



Article

Hybrid Bi-LSTM and Deep Reinforcement Learning for AI-Native Predictive Mobility Management in High-Mobility 5G/6G Networks

Normtawan Phuttaamart and Sunisa Kunarak *

Department of Electrical Engineering, Srinakharinwirot University, Nakhon Nayok 26120, Thailand; normtawan.phuttaamart@g.swu.ac.th

* Correspondence: sunisaku@g.swu.ac.th

Abstract

Reliable mobility management is critical for high-mobility 5G/6G applications such as smart cities, intelligent transportation systems (ITS), and connected vehicles, where reactive 3GPP handover mechanisms become unreliable as user equipment (UE) velocity increases. This paper proposes a hybrid Bidirectional LSTM (Bi-LSTM) and Deep Reinforcement Learning (DRL) framework for AI-native predictive handover management: a Bi-LSTM encoder extracts a knowledge embedding and forecasts future network observations, forming a predictive state that a Dueling Double DQN (Dueling DDQN) with prioritized experience replay uses to select handover actions. A closed-loop, dual-frequency refresh mechanism periodically updates both modules from accumulated network experience, without manual retuning. The framework is evaluated via Monte Carlo simulation across Smart City, ITS, and Connected Vehicle scenarios and Urban, Suburban, and Mixed deployments. The proposed Bi-LSTM predictor reduces RMSE by 51.9–68.6% relative to a persistence baseline and by 8.4–25.6% relative to a Transformer predictor, with the largest gains under high mobility. End-to-end evaluation shows the framework achieves a throughput of 67.0 Mbps, a latency of 48.1 ms, a handover failure rate of 3.47%, and an average utility of 0.678, outperforming all ablation variants; disabling closed-loop refresh causes the largest degradation (throughput loss: 7.0%, higher failure rate: 33.1%). Generalization experiments show bounded degradation under unseen conditions, with closed-loop adaptation recovering utility from approximately 0.588 to 0.678 within 100 refresh epochs after a domain shift. These results show that integrating predictive knowledge extraction, adaptive decision-making, and continual refresh provides a robust, closed-loop architecture for AI-native mobility management for proactive 5G/6G mobility management.

Academic Editor: Xiangyun Long

Received: 26 August 2026

Revised: 22 September 2026

Accepted: 25 September 2026

Published: 28 September 2026

Keywords: AI-native networks; Bidirectional LSTM; deep reinforcement learning; Dueling Double DQN; knowledge extraction; mobility prediction; predictive handover management; 5G/6G networks

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid evolution of wireless communication technologies has transformed how people, devices, and intelligent systems interact within modern digital societies. Over the past decade, the deployment of fifth-generation (5G) mobile communication systems has

delivered substantial gains in network capacity, latency, spectral efficiency, and device connectivity, accelerating applications such as autonomous transportation, industrial automation, extended reality (XR), the Internet of Things (IoT), smart healthcare, and intelligent urban infrastructure. Nevertheless, continued growth in mobile data traffic, increasingly dense heterogeneous infrastructures, and highly dynamic mobility scenarios have exposed clear limitations in the mobility management mechanisms of current 5G networks [1–4].

Among these limitations, mobility management remains one of the most critical determinants of communication reliability in dynamic wireless environments. As user equipment (UE) moves across neighboring cells, the network must continuously determine the optimal timing and target cell for handover while preserving seamless connectivity. Handover mechanisms defined by the Third Generation Partnership Project (3GPP) rely primarily on static threshold-based parameters, including the Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP), hysteresis margin, and Time-to-Trigger (TTT). These mechanisms perform adequately under moderate mobility but degrade significantly in environments marked by rapid propagation changes, fluctuating traffic loads, and high user mobility [4,5].

These limitations become particularly pronounced in high-mobility applications such as connected vehicles, intelligent transportation systems, high-speed railways, unmanned aerial vehicles (UAVs), and smart city infrastructures, where network conditions change faster than reactive mechanisms can accommodate. This has made predictive mobility management systems that anticipate future network states rather than merely reacting to instantaneous measurements as a central research direction for Beyond-5G (B5G) and sixth-generation (6G) wireless networks, in which artificial intelligence (AI) is expected to become an integral architectural component rather than an external optimization tool [3,6–8].

Deep learning models, particularly Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) networks, have demonstrated strong capability in learning temporal dependencies from sequential network measurements, enabling accurate prediction of mobility trajectories, signal quality evolution, and future network conditions [9–12]. In parallel, Deep Reinforcement Learning (DRL) has emerged as an effective paradigm for sequential decision-making under uncertainty, allowing intelligent agents to learn handover policies through continuous interaction with the wireless network [13–16].

Despite this progress, most existing studies treat mobility prediction and handover decision-making as independent problems. Prediction models typically estimate future trajectories or signal strength without directly informing handover decisions, whereas DRL-based approaches often rely solely on instantaneous observations without exploiting predictive information about future network evolution. This separation constrains the ability of current AI-enabled mobility management systems to act proactively, and relatively few studies integrate prediction, knowledge extraction, decision-making, and continuous learning within a unified AI-native control framework [3,8].

Motivated by these observations, this study proposes a hybrid predictive mobility management framework that combines the temporal learning capability of Bi-LSTM with the adaptive decision-making capability of DRL. Rather than reacting only after communication quality deteriorates, the proposed framework predicts future mobility and wireless channel conditions before executing handover decisions, thereby enabling proactive network optimization for high-mobility 5G/6G environments. The architecture further incorporates knowledge extraction and sequential learning alongside policy optimization, aligning with the AI-native networking vision for future 6G systems.

1.1. Evolution from 5G to AI-Native 6G

The evolution from 5G toward 6G represents a paradigm shift from communication-centric architectures to intelligence-native network infrastructures. While earlier generations focused primarily on transmission capacity, coverage, and latency, 6G networks are expected to embed artificial intelligence directly into the network architecture, enabling autonomous operation, predictive optimization, and self-adaptive resource management [3,8,17]. The three service categories introduced by 5G enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC) support applications ranging from high-definition multimedia to industrial IoT. However, the exponential growth in connected devices, heterogeneous radio access technologies, and dynamic mobility patterns has pushed network management complexity beyond the reach of traditional optimization algorithms [1,3,8].

In response, research and standardization communities have introduced the concept of AI-native 6G, in which artificial intelligence is a fundamental architectural component rather than an auxiliary module. AI-native networks envision tighter integration of learning, reasoning, prediction, decision-making, and autonomous control with network functions, enabling continuous observation of environmental changes, knowledge extraction from multidimensional data, and adaptive resource optimization [8]. The IMT-2030 framework established by ITU-R identifies Artificial Intelligence and Communication (AIAC) as one of the envisaged usage scenarios and recognizes applicable AI-related capabilities as an important dimension of future IMT systems [18]. This direction motivates the tighter integration of learning, prediction, and network control functions in future wireless systems. In the scope of the present study, however, the term “AI-native” is used specifically at the mobility-management function rather than to imply AI integration across the complete protocol stack.

The evolution toward AI-native wireless systems is also extending beyond conventional terrestrial cellular networks. Lyu et al. [17] investigated the deployment of large AI models in the emerging low-altitude economy and proposed a hierarchical architecture to address the interaction among AI intelligence, communication, sensing, and computational resources. Their study highlights important challenges associated with deploying sophisticated AI models in dynamic wireless environments, including constrained onboard resources, discrepancies between offline-trained models and real-world operating conditions, and the need for scalable and adaptive service execution. These observations reinforce the broader motivation for adaptive AI-native network architectures capable of continuously responding to environmental and mobility variations.

A defining characteristic of AI-native networking is closed-loop autonomous control: rather than relying on manually configured parameters, AI-native systems continuously collect measurements, construct intelligent state representations, learn from historical experience and update decision policies as conditions change. This supports self-configuration, self-optimization, self-healing, and self-protection—key enabling capabilities for 6G [6,17]. Mobility management becomes considerably more demanding within this paradigm. Future networks will combine terrestrial base stations, unmanned aerial platforms, low-Earth-orbit satellites, intelligent reflecting surfaces, edge computing servers, and multiple radio access technologies, while users exhibit increasingly diverse mobility patterns across smart cities, transportation systems, and industrial cyber-physical systems. Mobility management algorithms must therefore jointly account for temporal mobility, spatial topology, traffic dynamics, propagation variation, and service quality requirements [2,7].

This has driven a broader shift from reactive to predictive mobility management, in which sequential learning models forecast future conditions and proactively optimize handover policies before degradation occurs. Bi-LSTM architectures are particularly

effective here, as they exploit both historical and future contextual information within sequential measurements, while DRL allows agents to refine mobility policies according to predicted evolution and long-term cumulative reward [12,15]. Together, these complementary capabilities provide the foundation for the hybrid AI-native mobility management framework developed in this study [2,8].

1.2. Mobility Management

Mobility management is a fundamental function of cellular systems, ensuring continuous service as UE moves across radio coverage areas while optimizing resource utilization, minimizing signaling overhead, and preserving QoS and QoE. Conventional mobility management comprises location management and handover management, with the latter becoming increasingly critical as radio access networks densify and wireless infrastructures diversify [4,6]. The ultra-dense small-cell deployments introduced in 5G have significantly increased handover frequency relative to earlier generations. While densification improves spectral efficiency and capacity, it also raises the likelihood of unnecessary handovers, radio link failures, handover interruptions, packet loss, and the well-known ping-pong effect, particularly under high-speed mobility [5,13]. The static parameters specified by 3GPP—RSRP, Reference Signal Received Quality (RSRQ), hysteresis margin, and TTT—cannot adapt effectively to varying user velocity, traffic density, radio interference, and network congestion.

To overcome the limitations of fixed handover parameters, Karmakar et al. [19] proposed a Learning-Based Intelligent Mobility Management (LIM2) mechanism for 5G and beyond networks. LIM2 combines Kalman-filter-based prediction of future signal quality with SARSA-based target-cell selection and online adaptation of the time-to-trigger and hysteresis margin. The study demonstrated the potential of jointly exploiting prediction and reinforcement learning to improve handover operation under high-mobility conditions. Nevertheless, its predictive component is primarily based on Kalman filtering and handover-parameter adaptation, leaving opportunities for richer temporal knowledge extraction from multidimensional network observations and deep reinforcement-learning-based long-term policy optimization.

This has motivated a shift toward predictive mobility management, in which future wireless conditions and user trajectories are estimated before communication degrades, allowing handover decisions to be executed proactively and reducing interruption time [2,10]. Deep learning models such as LSTM and Bi-LSTM capture temporal dependencies within sequential wireless measurements to predict future network states with high accuracy [9–12], while DRL enables agents to learn optimal handover policies by balancing throughput, latency, handover failures, energy efficiency, and signaling overhead as long-term cumulative rewards [13–16].

Nevertheless, although recent approaches have begun to couple prediction with adaptive handover control [19], most existing methods either rely on relatively lightweight predictors and handover-parameter adaptation or address mobility prediction and handover policy optimization as partially separated components. The joint exploitation of deep temporal prediction, learned knowledge representations, and long-term sequential decision-making within a continuously adaptive AI-native framework remains insufficiently explored. The present study addresses this gap by integrating Bi-LSTM-based future-state prediction and temporal knowledge extraction with Dueling DDQN-based handover optimization and closed-loop model refresh.

1.3. Smart City

Smart cities are among the most important application domains driving the evolution of 5G and 6G systems. By integrating advanced information and communication

technologies with urban infrastructure, smart cities aim to improve sustainability, operational efficiency, public safety, environmental monitoring, healthcare, transportation, and quality-of-life services that require ubiquitous wireless connectivity supporting massive numbers of heterogeneous devices under strict latency, reliability, and scalability requirements [2,20]. Smart city environments are highly dynamic and heterogeneous: users move continuously among residential areas, business districts, transportation corridors, industrial zones, and suburban regions, producing frequent transitions across radio environments, while smartphones, connected vehicles, autonomous robots, surveillance systems, UAVs, and billions of IoT devices add substantial traffic diversity [1,3].

Managing mobility under such conditions is challenging as network conditions fluctuate continuously with user density, traffic demand, propagation characteristics, and interference, making conventional threshold-based mechanisms insufficient. Future smart city infrastructures therefore require systems capable of learning environmental characteristics, predicting network conditions, and autonomously adapting communication strategies [5,8]. The AI-native networking paradigm envisioned for 6G aligns naturally with these requirements, enabling continuous knowledge extraction from large-scale heterogeneous datasets and autonomous optimization of mobility management, traffic engineering, and resource allocation forming the technological foundation for self-evolving smart city communication infrastructures [6,20].

1.4. Intelligent Transportation

The rapid development of Intelligent Transportation Systems (ITS) has intensified demand for reliable wireless communication under high-speed mobility. Applications, such as autonomous driving, cooperative perception, intelligent traffic control, remote vehicle monitoring, platooning, emergency response, and infrastructure-assisted navigation, require simultaneous ultra-low latency, high reliability, high data rates, and uninterrupted service continuity [3,7]. High-speed mobility substantially complicates wireless communication as vehicles cross multiple cells at highway speeds. The channel characteristics change rapidly, producing signal fluctuations, Doppler effects, frequent handovers, and radio link failures. Reactive handover mechanisms often initiate too late under these conditions, causing interruptions and degraded performance [5,10].

Temporal sequence learning and reinforcement learning together address these challenges by predicting future trajectories, signal evolution, and network conditions while optimizing mobility decisions for long-term performance, allowing proactive resource allocation and handover scheduling before degradation occurs [11,15].

Intelligent transportation is also a flagship application for AI-native 6G, where edge intelligence, digital twins, network automation, semantic communications, and distributed AI are expected to enable autonomous cooperation among vehicles, roadside units, and communication infrastructure, driving mobility management from simple signal-based handover toward knowledge-driven autonomous orchestration [8,20].

1.5. Connected Vehicles

Connected vehicles present one of the most demanding mobility scenarios in future wireless systems. Vehicle-to-Everything (V2X) communications encompassing Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) interactions enable cooperative driving, collision avoidance, intelligent traffic management, remote diagnostics, and autonomous transportation services [7,15]. Compared with conventional mobile users, connected vehicles generate rapidly changing mobility patterns with frequent transitions between heterogeneous infrastructures. Variations in speed, traffic density, road topology, and environmental conditions introduce substantial channel uncertainty, making seamless connectivity difficult to

maintain through conventional handover mechanisms, particularly in mixed urban–sub-urban environments [5,15].

AI has shown considerable potential here: Deep learning captures temporal mobility patterns and channel evolution, while reinforcement learning enables adaptive strategies under uncertainty. However, existing AI-enabled methods often optimize isolated objectives, such as throughput or latency, without jointly considering mobility prediction, sequential decision optimization, and knowledge extraction within a unified framework [15,16]. As AI-native 6G evolves, connected vehicles are expected to become intelligent network entities interacting with edge servers, cloud platforms, digital twins, and autonomous radio access networks. Realizing this vision requires predictive mobility management frameworks that integrate temporal sequence learning, knowledge representation, and adaptive reinforcement learning into an end-to-end decision-making architecture, improving handover reliability and providing a scalable foundation for autonomous, self-optimizing, context-aware transportation ecosystems [5,8,15].

2. Related Works

This section reviews prior work relevant to the proposed framework, organized into four strands: AI-native networking architectures, general handover management surveys, deep learning-based mobility and signal prediction, and deep reinforcement learning for handover decision-making. Table 1 summarizes all twenty-seven studies at the end of this section.

Table 1. Comparison of recent AI-based mobility management studies.

Ref.	AI Prediction	Reinforcement Learning	Predictive Handover	Knowledge Extraction	AI-Native	High Mobility	Smart City/ITS	Limitation
Cui et al. (2025) [1]	✗	✗	✗	✓	✓	✗	Partial	Review only
Ogenyi et al. (2025) [2]	✗	✗	✗	✓	✓	Partial	✓	Survey
Li et al. (2025) [3]	Partial	✗	✗	✓	✓	Partial	Partial	Architecture only
Mollet et al. (2021) [4]	Partial	Partial	Partial	✗	✗	Partial	✗	Survey
Amirova et al. (2025) [5]	✗	✗	Partial	✗	Partial	✓	✓	Survey
Safiullin et al. (2026) [6]	✗	✓	Partial	✗	✓	Partial	✗	AI-RAN handover; no temporal prediction
Junejo et al. (2025) [7]	Partial	Partial	✓	✗	Partial	✓	✓	No temporal prediction
Feng et al. (2026) [8]	Partial	Partial	✗	✓	✓	Partial	Partial	AI-RAN overview; no mobility algorithm
Baz et al. (2024) [9]	✓	✗	Partial	✗	✗	✓	✗	Prediction only
Sulaiman & Al-Raweshidy (2025) [10]	✓	✗	✓	✗	✗	✓	✗	No RL
Dzaferagic et al. (2025) [11]	✓	✗	✓	✗	Partial	✓	✗	O-RAN only
Park et al. (2024) [12]	✓	✗	✓	✗	Partial	✓	✗	No adaptive optimization
Prado et al. (2023) [13]	✗	✓	✓	✗	Partial	✓	✗	Reactive RL
Park & Yoon (2024) [14]	✓	✓	Partial	✗	Partial	Partial	✗	Traffic prediction only

Shahid et al. (2025) [15]	✓	Partial	✓	✗	Partial	✓	✗	Limited knowledge representation
Alraih et al. (2023) [16]	Partial	Partial	Partial	✗	✗	✓	Partial	Survey
Karmakar et al. (2023) [19]	✓	✓	✓	✗	✗	✓	✗	Kalman-based prediction with adaptive TTT/HM; lacking deep temporal knowledge representation and joint closed-loop model adaptation.
Mao et al. (2017) [20]	✗	✗	✗	✓	✓	✗	✗	Edge computing survey only
Chen et al. (2023) [21]	✗	✗	✗	✓	✓	✗	✗	Foundation-model architecture only
Kholmatov et al. (2026) [22]	✗	✗	✗	Partial	✓	✗	✗	RAN architecture; no mobility model
Chabira et al. (2025) [23]	Partial	Partial	Partial	✗	Partial	✓	✓	Review
Yalin & Mumcu (2026) [24]	✗	✗	✗	✗	✗	✓	Partial	Survey
Lee et al. (2024) [25]	✓	✗	Partial	✗	✗	✓	✗	Bandwidth prediction only; no RL
Mehregan & De Grande (2025) [26]	Partial	✗	✓	Partial	✗	✓	✓	GCN-based handover; no RL policy learning
He et al. (2026) [27]	✓	✗	✓	✗	✗	✓	✗	LEO satellite only; no RL
Wang & Li (2024) [28]	✗	✓	Partial	✗	✗	✓	✗	mmWave dual connectivity only; no prediction
Amaira et al. (2024) [29]	✗	✓	✗	✗	Partial	Partial	✗	Reactive RL; slicing-specific
Amaira & Zarai (2025) [30]	✗	✓	✗	✗	Partial	✓	✓	Reactive RL; vehicular slicing
Proposed Work	✓	✓	✓	✓	✓	✓	✓	Unified AI-native predictive mobility management framework

Note: ✓ denotes that the corresponding feature or capability is explicitly considered in the study, whereas ✗ denotes that it is not considered.

