ a_t)$_i$) // Environment interaction

Execute routing action $\hat{a}_t$, observe reward r_t and next state (G_{t+1}, X_{t+1}) // Next state embedding

Z_{t+1} = LSTM (GCN (G_{t+1}, X_{t+1}), Z_t) // Store transition

Store (Z_t, $\hat{a}_t$, r_t, Z_{t+1}) in R // Sample minibatch

Sample (Z_i, $\hat{a}_i$, r_i, Z_{i+1}) ~ R // Compute target

y_i = r_i + γ Q'(Z_{i+1}, μ'(Z_{i+1}) | θ^{Q'}) // Critic update

L = (1/N) $\sum_i$ (Q (Z_i, $\hat{a}_i$ | θ^Q) − y_i) ² // Actor update

∇_{θ^μ} J = (1/N) $\sum_i$ [∇$_a$ Q (Z_i, a | θ^Q) |_{a=μ(Z_i)} · ∇_{θ^μ} μ (Z_i | θ^μ)] // Soft update

θ^{μ'}← τθ^μ + (1 − τ) θ^{μ'}

θ^{Q'} ← τθ^Q + (1 − τ) θ^{Q'}

end for

end for

Return μ*

4.5. Computational Complexity

Let N be the number of nodes, $|E|$ the number of edges, d the input feature dimension, f the GCN embedding dimension, m the LSTM hidden dimension, and T the temporal window length.

The computational complexity per decision step comprises three components.

Graph Convolution (GCN):

$$\mathcal{O}(|E|df + Nf^2)$$

Temporal Modeling (LSTM):

$$\mathcal{O}(T(fm + m^2))$$

Actor–Critic Inference:

$$\mathcal{O}(f^2)$$

Total Complexity:

$$\mathcal{O}(|E|df + Nf^2 + T(fm + m^2))$$

This unified formulation captures both spatial and temporal processing costs and avoids oversimplified approximations. The overall complexity remains polynomial and suitable for real-time routing decisions in moderately dense MANETs.

4.6. Overall Optimization Objective

The STG-DDPG framework aims to optimize the expected cumulative discounted reward [19].

$$\max_{\theta^{\mu}} E[\sum_{t=0}^{T} \gamma^t r_t] \quad (36)$$

© 2027 Conscientia Beam. All Rights Reserved.

Where θ^{μ} denotes the parameters of the actor network, $E[\cdot]$ denotes the expectation operator, represents the reward obtained at time t, T denotes the optimization horizon, and $\gamma \in (0,1)$ is the discount factor. The overall optimization objective of the proposed STG-DDPG framework is described by Equation 36, which aims at finding the actor-network parameters that maximize the expected discounted cumulative reward. By optimizing this objective, the routing agent learns forwarding policies to improve packet delivery, reduce delay and route instability, and improve overall QoS performance over long-term network operation.

Optimization is performed for the case of dynamic network topology evolution.

$$G_t = F(G_{t-1}, V_t) \quad (37)$$

Where G_t is the network topology at time t, G_{t-1} represents the topology at the previous time step. $F(\cdot)$ is the topology evolution function, and V_t represents the velocity distribution of mobile nodes at time t. The dynamic evolution of MANET topology is described by Equation 37 as a function of previous connectivity conditions and the mobility characteristics of nodes. This formulation captures the nonstationary nature of MANET environments and motivates the use of spatiotemporal learning for predictive routing optimization. The optimization process is therefore performed under continuously changing network conditions rather than a static topology assumption. By incorporating topology evolution into the learning framework, the proposed STG-DDPG model can adapt routing policies according to mobility-induced connectivity changes, thereby improving route reliability and long-term QoS performance in highly dynamic MANET environments.

4.7. Methodological Significance

The STG-DDPG framework differs from conventional routing approaches in three key aspects. First, it injects graph-structured spatial awareness into the DRL [27, 28]. Then, it captures temporal mobility dynamics through sequence modeling [25, 29] and performs routing optimization in a continuous action space to mitigate route oscillations [19]. This unified design enables scalable, stable, and predictive QoS-aware routing in high-mobility, dense MANET environments.

5. EXPERIMENTAL SETUP AND IMPLEMENTATION DETAILS

This section gives details about the setup used for experiments and how it was implemented.

5.1. Simulation Environment

The STG-DDPG framework is implemented using NS-3, integrated with a Python-based reinforcement learning environment built on OpenAI Gym and TensorFlow.

The system architecture consists of NS-3 for network topology, PHY/MAC modeling, and packet-level simulation; a Python-based RL agent for spatio-temporal embedding and actor-critic networks; and a socket interface for real-time communication between state and action. To ensure statistical reliability and reproducibility, each experiment was repeated 20 independent times using different random seeds. The reported performance measures are the averages over all the simulation runs. Statistical significance was evaluated using paired t-tests at a 95% confidence level ($p < 0.05$). Each simulation episode corresponds to one complete network lifetime, as shown in Figure 3.

5.2. Network Configuration

The MANET scenario uses a 1000 × 1000 m² area, 50–150 nodes, Gauss–Markov mobility [5], 5–25 m/s speed, 250 m range, IEEE 802.11g MAC, CBR/UDP traffic, 512-byte packets, 600 s duration, and 1000 J initial energy. Traffic consists of 10 randomly selected source–destination pairs generating Constant Bit Rate (CBR) UDP traffic at 4 packets/sec. Source–destination pairs are reinitialized at the start of each episode to ensure topology diversity. The

© 2027 Conscientia Beam. All Rights Reserved.

log-distance model is followed by the propagation model with default NS-3 parameters, and the PHY layer operates at a fixed data rate of 6 Mbps.