2.1. AI-Native Networking and Knowledge-Driven 6G Architectures

The first strand of the literature examines the architectural transition toward AI-native 6G, in which artificial intelligence becomes a core component of the network rather than an auxiliary optimization layer. Cui et al. [1] and Li et al. [3] outline broad AI-native 6G frameworks in which learning, prediction, knowledge extraction, and autonomous control are integrated across the communication system. Ogenyi et al. [2] further review semantic communications, reconfigurable intelligent surfaces, and edge intelligence within the AI-native paradigm. More recently, Feng et al. [8] survey the emerging AI-RAN paradigm, including AI-for-RAN, AI-and-RAN, and AI-on-RAN functions, with particular attention to distributed learning, edge inference, orchestration, and mobility-related convergence challenges. Mao et al. [20] provide the communication perspective on mobile edge computing that underlies much of the later edge-intelligence literature, while Chen et al. [21] propose a foundation-model-based cloud-edge-end collaboration framework for

native AI in 6G, and Kholmatov et al. [22] introduce an on-RAN edge-intelligence architecture. Collectively, these studies establish the architectural motivation for AI-native networking and knowledge-driven control; however, they do not provide an end-to-end predictive handover framework that jointly integrates temporal prediction, explicit knowledge representation, reinforcement-learning-based decision-making, and continual mobility adaptation.

2.2. Handover and Mobility Management: Surveys and Conventional Approaches

A second strand surveys handover and mobility management techniques without proposing new predictive models. Mollel et al. [4] review machine learning applications to handover management in 5G and beyond, cataloguing algorithmic families without extending them into a unified framework. Amirova et al. [5] and Junejo et al. [7] focus specifically on handover decision-making in ultra-dense and smart city networks, highlighting the density-driven challenges as frequent handovers, ping-pong effects, and radio link failures that motivate predictive approaches. Alraih et al. [16] similarly survey handover optimization techniques for beyond-5G networks, comparing threshold-based, fuzzy-logic, and early machine learning solutions. More recent surveys continue to map this space: Chabira et al. [23] review AI-driven handover and load-balancing techniques for ultra-dense 5G/6G networks, explicitly identifying standardized datasets and context-aware mobility strategies as open research gaps, while Yalin and Mumcu [24] survey mobility management techniques for beyond-5G ultra-dense deployments more broadly. These surveys are valuable for mapping the problem space and for confirming that conventional threshold-based mechanisms are inadequate under high mobility, but being reviews, they do not themselves contribute to predictive or reinforcement-learning-based solutions.

2.3. Deep Learning-Based Mobility and Signal Prediction

A third strand applies deep sequence models to forecast user trajectories, signal strength, and channel conditions. As a classical alternative to data-driven prediction, state-space models such as Kalman filtering recursively estimate UE motion states and extrapolate future positions under an assumed mobility model. Such model-based predictors provide an interpretable reference for distinguishing the benefit of prediction itself from the additional ability of learned sequence models to capture nonlinear temporal dynamics. Baz et al. [9] use a deep residual LSTM model to enhance mobility management in 5G networks, while Sulaiman and Al-Raweshidy [10] propose a predictive handover mechanism explicitly designed for seamless mobility in 5G and beyond. Dzaferagic et al. [11] demonstrate machine learning-based handover prediction over a real O-RAN deployment using the RAN Intelligent Controller, providing valuable field validation, and Park et al. [12] frame the broader paradigm shift from reactive to proactive handover using AI/ML. Complementary recent work extends deep sequence learning to new prediction targets: Lee et al. [25] develop a Bi-LSTM model with time distribution for per-network bandwidth prediction across 3G/4G/5G handovers, Mehregan and De Grande [26] apply a graph convolutional network to jointly model vehicles and base stations for throughput-oriented handover in dense vehicular networks, and He et al. [27] propose PreHO, a predictive handover mechanism for low-Earth-orbit satellite networks. These studies consistently report strong predictive capability for future network states and broaden the scope of predictive mobility modeling beyond terrestrial LSTM/Bi-LSTM approaches; however, as summarized in Table 1, none couples its predictive output to an adaptive decision-making mechanism—the predicted trajectories or signal levels are not used to directly optimize the handover policy itself.

2.4. Deep Reinforcement Learning for Adaptive Handover Decision-Making

A fourth strand applies DRL to learn handover policies directly. Prado et al. [13] propose a proportionally fair mobility management scheme using reinforcement learning in 5G networks, and Park and Yoon [14] combine federated LSTM-based traffic prediction with DRL for base station switching, representing one of the few studies that already couples prediction with decision-making, albeit for traffic load switching rather than user-level handover. Shahid et al. [15] incorporate mobility prediction directly into the handover procedure to mitigate frequent handovers in small-cell networks, making it the closest existing work to the framework proposed in this study. Recent work has also moved DRL closer to AI-native radio access operation: Safiullin et al. [6] develop a reinforcement-learning-based handover algorithm for 5G/6G AI-RAN, but without the explicit Bi-LSTM future-state prediction and knowledge-embedding mechanism considered here. Additional DRL-based studies address specific deployment contexts: Wang and Li [28] apply double-deep Q-learning to dual-connectivity handover in mmWave heterogeneous networks, and Amaira et al. [29,30] extend DRL-based handover management to sliced 5G and sliced vehicular networks. These works confirm the effectiveness of DRL for context-specific handover optimization; however, they generally construct the agent state from instantaneous or short-horizon observations and do not jointly provide explicit temporal knowledge extraction, proactive future-state representation, and closed-loop representation refresh across heterogeneous smart city, ITS, and connected vehicle scenarios.

Table 1 summarizes these twenty-seven studies according to whether they perform AI-based prediction, reinforcement learning, predictive (proactive) handover, knowledge extraction, and AI-native design, and whether they address high-mobility and smart city/ITS conditions. The comparison shows a consistent pattern: architectural and survey studies ([1–3,8,16,20–24]) establish AI-native and mobility-management requirements but do not implement a unified predictive mobility algorithm; prediction-focused studies ([9–12,25–27]) achieve accurate forecasting but generally do not optimize handover decisions through an adaptive policy; and DRL-based handover studies ([4,13–15,28–30]) learn adaptive decisions but largely without an explicit predictive knowledge representation and dual-frequency closed-loop refresh. Table 1 summarizes the reviewed literature across several dimensions relevant to predictive mobility management, including AI-based prediction, reinforcement learning, predictive handover, knowledge representation, AI-native design, high-mobility operation, and smart city/ITS applicability. The comparison is intended to characterize the methodological emphasis of existing studies rather than to define novelty solely by the presence or absence of individual features.

To complement the literature-level comparison in Table 1, Table 2 provides a more explicit algorithmic and architectural comparison of the representative handover paradigms considered in this study. Rather than relying solely on binary feature-presence criteria, the comparison focuses on the information available to the handover decision process, predictive outputs, temporal representation, learning objectives, and model-update mechanisms. The contribution of the present work therefore does not arise from introducing Bi-LSTM or deep reinforcement learning individually, but from coupling prediction, temporal representation, sequential decision-making, and continual adaptation within a unified mobility-management architecture.

Table 2. Algorithmic and architectural comparison of representative mobility-management paradigms.

Method	Decision State/Input	Prediction	Temporal Representation	Decision Mechanism	Learning Objective	Update Strategy
Reactive 3GPP (B1)	Current radio measurements	None	None	TTT/HM rules	None	Fixed
DRL-only (B2)	$\mathbf{o}_u(t)$	None	None	Dueling DDQN	TD loss	Policy learning
Prediction-only (B3)	Current/predicted measurements	Bi-LSTM	Internal predictor features	Rule/threshold based	Prediction loss	Predictor learning
Prediction-assisted DRL (B4)	Current + predicted information	Predictive model	No explicit $\mathbf{h}_t$ exposed to decision state	DRL	Prediction + DRL objectives	Method-specific
Proposed	$[\mathbf{o}_u(t); \hat{\mathbf{o}}_u(t + \Delta); \mathbf{h}_t]$	Bi-LSTM future-state prediction	Explicit learned temporal representation $\mathbf{h}_t$	Dueling DDQN + PER	Prediction loss + TD loss	Dual-frequency closed-loop refresh K_1, K_2

As shown in Table 2, the distinction among the evaluated paradigms lies primarily in how temporal information is made available to the handover decision process. Reactive 3GPP and DRL-only methods operate without explicit future-state information, whereas prediction-only approaches forecast future conditions without optimizing the long-term consequences of sequential handover actions. Prediction-assisted DRL partially bridges these paradigms by incorporating predictive information into reinforcement-learning-based decision-making. The proposed framework further couples three complementary information sources through the predictive state $\mathbf{s}_u(t) = [\mathbf{o}_u(t); \hat{\mathbf{o}}_u(t + \Delta); \mathbf{h}_t]$, where $\mathbf{o}_u(t)$ represents the current network and mobility observation, $\hat{\mathbf{o}}_u(t + \Delta)$ provides an explicit estimate of future conditions, and $\mathbf{h}_t$ supplies a compact learned representation of recent temporal dynamics. The Bi-LSTM prediction objective and Dueling-DDQN temporal-difference objective therefore serve complementary functions, while the dual-frequency refresh mechanism allows the predictive representation and decision policy to adapt at different timescales. Accordingly, the principal contribution is the integration and closed-loop coordination of these components rather than the introduction of a new standalone prediction or reinforcement-learning algorithm.

2.5. Research Motivation

The transition toward AI-native 6G networks has transformed artificial intelligence from an auxiliary optimization tool into a core architectural component responsible for autonomous network management, predictive decision-making, and continuous knowledge acquisition [6,8,20]. Within this paradigm, mobility management is among the most consequential network functions, directly shaping communication continuity, resource utilization, service reliability, and user experience in highly dynamic environments [2,5,10]. Existing machine learning approaches to handover performance remain fragmented along two largely separate lines. Temporal learning models such as LSTM and Bi-LSTM variants accurately forecast user trajectories, signal strength, and channel quality [9–12], but their outputs are rarely incorporated into adaptive decision-making. Conversely, DRL-based approaches learn adaptive handover strategies that jointly optimize communication quality, latency, throughput, signaling overhead, and handover failures [13–16], but typically construct network states from instantaneous observations alone,

causing agents to react only after quality has already deteriorated rather than anticipate future conditions [13,15].

A further limitation is the absence of explicit knowledge extraction. AI-native networks require not only prediction and adaptive decision-making but also the intelligent extraction of representative knowledge from multidimensional measurements as temporal mobility patterns, propagation characteristics, traffic dynamics, user behavior, and network topology [6,20] to construct informative state representations and improve learning efficiency and generalization. Most existing studies instead treat feature engineering as a preprocessing step rather than explicitly exposing learned temporal information to the downstream handover decision process, thereby limiting the ability of the decision agent to exploit recent mobility and network dynamics. These gaps are amplified in practical deployments. Smart city and intelligent transportation environments expose users to continuously varying cell densities, traffic loads, propagation conditions, and service requirements across mixed urban–suburban regions [2,3], and optimal handover decisions in such settings depend on anticipated future network evolution as much as on current signal quality [2,7,15]. Reactive mechanisms are therefore insufficient for autonomous driving, connected vehicles, ITS, industrial automation, and immersive XR applications that demand uninterrupted, low-latency, high-reliability communication [2,6,13]. Although mobility management is gradually shifting from signal-strength-based optimization toward context-aware, prediction-driven decision-making, few studies jointly investigate temporal prediction, sequential reinforcement learning, and AI-native knowledge extraction within a single end-to-end framework.

Motivated by these gaps, this study proposes a hybrid Bi-LSTM and Deep Reinforcement Learning framework for AI-native predictive mobility management in high-mobility 5G/6G networks. The framework integrates Bi-LSTM-based temporal prediction with DRL-based adaptive handover decision-making, predicting future mobility patterns and channel evolution before handover execution so that the learning agent optimizes policies proactively rather than reactively. A learned temporal representation mechanism further constructs a compact latent representation from heterogeneous network measurements, providing the downstream decision agent with information on recent mobility and network dynamics that is not explicitly available from instantaneous observations alone. By jointly integrating temporal prediction, learned representation, and reinforcement-learning-based decision-making, the proposed framework establishes a unified predictive-decision architecture for mobility management in smart city, intelligent transportation systems, connected vehicle, and B5G/6G wireless environments.

2.6. Research Gap

Although artificial intelligence has become a promising solution for mobility management in B5G and 6G networks, existing studies remain fragmented across different research directions. Most studies address only one stage of the mobility management pipeline prediction, handover optimization, or reinforcement learning-based decision-making without forming an end-to-end AI-native framework. Deep learning approaches based on LSTM or Bi-LSTM achieve high prediction accuracy for user trajectories, signal strength, and future network conditions, but generally terminate at the prediction stage without using this information to optimize subsequent handover decisions, leaving predictive accuracy underexploited for long-term communication performance. DRL-based mobility management, conversely, adapts handover policies effectively under dynamic conditions, but typically constructs state representations from instantaneous observations alone current RSRP, SINR, throughput, and cell load without incorporating predicted future evolution, leaving agents reactive rather than proactive, particularly under high-speed mobility. A further limitation concerns how temporal information is represented

and exposed to the handover decision process. Although sequence-learning models can implicitly encode temporal dependencies in their hidden states, these representations are not always explicitly incorporated into the downstream handover decision state. Consequently, potentially useful information about recent mobility and network evolution may remain internal to the prediction model rather than being directly available to the decision-making agent. This motivates a tighter coupling between temporal prediction, learned state representation, and adaptive handover optimization.

Finally, few studies jointly evaluate smart city, intelligent transportation, and connected vehicle scenarios, where communication conditions vary substantially across mixed urban–suburban regions. Such settings require simultaneous modeling of temporal mobility, heterogeneous radio access technologies, dynamic traffic, and adaptive handover strategies, whereas existing work typically assumes simplified mobility patterns or homogeneous network conditions. Taken together, the literature suggests that the remaining challenge is not simply the absence of individual prediction or reinforcement-learning components, but the limited integration among future-state prediction, temporal state representation, sequential handover optimization, and continual adaptation under heterogeneous high-mobility conditions. The research gap addressed in this study is therefore architectural: how to construct a predictive mobility-management pipeline in which current observations, predicted future conditions, and learned temporal information are jointly exposed to the handover agent and continuously adapted through operational feedback. Table 1 summarizes the broader literature-level positioning of this gap, while Table 2 provides a more explicit algorithmic and architectural comparison of the representative handover paradigms considered in this study.

3. System Model

We consider a predictive mobility management framework for high-mobility 5G/6G wireless networks spanning heterogeneous smart city, intelligent transportation systems (ITS), and connected vehicle (CV) environments under mixed urban–suburban conditions, consistent with the scenarios motivated in Sections 1.3–1.5. This section formalizes the network architecture, the high-mobility user model, the wireless channel model, and the Markov Decision Process (MDP) that underlies the proposed Bi-LSTM–DRL predictive handover framework. Table 3 summarizes the notation used throughout this section.

Table 3. Summary of notation used in Section 3.

Symbol	Description
$\mathcal{B}, b$	Set of cells (MBS/SBS/RSU); a generic cell
$\mathcal{U}, u$	Set of UEs; a generic UE
$\mathcal{N}_u(t), K$	Ranked candidate neighbor-cell set of UE u at time t ; fixed neighbor-list size
$t, \Delta t$	Discrete decision epoch; epoch duration
P_b^{tx}, f_c, R_b	Transmit power, carrier frequency, and coverage radius of cell b
$v_u(t), \theta_u(t)$	Velocity magnitude and direction of UE u at time t
$\alpha, \bar{v}, \bar{\theta}, \sigma_v, \sigma_\theta$	Gauss–Markov memory parameter, asymptotic means, and standard deviations
$x_u(t), y_u(t)$	Cartesian position of UE u at time t
$d_{u,b}(t)$	Euclidean distance between UE u and cell b
$PL_{u,b}(t), PL_0, d_0, n, X_\sigma$	Path loss, reference path loss, reference distance, path-loss exponent, and shadow fading
$f_{D,u}(t), c, \varphi_u(t)$	Doppler frequency, speed of light, and UE–cell angle of arrival

$h_{u,b}(t)$	Small-scale (Rayleigh/Doppler) fading coefficient
$RSRP_{u,b}(t), RSRQ_{u,b}(t), SINR_{u,b}(t)$	Reference Signal Received Power, Received Quality, and Signal-to-Interference-plus-Noise Ratio
$N_0, P_b^{rx}(t)$	Noise power; received power from cell b
$\mathbf{o}_u(t), b^*$	Instantaneous observation vector; current serving cell
$T, \mathbf{h}_t$	Bi-LSTM input window length; concatenated forward/backward hidden state
$\Delta, \hat{\mathbf{o}}_u(t + \Delta)$	Prediction horizon; Bi-LSTM-predicted future observation
$\mathbf{W}_p, \mathbf{b}_p$	Prediction-head weight matrix and bias
$\mathbf{s}_u(t)$	Predictive, knowledge-extracted state vector
$a_u(t), \mathcal{A}$	Handover action; action space
π_θ, π^*	DRL policy parameterized by θ ; optimal policy
R_t, w_i	Instantaneous reward; reward weights
$r_{thr}, r_{lat}, r_{fail}, r_{pp}, r_{sig}$	Throughput, latency, failure, ping-pong, and signaling reward terms
τ_{pp}, γ	Ping-pong detection window; discount factor
$\mathcal{M}, \mathcal{S}, \mathcal{P}$	MDP tuple; state space; state-transition probability

3.1. Network Architecture and Assumptions

We consider a heterogeneous radio access network (HetNet) composed of a set of macro base stations (MBSs) and small-cell base stations (SBSs), denoted collectively as $\mathcal{B} = \{b_1, b_2, \dots, b_B\}$, overlaid with roadside units (RSUs) deployed along transportation corridors to support ITS communication, consistent with the ultra-dense small-cell deployments that motivate this study [5,7]. Each cell $b \in \mathcal{B}$ is characterized by a transmit power P_b^{tx} , a carrier frequency f_c , and a nominal coverage radius R_b , with substantial overlap among neighboring cells typical of dense 5G/6G deployments. A set of mobile user equipment (UEs), $\mathcal{U} = \{u_1, u_2, \dots, u_U\}$, moves continuously across this heterogeneous network. Consistent with the three application domains addressed in this study, $\mathcal{U}$ comprises pedestrians and static Internet-of-Things (IoT) devices in smart city zones, vehicles equipped with onboard units interacting with RSUs in ITS scenarios and connected vehicles exchanging Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) messages as shown in Figure 1.

Time is discretized into decision epochs $t = 0, 1, 2, \dots$ of duration Δt . At every epoch, each UE u measures signal quality from its serving cell and from a candidate neighbor set $\mathcal{N}_u(t) \subset \mathcal{B}$, obtained through standard 3GPP measurement reporting and ranked by received signal strength; to keep the decision space tractable (Section 3.4.2), $\mathcal{N}_u(t)$ is capped at a fixed size K , i.e., $|\mathcal{N}_u(t)| = K$.