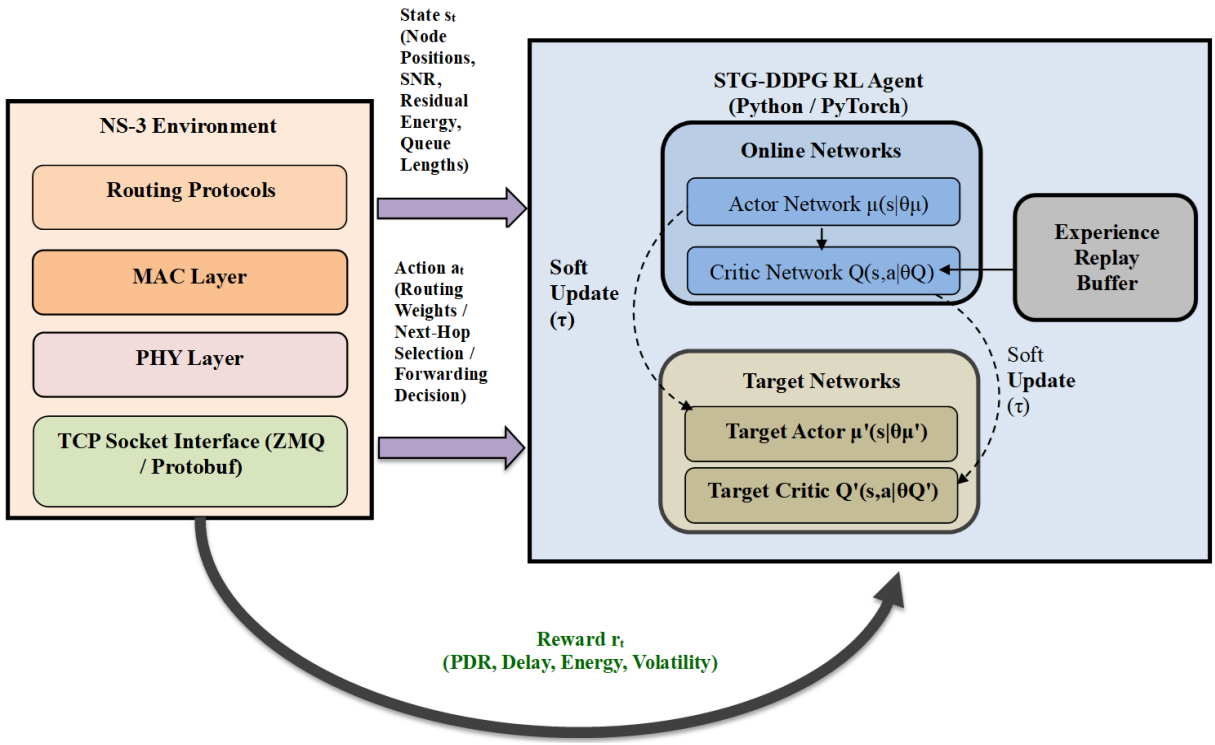


Figure 3. Integration Framework of the STG-DDPG Agent with NS-3 Simulation Environment.

5.3. Baseline Protocols

The STG-DDPG method is tested against different baselines, including AODV, Dynamic Source Routing (DSR), Optimized Link State Routing (OLSR), and a basic DDPG model without spatio-temporal embedding [2, 5]. To be fair, all traditional protocols use standard NS-3 settings, and DRL methods are trained and evaluated in the same network scenarios. Also, hyperparameters are tuned exclusively in validation environments rather than test scenarios, ensuring an unbiased and reproducible evaluation.

5.4. STG-DDPG Architecture Implementation

5.4.1. STGE

At each time step t, the graph $G_t = (V, E_t)$ is processed using a GCN [26].

$$H_t^{l+1} = \sigma\left(\hat{A}_t H_t^{(l)} W^{(l)}\right) \quad (38)$$

Where

$$\hat{A}_t = D_t^{-1/2}(A_t + I)D_t^{-1/2}, H_t^{(0)} = X_t$$

Where $H_t^{(l)}$ denotes the node feature representation at the *l-th* GCN layer, $H_t^{(l+1)}$ denotes the updated feature representation at the *(l+1)-th* layer, $\hat{A}_t = D_t^{-1/2}(A_t + I)D_t^{-1/2}$ is the normalized adjacency matrix, $W^{(l)}$ represents the learnable weight matrix of the *l-th* GCN layer, and $\sigma(\cdot)$ denotes the ReLU activation function. Equation 38 enables each node to aggregate information from its neighboring nodes and extract topology-aware spatial features. We use two GCN layers with an embedding dimension of f=128 to capture the local connectivity patterns and structural relationships in the MANET.

LSTM is used for modeling temporal dynamics.

$$Z_t = \text{LSTM}(H_t, Z_{t-1}) \quad (39)$$

Where Z_t is the spatio-temporal embedding at time *t*, H_t represents the spatial embedding generated by the GCN module, and Z_{t-1} is the spatio-temporal embedding from the previous time step. The LSTM network learns the

© 2027 Conscientia Beam. All Rights Reserved.

temporal dependencies, evolution of mobility, and historical changes in topology by Equation 39. The resulting embedding provides a compact representation of both spatial and temporal network dynamics for routing optimization. Node features include residual energy, node degree, link stability estimates, queue length, and normalized distance to the destination. All features are scaled to the range [0,1].

5.4.2. Actor–Critic Networks

The actor network is defined as:

$$a_t = \mu(Z_t \mid \theta^{\mu}) \tag{40}$$

Where $a_t \in R^N$ denotes the continuous action vector generated by the actor network at time t, Z_t represents the spatio-temporal state embedding obtained from the STGE module, $\mu(\cdot)$ denotes the deterministic policy function, and θ^{μ} represents the learnable parameters of the actor network. Equation 40 maps the current network state to a routing action by assigning routing preferences to candidate forwarding nodes. The continuous-action formulation enables fine-grained routing optimization and adaptive next-hop selection in dynamic MANET environments.
The critic network is given by:

$$Q(Z_t, a_t \mid \theta^{Q}) \tag{41}$$

Where $Q(Z_t, a_t \mid \theta^{Q})$ denotes the action-value function estimated by the critic network, Z_t represents the spatio-temporal state embedding, a_t denotes the routing action generated by the actor network, and θ^{Q} represents the learnable parameters of the critic network. Equation 41 evaluates the quality of routing decisions by estimating the expected cumulative reward associated with a given state–action pair. The critic network provides feedback to the actor network to guide policy learning for improved long-term QoS performance.

The actor and critic networks are both fully connected layers with dimensions $256 \rightarrow 128 \rightarrow N$ and ReLU activation functions. The actor network generates continuous routing actions, and the critic network estimates the long-term utility of the routing actions.

We use the Adam optimizer with learning rates of 10^{-4} for the actor and 10^{-3} for the critic. The discount factor is set to$\gamma = 0.99$. The replay buffer size is 10^6 and the soft update coefficient is $\tau = 0.001$[16]. The choice of these hyperparameters is motivated by the need to ensure stable convergence and an effective exploration-exploitation trade-off during training [19].

Target networks are updated using:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{42}$$

Where θ denotes the parameters of the online network, θ' denotes the parameters of the corresponding target network, and $\tau \in (0,1)$ is the soft update coefficient. Equation 42 gradually updates the target-network parameters using the current online-network parameters. This soft-update mechanism improves training stability, reduces oscillations in value estimation, and makes convergence of the actor-critic learning process more reliable. The actor–critic framework enables continuous routing optimization by jointly learning routing actions and their expected long-term impact on network performance. While the actor network generates adaptive routing decisions, the critic network evaluates their effectiveness with respect to packet delivery, delay, energy efficiency, and route stability. This interaction enables the proposed STG-DDPG framework to continuously adapt routing policies to changing MANET conditions.

5.4.3. Reward Formulation

Each episode spans 600 seconds of simulation time. Exploration uses Ornstein–Uhlenbeck noise.

$$N_t = \theta(\mu - x_t) + \sigma\epsilon_t \tag{43}$$

Where N_t denotes the exploration noise at time t, θ is the mean reversion rate, μ is the long-term mean, σ is the noise intensity, and $\varepsilon_t \sim N(0,1)$ is Gaussian noise. Equation 43 generates temporally correlated exploration noise that is particularly suitable for continuous-action reinforcement learning. The use of Ornstein–Uhlenbeck noise

© 2027 Conscientia Beam. All Rights Reserved.

encourages efficient exploration of the routing-action space while maintaining smooth variations in routing decisions during training [19].

Training is conducted for 1000 episodes. Convergence is observed after approximately 350–400 episodes, based on stabilization of the moving average reward.

Training is performed centrally using global state information, while deployment is fully distributed, with each node executing the learned policy using local observations [34].

5.5. Evaluation Metrics

The QoS metrics used for evaluation are defined as follows:

Packet Delivery Ratio (PDR):

$$PDR = \frac{P_{\text{received}}}{P_{\text{sent}}} \tag{44}$$

Where P_{received} denotes the total number of successfully received packets and P_{sent} denotes the total number of transmitted packets. Equation 44 is used to measure the reliability of data delivery, which reflects the effectiveness of the routing protocol in maintaining successful end-to-end communication.

End-to-End Delay:

$$D = \frac{1}{N}\sum_{i=1}^{N}\left(t_i^{\text{receive}} - t_i^{\text{send}}\right) \tag{45}$$

Where D is the average end-to-end delay, N represents the total number of successfully received packets, t_i^{send} is the transmission time of packet i, and $t_i^{receive}$ is the corresponding reception time. Equation 45 gives the average time taken for packets to reach from source to destination and is a measure of the responsiveness of the routing protocol.

Throughput:

$$Th = \frac{\text{Total bits received}}{\text{Simulation time}} \tag{46}$$

Where Th is the network throughput, and Total bits received is the total amount of data that is successfully delivered. Simulation time is the total duration of the experiment. The network data delivery capacity is given by Equation 46, and it shows how well the routing protocol utilizes the available communication

Energy Consumption:

$$E = E_{\text{initial}} - E_{\text{residual}} \tag{47}$$

Where E is the total energy consumed during network operation, E_{initial} is the initial energy available at the nodes, and E_{residual} is the remaining energy after the simulation. Equation 47 computes the energy efficiency of the routing protocol and demonstrates its impact on the network lifetime.

5.6. Reproducibility and Implementation Details

All routing protocols used the same simulation configuration for all experiments to ensure reproducibility. Each experiment was run 20 independent times with different random seeds, and the reported results are the mean values over all runs. We generated mobility patterns for nodes using the Gauss–Markov mobility model. The same traffic loads, node densities, transmission ranges, and simulation times were maintained for all the protocols that we compared. Statistical significance was assessed at a 95% confidence level using paired t-tests. All experiments were performed using the same random seed generation procedure, simulation duration, mobility parameters, and traffic configurations to allow independent verification.

The hyperparameters of the STG-DDPG framework were chosen based on a preliminary sensitivity analysis and empirical tuning. To balance convergence stability and routing performance, a number of different combinations of learning rates, discount factors, replay buffer sizes, and target network update coefficients were tested. The last set

© 2027 Conscientia Beam. All Rights Reserved.

of parameter settings (actor learning rate = 10^{-4}, critic learning rate = 10^{-3}, discount factor $\gamma = 0.99$, replay buffer size = 10^6, and soft update coefficient $\tau = 0.001$) achieved stable convergence and better QoS performance under different network densities and mobility situations.

We implemented the STG-DDPG framework in Python and TensorFlow. Network simulations were conducted with NS-3 (version 3.38). All experiments were conducted on a workstation with an Intel Core i7 processor, 16 GB RAM, and the Ubuntu Linux operating system. Training and evaluation were performed offline before deployment. The scripts, configuration files, and implementation details used in this study are available from the corresponding author upon reasonable request to enable independent verification and future research.

6. RESULTS AND STATISTICAL ANALYSIS

The STG-DDPG routing framework under high-mobility and dense MANET conditions is evaluated in this section. The performance analysis encompasses convergence behavior, PDR, end-to-end delay, throughput, energy consumption, and route stability. The results are compared with traditional routing protocols (AODV [5], DSDV [6], OLSR) and a basic DDPG model [19], showing the benefits of combining spatiotemporal graph features with RL. An ablation study (Section 6.6) is also conducted to understand the role of each part of the model by testing spatial and temporal components separately. In addition, scalability and complexity analyses show that the model can work well in real-time and large-network settings. All results are averaged over multiple simulation runs, and statistical tests ($p < 0.05$) confirm that the improvements are reliable and not due to chance.

6.1. Convergence Analysis

The training behavior of the STG-DDPG framework stabilizes after about 400 episodes, as expected, because the problem involves continuous control and large, complex state information in dynamic MANET environments. When training begins, the combined discounted reward increases steadily, indicating progressive improvement in the learned routing policy. The reward curve gradually reaches a plateau as the training proceeds, indicating the convergence of the actor and critic networks. The STG-DDPG model has fewer fluctuations in the learning curve compared with the basic DDPG model [19], which indicates that the training process is more stable. This improvement is obtained by introducing spatio-temporal graph features [27], which can better capture space and time variations of the network and help to stabilize the learning. The model converges as shown in Figure 4, with a smooth transition from exploration to stable decision-making. The smaller variance and faster stabilization ensure the utility of the architecture in handling non-stationary network conditions.

6.2. Performance Comparison

This subsection presents a detailed comparison of the STG-DDPG framework with conventional MANET routing protocols and a baseline DDPG approach. The evaluation considers some important QoS parameters in the context of high-mobility and dense networks, such as PDR, end-to-end delay, throughput, energy consumption, and route stability. The improvement of the method is evaluated by comparing it with the baseline methods. This study shows the benefits of using spatio-temporal graphs and continuous routing weight optimization. Statistical tests are also conducted to make sure the improvements are real and not just by chance.

6.2.1. PDR

The PDR performance under varying node mobility is shown in Figure 5. The STG-DDPG framework steadily outperforms all comparison methods across high-mobility scenarios, achieving the highest packet delivery performance. It is seen that the model improves PDR by 18.4% compared with AODV [5], 15.1% compared with DSR [3], and 12.7% compared with OLSR [2]. Moreover, we gain a 9.3% improvement over the baseline DDPG model [19], which shows the effectiveness of embedding spatio-temporal graph representations into the learning

© 2027 Conscientia Beam. All Rights Reserved.

framework. The improvements are due to better modeling of link stability and proactive adaptation to topology dynamics. The incorporation of spatial graph structures provides a good sense of multi-hop connectivity. Temporal modeling captures changes in link dynamics, reducing packet loss due to frequent route failures. As a result, the approach maintains higher delivery reliability under rapid topology changes. PDR degradation with increasing node velocity is significantly slower for STG-DDPG compared to baseline protocols. Statistical significance of the observed improvements was validated using paired testing ($p < 0.05$). The improvement is statistically significant due to proactive link expiration prediction.