The following assumptions are adopted:

- A1 (Synchronized reporting): Measurement reports are generated periodically every Δt with negligible processing delay relative to Δt .
- A2 (Ideal backhaul): The backhaul network interconnecting cells and RSUs is not capacity-constrained, so the study focuses on radio-access mobility management rather than backhaul provisioning.
- A3 (Single connectivity): Each UE is served by exactly one cell at a time; extension to multi-connectivity is left for future work.
- A4 (Fixed-size candidate set): The neighbor list is capped at K cells through RSRP-based ranking, ensuring a fixed-size action space (Section 3.4.2).

spread, consistent with contemporary 5G channel-modeling practice [32] and with the rapid channel fluctuations identified as a core challenge for high-mobility handover in Section 1.4. The distance-dependent path loss is

$$PL_{u,b}(t) [\text{dB}] = PL_0 + 10n \log_{10} \left(\frac{d_{u,b}(t)}{d_0} \right) + X_\sigma \quad (5)$$

where $d_{u,b}(t)$ is the Euclidean distance between UE u and cell b at time t , PL_0 is the reference path loss at distance d_0 , n is the path-loss exponent, and $X_\sigma \sim \mathcal{N}(0, \sigma_{sf}^2)$ represents log-normal shadow fading. Because relative motion between UE and cell shifts the received carrier frequency, small-scale fading is modeled as a Rayleigh process with mobility-induced Doppler variation, consistent with recent high-mobility V2V channel analyses [33], with a maximum Doppler frequency

$$f_{D,u}(t) = \frac{v_u(t) f_c}{c} \cos \varphi_u(t) \quad (6)$$

where f_c is the carrier frequency, c is the speed of light, and $\varphi_u(t)$ is the angle between the UE velocity vector and the line-of-sight to cell b . Larger $v_u(t)$ as in the highway/ITS regime of Section 3.2 produces faster channel decorrelation, directly motivating predictive rather than reactive handover decisions. Combining large and small-scale effects, the instantaneous received power and Reference Signal Received Power (RSRP) as the same metric used by the conventional 3GPP threshold-based mechanisms discussed in Section 1.2 is

$$RSRP_{u,b}(t) = P_b^{tx} - PL_{u,b}(t) + 20 \log_{10} |h_{u,b}(t)| \quad (7)$$

where $h_{u,b}(t)$ is the small-scale (Rayleigh/Doppler) fading coefficient. The Signal-to-Interference-plus-Noise Ratio (SINR) at UE u from serving cell b is

$$SINR_{u,b}(t) = \frac{P_b^{rx}(t)}{N_0 + \sum_{b' \neq b} P_{b'}^{rx}(t)} \quad (8)$$

where $P_b^{rx}(t)$ is the received power computed from Equation (7) and N_0 is the noise power. Equations (5)–(8) generate the raw measurement quantities as RSRP, Reference Signal Received Quality (RSRQ, derived from RSRP and total received power), and SINR that constitute the instantaneous observation vector used as the baseline input in Section 3.4.1, allowing a direct, like-for-like comparison against the reactive 3GPP mechanism introduced in Section 1.2.

The analytical propagation model in Equations (5)–(8) is intentionally retained as the baseline channel abstraction for the large-scale Monte Carlo experiments, because it enables controlled isolation of mobility-management effects across a wide range of UE velocities, traffic loads, and deployment scenarios. However, this abstraction does not reproduce the complete delay, angle, and cluster-domain characteristics of standardized 5G NR channels and is therefore not intended to replace a link-level 3GPP-compliant channel model. To assess whether the conclusions of the mobility-management evaluation depend on this abstraction, an additional validation experiment using 3GPP TR 38.901 CDL channel profiles is presented in Section 5.5.7.

Throughput and Handover Latency Model

In addition to radio-link quality, the proposed handover framework explicitly considers end-to-end service performance. Let $B_{u,b}(t)$ denote the bandwidth allocated to UE u by cell b at decision epoch t . The instantaneous achievable downlink throughput is modeled as

$$R_{u,b}(t) = \eta B_{u,b}(t) \log_2 \left(1 + SINR_{u,b}(t) \right), \quad (9)$$

where $0 < \eta \leq 1$ denotes the effective spectral-efficiency factor accounting for practical coding, scheduling, and protocol overhead. The effective bandwidth $B_{u,b}(t)$ depends on the available cell resources and the current cell utilization, thereby capturing the effect of network load on the achievable per UE throughput.

The handover latency is defined as the service-interruption interval from handover initiation until successful communication through the selected target cell. For a handover from the serving cell b_s to a target cell b_t , it is expressed as

$$L_u^{\text{HO}}(t) = L_{\text{prep}}(t) + L_{\text{exec}}(t) + L_{\text{access}}(t), \quad (10)$$

where $L_{\text{prep}}(t)$, $L_{\text{exec}}(t)$, and $L_{\text{access}}(t)$ represent the handover preparation, execution/interruption, and target-cell access delays, respectively. When no handover is triggered at decision epoch t , $L_u^{\text{HO}}(t) = 0$. These throughput and handover-latency quantities are subsequently incorporated into the state and reward formulation and are also used as end-to-end performance metrics in Section 5.

3.4. Problem Formulation: Predictive Handover as a Markov Decision Process

Building on the network, mobility, and channel models above, the predictive handover management problem is formulated as a Markov Decision Process [34]

$$\mathcal{M} \triangleq (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma) \quad (11)$$

where $\mathcal{S}$ is the state space (Section 3.4.1), $\mathcal{A}$ is the action space (Section 3.4.2), $\mathcal{P}(s_{t+1} | s_t, a_t)$ is the state-transition probability induced by the mobility and channel models of Sections 3.2 and 3.3, $\mathcal{R}$ is the reward function (Section 3.4.3), and $\gamma \in [0,1)$ is a discount factor. The DRL agent seeks a policy π_θ that maximizes the expected discounted cumulative reward

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_t | \pi \right] \quad (12)$$

which is implemented in the following section using a value-based deep reinforcement learning method consistent with recent 5G handover and network-selection studies [34,35]. Unlike prior DRL-based handover formulations that build $\mathcal{S}$ solely from instantaneous measurements [13–15], the state defined next explicitly incorporates Bi-LSTM-based predicted future conditions, so that π_θ can act proactively.

3.4.1. State Representation via Bi-LSTM-Based Knowledge Extraction

To ensure consistency between the system model and the measurements collected by the proposed framework, the instantaneous observation is reformulated by explicitly separating cell-specific network features from UE-specific mobility information. For each serving or candidate cell b , the cell-specific feature vector of UE u at decision epoch t is defined as

$$\mathbf{x}_{u,b}(t) = [RSRP_{u,b}(t), RSRQ_{u,b}(t), SINR_{u,b}(t), \rho_b(t), R_{u,b}(t), d_{u,b}(t), D_{u,b}(t)]^T \quad (13)$$

where $\rho_b(t)$ denotes the resource utilization (or cell load) of cell b , $R_{u,b}(t)$ is the achievable throughput defined in Section 3.3, $d_{u,b}(t)$ is the UE-to-cell distance, and $D_{u,b}(t)$ denotes the experienced service delay. The UE-specific mobility information is separately represented as

$$\mathbf{m}_u(t) = [v_u(t), \phi_u(t)]^T, \quad (14)$$

where $v_u(t)$ and $\phi_u(t)$ denote the UE velocity magnitude and movement direction, respectively.

Accordingly, the complete instantaneous observation is constructed by concatenating the features of the serving cell b_s , the K candidate cells, and the UE-specific mobility vector:

$$\mathbf{o}_u(t) = [\mathbf{x}_{u,b_s}^T(t), \mathbf{x}_{u,c_1}^T(t), \dots, \mathbf{x}_{u,c_K}^T(t), \mathbf{m}_u^T(t)]^T \quad (15)$$

where $c_1, \dots, c_K$ denote the candidate cells associated with UE u . The resulting observation dimension is $d_0 = 7(K + 1) + 2$.

This unified observation representation ensures that the same radio-quality, resource-utilization, service-performance, and mobility information is consistently used by the measurement layer, the Bi-LSTM predictor, and the DRL-based handover decision module. In contrast to the previous formulation, UE velocity and movement direction are represented only once as UE-level features rather than being redundantly repeated for each serving or candidate cell.

Cell-identity alignment: The instantaneous RSRP rank is not used as the temporal identity of a candidate cell. Each physical cell b is associated with a persistent identifier ID_b . For each Bi-LSTM input window, the selected candidate cells are arranged in identity-consistent slots such that the feature trajectory of a physical cell remains in the same slot throughout the window. If a candidate is temporarily unavailable, its slot is retained using a validity mask rather than reassigned to another cell solely according to instantaneous RSRP rank. This representation prevents temporal slot permutation and allows the Bi-LSTM to learn the evolution of physically consistent candidate-cell conditions.

Accordingly, action $a_u(t) = k$ selects the physical target cell $\mu_u^t(k) = ID_{c_k(t)}$. Hence, k is an epoch-specific action index rather than a persistent cell identity. Before evaluating the Q-values, the predicted cell-specific features are gathered according to the same current ID-to-action mapping, and invalid candidate positions are excluded using an action mask. This guarantees alignment between the Bi-LSTM prediction slots and the physical candidate cells represented by the DRL actions.

To obtain a predictive, knowledge-extracted state, the sequence of the past T observations $\{\mathbf{o}_u(t - T + 1), \dots, \mathbf{o}_u(t)\}$ is passed through a Bidirectional LSTM (Bi-LSTM) encoder [9,23], which processes the sequence in both temporal directions,

$$\mathbf{h}_t^{\rightarrow} = \text{LSTM}_{fwd}(\mathbf{o}_u(t), \mathbf{h}_{t-1}^{\rightarrow}), \quad \mathbf{h}_t^{\leftarrow} = \text{LSTM}_{bwd}(\mathbf{o}_u(t), \mathbf{h}_{t+1}^{\leftarrow}), \quad \mathbf{h}_t = [\mathbf{h}_t^{\rightarrow}; \mathbf{h}_t^{\leftarrow}] \quad (16)$$

The concatenated hidden state $\mathbf{h}_t \in \mathbb{R}^{d_h}$ constitutes the knowledge-extraction step: rather than treating the heterogeneous raw measurements of Equation (12) as hand-engineered features, $\mathbf{h}_t$ compresses temporal mobility, propagation, and traffic-related patterns into a compact, informative embedding with $d_h \ll \dim(\mathbf{o}_u(t)) \times T$, directly operationalizing the knowledge-extraction contribution set out in Section 1 and Table 1. A lightweight prediction head maps this embedding to an explicit forecast of the future observation at prediction horizon Δ ,

$$\hat{\mathbf{o}}_u(t + \Delta) = \mathbf{W}_p \mathbf{h}_t + \mathbf{b}_p \quad (17)$$

trained by minimizing the mean-squared error between $\hat{\mathbf{o}}_u(t + \Delta)$ and the realized $\mathbf{o}_u(t + \Delta)$ over historical trajectories. The final, predictive state fed to the DRL agent concatenates the instantaneous observation, the predicted future observation, and the knowledge-extracted embedding

$$\mathbf{s}_u(t) = [\mathbf{o}_u(t); \hat{\mathbf{o}}_u(t + \Delta); \mathbf{h}_t] \in \mathcal{S} \quad (18)$$

so that $\mathbf{s}_u(t)$ reflects both current network conditions and their anticipated evolution, distinguishing this formulation from the instantaneous only states of prior DRL-based handover studies [13–15]. Here, $\mathbf{h}_t$ denotes a compact learned temporal representation of recent mobility and network dynamics. It is not intended to provide a directly human-

interpretable explanation of individual handover decisions. Instead, its role is to expose temporally encoded information to the downstream Dueling DDQN agent, complementing the instantaneous observation $\mathbf{o}_u(t)$ and explicit future-state prediction $\hat{\mathbf{o}}_u(t + \Delta)$.

3.4.2. Action Space

Given the predictive state $\mathbf{s}_u(t)$, the DRL agent selects a handover action from a fixed-size discrete action space of cardinality $K + 1$

$$a_u(t) = \pi_\theta(\mathbf{s}_u(t)) \in \mathcal{A} = \{0, 1, 2, \dots, K\} \quad (19)$$

where $a_u(t) = 0$ denotes remain with the current serving cell b^* and $a_u(t) = k$ that is $k \in \{1, \dots, K\}$ denotes handover to the k -th ranked neighbor cell in $\mathcal{N}_u(t)$ (Assumption A4, Section 3.1). Because π_θ is conditioned on $\mathbf{s}_u(t)$, which embeds the predicted future observation $\hat{\mathbf{o}}_u(t + \Delta)$ of Equation (17), the resulting handover decision is executed in anticipation of future channel and mobility conditions rather than only after instantaneous signal quality has already degraded, operationalizing the predictive/proactive handover criterion in Table 1.

3.4.3. Reward Function

The reward function extends the objectives already balanced by existing DRL-based handover schemes (Section 2.6) as throughput, latency, handover failures, ping-pong effects, and signaling overhead [13–16], while crediting the proactive component enabled by the predictive state of Section 3.4.1. The instantaneous reward is defined as the weighted combination

$$R_t = w_1 r_{thr}(t) - w_2 r_{lat}(t) - w_3 r_{fail}(t) - w_4 r_{pp}(t) - w_5 r_{sig}(t), \sum_{i=1}^5 w_i = 1, w_i \geq 0 \quad (20)$$

where $r_{thr}(t)$ is the normalized post-handover throughput; $r_{lat}(t)$ is the interruption/handover-execution latency, which the predictive state of Section 3.4.1 is designed to reduce by triggering handover before quality deteriorates; $r_{fail}(t) \in \{0, 1\}$ indicates a radio-link failure caused by a premature or an overly late handover; $r_{pp}(t) \in \{0, 1\}$ flags a ping-pong event, defined as a handover back to the previously serving cell within a short window τ_{pp} (Section 1.2); and $r_{sig}(t)$ counts handover-signaling messages. Weights $w_i \geq 0$ are tunable design parameters. Because r_{lat} and r_{pp} are evaluated relative to the future network state predicted in Equation (17), a successful early handover that avoids future degradation is rewarded before the corresponding reactive penalty would otherwise have been incurred, distinguishing this reward design from the purely reactive reward formulations of prior DRL-based handover work [13,15]. The default reward weights are intended to represent a balanced service-oriented operating point rather than a universal optimum. Throughput is assigned the largest weight to preserve service capacity under high mobility, while handover latency and failure are given substantial penalties to maintain service continuity. Ping-pong and signaling terms receive smaller weights because they primarily regularize excessive handover activity. The sensitivity of the resulting policy to these weights is evaluated in Section 5.1.

4. Proposed Hybrid AI-Native Predictive Mobility Management Framework

Section 3 formalized the predictive handover problem as an MDP and defined, in the abstract, the state $\mathbf{s}_u(t)$, action $a_u(t)$, and reward R_t that the agent operates on. This section presents the concrete, trainable architecture that computes these quantities in practice. Section 4.1 gives the overall framework architecture; Section 4.2 details the Bi-LSTM-based module that implements the knowledge-extraction and prediction mechanism underlying Equations (12)–(17); Section 4.3 details the DRL agent that implements the decision policy of Equations (18)–(20); Section 4.4 gives the end-to-end algorithm; and Section 4.5 analyzes computational complexity. The notation defined in Section 3 is retained throughout this section, while additional symbols introduced for the Bi-LSTM prediction module and the DRL optimization framework are summarized in Table 4.

Table 4. Additional notation introduced in Section 4.

Symbol	Description
d_o, d_h	Dimensionality of the instantaneous observation, Bi-LSTM hidden/embedding size
L_{bi}, L_q	Number of Bi-LSTM layers, number of Q-network fully connected layers
$\mathcal{L}_{pred}, N$	Bi-LSTM pretraining loss (MSE), number of training sequences
θ, θ^-	Online and target Q-network parameters
$V(\cdot), A(\cdot, \cdot)$	Dueling value and advantage streams
ε	Exploration rate (ε -greedy)
y_t	Double DQN training target
$p_i, \omega, P(i)$	Replay-buffer priority of transition i , prioritization exponent, sampling probability
$\mathcal{L}(\theta)$	Q-network training loss
τ	Target-network Polyak averaging rate
N_{update}	Number of epochs between target-network/encoder refresh
B, N_{buf}	Minibatch size, replay buffer size

4.1. Overall Framework Architecture

Figure 2 illustrates the overall architecture and closed-loop operation of the proposed hybrid Bi-LSTM-DRL predictive handover framework, integrating the network-measurement, knowledge-extraction-and-prediction, decision-making, and periodic-refresh stages of Sections 3, 4.2, and 4.3 into a single end-to-end pipeline. Within the scope defined in Section 1.1, the proposed architecture is AI-native specifically at the mobility-management-function level, where prediction, learned temporal representation, sequential handover decision-making, operational feedback, and model adaptation form an integrated closed control loop.

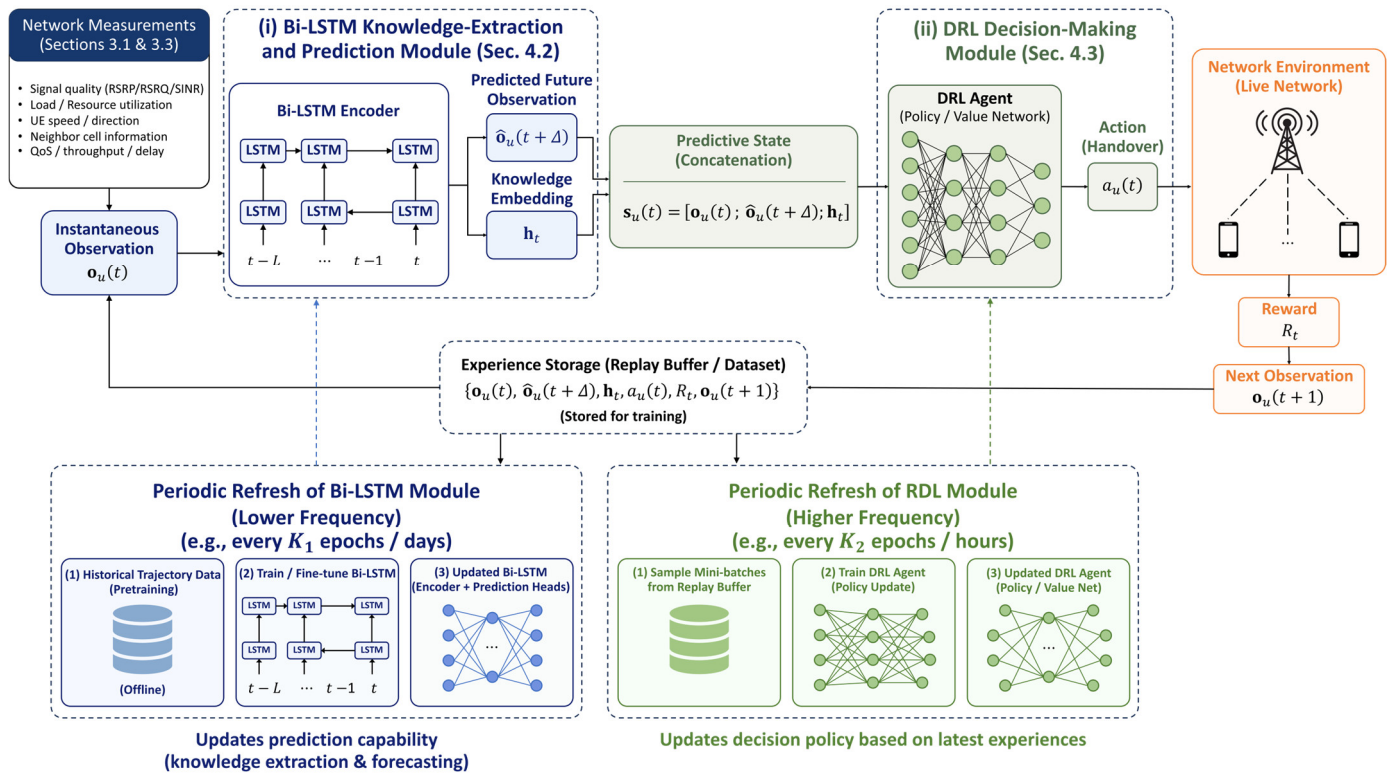


Figure 2. Overall architecture of the proposed hybrid Bi-LSTM and Deep Reinforcement Learning framework for AI-native predictive mobility management. The framework consists of two coupled modules: (i) a Bi-LSTM-based knowledge extraction and future mobility prediction module and (ii) a DRL-based handover decision module operating in a closed-loop with the live network. Sequential network measurements are transformed into a predictive state by combining the current observation, predicted future observation, and knowledge embedding before being processed by the DRL agent to determine the optimal handover action. Interaction experiences are continuously stored and used to periodically refresh both modules at different update frequencies, enabling continual learning and adaptive mobility management in dynamic 5G/6G environments.