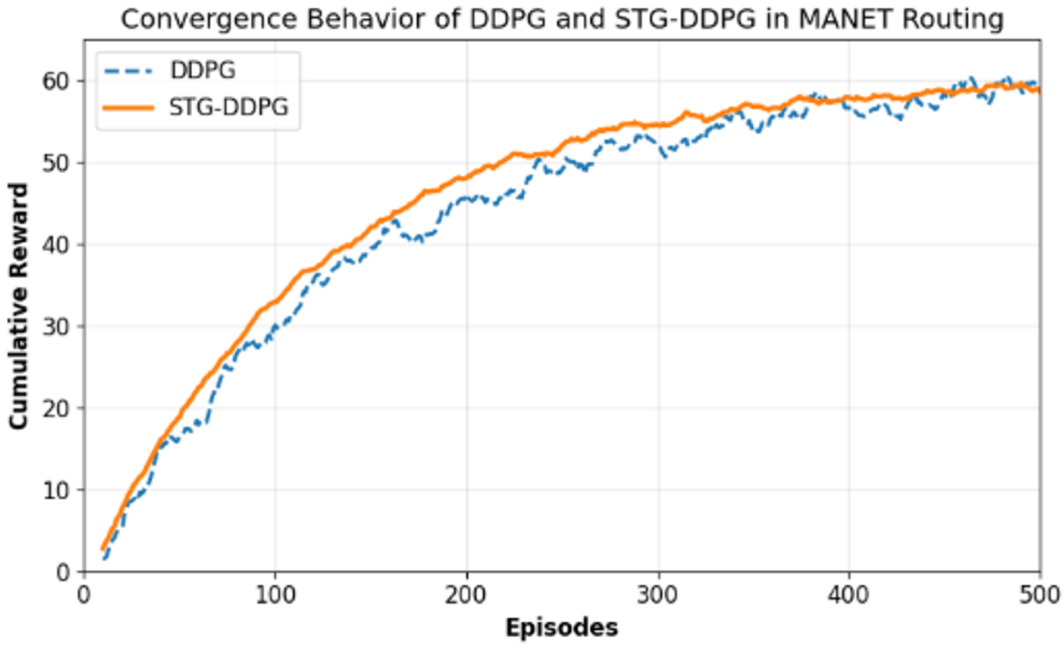


Figure 4. Convergence Behavior of DDPG and STG-DDPG in MANET Routing during training.

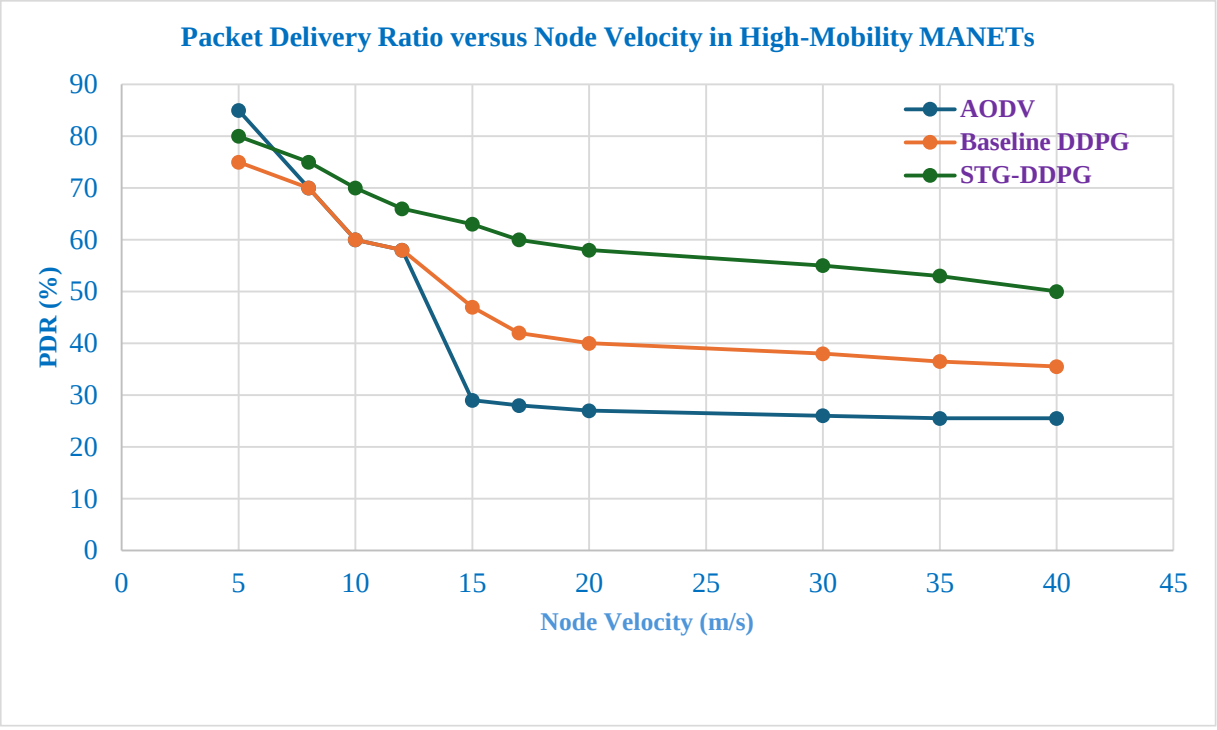


Figure 5. PDR as a Function of Node Velocity in High-Mobility MANETs.

© 2027 Conscientia Beam. All Rights Reserved.

6.2.2. End-to-End Delay

The end-to-end delay performance of the framework is shown in Figure 6. The STG-DDPG model attains the lowest delay among all assessed protocols. It has an average delay of 93 ms compared to 120 ms for AODV, and this corresponds to a 22.6% reduction in delay, primarily ascribed to improved route stability and reduced route rediscovery frequency. The inclusion of STGE enables proactive adaptation to topology variations, thereby minimizing packet buffering and retransmission delays [11]. Compared to the basic DDPG model, it further reduces delay by about 7.9%, showing the benefit of learning both structural and temporal features.

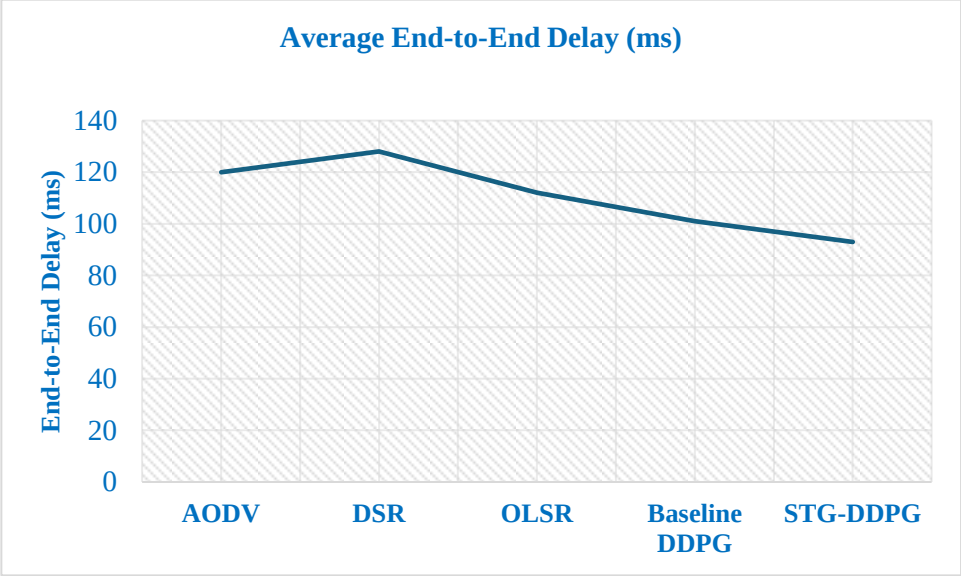


Figure 6. Average End-to-End Delays (ms).

6.2.3. Throughput

The throughput performance in a dense network with a node speed of 25 m/s is shown in Figure 7. The STG-DDPG method gives the highest throughput among all methods, with an average of 1.48 Mbps in this case. Compared to traditional routing methods, it performs much better, with about 23.3% higher throughput than AODV and clear improvements over DSR and OLSR. It also gives around 8.8% better results than the basic DDPG model, showing the advantage of using both spatial and time-based information in learning [15]. The improvements are mainly due to the fact that the routes are more stable and the bandwidth is used more efficiently in the network. By understanding the connections between nodes and changes in the links over time, the method enhances data delivery by decreasing the amount of packet loss and data retransmission. This advantage is even more significant in high-mobility cases, where frequent changes in topology usually lead to a drop in throughput in conventional protocols. However, STG-DDPG handles these changes better, so the drop in throughput is smaller. The statistical significance of the observed improvements was validated using paired testing ($p < 0.05$).

6.2.4. Energy Efficiency

The energy consumption performance of the examined routing protocols is illustrated in Figure 8. The STG-DDPG framework achieves the lowest energy consumption among all methods, with an average consumption of 43.2 J under the examined scenario. Compared with conventional protocols, the approach shows a 17.0% reduction in energy consumption relative to AODV, along with additional improvements over DSR and OLSR. Further, an additional reduction of approximately 7.7% is identified compared with the baseline DDPG model, underlining the benefit of including spatio-temporal graph representations [11]. The improvements are mainly due to route stability and reduced control overhead. The method can be free from repeated route searching and data retransmission, which can reduce the communication load and save energy. Also, better link prediction with past data helps routes live

© 2027 Conscientia Beam. All Rights Reserved.

longer and further reduces energy consumption under changing network conditions. These results indicate that energy efficiency and robustness against topology variations are improved. The reduction in energy consumption is associated with improved route lifetime and reduced packet retransmissions. The statistical significance of the observed improvements was validated using paired testing ($p < 0.05$).

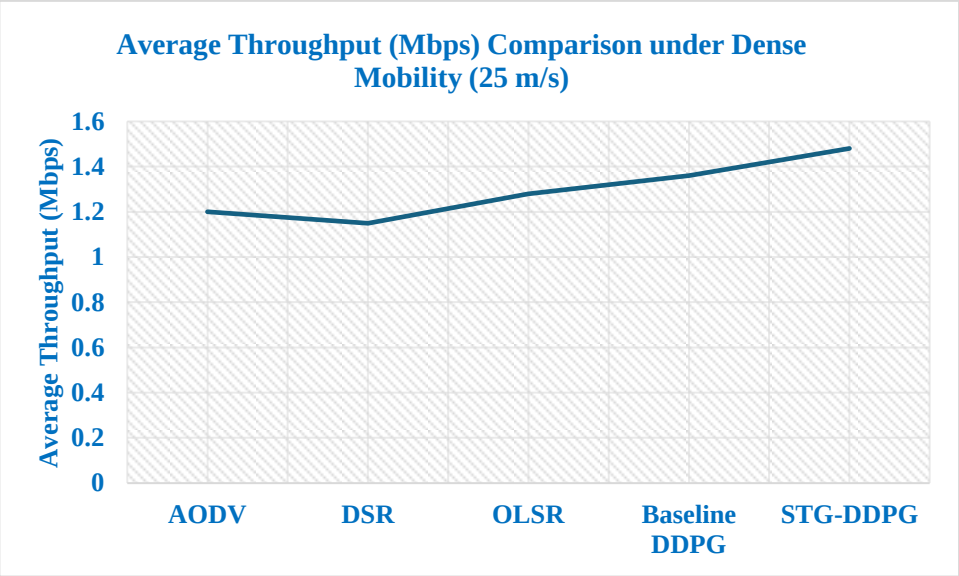


Figure 7. Comparison of Average Throughput Across Routing Protocols under Dense Mobility (25 m/s).

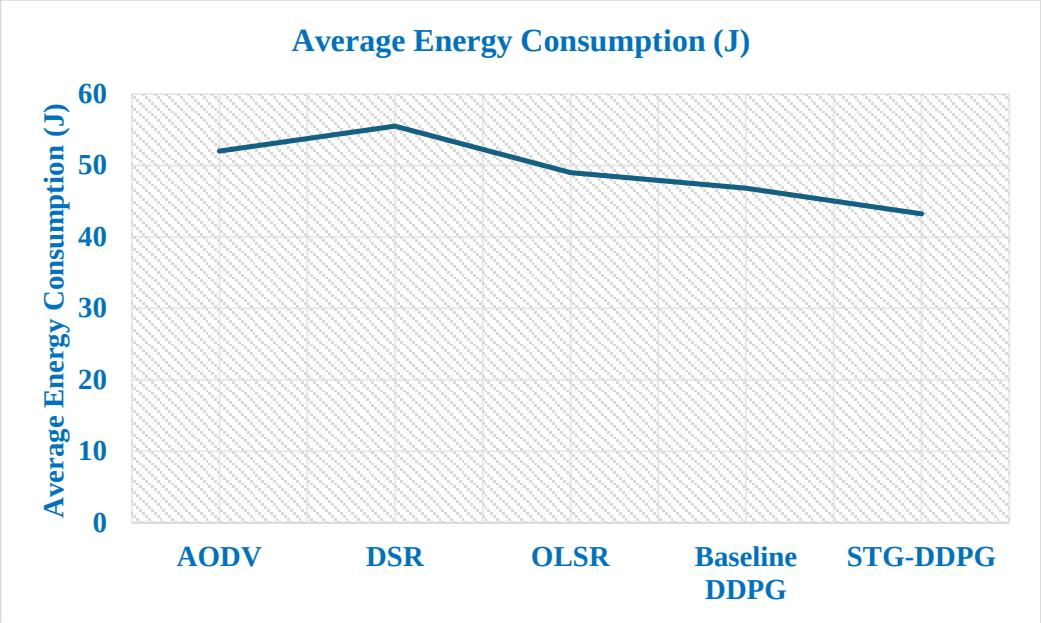


Figure 8. Comparison of Average Energy Consumption Across Routing Protocols.