At each decision epoch t , the network measurements stage (Sections 3.1 and 3.3) collects signal-quality indicators (RSRP/RSRQ/SINR), load and resource-utilization statistics, UE speed and direction, neighbor-cell information, and QoS metrics such as throughput and delay, which are assembled into the instantaneous observation $\mathbf{o}_u(t)$. This observation is passed to the Bi-LSTM Knowledge-Extraction and Prediction Module (Section 4.2), where a Bidirectional LSTM encoder processes the sliding window of past observations, from epoch $t-L$ to t , in both forward and backward directions. The encoder produces two outputs: the predicted future observation $\hat{\mathbf{o}}_u(t+\Delta)$ and the knowledge-embedding vector $\mathbf{h}_t$. Together with the instantaneous observation, these are concatenated into the predictive state $\mathbf{s}_u(t) = [\mathbf{o}_u(t); \hat{\mathbf{o}}_u(t+\Delta); \mathbf{h}_t]$.

The predictive state is provided to the DRL decision-making module (Section 4.3), where the DRL agent's policy/value network computes the handover action $a_u(t)$. This action is executed in the network environment, producing a reward R_t and a next observation $\mathbf{o}_u(t+1)$, which feeds back into the network measurements stage, thereby closing the loop from perception through prediction, decision, and action to reward at every decision epoch.

Each transition, $\{\mathbf{o}_u(t), \hat{\mathbf{o}}_u(t+\Delta), \mathbf{h}_t, a_u(t), R_t, \mathbf{o}_u(t+1)\}$, is retained in the experience storage (replay buffer/dataset) for training. From this shared storage, the two learnable modules are refreshed independently and at different frequencies, reflecting the different rates at which mobility/channel statistics and the optimal handover policy are

expected to drift: The Bi-LSTM module is refreshed at a lower frequency (every K_1 epochs, for example on the order of days), retraining or fine-tuning the encoder and prediction heads offline on historical trajectory data; the DRL module is refreshed at a higher frequency (every K_2 epochs, for example on the order of hours), sampling mini-batches from the replay buffer to update the policy/value network online. Because $K_1 \neq K_2$, the two modules are coupled, in that the DRL agent always acts on the Bi-LSTM's latest state representation, yet decoupled in their update schedules, allowing prediction and decision-making to adapt at the pace appropriate to each.

This closed-loop, dual-frequency refresh design operationalizes the “AI-native” criterion introduced in Table 1. Rather than being trained once offline and deployed statically, the framework continuously ingests live network measurements and updates its own predictive and decision-making components without manual reconfiguration of thresholds or rules, adapting to non-stationary mobility and channel conditions over the deployment horizon.

4.2. Bi-LSTM-Based Predictive Knowledge Extraction Architecture

At each decision epoch, the Bi-LSTM receives the instantaneous observation defined in Equation (15), consisting of identity-aligned cell-specific radio-quality, resource-utilization, throughput, distance, and delay features for the serving and candidate cells, together with UE-level velocity and movement direction. A sliding window of the previous T observations is processed by the two-layer Bi-LSTM encoder to capture the joint temporal evolution of radio conditions, mobility, resource congestion, and QoS. A sliding window of the past T observations, $\{\mathbf{o}_u(t - T + 1), \dots, \mathbf{o}_u(t)\} \in \mathbb{R}^{T \times d_o}$, is processed by a two-layer Bi-LSTM encoder with $d_h/2$ hidden units per direction per layer, producing the forward and backward hidden states of Equation (15) and the concatenated embedding $\mathbf{h}_t \in \mathbb{R}^{d_h}$. Choosing $d_h \ll T \cdot d_o$ (e.g., $d_h = 128$ against $T \cdot d_o$ in the low hundreds for typical T and K) forces the encoder to compress temporal mobility, propagation, and traffic patterns into a compact representation rather than passing raw measurements forward unchanged; this compression is the concrete mechanism realizing the knowledge extraction contribution of Section 1 and Table 1, as opposed to the hand-selected instantaneous features used in prior DRL-based handover work [13–16].

A fully connected prediction head (Equation (16)) maps $\mathbf{h}_t$ to the forecast $\hat{\mathbf{o}}_u(t + \Delta)$. The encoder and prediction head are pretrained offline by minimizing the mean squared prediction error over historical trajectories collected under the mobility and channel models of Sections 3.2 and 3.3

$$\mathcal{L}_{pred} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{o}}_u^{(i)}(t + \Delta) - \mathbf{o}_u^{(i)}(t + \Delta)\|^2 \quad (21)$$

where N is the number of training sequences. Two operating modes are supported for the pretrained encoder once embedded in the closed loop of Section 4.3 as a frozen mode, in which $\mathbf{h}_t$ is treated as a fixed feature extractor while only the DRL agent is trained, favored for training stability; and a fine-tuned mode, in which encoder weights continue to be updated at a reduced learning rate during closed-loop operation (Section 4.3), favored when mobility statistics are expected to drift over the deployment horizon. Both modes are evaluated in Section 5.

4.3. DRL-Based Handover Decision and Closed-Loop Policy Update

Given the discrete action space of Equation (19), the handover policy is realized using a Dueling Double Deep Q-Network (Dueling DDQN) with prioritized experience replay [34–37]. This choice is deliberate: Because the action space is discrete and moderate in size ($K + 1$ candidate decisions), a value-based method is a natural fit, and it allows the

framework's novelty to be attributed to the predictive, knowledge-extracted state $\mathbf{s}_u(t)$ of Section 3.4.1 rather than to the decision algorithm itself, which extends the double-deep Q-learning baseline of Wang and Li [28] rather than competing with it. To avoid confusion with the mobility heading angle $\theta_u(t)$ of Section 3.2, DRL network parameters are denoted using the bare symbol θ (online network) and θ^- (target network), consistent with standard notation and with the policy π_θ already introduced in Equation (11).

The online Q-network takes $\mathbf{s}_u(t)$ as input and splits it into two a state-value stream $V(\mathbf{s}_u(t); \theta)$ and an action-advantage stream $A(\mathbf{s}_u(t), a; \theta)$, which are recombined as

$$Q(\mathbf{s}_u(t), a; \theta) = V(\mathbf{s}_u(t); \theta) + \left(A(\mathbf{s}_u(t), a; \theta) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} A(\mathbf{s}_u(t), a'; \theta) \right) \quad (22)$$

which improves policy evaluation stability by separating the overall value of being in each state from the relative advantage of each action within that state, a distinction that matters here because the value of remaining with the current cell ($a_u(t) = 0$) is often similar across nearby candidate actions. Actions are selected by an ε -greedy rule over $Q(\mathbf{s}_u(t), a; \theta)$ during training, with ε annealed over epochs, and greedily at deployment.

Training uses the Double DQN target, which decouples action selection from action evaluation to reduce the overestimation bias of standard Q-learning

$$y_t = R_t + \gamma Q \left(\mathbf{s}_u(t+1), \left| \arg \max_a Q(\mathbf{s}_u(t+1), a; \theta) \right|, \theta^- \right). \quad (23)$$

Transitions $(\mathbf{s}_u(t), a_u(t), R_t, \mathbf{s}_u(t+1))$ are stored in a prioritized replay buffer, sampled with probability proportional to their temporal-difference error,

$$P(i) = \frac{p_i^\omega}{\sum_k p_k^\omega} \quad (24)$$

where p_i is the stored priority of transition i and ω controls the degree of prioritization ($\omega = 0$ recovers uniform sampling), so that transitions associated with rare events such as handover failures and ping-pong occurrences are replayed more frequently than the abundant “no-op” transitions. The online network is updated by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}_{i \sim P} \left[(y_t - Q(\mathbf{s}_u(t), a_u(t); \theta))^2 \right] \quad (25)$$

and the target network is refreshed by $\theta^- \leftarrow \tau \theta + (1 - \tau) \theta^-$ with $\tau \ll 1$. Closed-loop operation (Section 4.1) is realized by continuing to append live transitions to the replay buffer during deployment and periodically repeating the update every N_{update} epochs, so that the policy adapts to gradual, unannounced shifts in mobility patterns, cell load, or propagation conditions without manual re-tuning of thresholds, which is the behavior that Table 1 records as the “AI-native” criterion.

4.4. Algorithm of the Hybrid Bi-LSTM-DRL Predictive Handover

Offline pretraining phases are as follows:

1. Collect historical trajectories of $\mathbf{o}_u(t)$ under the mobility and channel models of Sections 3.2 and 3.3.
2. Train the Bi-LSTM encoder and prediction head by minimizing $\mathcal{L}_{pred}$ via stochastic gradient descent.
3. Initialize the online Q-network with parameters θ and set the target network $\theta^- \leftarrow \theta$; initialize an empty prioritized replay buffer.

Online closed-loop phase, for each decision epoch $t = 0, 1, 2, \dots$ are as follows:

4. Measure the serving and neighboring-cell features, construct the Top- K candidate set, and align the candidate observations using persistent cell IDs and the validity mask.

5. Process the identity-aligned observation window using the Bi-LSTM to obtain $\hat{\mathbf{o}}_u(t + \Delta)$ and $\mathbf{h}_t$, and construct $\mathbf{s}_u(t)$.
6. Construct the current ID-to-action mapping and action mask; select $a_u(t)$ using the Dueling DDQN and execute the action on the corresponding physical target cell.
7. Execute $a_u(t)$; observe reward R_t and next observation $\mathbf{o}_u(t + 1)$.
8. Compute $\mathbf{s}_u(t + 1)$ as in Steps 4–5; store the transition $(\mathbf{s}_u(t), a_u(t), R_t, \mathbf{s}_u(t + 1))$ in the prioritized replay buffer with maximal priority.
9. Sample a minibatch according to Equation (24); compute targets via Equation (23); update θ by gradient descent on Equation (25); update sampled transitions' priorities from the resulting temporal difference error.
10. Every N_{update} epochs: Update θ^- via Equation (26) and, if operating in fine-tuned mode (Section 4.2), refresh the Bi-LSTM encoder on the most recent trajectory window.
11. Anneal ε and return to Step 4.

4.5. Computational Complexity Analysis

At deployment (inference), each decision epoch requires one forward pass through the Bi-LSTM encoder and one forward pass through the dueling Q-network. For an L_{bi} -layer Bi-LSTM with $d_h/2$ units per direction processing a window of length T over d_o dimensional input, the forward-pass cost is $\mathcal{O}(L_{bi} T d_h (d_o + d_h))$; the fully connected dueling Q-network with L_q layers of width d_h adds $\mathcal{O}(L_q d_h^2)$. Since the sequence length T , input dimension d_o , hidden dimension d_h , the number of Bi-LSTM layers L_{bi} , the number of Q-network layers L_q , and the candidate neighbor size K are fixed design parameters, the computational cost of a single inference does not scale with either the number of cells B or the number of users U . Therefore, the per-UE inference complexity is $\mathcal{O}(1)$ with respect to the network size, whereas the total inference complexity grows linearly as $\mathcal{O}(U)$. This theoretical network-size independence is empirically examined in Section 5.5.9 by varying B from 20 to 200 cells at fixed UE density and measuring both per-UE inference latency and total training time. This is the same asymptotic order as the DRL only baselines of Section 2.4 [13–15,28–30], whose Q-networks operate on instantaneous states of comparable dimensionality; the additional cost introduced by the Bi-LSTM module is the $\mathcal{O}(L_{bi} T d_h (d_o + d_h))$ term, which is small relative to $\mathcal{O}(L_q d_h^2)$ for the window lengths T considered in Section 5, and negligible relative to the Δt decision interval, supporting real-time deploy ability. Training complexity is dominated by the minibatch gradient update of Equation (25), $\mathcal{O}(B L_q d_h^2)$ per step for batch size B , plus $\mathcal{O}(\log N_{buf})$ per transition for sum-tree-based prioritized sampling and priority update over a buffer of size N_{buf} (Equation (24)). Periodic Bi-LSTM fine-tuning (Step 10 of the algorithm), when enabled, adds $\mathcal{O}(B L_{bi} T d_h (d_o + d_h))$ every N_{update} epochs rather than at every decision epoch, keeping the amortized per-epoch training overhead small. Overall, both inference and training complexity are governed by fixed architectural constants rather than by the size of the network topology, which is consistent with the scalability required for the multi-cell smart city, ITS, and connected vehicle deployments motivating this study (Sections 1.3–1.5). The additional computational cost introduced by the Bi-LSTM module is therefore bound by fixed architectural parameters. Its practical execution-time impact is evaluated empirically in Section 5.5 through measured per-UE inference latency, total runtime per simulation episode, and runtime overhead, rather than inferred solely from asymptotic complexity.

5. Performance Evaluation and Discussion

This section evaluates the proposed hybrid Bi-LSTM-DRL predictive handover framework against the system model and architecture defined in Sections 3 and 4. Section 5.1 describes the experimental setup, including the simulation environment, hyperparameters, and evaluation protocol used throughout this section. Section 5.2 introduces the baseline and ablation methods used for comparison. Section 5.3 evaluates the accuracy of the Bi-LSTM future-mobility-prediction module in isolation. Section 5.4 analyzes the knowledge representation learned by the Bi-LSTM encoder. Section 5.5 evaluates end-to-end handover performance and statistical significance across independent simulation runs. Section 5.6 evaluates generalization across the smart city, ITS, and connected vehicle scenarios of Sections 1.3–1.5 and across urban, suburban, and mixed deployment regimes.

5.1. Experimental Setup

Simulation environment: Because no public dataset jointly provides fine-grained mobility traces, wireless channel measurements, and handover signaling logs across smart city, ITS, and connected vehicle scenarios, evaluation data are generated through Monte Carlo simulation that directly instantiates the network architecture (Section 3.1), high-mobility model (Section 3.2), and wireless channel model (Section 3.3). At each decision epoch, every simulated UE moves according to Equations (1)–(4), its channel to each cell is generated according to Equations (5)–(8), and the resulting instantaneous observation $\mathbf{o}_u(t)$ (Equation (19)) are logged together with the ground-truth future observation $\mathbf{o}_u(t + \Delta)$, enabling supervised evaluation of the prediction module (Section 5.3) independently of the DRL agent. For the main Monte Carlo evaluation, Equations (5)–(8) are used as the baseline system-level propagation abstraction to enable controlled comparison across all methods and mobility scenarios. In addition, a representative subset of the experiments is repeated using standardized 3GPP TR 38.901 CDL channel profiles to assess the robustness of the reported handover gains under richer multipath conditions, as described in Section 5.5.7.

Scenarios and deployment scenarios: Three application scenarios are simulated, corresponding to Sections 1.3–1.5: smart city (pedestrian and static IoT UEs, $v_u(t) \in [0, 5]$ m/s), ITS (road-constrained vehicular UEs, $v_u(t) \in [5, 20]$ m/s), and connected vehicle (highway-speed UEs exchanging V2V/V2I/V2N messages, $v_u(t) \in [20, 33]$ m/s), following the mobility regimes defined in Section 3.2. Each scenario is further evaluated under three deployment regimes that vary the network architecture of Section 3.1: urban (high cell density, small coverage radius R_b , shadow-fading standard deviation $\sigma_{SF} = 8$ dB), suburban (lower cell density, larger R_b , $\sigma_{SF} = 6$ dB), and mixed (a corridor transitioning between urban and suburban cell density, used specifically for the generalization evaluation of Section 5.6). This gives nine scenario/regime combinations in total, summarized in Table 5.

Bi-LSTM prediction module configuration: The encoder of Section 4.2 uses an input window of $T = 10$ epochs, two Bi-LSTM layers with 64 hidden units per direction ($d_h = 128$), a dropout rate of 0.2 between layers, and a prediction horizon of $\Delta = 5$ epochs. The encoder and prediction head are pretrained by minimizing $\mathcal{L}_{pred}$ using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 64, for up to 100 epochs with early stopping on validation loss (patience of 10 epochs).

DRL agent configuration: The Dueling DDQN agent of Section 4.3 uses a discount factor $\gamma = 0.95$, a replay buffer of size $N_{buf} = 10^5$ transitions with prioritization exponent $\omega = 0.6$, minibatch size $B = 64$, Adam optimizer with a learning rate of 5×10^{-4} , and target-network update rate $\tau = 0.01$. The exploration rate ε is annealed linearly from 1.0 to 0.05 over the first 20% of training epochs. Following the two-frequency refresh design of Section 4.3, the DRL policy is updated every K_2 epochs (on the order of hours of simulated time) and the Bi-LSTM encoder is fine-tuned every K_1 epochs (on the order of

days of simulated time), with $K_1 \gg K_2$. The reward weights are set to $w_{thr} = 0.35$, $w_{lat} = 0.25$, $w_{fail} = 0.20$, $w_{pp} = 0.10$, and $w_{sig} = 0.10$, and the ping-pong detection window is $\tau_{pp} = 5$ s.

Data split and evaluation protocol: For each scenario combination, independent simulation runs are generated using distinct random seeds and partitioned 70%/15%/15% into training, validation, and test sets, respectively. The training and validation sets were used for Bi-LSTM pretraining and DRL policy learning, whereas the held-out test set was generated from seeds never used during training and reserved for performance and generalization evaluation. Statistical significance between the proposed framework and the baseline methods was evaluated using the procedure described in Section 5.5.

Hardware and software configuration: All experiments were implemented in MATLAB R2025b, where the proposed hybrid Bi-LSTM and Dueling DDQN framework was trained and evaluated. Simulations were conducted on a workstation equipped with an Intel Core i9-14900K CPU, NVIDIA RTX 4090 GPU (24 GB VRAM), and 64 GB RAM. The average training time, inference latency, and runtime overhead for each experimental scenario are reported in Section 5.5.8 and discussed together with the computational complexity analysis presented in Section 4.5.