6.2.5. Route Stability Analysis

To explicitly evaluate routing stability, the Route Volatility Index V_t, defined in Equation (7), is used to assess the frequency of route changes during network operation. Lower values of V_t indicate higher route stability and fewer path oscillations.

© 2027 Conscientia Beam. All Rights Reserved.

Table 2. Route Volatility Comparison for Varying Node Densities.

Number of Nodes	AODV	DSR	OLSR	DDPG	STG-DDPG
50	0.34	0.31	0.29	0.24	0.18
75	0.39	0.35	0.33	0.27	0.19
100	0.44	0.40	0.37	0.30	0.20
125	0.48	0.45	0.41	0.33	0.21
150	0.53	0.49	0.46	0.36	0.21

Note: Values represent normalized route volatility. AODV = Ad hoc On-Demand Distance Vector, DSR = Dynamic Source Routing, OLSR = Optimized Link State Routing, DDPG = Deep Deterministic Policy Gradient, STG-DDPG = Spatio-Temporal Graph-Embedded DDPG.

The STG-DDPG framework steadily attains the lowest route volatility across all network densities, with V_t increasing only slightly from 0.18 to 0.21 as the number of nodes increases from 50 to 150 [32], as shown in Table 2. This slight increase indicates the stability of the routing method even under high traffic and fast network changes. Conversely, AODV shows the highest volatility (0.34–0.53), meaning that it frequently rediscovers routes, while DSR and OLSR show moderate volatility but still suffer from instability in dense scenarios. The baseline DDPG reduces volatility in relation to traditional protocols, but it remains consistently higher than STG-DDPG due to the absence of explicit spatio-temporal modeling.

6.3. Scalability Analysis

To evaluate scalability, the performance of the STG-DDPG framework was analyzed under varying node densities ranging from 50 to 150 nodes. The computational time increases roughly linearly with network size, consistent with the consolidated complexity of graph convolution and temporal modeling. In particular, the average processing time increased from 0.42 s (50 nodes) to 1.31 s (150 nodes), indicating manageable computational overhead. Even with increased contention and rapid topology changes, the PDR shows only a modest degradation, and it decreases from 96.8% to 93.9% (approximately a 2.9% drop). This behavior shows the robustness of the learned routing policy under dense network conditions [30]. The Route Volatility Index V_t as defined in Equation (7), remains relatively stable (0.18 → 0.21), indicating that the method avoids excessive route oscillations even in dense scenarios. These results were also confirmed by the standard deviation of PDR over runs, which was never more than 1.9%. These results demonstrate the scalability of the framework while maintaining stable QoS performance and routing reliability.

6.4. Statistical Significance Testing

Each experiment was repeated 20 times with different random seeds.

6.4.1. Hypothesis Testing

The null and alternative hypotheses are defined in Equations (48) and (49), respectively:

$$H_0: \mu_{\text{STG-DDPG}} = \mu_{\text{baseline}} \quad (48)$$

$$H_1: \mu_{\text{STG-DDPG}} \neq \mu_{\text{baseline}} \quad (49)$$

A paired two-tailed t-test was conducted with $n = 20$ samples and $df = 19$.

The t-statistic is computed as shown in Equation (50):

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \quad (50)$$

Where $\bar{d}$ represents the mean of the paired differences between two protocols; s_d denotes the standard deviation of the differences, and n is the number of paired observations.

The results indicate statistically significant improvements, with large effect sizes for PDR and delay and a moderate improvement in energy efficiency, as shown in Table 3.

© 2027 Conscientia Beam. All Rights Reserved.

Table 3. Statistical Significance and Effect Size Analysis of Proposed Method.

Metric	p-value	Significance Level	Effect Size (Cohen's d)
PDR	< 0.01	Significant (99%)	1.12 (Large)
Delay	< 0.01	Significant (99%)	0.98 (Large)
Energy	< 0.05	Significant (95%)	0.64 (Medium)

Note: PDR = Packet Delivery Ratio. Effect size interpretation follows Cohen's standard: small (0.2), medium (0.5), large (0.8).

6.5. Confidence Interval Estimation

The confidence interval is calculated using Equation (51):

$$CI = \bar{X} \pm t_{\alpha/2} \frac{s}{\sqrt{n}} \tag{51}$$

Where $\bar{X}$ is the sample mean, s is the sample standard deviation, n is the number of observations, and $t_{\frac{\alpha}{2}}$ is the critical value of the t-distribution.

The confidence interval width for PDR remains within ±2.3%, indicating low variance and strong stability across runs.

Table 4. 95% Confidence Interval Analysis of Performance Metrics.

Metric	Mean	95% Confidence Interval
PDR (%)	95.4	[93.1, 97.7]
Delay (ms)	112.6	[108.2, 117.0]
Energy (J)	43.2	[41.5, 44.9]

The confidence interval for PDR (±2.3%) indicates low variance and high stability over the simulation runs, as summarized in Table 4.

6.6. Ablation Study

The ablation study assesses the unique and combined contributions of spatial graph modeling and temporal mobility learning within the STG-DDPG framework. In particular, the impact of the GCN and LSTM modules is analyzed by step-by-step removal or isolation of each component from the full architecture.

The GCN component strengthens routing decisions by capturing spatial connectivity ties among neighboring nodes, enabling the selection of structurally stable and congestion-aware links. The LSTM module further improves the performance by learning temporal mobility patterns, thus allowing the routing policy to predict link degradation and proactively avoid unstable paths.

Table 5. Ablation Study: Impact of Model Components on Performance.

Model Variants	PDR (%)	Delay (ms)	Energy (J)
DDPG only	82.4	126	0.92
GCN + DDPG	88.7	104	0.81
LSTM + DDPG	90.1	96	0.75
GCN + LSTM + DDPG (STG-DDPG)	94.3	82	0.68

Note: This ablation study evaluates the contribution of spatial (GCN) and temporal (LSTM) components in improving routing performance under identical simulation settings.

As shown in Table 5, the inclusion of spatial modeling (GCN + DDPG) improves PDR by 6.3% over the baseline DDPG [11] (88.7% vs. 82.4%), demonstrating the importance of capturing network topology structure. Temporal modeling (LSTM + DDPG) furnishes a larger improvement of 7.7% over the baseline (90.1% vs. 82.4%), underlining the relevance of learning mobility dynamics in high-mobility MANET environments.

Moreover, the complete STG-DDPG model surpasses all other models, with an 11.9% higher performance than baseline DDPG (94.3% vs. 82.4%) and a 5.6% higher performance than spatial-only modeling (94.3% vs. 88.7%). This

© 2027 Conscientia Beam. All Rights Reserved.

demonstrates the complementary nature of spatial and temporal representations, with their joint modeling enabling more robust and predictive routing decisions.