Implementation and reproducibility details: To improve experimental reproducibility, Table 5 summarizes the principal network, channel, mobility, handover, learning, and implementation parameters used throughout the simulations. The simulation operates at discrete decision epochs, at which UE mobility, radio-link measurements, candidate-cell information, cell load, achievable throughput, and service delay are updated according to the models defined in Section 3. Unless otherwise stated, the same network topology, channel conditions, mobility configuration, decision interval, candidate-cell selection procedure, and computational environment are applied to all baseline, ablation, and proposed methods to ensure a fair comparison. The Bi-LSTM and Dueling DDQN models are independently initialized for each run, with different random seeds used to capture mobility, channel, and learning stochasticity; test seeds are excluded from model training and hyperparameter selection. Each experimental configuration is independently repeated 10 times, and the results are reported as the mean with 95% confidence intervals.

Table 5. Simulation and implementation parameters used for performance evaluation.

Category	Parameters and Values
Network deployment	Area: 2500×2500 m ² ; MBS/SBS/RSU: 1/12/8; active UEs: 20–500; f_c : 3.5 GHz;
Channel	bandwidth: 100 MHz; Tx power (MBS/SBS/RSU): 46/30/20 dBm Path loss: 3GPP UMa/UMi; shadowing SD: 4–7 dB; fading: Rayleigh; noise PSD: −174 dBm/Hz; noise figure: 6 dB; update interval: 100 ms
Mobility	Smart city/ITS/connected vehicle speeds: 0–5/5–20/20–33 m/s; model: Gauss–Markov; update interval: 100 ms
Handover	Decision interval: 100 ms; $K = 5$; hysteresis: 3 dB; TTT: 160 ms; minimum serving SINR: −6 dB; failure: SINR below the threshold or unsuccessful target access
Bi-LSTM	$T = 10$; 2 layers; hidden size: 128 units/direction; dropout: 0.2; prediction horizons $\Delta = \{1, 3, 5, 7, 10\}$ epochs; selected horizon: 5 epochs; Adam; LR: 10^{-3} ; batch: 128; max epochs: 200; early-stopping patience: 15
Dueling DDQN	$\gamma = 0.95$; replay buffer: 10^5 ; batch: 128; LR: 5×10^{-4} ; PER $\alpha = 0.6$, $\beta_0 = 0.4$; $\epsilon_0/\epsilon_{\min} = 1.0/0.05$; target update: 500 steps

Closed-loop adaptation	DRL/Bi-LSTM refresh: 10/50 epochs; encoder fine-tuning LR: 10^{-4}
Reward	$w_{thr}/w_{lat}/w_{fail}/w_{pp}/w_{sig}/\tau_{pp}$: 0.35/0.25/0.20/0.10/0.10/5 s
Evaluation	Train/validation/test: 70/15/15%; 10 independent runs; 95% CI; different random seed/run; test-seed reuse during training: no

5.2. Baseline and Ablation Methods

The proposed framework is evaluated against two families of comparison methods: external baselines drawn from the literature reviewed in Section 2 and internal ablation variants of the proposed framework itself. All methods are trained and evaluated under the identical simulation environment, network architecture, mobility and channel models, and train/validation/test split described in Section 5.1, and are scored using the same reward components of Equation (20) to ensure a like-for-like comparison; for baselines that do not optimize a reward signal directly (B1, B3), Equation (20) is applied post hoc for evaluation purposes only.

External baselines: Five baselines are reimplemented, each representative of one of the five regions identified in the Table 1 comparison (reactive threshold-based, DRL-only, prediction-only, prediction-assisted DRL, and prediction-Kalman Filter) as follows:

- B1 (Reactive 3GPP): the conventional threshold-based handover mechanism described in Section 1.2, triggering handover when the neighbor-cell RSRP exceeds the serving-cell RSRP by a fixed hysteresis margin sustained for a Time-to-Trigger (TTT) interval, with parameter values following standard practice reported in the handover surveys of Amirova et al. [5] and Alraih et al. [16]. This baseline uses no learning of any kind and represents the non-AI floor against which the benefit of AI-based mobility management is measured.
- B2 (DRL-only, reactive state): a reimplementation of the double-deep Q-learning approach of Wang and Li [28], using the same Dueling DDQN training procedure as the proposed framework (Section 4.3, Equations (22)–(26)) but operating solely on the instantaneous observation $\mathbf{o}_u(t)$ of Equation (12), without the predicted future observation $\hat{\mathbf{o}}_u(t + \Delta)$ or the knowledge-extracted embedding $\mathbf{h}_t$. Because B2 shares the same DRL algorithm and hyperparameters as the proposed framework, differing only in the state representation, it isolates the contribution of the predictive, knowledge-extracted state of Section 3.4.1 from the contribution of the decision algorithm itself.
- B3 (Prediction-only, reactive decision): a reimplementation of the Bi-LSTM/LSTM-based proactive triggering approach of Sulaiman and Al-Raweshidy [10], in which the same Bi-LSTM encoder as the proposed framework (Section 4.2) forecasts $\hat{\mathbf{o}}_u(t + \Delta)$, but the handover decision is made by applying the reactive threshold rule of B1 to the forecasted RSRP rather than by a learned DRL policy. B3 isolates the contribution of the DRL decision-making module (Section 4.3) from the contribution of prediction alone.
- B4 (Prediction-assisted DRL): a reimplementation of Shahid et al. [15], the closest existing work identified in Section 2.4, which incorporates mobility prediction into the handover procedure but, per Table 1, without an explicit knowledge-extraction mechanism or closed-loop refresh. B4 represents the closest prior-art comparator among the implemented baselines and is used to assess the incremental contribution of the full proposed framework over a prediction-assisted handover approach.
- B5 (Model-based predictive handover using a Kalman filter): A constant-velocity Kalman filter (CV-KF) serves as a non-learning predictive baseline, recursively estimating UE position and velocity and forecasting future position over the same horizon as the proposed framework. The forecast is mapped to candidate-cell radio

conditions using the models in Section 3 to select the target cell under identical hand-over constraints. By excluding learned temporal representation, knowledge embedding, and DRL optimization, B5 isolates the contribution of classical model-based prediction from the additional gains of the proposed framework.

Rationale for baseline selection: The five external baselines were selected to represent complementary mobility-management paradigms. B1 represents conventional reactive handover, B2 isolates DRL-based decision-making without explicit mobility prediction, B3 represents prediction-driven mobility management without DRL optimization, and B4 represents the closest prediction-assisted DRL paradigm. B5 provides a classical non-learning predictive reference, allowing the benefit of mobility prediction itself to be distinguished from that of learned temporal representation and DRL-based decision optimization. Together, these baselines enable a structured comparison of reactive, learning-based, prediction-based, and hybrid predictive decision-making approaches. Collectively, these baselines span the progression from reactive handover to prediction-assisted learning-based mobility management and therefore provide a controlled basis for assessing the incremental contributions of future-state prediction, temporal knowledge extraction, and DRL-based long-term decision optimization. Nevertheless, the comparison should be interpreted as representative rather than exhaustive, since rapidly evolving AI-native 5G/6G mobility management encompasses additional architectures and deployment-specific methods.

Ablation variants: In addition to the external baselines, seven internal variants of the proposed framework are evaluated to isolate the contribution of each architectural component introduced in Section 4 as follows:

- AV1 (w/o predicted observation): The predictive state of Equation (17) is reduced to $\mathbf{s}_u(t) = [\mathbf{o}_u(t); \mathbf{h}_t]$, removing the explicit forecast $\hat{\mathbf{o}}_u(t + \Delta)$ while retaining the Bi-LSTM embedding, to test whether an explicit prediction target adds value beyond the embedding alone.
- AV2 (w/o knowledge-extraction embedding): The predictive state is reduced to $\mathbf{s}_u(t) = [\mathbf{o}_u(t); \hat{\mathbf{o}}_u(t + \Delta)]$, with $\hat{\mathbf{o}}_u(t + \Delta)$ produced by a lightweight feedforward regressor over the same input window rather than a Bi-LSTM encoder, removing the compressed embedding $\mathbf{h}_t$ while retaining an explicit forecast, to test whether the knowledge-extraction mechanism of Section 4.2 contributes beyond the forecast itself.
- AV3 (Unidirectional LSTM): The Bi-LSTM encoder of Section 4.2 is replaced with a forward-only LSTM of matched hidden dimensionality d_h , isolating the specific contribution of bidirectional temporal processing.
- AV4 (Frozen encoder): The Bi-LSTM encoder parameters are fixed after offline pre-training, and the pretrained predictive representation is used without subsequent encoder updating during evaluation. The remaining decision framework follows the standard evaluation protocol. This variant is retained as an architectural ablation to assess the effect of using a fixed predictive representation and is interpreted separately from the controlled refresh-specific experiment described below.
- AV5 (w/o closed-loop refresh): Both periodic adaptation channels are disabled after offline training. Neither the Bi-LSTM encoder nor the Dueling DDQN policy is refreshed using newly accumulated deployment experience. This variant serves as the no-refresh control.
- AV6 (DRL refresh-only): Periodic encoder refresh is disabled, while the Dueling DDQN policy is periodically updated using newly accumulated transition experience according to the same policy-refresh schedule as the proposed framework. This variant isolates the contribution of DRL-policy adaptation. Unlike AV4, which is retained as the original frozen-representation architectural ablation under the standard

evaluation protocol, AV6 specifically serves as the DRL-refresh-only condition in the controlled refresh experiment.

- AV7 (Encoder refresh-only): The Bi-LSTM encoder is periodically updated using newly accumulated observations according to the same encoder-refresh schedule used by the proposed framework, while the Dueling DDQN policy remains fixed after offline training. This variant isolates the contribution of predictive-representation adaptation.

The ablation study is organized into two complementary groups. AV1–AV4 assess the contributions of individual architectural components, whereas AV5–AV7 together with the complete proposed framework form a controlled 2×2 refresh-specific experiment covering all combinations of encoder and DRL-policy refresh. This factorial design enables the individual and complementary contributions of the two adaptation channels to be evaluated independently of the AV4–AV5 comparison.

The complete framework, comprising Bi-LSTM-based prediction and knowledge extraction, Dueling DDQN decision-making, and both periodic encoder and DRL-policy refresh mechanisms, is denoted as “Proposed” throughout Sections 5.3–5.7. Table 6 summarizes the external baselines and internal ablation variants together with the components and refresh mechanisms enabled in each configuration.

Table 6. Baseline and ablation methods evaluated in Section 5.

Method	Predicted Observation $\hat{o}_u(t + \Delta)$	Knowledge Embedding; h_t	DRL Decision	Encoder Refresh	DRL Policy Refresh	Source
B1: Reactive 3GPP	×	×	×	×	×	Section 1.2 [5,16]
B2: DRL-only	×	×	✓	N/A	✓	Wang and Li [26]
B3: Prediction-only	✓	×	×	×	N/A	Sulaiman and Al-Raweshidy [10]
B4: Prediction-assisted DRL	✓	×	Partial	×	×	Shahid et al. [15]
B5: Model-based predictive (CV-KF)	✓	×	×	N/A	N/A	This study
AV1: w/o predicted observation	×	✓	✓	N/A	✓	This study
AV2: w/o knowledge embedding	✓	×	✓	✓	✓	This study
AV3: Unidirectional LSTM	✓	✓ (Forward only)	✓	✓	✓	This study
AV4: Frozen encoder	✓	✓ (Frozen)	✓	×	Standard	This study
AV5: w/o closed-loop refresh	✓	✓	✓	×	×	This study
AV6: DRL refresh-only	✓	✓	✓	×	✓	This study
AV7: Encoder refresh-only	✓	✓	✓	✓	×	This study
Proposed	✓	✓	✓	✓	✓	This study

Note: ✓ and × indicate that the corresponding component or periodic refresh mechanism is enabled or disabled, respectively; N/A indicates that the corresponding mechanism is not applicable because the associated module is absent.

5.3. Prediction Performance Evaluation

The predictive accuracy of the proposed Bi-LSTM module was first evaluated independently of the DRL decision-making process, to determine whether the predictive state used for proactive handover carries sufficiently accurate information regarding future network evolution. Consistent with Section 4.2, the predictor receives a sequence of the previous $T = 10$ instantaneous observations and estimates the future observation $\hat{\mathbf{o}}_u(t + \Delta)$ at a prediction horizon of $\Delta = 5$ epochs. Each observation contains the cell-specific and UE-level features defined in Equation (15), including RSRP, RSRQ, SINR, resource utilization, achievable throughput, UE-to-cell distance, delay, UE velocity, and movement direction. The candidate-cell features are maintained using the identity-alignment mechanism described in Section 4.2 to ensure temporal consistency across the Bi-LSTM input sequence. The prediction task therefore evaluates the ability of each model to jointly capture temporal mobility dynamics and wireless-channel evolution, rather than predicting a single signal indicator in isolation. Five comparators are considered alongside the proposed Bi-LSTM: a persistence baseline (naive no-change forecast), a classical constant-velocity Kalman filter (CV-KF), a unidirectional LSTM, a GRU, and a Transformer-based sequence predictor. The CV-KF provides a non-learning model-based reference, whereas the neural predictors are configured using comparable sequence information and matched hidden dimensionality where applicable.

Accuracy across mobility scenarios: Table 7 reports the MAE, RMSE, and R^2 of the persistence, CV-KF, LSTM, GRU, Transformer, and proposed Bi-LSTM predictors across the smart city, ITS, and connected vehicle scenarios defined in Section 5.1. The newly included CV-KF provides a non-learning model-based reference for distinguishing the benefit of mobility prediction itself from that of learned temporal modeling. Compared with persistence, CV-KF substantially reduces the RMSE from 0.1711 to 0.1126 in the smart city scenario, from 0.3526 to 0.1875 in the ITS scenario, and from 0.3596 to 0.3015 in the connected vehicle scenario, confirming that explicit model-based mobility prediction is beneficial even without learned temporal representations. Nevertheless, the proposed Bi-LSTM achieves the lowest MAE and RMSE and the highest R^2 in all three scenarios. Relative to CV-KF, it reduces the RMSE by approximately 26.9%, 40.9%, and 48.5% in the smart city, ITS, and connected vehicle scenarios, respectively. The increasing performance gap under higher mobility indicates that the gain of the proposed predictor cannot be attributed solely to future-state prediction; rather, learned bidirectional temporal modeling provides additional robustness when mobility and channel dynamics become increasingly non-stationary. The Transformer remains the strongest competing learned predictor, whereas the larger advantage over CV-KF in the connected vehicle scenario further highlights the limitation of a fixed constant-velocity motion model under rapidly varying mobility conditions.

Accuracy across prediction horizons: Table 8 evaluates prediction accuracy over horizons of $\Delta = \{1, 3, 5, 7, 10\}$ epochs. All predictors exhibit an increasing RMSE as the horizon grows, reflecting the increasing uncertainty of longer-range forecasting. The CV-KF consistently outperforms persistence, confirming that classical model-based prediction remains beneficial across all evaluated horizons; however, its error increases more rapidly than that of the learned sequence predictors as the horizon becomes longer. In contrast, the proposed Bi-LSTM maintains the lowest RMSE at every evaluated horizon. At the operating horizon of $\Delta = 5$, it achieves an RMSE of 0.0935, corresponding to a 38.3% reduction relative to CV-KF, in addition to reductions of 52.2%, 22.9%, 30.6%, and 16.9% relative to persistence, LSTM, GRU, and Transformer, respectively. Even at $\Delta = 10$, the proposed Bi-LSTM maintains an RMSE of 0.1555 compared with 0.2590 for CV-KF, demonstrating greater robustness to increasing prediction horizon.

Taken together, Tables 7 and 8 establish two complementary findings. First, the consistent improvement of CV-KF over persistence confirms that explicit mobility prediction itself provides measurable benefit. Second, the further reduction in prediction error achieved by the proposed Bi-LSTM, particularly under high mobility and longer prediction horizons, demonstrates the additional value of learned multivariate and bidirectional temporal representation beyond classical model-based prediction. These findings support the design decision in Section 3.4.1 to incorporate the predicted observation into the DRL state and motivate the end-to-end evaluation in Section 5.5, where the prediction-level advantage is examined in terms of actual handover performance.

Table 7. Prediction performance comparison of persistence, CV-KF, LSTM, GRU, Transformer, and the proposed Bi-LSTM across different mobility scenarios.

Scenario	Predictor	MAE	RMSE	R ²
Smart city	Persistence	0.1282	0.1711	0.8425
	CV-KF	0.0835	0.1126	0.9038
	LSTM	0.0711	0.0967	0.9216
	GRU	0.0755	0.1010	0.8944
	Transformer	0.0664	0.0904	0.9451
	Proposed Bi-LSTM	0.0611	0.0823	0.9558
ITS	Persistence	0.2174	0.3526	0.7781
	CV-KF	0.1408	0.1875	0.8696
	LSTM	0.1174	0.1480	0.9079
	GRU	0.1239	0.1554	0.8985
	Transformer	0.1036	0.1209	0.9091
	Proposed Bi-LSTM	0.0977	0.1108	0.9312
Connected vehicle	Persistence	0.2901	0.3596	0.7104
	CV-KF	0.2578	0.3015	0.8217
	LSTM	0.2456	0.2809	0.8661
	GRU	0.2820	0.3177	0.8469
	Transformer	0.1713	0.2087	0.8709
	Proposed Bi-LSTM	0.1221	0.1553	0.8960

Table 8. RMSE performance of the proposed Bi-LSTM and baseline predictors across different prediction horizons.

Prediction Horizon Δ (Epochs)	Persistence	CV-KF	LSTM	GRU	Transformer	Proposed Bi-LSTM
1	0.1026	0.0838	0.0611	0.0620	0.0567	0.0494
3	0.1511	0.1089	0.0871	0.0901	0.0801	0.0723
5	0.1955	0.1516	0.1213	0.1348	0.1125	0.0935
7	0.2418	0.1967	0.1428	0.1503	0.1287	0.1209
10	0.3019	0.2590	0.1894	0.1968	0.1765	0.1555

5.4. Knowledge Representation Analysis

Beyond prediction accuracy, an important objective of the proposed framework is to determine whether the Bi-LSTM encoder learns a compact representation that captures meaningful temporal structure from heterogeneous network observations. This aspect is particularly relevant to the AI-native design considered in this study because the encoder output $\mathbf{h}_t$ is incorporated into the predictive state $\mathbf{s}_u(t)$ of Equation (18), rather than serving solely as an intermediate feature for forecasting. Accordingly, the analysis in this

subsection examines whether the learned embedding preserves mobility- and channel-related information that is useful for distinguishing network operating conditions associated with different handover decisions.

To examine the structure of the learned representation, the high dimensional Bi-LSTM embeddings obtained from the held-out test trajectories were projected onto a two-dimensional latent space using dimensionality reduction analysis. Each point represents the embedding $\mathbf{h}_t$ generated from a temporal window of network observations, while the samples are grouped according to the mobility condition and subsequent handover outcome. The resulting representation reveals that the encoder does not map observations into an unstructured latent space. Instead, samples associated with relatively stable serving-cell conditions occupy distinguishable regions from those corresponding to deteriorating serving links and approaching handover conditions. The separation becomes particularly evident for high-mobility ITS and connected vehicle samples, for which instantaneous radio measurements fluctuate more rapidly because of mobility-induced channel variation. This behavior is important because individual instantaneous measurements can be ambiguous from a handover perspective. For example, a temporary decrease in RSRP or SINR may arise from short-term fading and does not necessarily indicate that a handover should be initiated. Conversely, a similar instantaneous signal level accompanied by a persistent temporal trend in UE velocity, UE-to-cell distance, and neighboring-cell signal evolution may indicate that the serving link is approaching sustained degradation. By encoding a sequence of observations rather than an isolated measurement, $\mathbf{h}_t$ provides the DRL agent with information about the temporal context underlying the current radio state. The learned representation therefore complements the explicit future prediction $\hat{\mathbf{o}}_u(t + \Delta)$. Whereas the latter provides an interpretable estimate of the anticipated network condition, the embedding retains additional latent temporal dependencies that need not be explicitly represented by individual forecast variables.

To quantify this property, representation separability can be evaluated using the silhouette coefficient of the projected embeddings. Let $a(i)$ denote the mean intra-cluster distance of sample i , and $b(i)$ denote the minimum mean distance between sample i and samples belonging to another cluster. The silhouette coefficient is defined as

$$SC(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (26)$$

where $SC(i) \in [-1, 1]$. Values approaching one indicate that a representation is well matched to its assigned operating regime and clearly separated from neighboring regimes, whereas values close to zero indicate overlapping representations. The average silhouette coefficient over the test samples can therefore be used as a quantitative indicator of whether the learned latent space discriminates mobility and handover-relevant network conditions.

The representation analysis should further distinguish between the three mobility scenarios considered throughout this study. Smart city trajectories are expected to exhibit relatively compact latent structures because low UE velocities produce slower variations in distance and channel quality. In contrast, ITS and connected vehicle trajectories introduce substantially faster transitions between network states. Consequently, their latent distributions are expected to span broader regions of the representation space. Importantly, such increased dispersion should not be interpreted solely as poorer representation quality; rather, it reflects the greater diversity and temporal variability of the underlying high-mobility observations. The relevant criterion is whether states associated with different mobility or handover conditions remain distinguishable despite this increased variability. The functional value of the learned representation is further assessed through the AV1 and AV2 ablation variants defined in Section 5.2, which isolate the

knowledge embedding and the explicit forecast from the complete predictive state, respectively. Comparing their handover performance provides a direct means of determining whether $\mathbf{h}_t$ contributes information beyond that already contained in the predicted future observation. If AV1 maintains stronger handover performance than AV2 despite lacking the explicit forecast, this would indicate that $\mathbf{h}_t$ captures temporal information that cannot be represented completely by point prediction; conversely, superiority of the complete predictive state of Equation (18) over both variants would demonstrate the complementary roles of explicit forecasting and latent knowledge representation.

The comparison with the AV3 unidirectional-LSTM variant provides an additional test of representation quality because AV3 differs from the proposed framework only in its temporal encoding mechanism, differences between their latent structures can be attributed primarily to bidirectional versus forward-only processing rather than to state dimensionality or DRL architecture. A more discriminative representation produced by the Bi-LSTM would support the hypothesis that bidirectional encoding provides a richer characterization of temporal dependencies, consistent with the lower prediction error achieved by the Bi-LSTM over the unidirectional LSTM.

Table 9 reports the quantitative outcome of this analysis, presenting the silhouette coefficient of Equation (26) together with the mean intra-cluster and inter-cluster distances of the projected embeddings, for the proposed Bi-LSTM and the AV3 unidirectional-LSTM baseline across the three mobility scenarios. Across all three scenarios, the proposed Bi-LSTM achieves a higher silhouette score, a smaller intra-cluster distance, and a larger inter-cluster distance than the unidirectional LSTM, confirming the hypothesis advanced above that bidirectional processing yields a more discriminative latent representation than forward-only encoding: Embeddings produced by the Bi-LSTM form tighter, more clearly separated clusters corresponding to distinct mobility- and handover-relevant operating conditions. The magnitude of this improvement varies across scenarios rather than increasing monotonically with mobility. The relative gain in the silhouette score is smallest in the smart city scenario (4.4%, from 0.777 to 0.811), because the unidirectional LSTM already attains a comparatively high score under low-velocity conditions, leaving limited room for improvement, whereas the gain is substantially larger in the ITS scenario (29.8%, from 0.604 to 0.784) and remains considerable for the connected vehicle scenario (23.6%, from 0.513 to 0.634). This pattern indicates that the benefit of bidirectional encoding is most pronounced precisely where forward-only processing struggles most, that is, under the faster and more irregular state transitions associated with high-mobility ITS and connected vehicle trajectories, consistent with the prediction-accuracy results of Section 5.3. Consistent with the dispersion argument advanced above, the absolute silhouette score and inter-cluster distance achieved by the proposed Bi-LSTM decrease monotonically from smart city (0.811; 1.722) to ITS (0.784; 1.457) to connected vehicle (0.634; 1.310) scenarios, while the intra-cluster distance increases in the same order (0.317, 0.418, and 0.455). Rather than indicating degraded representation quality, this trend reflects the greater intrinsic variability of high-mobility observations noted earlier: Even as absolute separability declines with increasing mobility, the relative improvement of the Bi-LSTM over the unidirectional LSTM remains positive throughout, and for inter-cluster distance is in fact largest for the connected vehicle scenario (29.6% versus 22.7% for the smart city scenario and 20.7% for the ITS scenario). This indicates that the encoder continues to separate operating regimes more effectively than the unidirectional baseline even as the underlying discrimination task becomes intrinsically harder.

Overall, the knowledge-representation analysis complements the forecasting evaluation of Section 5.3 by showing that the role of the Bi-LSTM encoder extends beyond minimizing prediction error as quantitatively confirmed in Table 9, the embedding $\mathbf{h}_t$ provides contextual information that is not explicitly available from either the instantaneous

observation or the point forecast alone, supporting the complementary three-component design of the predictive state in Equation (18). The extent to which this richer representation improves actual mobility-management decisions is evaluated next in Section 5.5 through end-to-end handover performance.

Table 9. Quantitative analysis of learned knowledge representations across mobility scenarios.

Scenario	Encoder	Silhouette Score	Intra-Cluster Distance	Inter-Cluster Distance
Smart city	LSTM	0.777	0.508	1.404
	Proposed Bi-LSTM	0.811	0.317	1.722
ITS	LSTM	0.604	0.552	1.207
	Proposed Bi-LSTM	0.784	0.418	1.457
Connected vehicle	LSTM	0.513	0.599	1.011
	Proposed Bi-LSTM	0.634	0.455	1.310

5.5. Handover Performance and Statistical Evaluation

The end-to-end effectiveness of the proposed hybrid Bi-LSTM–DRL framework was evaluated under two complementary stress conditions. First, UE velocity was varied from 0 to 33 m/s to determine the robustness of the handover policy against increasingly rapid mobility and channel variation. Second, the number of simultaneously active UEs was increased from 50 to 500 to examine scalability under increasing network load. The proposed framework was compared with B1 (Reactive 3GPP), B2 (DRL-only), B3 (Prediction-only), and B4 (Prediction-assisted DRL), as defined in Section 5.2. Performance was evaluated using throughput, handover latency, handover failure rate (HFR), ping-pong rate (PPR), and average utility, which collectively reflect the principal objectives incorporated into the reward formulation of Equation (20). Each experimental configuration was independently repeated 10 times ($n = 10$) using different random seeds to account for stochastic variations in UE mobility, wireless-channel realizations, and learning behavior. All reported results represent the mean over these 10 independent simulation runs, with error bars indicating the corresponding 95% confidence intervals (CIs).

5.5.1. Impact of UE Velocity on Handover Performance

Figures 3–7 investigate the effect of increasing UE velocity on end-to-end mobility-management performance. This experiment is particularly important because increasing velocity directly increases the rate at which UE–cell distance, RSRP, SINR, and Doppler-dependent channel conditions evolve, as described by the mobility and channel models in Sections 3.2 and 3.3. Consequently, the available decision interval becomes progressively shorter, making reactive handover mechanisms increasingly susceptible to delayed handovers and unstable cell selection.

Throughput performance: Figure 3 shows the average throughput as UE velocity increases from stationary conditions to 33 m/s. A progressive throughput degradation is observed for all methods as mobility increases. This trend is theoretically expected because faster UE movement shortens the residence time within individual cells and produces more rapid channel variations, thereby increasing the frequency of handover interruptions and reducing the time available for stable data transmission. The degradation is most pronounced for B1. Because the conventional 3GPP mechanism relies on instantaneous signal measurements, hysteresis, and TTT rather than anticipated channel evolution, its handover decisions are initiated only after the predefined triggering condition has been satisfied. At high velocities, the serving link may therefore deteriorate considerably before the handover procedure is completed. B2 improves upon B1 through adaptive DRL-based decision-making, demonstrating that policy learning can provide greater robustness than static threshold rules. However, because B2 operates only on the instantaneous observation $\mathbf{o}_u(t)$, its decisions remain fundamentally reactive. B3 benefits from future-condition prediction but lacks the long-term policy optimization of DRL. B4 therefore performs more favorably by combining prediction with adaptive decision-making.

The proposed framework maintains the highest throughput over the evaluated velocity range. More importantly, its relative advantage becomes more evident toward the high-mobility region. This behavior is consistent with the prediction results in Section 5.3, where the Bi-LSTM demonstrated its strongest relative advantage under ITS and connected vehicle conditions. By combining the instantaneous observation $\mathbf{o}_u(t)$, predicted observation $\hat{\mathbf{o}}_u(t + \Delta)$, and latent embedding $\mathbf{h}_t$, the DRL agent can initiate a handover before the serving link enters a severely degraded state rather than waiting for a reactive threshold violation.

Handover latency: Figure 4 provides complementary evidence by examining handover latency. Unlike throughput, lower values indicate better performance. Handover latency increases with UE velocity for all methods, reflecting the greater difficulty of maintaining a stable target-cell association as mobility becomes more rapid. The steepest increase is observed for B1, whereas the learning-based methods exhibit progressively smaller latency penalties. Although B2 can learn more appropriate actions than fixed 3GPP rules, it cannot explicitly anticipate future signal deterioration. B3 can forecast such deterioration, but its subsequent decision remains threshold-based. The superior performance of B4 relative to B2 and B3 therefore demonstrates that prediction and adaptive decision-making are complementary rather than interchangeable components. The proposed method exhibits the lowest handover latency throughout the velocity range and a comparatively slower rate of degradation at high velocities. The result indicates that the predictive state of Equation (18) provides the Dueling DDQN with sufficient temporal context to identify an appropriate target cell before severe link degradation occurs. This reduces the likelihood of late handover execution and consequently decreases interruption time.

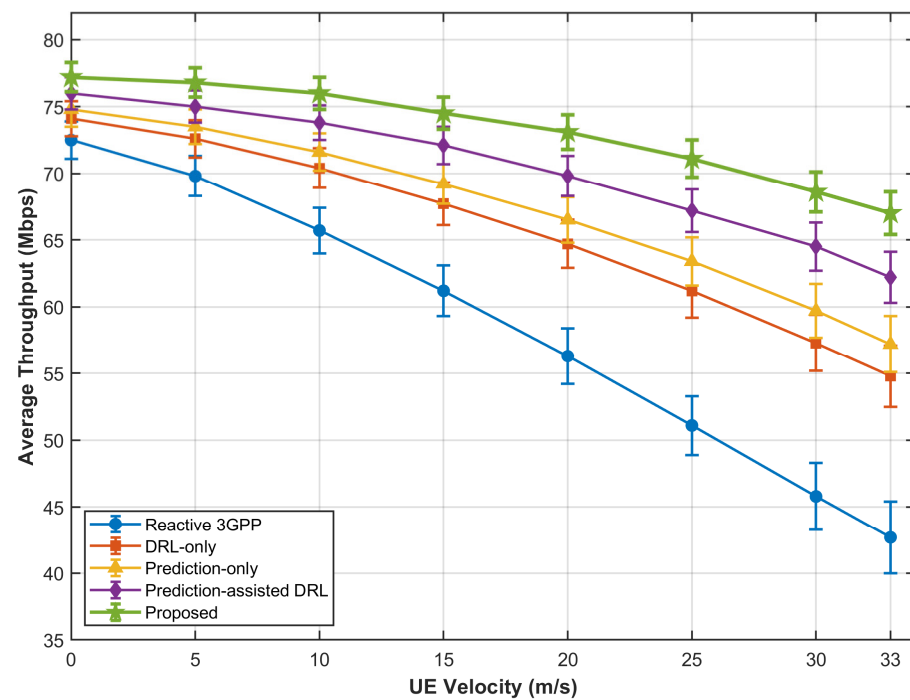


Figure 3. Average throughput performance of the proposed framework and baseline methods under increasing UE velocity. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

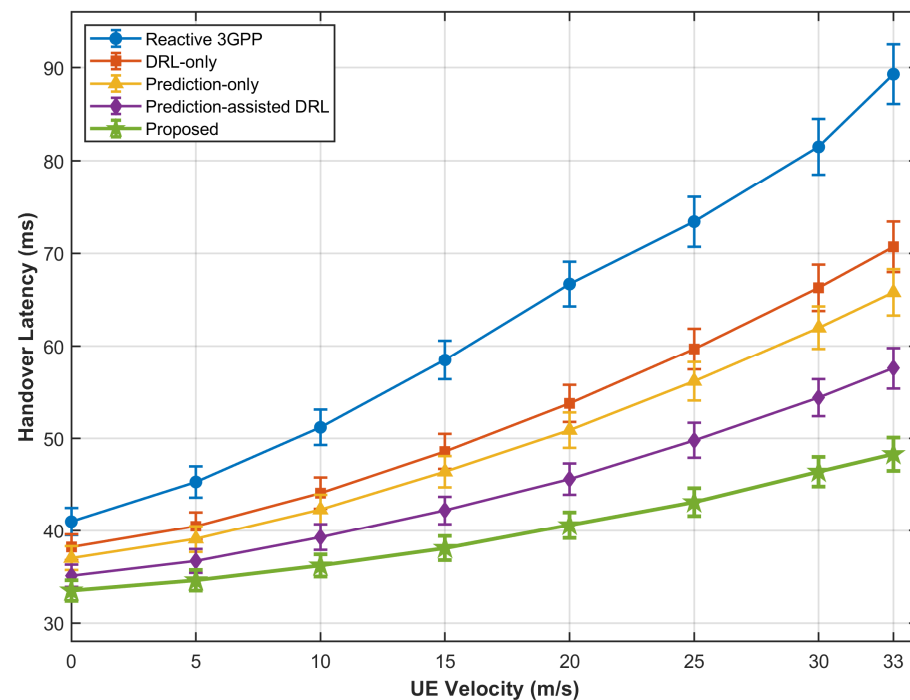


Figure 4. Handover latency of the proposed framework and baseline methods under varying UE velocities. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

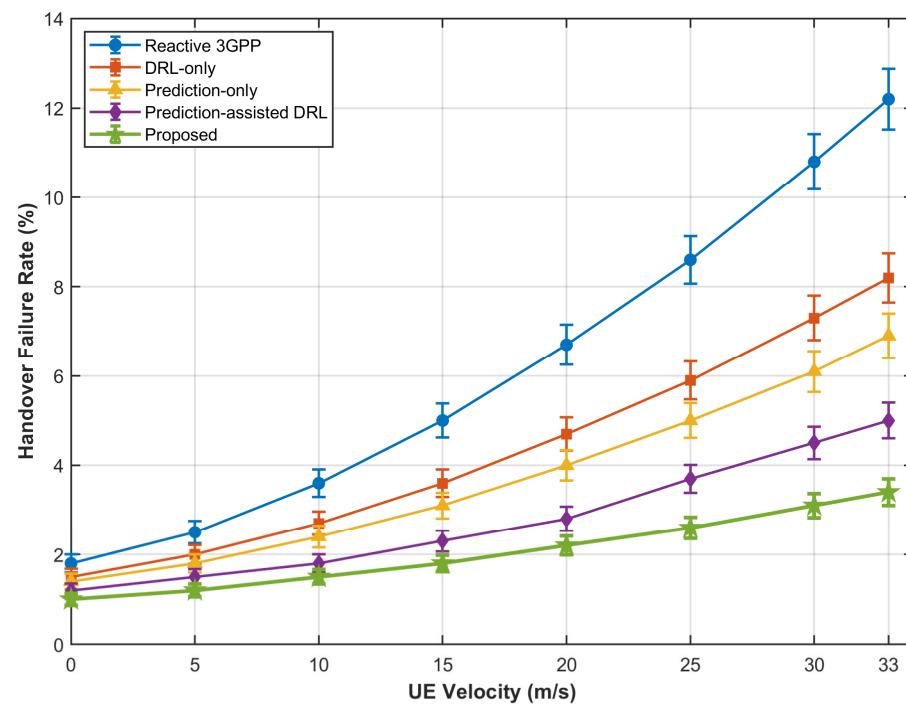


Figure 5. Handover failure rate of the proposed framework and baseline methods under varying UE velocities. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

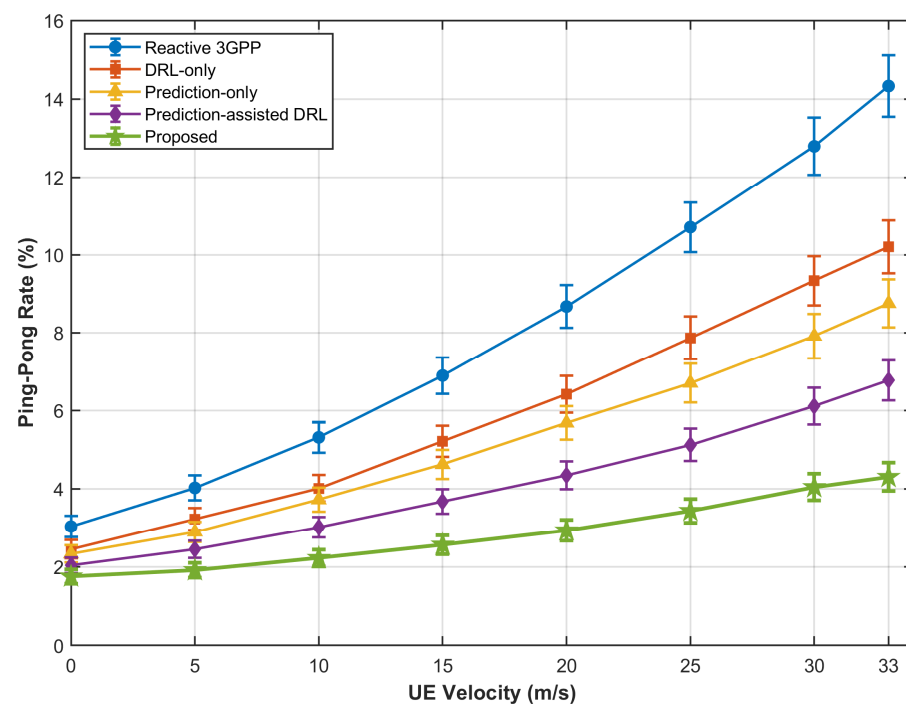


Figure 6. Ping-pong rate of the proposed framework and baseline methods under varying UE velocities. Error bars indicate 95% confidence intervals across independent simulation runs.