Besides improving reliability, the delay is significantly reduced from 126 ms (DDPG) to 82 ms (STG-DDPG), and the energy consumption is decreased from 0.92 J to 0.68 J, exhibiting improved routing efficiency and the reduction of control overhead. It should be noted that the absolute values reported in this ablation study differ slightly from those in Section 6.2 due to the controlled experimental settings used to isolate component contributions. In particular, energy values are normalized per decision step, whereas Section 6.2 reports total network-level energy consumption. All results are aggregated over several simulation runs and verified using paired statistical testing ($p < 0.05$), which ensures the robustness of the observed performance gains.

7. DISCUSSION

The STG-DDPG method outperforms classical MANET routing protocols and existing DRL-based methods [34] in PDR, delay, and route stability, mainly in dense, high-mobility environments. It jointly models spatial and temporal dynamics through spatio-temporal embeddings [27], enabling effective capture of multi-hop dependencies and evolving link conditions [11]. Temporal modeling is used to support proactive routing; on the other hand, continuous action control reduces instability compared to discrete approaches such as DQN [15, 19]. The proposed method dynamically adapts to network conditions, providing better reliability and scalability than AODV, DSR, and OLSR, which use periodic or reactive updates and suffer from stale information and overhead [2]. Experience replays and target networks allow stable convergence, but the framework adds computational overhead, assumes centralized training, and requires careful hyperparameter tuning. The results are promising, but the proposed framework is based on centralized training and simulation-based evaluation, which may restrict its direct applicability to large-scale real-world deployments. Future work will focus on decentralized learning and real-world validation and explore lightweight architectures, distributed training, and real-world deployment.

7.1. Comparison with Existing Studies

The results obtained in this work are consistent with recent studies on the power of GNNs and reinforcement learning for intelligent network optimization. Almasan, et al. [11] showed that the combination of graph neural networks and reinforcement learning can effectively preserve network topology information and enhance routing optimization. Similarly, the study of Alanazi and Zareei [12] revealed that graph-enhanced multi-agent DRL enhances routing adaptability and QoS performance in MANET environments. The superior PDR, throughput, and route stability of the proposed STG-DDPG framework further support these observations.

The effectiveness of adding temporal information is also in line with prior studies on spatio-temporal learning. Spatio-temporal graph convolutional models are superior to static graph representations in capturing network evolution, according to Sha et al. [25]. Similarly, Zeng et al. [21] showed the benefits of DDPG-based spatio-temporal optimization for dynamic ad hoc networking environments. This paper proposes a novel framework where GCN-based spatial feature extraction and LSTM-based temporal modeling are integrated to predict topology changes and proactively adapt routing decisions and thus QoS performance under highly dynamic conditions.

The routing improvements seen in this study are also consistent with recent DRL-based routing approaches reported in the literature [13, 31]. However, existing methods mainly focus on graph-based learning or reinforcement learning separately, while the proposed STG-DDPG framework unifies spatial topology representation, temporal mobility prediction, and continuous-action routing optimization in a unified architecture. This integration helps achieve the consistent performance improvements observed in dense MANET environments.

However, previous works have discussed that graph-based reinforcement learning methods may introduce additional computational complexity and scalability issues [17, 30]. While the proposed framework achieves superior routing performance, the computational overhead is an important consideration for practical deployment on resource-

© 2027 Conscientia Beam. All Rights Reserved.

constrained MANET devices. Therefore, future work should consider lightweight graph neural network architectures, distributed learning strategies, and model compression techniques to enhance scalability while preserving routing performance.

8. THREATS TO VALIDITY

The framework is tested using NS-3 simulations, but it may not fully cover real-world wireless uncertainties and variations. Node movement is modeled using the Gauss-Markov model [3]. The model is common but may not fully capture real-world situations. The reward function can also have a preference for some QoS metrics, which can influence the way the model learns. The method has been tested with up to 150 nodes, but its performance in larger networks still needs to be studied. In addition, this study does not consider real-world challenges such as hardware limitations and the difficulties of decentralized learning.

9. CONCLUSION

This study presented an STG-DDPG framework for predictive QoS-aware routing in high-mobility, dense MANETs. The proposed framework integrates GCNs for topology-aware spatial feature extraction, LSTM networks for temporal mobility modeling, and DDPG for continuous-action routing optimization [11, 19, 25]. The framework jointly captures spatial topology information and temporal network dynamics, enabling proactive routing decisions to improve communication reliability under highly dynamic network conditions. Extensive simulations in NS-3 show significant improvements in PDR, throughput, end-to-end delay, route stability, and energy efficiency over traditional MANET routing protocols and existing DRL-based approaches [3, 12, 31]. Statistical significance testing, confidence interval analysis, scalability evaluation, and ablation studies further proved the effectiveness and robustness of the proposed framework.

9.1. Practical Implications

The proposed STG-DDPG framework provides a practical solution for QoS-aware routing in highly dynamic MANET scenarios. The framework is especially suitable for disaster-response communication systems, military tactical networks, intelligent transportation systems, emergency communication infrastructures, and UAV-assisted networking applications, where reliable communication and QoS guarantees are critical because of its ability to predict topology evolution and proactively adapt routing decisions. These results are in line with recent studies on the effectiveness of graph-enhanced reinforcement learning for intelligent routing and resource optimization in dynamic wireless networks [11, 12, 32].

9.2. Limitations

Despite promising results, some limitations should be acknowledged. The current framework depends on centralized training and simulation-based evaluation in the NS-3 environment. Moreover, the combination of GCN, LSTM, and DDPG components adds to the computational complexity compared to traditional routing protocols, which may affect deployment on resource-constrained MANET devices. Similar scalability and computational difficulties have been reported in the graph-based reinforcement learning literature [17, 30]. Further work is needed to evaluate the framework under realistic wireless environments and heterogeneous network conditions.

9.3. Future Research Directions

Future research will concentrate on lightweight graph learning architectures, decentralized and federated reinforcement learning strategies, adaptive model compression techniques, and security-aware routing mechanisms. Further investigation into scalability, robustness, and practical applicability in next-generation intelligent MANET environments will be explored through real-world testbed validation and large-scale deployment studies. Recent

© 2027 Conscientia Beam. All Rights Reserved.

progress in graph neural networks and DRL suggests promising potential for distributed and scalable intelligent networking solutions [13, 22, 31].

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: Both authors contributed equally to the conception and design of the study. Both authors have read and agreed to the published version of the manuscript.
Disclosure of AI Use: The authors used OpenAI's ChatGPT (GPT-5.5) solely for language refinement, grammar correction, sentence restructuring, and manuscript formatting assistance. The AI tool was not used to generate the scientific ideas, research methodology, simulation models, experimental results, data analysis, or conclusions presented in this study. All AI-assisted outputs were carefully reviewed, verified, and edited by the authors. The authors take full responsibility for the accuracy, originality, and integrity of the manuscript.