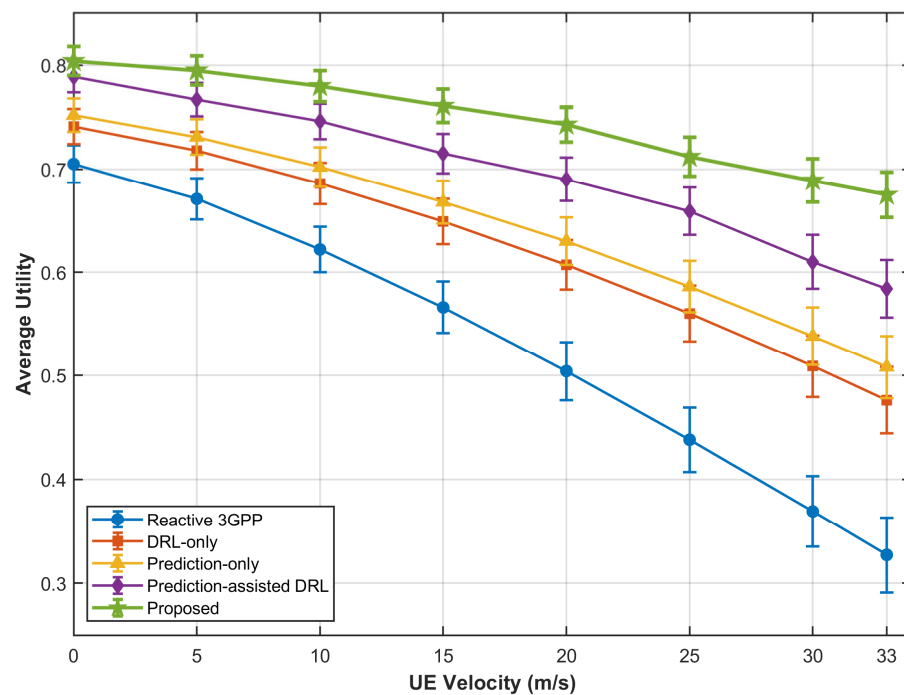


Figure 7. Average utility of the proposed framework and baseline methods under varying UE velocities. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

5.5.2. Handover Reliability Under Increasing Mobility

Figure 5 shows that the handover failure rate increases with UE velocity for all five methods. This trend follows directly from the decreasing time available to detect serving-link deterioration, select a target cell and complete the handover procedure. B1 exhibits the largest increase because its reactive trigger is particularly vulnerable to rapidly changing channel conditions. B2 reduces failures through learned action selection, while B3 benefits from anticipating future signal conditions. Nevertheless, neither mechanism alone provides the same level of robustness as the integrated approaches. The proposed framework consistently yields the lowest HFR, with its advantage becoming increasingly relevant in the high-velocity region. This result provides direct evidence that the prediction improvements reported in Tables 7 and 8 are not merely reductions in forecasting error; they translate into a measurable improvement in an end-to-end network reliability metric. In particular, the Bi-LSTM forecast allows the agent to identify deteriorating serving links earlier, while the learned knowledge embedding provides additional temporal context that assists the Dueling DDQN in distinguishing persistent degradation from transient fading.

Figure 6 evaluates the ping-pong rate. The PPR increases with velocity because rapidly varying RSRP and SINR values can repeatedly change the apparent ranking of neighboring cells, particularly around overlapping cell boundaries. Such behavior is especially problematic in ultra-dense HetNets, where several candidate cells may exhibit similar instantaneous signal strengths. The comparatively high PPR of B1 illustrates this sensitivity to short-term measurement fluctuations. DRL-based B2 reduces unnecessary transitions because the learned policy accounts for the long-term reward penalties associated with ping-pong events. Prediction-based B3 further demonstrates that anticipated signal evolution can stabilize target-cell selection.

The proposed framework achieves the lowest PPR because its handover decision is conditioned not only on instantaneous measurements but also on predicted network evolution and the temporal knowledge embedding. The latter is particularly important in this experiment. As discussed in Section 5.4, a temporary RSRP reduction and a sustained deterioration trend may produce similar instantaneous observations but represent fundamentally different mobility states. The latent representation provides the decision agent with temporal context for discriminating between these situations, thereby reducing unnecessary back-and-forth handovers.

5.5.3. Joint Performance Trade-Off

Whereas Figures 3–6 examine individual performance dimensions, Figure 7 reports the average utility obtained from the common reward formulation. For B1 and B3, which do not directly optimize a reinforcement-learning objective, the same reward function is evaluated post hoc solely to provide a common performance criterion, as specified in Section 5.2. The average utility decreases as UE velocity increases for all methods. This decline is consistent with the simultaneous reduction in throughput and increases in latency, handover failures, and ping-pong events observed in Figures 3–6. Importantly, the proposed framework maintains the highest utility across the complete mobility range.

This result is particularly relevant because optimizing throughput alone could produce overly aggressive handover policies, whereas minimizing the number of handovers alone could retain UEs on deteriorating serving cells. The reward formulation of Equation (20) instead evaluates the combined consequences of throughput, interruption latency, handover failure, ping-pong behavior, and signaling cost. The superior utility of the proposed framework therefore indicates that its improvement does not arise from optimizing one metric at the expense of the remaining objectives. The increasing separation between the proposed framework and the reactive baseline toward high velocities further supports the central hypothesis of this work: The value of predictive mobility information increases as the network dynamics become faster relative to the handover decision interval. Under low mobility, current observations remain informative for a comparatively long period, reducing the incremental value of prediction. Under high mobility, however, instantaneous measurements become obsolete more rapidly, making anticipated network evolution substantially more valuable for decision-making.

5.5.4. Scalability Under Increasing Network Load

Figures 8–10 extend the evaluation from mobility robustness to network-load scalability by increasing the number of active UEs from 50 to 500 while maintaining the network configuration and resource assumptions specified in Section 5.1. This experiment determines whether the benefits of predictive handover remain observable when radio resources become increasingly contested.

Figure 8 shows the average per-UE throughput. As expected, the throughput decreases with increasing numbers of active UEs because a larger user population competes for finite radio resources. At relatively light load, the differences among the learning-based methods are moderate because sufficient resources remain available to accommodate suboptimal handover decisions. As the number of active UEs increases, however, target-cell selection becomes increasingly consequential. B1 exhibits the strongest degradation because its RSRP-based trigger does not explicitly account for the temporal evolution of the network state. B2 improves resource utilization through adaptive decision-making but remains dependent on instantaneous information. B3 can anticipate future link conditions but lacks long-term decision optimization. B4 provides stronger performance by integrating prediction and DRL. The proposed framework maintains the highest average per-UE throughput across the entire load range. Its advantage under heavy

load suggests that the predictive state enables the agent to avoid handovers toward candidate cells that may appear attractive at the current epoch but are likely to become unfavorable shortly afterward. The result therefore complements Figure 3 such as the proposed architecture remains effective not only when mobility dynamics intensify but also when network-resource contention increases.

Figure 9 examines the handover latency under the same load conditions. The handover latency increases with the number of active UEs because greater resource contention can delay radio-resource allocation and handover execution. The increase is largest for B1 and progressively smaller for the AI-assisted approaches. The proposed framework again achieves the lowest latency and exhibits the smallest degradation as network load increases. The result suggests that proactive target selection reduces the probability that a UE initiates or completes a handover only after the serving link has entered a critical state. This advantage becomes increasingly important under heavy load because recovery from a poorly timed handover becomes more costly when neighboring cells simultaneously serve large numbers of users.

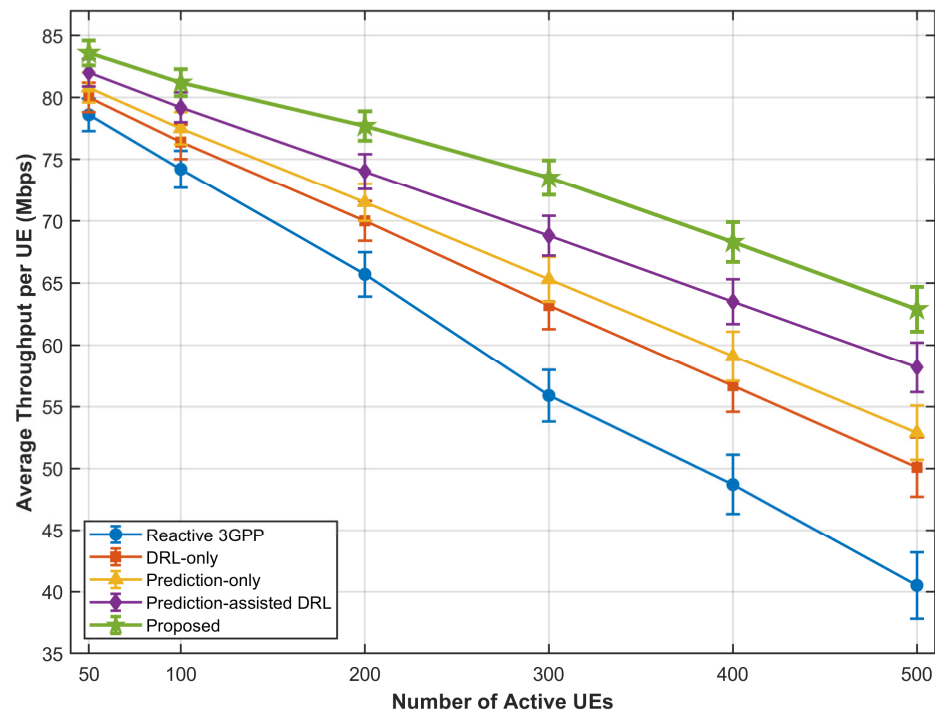


Figure 8. Average per-UE throughput of the proposed framework and baseline methods under increasing numbers of active UEs. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

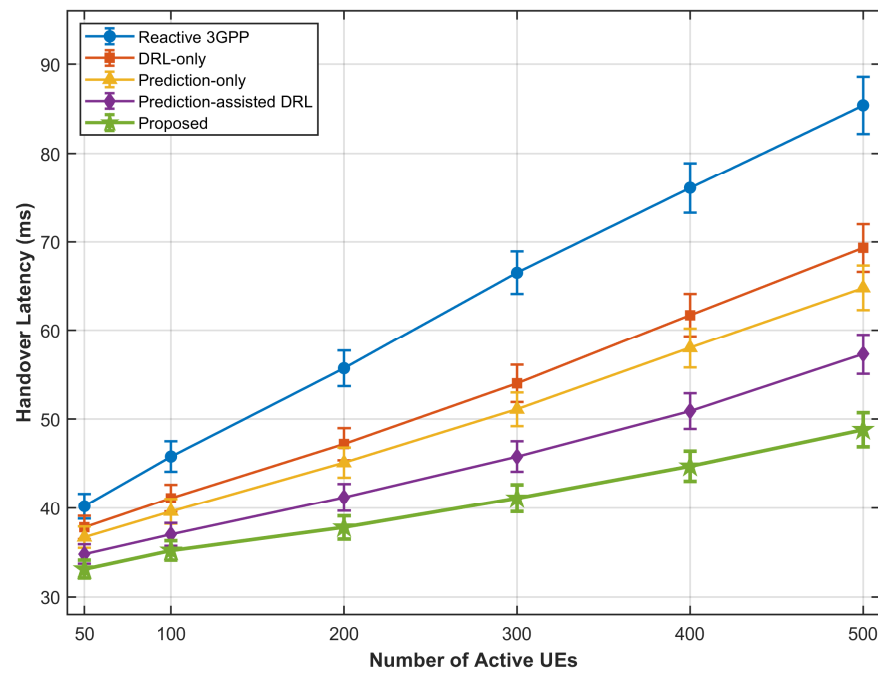


Figure 9. Handover latency of the proposed framework and baseline methods under increasing numbers of active UEs. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

The scalability analysis is completed by the HFR results in Figure 10. The HFR increases nonlinearly with the number of active UEs, particularly at the upper end of the evaluated range. Such behavior is theoretically plausible because increasing contention not only reduces available resources but also decreases the margin available for successful handover completion under rapidly changing radio conditions.

The proposed framework demonstrates the lowest HFR throughout the load range and, critically, its performance deteriorates more slowly than that of the baselines. This indicates that the proposed approach does not obtain its mobility benefit only under lightly loaded conditions. Rather, the combination of temporal prediction and policy optimization remains effective when handover decisions must be made under simultaneous mobility and resource contention. The growing separation between the proposed method and B4 at high loads is also important. Because B4 already incorporates prediction-assisted DRL, this difference cannot be attributed simply to the presence of prediction. Instead, it is consistent with the additional contributions introduced in the proposed framework, namely the explicit knowledge embedding and closed-loop adaptive refresh. This interpretation should subsequently be verified through the ablation experiments defined in Section 5.2.

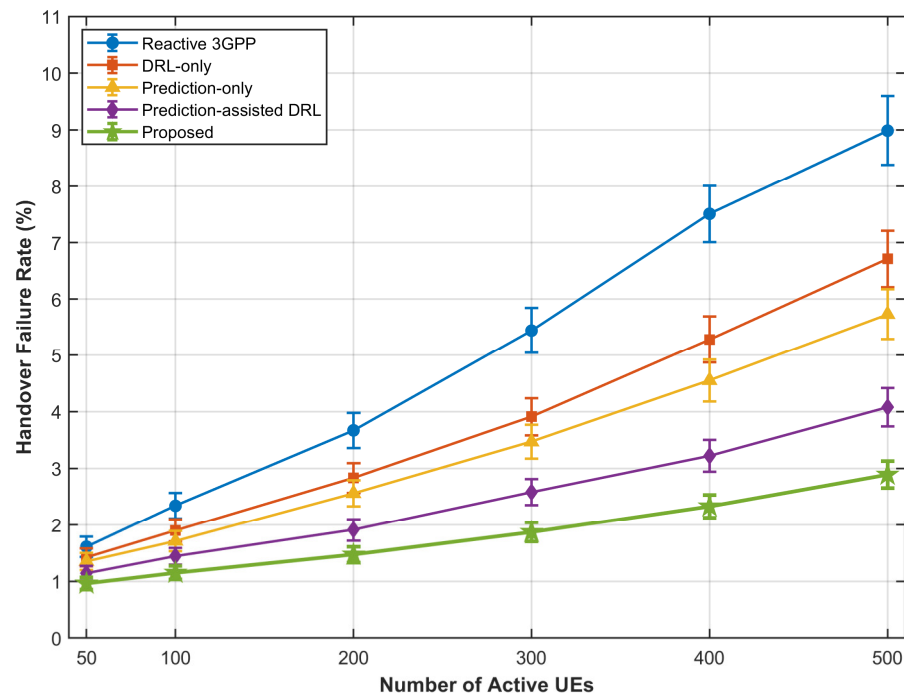


Figure 10. Handover failure rate of the proposed framework and baseline methods under increasing numbers of active UEs. Error bars indicate 95% confidence intervals computed across 10 independent simulation runs.

5.5.5. Comparison with the Model-Based Predictive Baseline

To further distinguish the benefit of mobility prediction itself from that of learned temporal representation and DRL-based decision optimization, an additional end-to-end comparison was conducted against the model-based CV-KF baseline defined in Section 5.2. Rather than repeating the CV-KF results across Figures 3–10, Table 10 summarizes its performance under the representative high-mobility condition using the same network, channel, handover, and evaluation settings as the other methods.

Table 10. End-to-end performance comparison with the model-based predictive baseline under high-mobility conditions.

Method	Throughput (Mbps)	Handover Latency (ms)	HFR (%)	PPR (%)	Average Utility
B1: Reactive 3GPP	52.1 ± 1.40	67.5 ± 2.70	7.20 ± 0.68	8.40 ± 0.80	0.515 ± 0.035
B2: DRL-only	59.8 ± 1.30	58.7 ± 2.20	5.20 ± 0.55	6.45 ± 0.68	0.592 ± 0.032
B3: Prediction-only	60.9 ± 1.30	56.4 ± 2.10	4.91 ± 0.50	6.10 ± 0.62	0.610 ± 0.030
B4: Prediction-assisted DRL	64.9 ± 1.20	51.5 ± 1.90	4.02 ± 0.40	4.95 ± 0.50	0.653 ± 0.028
B5: CV-KF-based predictive HO	63.8 ± 1.20	53.1 ± 1.70	4.28 ± 0.50	5.31 ± 0.40	0.641 ± 0.025
Proposed	67.0 ± 1.10	48.1 ± 1.40	3.47 ± 0.30	4.33 ± 0.35	0.678 ± 0.022

Table 10 shows that B5 (CV-KF-based predictive handover) improves substantially upon the reactive B1 baseline, increasing the throughput from 52.1 to 63.8 Mbps while reducing the handover latency from 67.5 to 53.1 ms, the HFR from 7.20% to 4.28%, and the PPR from 8.40% to 5.31%. This confirms that explicit model-based mobility anticipation alone provides meaningful end-to-end handover benefits. However, the proposed framework further improves the throughput to 67.0 Mbps and reduces the handover latency, the HFR, and the PPR to 48.1 ms, 3.47%, and 4.33%, respectively, while achieving the highest average utility of 0.678. Together with Tables 7 and 10, these results demonstrate that

the performance gain of the proposed framework is not attributable solely to prediction; learned temporal representation and DRL-based sequential decision optimization provide additional and complementary benefits under high-mobility conditions.

Statistical evaluation: To determine whether the observed performance differences resulted from systematic improvements rather than stochastic simulation variability, each experimental configuration was independently repeated 10 times ($n = 10$) using different random seeds. For each performance metric and operating point, the sample mean and corresponding 95% confidence interval were calculated across the 10 independent runs as

$$CI_{95\%} = \bar{x} \pm t_{0.975, n-1} \frac{s}{\sqrt{n}}, \quad (27)$$

where $\bar{x}$ denotes the sample mean, s is the sample standard deviation, and $n = 10$ is the number of independent simulation runs. Because the number of independent runs is relatively small, the confidence intervals were calculated using Student's t -distribution rather than the normal approximation. For $n = 10$, the corresponding critical value is $t_{0.975, 9} = 2.262$. Accordingly, the error bars reported in Figures 3–10 represent 95% confidence intervals of the estimated mean rather than standard deviations.

To further determine whether the performance gains of the proposed framework were statistically significant, identical operating conditions were maintained across competing methods within each independent run, enabling paired statistical comparisons. The normality of the paired differences was first examined using the Shapiro–Wilk test. When the normality assumption was satisfied, a two-sided paired t -test was applied; otherwise, the Wilcoxon signed-rank test was employed. Statistical significance was assessed at $\alpha = 0.05$. Because the proposed framework was compared against four baseline methods (B1–B4), the Holm–Bonferroni correction was applied to control the family-wise error rate associated with multiple comparisons. In addition to the corrected p -values, Cohen's d_z was calculated to quantify the magnitude of the paired performance differences.

Table 11 presents the ablation analysis under high-mobility conditions to quantify the individual and complementary contributions of the principal components of the proposed framework. Removing or modifying any key component degrades performance across the evaluated metrics. The complete framework achieves the highest throughput (67.0 Mbps) and average utility (0.678), while attaining the lowest handover latency (48.1 ms), HFR (3.47%), and PPR (4.33%). These results indicate that the overall gain arises from the integration of predictive mobility information, temporal knowledge representation, bidirectional sequence modeling, and closed-loop adaptation rather than from any single component.