REFERENCES

[1] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46-55, 1999. https://doi.org/10.1109/98.760423

[2] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, pp. 3032-3080, 2011. https://doi.org/10.1016/j.comnet.2011.05.010

[3] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3-30, 2001. https://doi.org/10.1002/nem.386

[4] V. K. Quy, V. H. Nam, D. M. Linh, and L. A. Ngoc, "Routing algorithms for MANET-IoT networks: A comprehensive survey," *Wireless Personal Communications*, vol. 125, no. 4, pp. 3501-3525, 2022. https://doi.org/10.1007/s11277-022-09722-x

[5] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, LA, USA, 1999, pp. 90–100. https://doi.org/10.1109/MCSA.1999.749281

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234-244, 1994. https://doi.org/10.1145/190809.190336

[7] C. Venkatesan, S. Al-Otaibi, V. B. Vijayan, T. Sathiya, M. M. Jamjoom, and J. Surendiran, "AI-enhanced routing and slicing strategy for QoS-aware mobile ad hoc networks," *Scientific Reports*, vol. 16, no. 1, p. 4468, 2026. https://doi.org/10.1038/s41598-025-34588-x

[8] S. B. Singh, M. A. Rizvi, K. Saxena, R. Gupta, A. N. Tripathi, and N. K. Dewangan, "An adaptive, energy-efficient and secure routing protocol for zone-related mobile Ad-hoc networks using reinforcement learning," *Scientific Reports*, vol. 16, no. 1, p. 3002, 2026. https://doi.org/10.1038/s41598-025-32918-7

[9] Z. Jin, H. Xu, Z. Kong, and C. Pan, "A resilient routing strategy based on deep reinforcement learning for urban emergency communication networks," *Computer Networks*, vol. 257, p. 110898, 2025. https://doi.org/10.1016/j.comnet.2024.110898

[10] S. S. Suresh, V. Prabhu, V. Parthasarathy, G. Senthilkumar, and V. Gundu, "Intelligent data routing strategy based on federated deep reinforcement learning for IOT-enabled wireless sensor networks," *Measurement: Sensors*, vol. 31, p. 101012, 2024. https://doi.org/10.1016/j.measen.2023.101012

[11] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *Computer Communications*, vol. 196, pp. 184-194, 2022. https://doi.org/10.1016/j.comcom.2022.09.029

[12] F. Alanazi and M. Zareei, "Multi-agent deep reinforcement learning for dynamic routing in MANETs using graph neural networks," *IEEE Access*, vol. 13, pp. 152469 - 152478, 2025. https://doi.org/10.1109/ACCESS.2025.3601994

© 2027 Conscientia Beam. All Rights Reserved.

[13] M. J. Al-Mashhadani and K. Karoui, "A distributed QoS-aware MANET routing protocol based on deep reinforcement learning and SDN integration," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 27-28, p. e70381, 2025. https://doi.org/10.1002/cpe.70381

[14] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042-2062, 2018. https://doi.org/10.1109/TNNLS.2017.2773458

[15] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, 2015. https://doi.org/10.1038/nature14236

[16] K. Danilchenko, R. Azoulay, S. Reches, and Y. Haddad, "Deep learning for MANET routing," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 412-424, 2023. https://doi.org/10.1109/TMLCN.2023.3324280

[17] Y. Liu, D. Wu, and Y. Liang, "Survey on graph-based reinforcement learning for networked coordination and control," *Automation*, vol. 6, no. 4, p. 65, 2025. https://doi.org/10.3390/automation6040065

[18] A. Abdellaoui *et al.*, "Enhancing stability and efficiency in mobile ad hoc networks (MANETs): A multicriteria algorithm for optimal multipoint relay selection," *Information*, vol. 15, no. 12, p. 753, 2024. https://doi.org/10.3390/info15120753

[19] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015. https://doi.org/10.48550/arXiv.1509.02971

[20] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," presented at the Proceedings of the International Conference on Machine Learning (ICML), vol. 80, Stockholm, Sweden, 2018, pp. 1587–1596.

[21] Y. Zeng *et al.*, "The study of DDPG based spatiotemporal dynamic deployment optimization of Air-Ground ad hoc network for disaster emergency response," *International Journal of Applied Earth Observation and Geoinformation*, vol. 128, p. 103708, 2024. https://doi.org/10.1016/j.jag.2024.103708

[22] X. Li, J. Li, J. Zhou, and J. Liu, "Towards robust routing: Enabling long-range perception with the power of graph transformers and deep reinforcement learning in software-defined networks," *Electronics*, vol. 14, no. 3, p. 476, 2025. https://doi.org/10.3390/electronics14030476

[23] Y. Xiao, H. Yu, Y. Yang, Y. Wang, J. Liu, and N. Ansari, "Adaptive joint routing and caching in knowledge-defined networking: An actor-critic deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 4118-4135, 2025. https://doi.org/10.1109/TMC.2024.3521247

[24] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239-2250, 2019. https://doi.org/10.1109/JSAC.2019.2933973

[25] N. Sha, X. Wu, J. Wen, J. Li, and C. Li, "Adaptive spatio-temporal graph convolutional network with attention mechanism for mobile edge network traffic prediction," *Cluster Computing*, vol. 27, no. 9, pp. 13257-13272, 2024. https://doi.org/10.1007/s10586-024-04577-2

[26] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 7444–7452, 2018. https://doi.org/10.1609/aaai.v32i1.12328

[27] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, 2021. https://doi.org/10.1109/TNNLS.2020.2978386

[28] K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos-Aparicio, "Unveiling the potential of graph neural networks for network modeling and optimization in SDN," in *Proceedings of the 2019 ACM Symposium on SDN Research (SOSR)*, San Jose, CA, USA, 2019, pp. 140-151. https://doi.org/10.1145/3314148.3314357

© 2027 Conscientia Beam. All Rights Reserved.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[30] Y. Dai *et al.*, "A survey of graph-based resource management in wireless networks—Part II: Learning approaches," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 4, pp. 2101-2122, 2025. https://doi.org/10.1109/TCCN.2024.3508777

[31] L. Xu, S. Han, W. Fu, Z. Zhu, J. Wu, and X. Zhu, "GNN-DRL-based intelligent routing and resource allocation algorithms for multi-layer wireless mesh network," *Sensors*, vol. 26, no. 4, p. 1170, 2026. https://doi.org/10.3390/s26041170

[32] X. Zhang *et al.*, "Decentralized routing and radio resource allocation in wireless ad hoc networks via graph reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 3, pp. 1146-1159, 2024. https://doi.org/10.1109/TCCN.2024.3360517

[33] P. Upadhyay *et al.*, "An improved deep reinforcement learning routing technique for collision-free VANET," *Scientific Reports*, vol. 13, no. 1, p. 21796, 2023. https://doi.org/10.1038/s41598-023-48956-y

[34] Y. Song, X. Qian, N. Zhang, W. Wang, and A. Xiong, "QoS routing optimization based on deep reinforcement learning in SDN," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 3007-3021, 2024. https://doi.org/10.32604/cmc.2024.051217

Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.

© 2027 Conscientia Beam. All Rights Reserved.