Table 11. Ablation analysis of the proposed framework under high-mobility conditions.

Method	Throughput (Mbps)	Handover Latency (ms)	HFR (%)	PPR (%)	Average Utility
AV1: w/o predicted observation	61.9 ± 1.90	56.5 ± 2.10	4.82 ± 0.39	6.21 ± 0.48	0.611 ± 0.026
AV2: w/o knowledge embedding	60.6 ± 1.80	58.1 ± 2.00	5.11 ± 0.41	6.65 ± 0.51	0.596 ± 0.028
AV3: Unidirectional LSTM	63.5 ± 1.80	54.6 ± 2.00	4.34 ± 0.36	5.79 ± 0.44	0.637 ± 0.025
AV4: Frozen encoder	65.1 ± 1.75	52.9 ± 1.90	4.05 ± 0.34	5.10 ± 0.40	0.652 ± 0.024
AV5: w/o closed-loop refresh	62.3 ± 1.85	55.3 ± 2.05	4.62 ± 0.37	5.94 ± 0.46	0.624 ± 0.026
AV6: DRL refresh-only	65.9 ± 1.72	51.2 ± 1.92	3.93 ± 0.32	4.93 ± 0.37	0.664 ± 0.023
AV7: Encoder refresh-only	64.3 ± 1.76	53.5 ± 1.96	4.27 ± 0.35	5.56 ± 0.40	0.649 ± 0.025
Proposed	67.0 ± 1.60	48.1 ± 1.80	3.47 ± 0.30	4.33 ± 0.36	0.678 ± 0.022

The importance of the predictive state components is particularly evident from AV1 and AV2. Removing the explicit predicted observation in AV1 decreases the throughput from 67.0 to 61.9 Mbps and increases the handover latency from 48.1 to 56.5 ms. The HFR and PPR also increase from 3.47% and 4.33% to 4.82% and 6.21%, respectively. These degradations demonstrate that explicit knowledge of anticipated network conditions enables the DRL agent to act before substantial deterioration of the serving link occurs. AV2, which removes the knowledge embedding $\mathbf{h}_t$, exhibits an even greater degradation, achieving a throughput of only 60.6 Mbps and an average utility of 0.596, together with the highest latency (58.1 ms), HFR (5.11%), and PPR (6.65%) among the evaluated ablation variants. This observation suggests that the learned temporal embedding provides decision-relevant contextual information that cannot be fully represented by the explicit future prediction alone. Consequently, the results of AV1 and AV2 support the use of both $\hat{\mathbf{o}}_u(t + \Delta)$ and $\mathbf{h}_t$ in the predictive state representation rather than treating these components as interchangeable sources of mobility information.

AV3 further demonstrates the contribution of bidirectional temporal modeling. Replacing the Bi-LSTM with a unidirectional LSTM reduces the throughput to 63.5 Mbps and the average utility to 0.637, while increasing the handover latency, HFR, and PPR to 54.6 ms, 4.34%, and 5.79%, respectively. This consistent degradation indicates that bidirectional temporal representation improves mobility-state characterization under rapidly varying network conditions. AV4 is retained as an architectural ablation that evaluates the effect of freezing the predictive encoder and is therefore interpreted separately from the refresh-specific factorial experiment. With the encoder frozen after pretraining, AV4 achieves a throughput of 65.1 Mbps, a handover latency of 52.9 ms, an HFR of 4.05%, a PPR of 5.10%, and an average utility of 0.652. Its degradation relative to the complete framework indicates that adapting the predictive representation to evolving mobility and channel conditions provides additional benefit beyond a fixed pretrained representation. To separately quantify the two adaptation channels, AV5, AV6, AV7, and the proposed framework form a controlled refresh-specific comparison. AV5 disables both refresh mechanisms, whereas AV6 enables only DRL-policy refresh and AV7 enables only encoder refresh. Relative to AV5, AV6 increases the throughput from 62.3 to 65.9 Mbps and the average utility from 0.624 to 0.664, while reducing the handover latency from 55.3 to 51.2 ms and the HFR from 4.62% to 3.93%. AV7 also improves upon AV5, achieving a throughput of 64.3 Mbps, a handover latency of 53.5 ms, an HFR of 4.27%, and an average utility of 0.649. These results demonstrate independent benefits from both adaptation channels, with DRL-policy refresh providing a larger individual gain under the evaluated conditions. Importantly, jointly enabling both mechanisms yields the best overall performance (a throughput of 67.0 Mbps, a latency of 48.1 ms, an HFR of 3.47%, and an average utility of 0.678), demonstrating that encoder and policy adaptation provide complementary rather than interchangeable benefits. Relative to AV5, DRL-only and encoder-only refresh reduce the HFR by approximately 14.9% and 7.6%, respectively, whereas their joint operation reduces the HFR by approximately 24.9%.

Collectively, the ablation results reveal complementary contributions from the major architectural components. Explicit future prediction supports anticipatory decisions, the knowledge embedding supplies latent temporal context, bidirectional modeling improves mobility-state representation, and the refresh-specific experiments demonstrate independent contributions from encoder and DRL-policy adaptation. The superior performance obtained when both refresh mechanisms are enabled further supports the closed-loop AI-native design of the proposed hybrid Bi-LSTM–DRL framework under non-stationary high-mobility conditions.

Taken together, Figures 3–10 and the ablation results in Table 11 demonstrate the robustness and scalability of the proposed framework under both increasing mobility and

network load. While all methods experience performance degradation under more challenging conditions, the proposed framework consistently maintains superior throughput, lower handover latency, HFR, and PPR, and higher aggregate utility. These improvements are consistent with the prediction and knowledge-representation results reported in Sections 5.3 and 5.4, indicating that the integration of Bi-LSTM-based future-state prediction and temporal knowledge extraction with Dueling DDQN translates predictive information into more effective end-to-end handover decisions. The ablation results further confirm the complementary contributions of prediction, knowledge embedding, bidirectional temporal modeling, and the independently evaluated encoder- and DRL-policy adaptation mechanisms. Collectively, these findings support the robustness and scalability of the proposed framework for AI-native mobility management in high-mobility and densely connected 5G/6G environments.

5.5.6. Reward-Weight Sensitivity Analysis

To evaluate the sensitivity of the proposed framework to reward design, the default balanced configuration was compared with throughput-, latency-, and reliability-oriented profiles. Table 12 summarizes the four representative reward-weight configurations, where the ping-pong and signaling weights were fixed at 0.10 and the remaining weights were redistributed among throughput, handover latency, and handover failure. These configurations represent distinct service priorities and provide interpretable reference operating points for the sensitivity analysis.

Table 12. Representative reward-weight configurations used in the sensitivity analysis.

Configuration	w_{thr}	w_{lat}	w_{fail}	w_{pp}	w_{sig}
Balanced	0.35	0.25	0.20	0.10	0.10
Throughput-oriented	0.50	0.15	0.15	0.10	0.10
Latency-oriented	0.25	0.40	0.15	0.10	0.10
Reliability-oriented	0.25	0.15	0.40	0.10	0.10

To examine the trade-off more systematically, the analysis was extended to 12 reward-weight configurations, including intermediate and mixed-priority operating points around the four representative profiles in Table 12. Figure 11 presents the resulting Pareto analysis in terms of the throughput, HFR, and handover latency. Throughput-oriented configurations achieve higher throughput at the expense of increased HFR and latency, whereas reliability- and latency-oriented configurations favor their respective objectives. The balanced configuration provides a competitive compromise across the three KPIs and remains close to the Pareto-efficient region, supporting its use as the nominal operating point without implying universal optimality.

5.5.7. Robustness Under 3GPP TR 38.901 Channel Profiles

To assess whether the observed handover gains are specific to the simplified propagation abstraction of Equations (5)–(8), a representative subset of the evaluation was repeated using standardized 3GPP TR 38.901 Clustered Delay Line (CDL) channel profiles. CDL-C was considered for the urban scenario, whereas CDL-D was used for the high-mobility scenario. The mobility-management algorithms, reward formulation, learning configuration, and evaluation protocol were otherwise kept unchanged so that the effect of the channel-model refinement could be isolated. Each configuration was evaluated over 10 independent runs, and the results are reported as the mean with 95% confidence intervals.

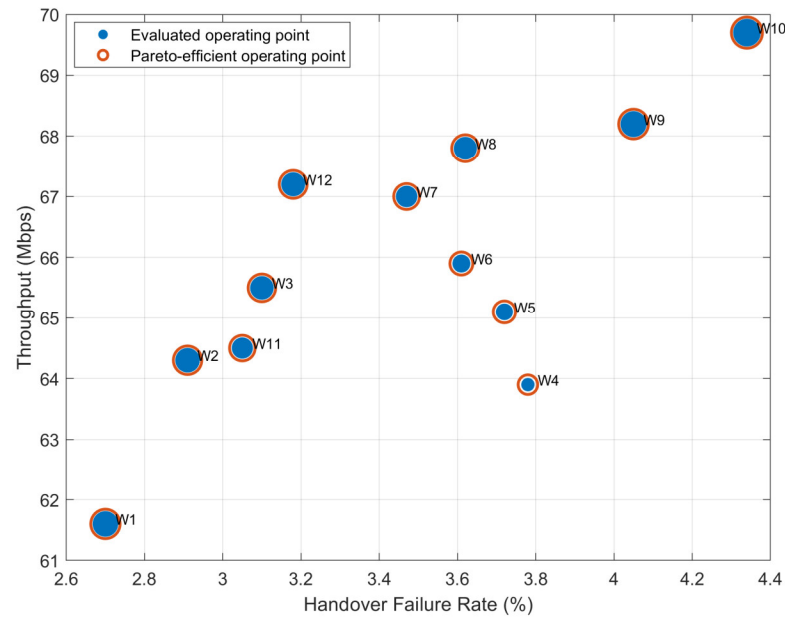


Figure 11. Pareto analysis of the throughput, handover failure rate (HFR), and handover latency across 12 reward-weight configurations. The horizontal and vertical axes represent the HFR and throughput, respectively, while marker size represents the handover latency. Outlined markers indicate Pareto-efficient operating points.

Table 13 evaluates the robustness of the proposed framework under standardized 3GPP TR 38.901 CDL channel profiles. Although the richer multipath conditions degrade the absolute performance of both methods relative to the baseline model in Equations (5)–(8), the proposed framework consistently outperforms B4. Under urban CDL-C, it achieves 5.6% higher throughput, 20.9% lower HFR, and 8.5% lower handover latency than B4, while under high-mobility CDL-D, the corresponding improvements are 7.9%, 19.9%, and 12.7%, respectively. The proposed framework also maintains higher average utility under both CDL profiles. Thus, despite the degradation in absolute KPIs under more realistic channel conditions, the relative performance ordering remains unchanged, providing evidence that the observed mobility-management gains are robust to standardized channel modeling and are not solely attributable to the simplified propagation model in Equations (5)–(8).

Table 13. Robustness comparison under the baseline propagation model and 3GPP TR 38.901 CDL channel profiles.

Scenario	Channel Model	Method	Throughput (Mbps)	HFR (%)	HO Latency (ms)	Average Utility
Urban	Baseline Equations (5)–(8)	B4	64.2 ± 1.72	4.05 ± 0.34	52.9 ± 1.90	0.652 ± 0.024
Urban	Baseline Equations (5)–(8)	Proposed	67.0 ± 1.60	3.47 ± 0.30	48.1 ± 1.80	0.678 ± 0.022
Urban	3GPP CDL-C	B4	58.8 ± 1.70	5.03 ± 0.38	56.7 ± 2.10	0.621 ± 0.026
Urban	3GPP CDL-C	Proposed	62.1 ± 1.66	3.98 ± 0.35	51.9 ± 2.02	0.648 ± 0.024
High mobility	Baseline Equations (5)–(8)	B4	60.7 ± 1.88	5.10 ± 0.41	57.8 ± 2.10	0.613 ± 0.027

High mobility	Baseline Equations (5)–(8)	Proposed	63.9 ± 1.76	4.29 ± 0.36	52.6 ± 1.95	0.646 ± 0.025
High mobility	3GPP CDL-D	B4	54.2 ± 1.89	6.53 ± 0.46	63.6 ± 2.28	0.567 ± 0.029
High mobility	3GPP CDL-D	Proposed	58.5 ± 1.71	5.23 ± 0.43	55.5 ± 2.11	0.603 ± 0.027

5.5.8. Computational Efficiency and Runtime Analysis

In addition to handover performance, the computational cost of each learning-based method was evaluated on the workstation described in Section 5.1. Three quantities were measured: training time, inference latency, and runtime overhead. Training time denotes the wall-clock duration required to complete model training under the stopping criterion defined in Section 5.1. Inference latency denotes the average wall-clock time required to process the current observation and generate a handover action for a single UE. This computational inference latency is distinct from the network-level handover latency reported in Figures 4 and 9, which represents communication-service interruption during handover execution. Runtime overhead is defined relative to B2 (DRL-only) as

$$O_{runtime}(\%) = \frac{T_{method} - T_{B2}}{T_{B2}} \times 100. \quad (28)$$

This empirical evaluation complements the asymptotic complexity analysis in Section 4.5 and quantifies the additional computational cost introduced by temporal prediction and closed-loop adaptation.

Table 14 summarizes the empirical computational efficiency of the evaluated methods under the common hardware and software environment described in Section 5.1. As expected, B1 incurs the lowest computational cost because it requires no model training. Among the learning-based baselines, B3 exhibits lower computational demand than B2, with a total runtime of 15.3 ± 0.6 s/episode, corresponding to a 13.1% reduction relative to B2. Combining prediction and reinforcement learning increases the computational cost. B4 requires 18.8 ± 0.8 s/episode (+6.8%), whereas the proposed framework requires 61.3 ± 2.2 min for training, 1.35 ± 0.05 ms/UE for inference, and 20.1 ± 0.7 s/episode, corresponding to a 14.2% runtime overhead relative to B2. This additional cost mainly results from integrating Bi-LSTM-based prediction and temporal knowledge extraction with Dueling DDQN-based handover decision-making.

Table 14. Computational efficiency comparison of the proposed framework and baseline methods in terms of training time, per-UE inference latency, total runtime per simulation episode, and runtime overhead relative to the DRL-only baseline (B2).

Method	Training Time (min)	Inference Latency (ms/UE)	Total Runtime (s/Episode)	Runtime Overhead vs. B2 (%)
B1: Reactive 3GPP	N/A	0.18 ± 0.02	10.3 ± 0.5	N/A
B2: DRL-only	40.1 ± 1.7	0.91 ± 0.04	17.6 ± 0.7	0
B3: Prediction-only	26.6 ± 1.3	0.59 ± 0.03	15.3 ± 0.6	−13.1
B4: Prediction-assisted DRL	56.5 ± 2.1	1.09 ± 0.05	18.8 ± 0.8	+6.8
Proposed	61.3 ± 2.2	1.35 ± 0.05	20.1 ± 0.7	+14.2

Despite the additional computation, the proposed framework maintains a low absolute inference latency. The measured 1.35 ms/UE corresponds to only approximately 1.35% of the 100 ms decision interval adopted in the simulations, indicating that the decision can be completed well within each decision epoch under the evaluated implementation. Together with the performance improvements reported in Figures 3–10, these results

demonstrate a favorable performance–complexity trade-off. Moreover, the empirical measurements complement the asymptotic analysis in Section 4.5 by quantitatively showing that the additional prediction and knowledge-extraction components introduce measurable but bounded computational overhead.

5.5.9. Empirical Computational Scalability with Network Size

To empirically evaluate the network-size scalability discussed in Section 4, an additional experiment was conducted by varying the number of cells as $B \in \{20, 50, 100, 200\}$ while maintaining a fixed UE density of five UEs per cell. Accordingly, the corresponding UE populations were $U \in \{100, 250, 500, 1000\}$. The candidate-set size, observation dimension, Bi-LSTM sequence length, embedding dimension, Dueling DDQN architecture, decision interval, and hardware/software environment were kept unchanged across all configurations. Each configuration was independently repeated 10 times, and the results are reported as the mean with 95% confidence intervals. Both per-UE inference latency and total inference time per decision epoch were measured as wall-clock execution times. per-UE latency represents the elapsed time required to generate a mobility-management decision for an individual UE, whereas total inference time represents the elapsed wall-clock time required to process all active UEs within one decision epoch. The latter is measured directly at the system level and is therefore not calculated as the product of the per-UE latency and the number of UEs. Table 15 summarizes the resulting wall-clock inference and training measurements.

Table 15. Wall-clock computational scalability with increasing network size at a fixed UE density of five UEs per cell.

Cells B	UEs U	UE Density	Per-UE Wall-Clock Inference Latency (ms)	Total Wall-Clock Inference Time per Decision Epoch (ms)	Total Training Time (min)
20	100	5	2.75 ± 0.10	244.10 ± 0.12	45.4 ± 1.6
50	250	5	2.81 ± 0.09	617.55 ± 0.10	67.5 ± 2.1
100	500	5	2.90 ± 0.09	1245.50 ± 0.10	100.9 ± 3.1
200	1000	5	2.95 ± 0.11	2499.16 ± 0.11	185.6 ± 5.2

As shown in Table 15, the measured per-UE wall-clock inference latency remains relatively stable as the network size increases from 20 to 200 cells, rising from 2.75 to 2.95 ms (approximately 7.3%) despite a tenfold increase in network size. Because the candidate-set size, observation dimension, sequence length, and neural-network architecture remain fixed, this behavior is consistent with the $O(1)$ algorithmic per-UE inference complexity with respect to B . The modest increase in wall-clock latency reflects practical execution overhead and does not indicate growth in the dimensionality of the neural inference problem. At the aggregate level, the measured total wall-clock inference time per decision epoch increases from 244.10 ms for 100 UEs to 2499.16 ms for 1000 UEs. A tenfold increase in the UE population therefore produces an approximately 10.24-fold increase in aggregate wall-clock inference time, which is empirically consistent with approximately $O(U)$ scaling of the overall inference workload. Importantly, the total wall-clock time is measured directly for the complete decision epoch rather than being obtained by multiplying the individual per-UE latency by U .

5.5.10. Sensitivity to Dual-Frequency Closed-Loop Refresh

The effect of the dual-frequency refresh mechanism was evaluated by varying the ratio K_1/K_2 from frequent joint adaptation to increasingly separated update timescales. The results show that performance is relatively stable over a moderate range of refresh ratios, whereas the two extremes exhibit different limitations. Frequent encoder refresh

increases computational cost with diminishing performance gains, while infrequent encoder adaptation reduces the ability of the temporal representation to track evolving mobility and channel statistics. The selected operating point therefore represents a compromise between adaptation responsiveness and computational efficiency as shown in Table 16.

Table 16. Sensitivity of handover performance and measured wall-clock computational overhead to the dual-frequency refresh ratio K_1/K_2 . Results are reported as the mean $\pm$ 95% confidence interval over 10 independent runs; AV5 denotes the no-refresh configuration.

K_1/K_2	Throughput (Mbps)	HFR (%)	Average Utility	Encoder Refresh Cost (s/refresh)	DRL Refresh Cost (s/Refresh)	Amortized Wall-Clock Refresh Overhead (ms/Epoch)
1	68.5 ± 1.58	3.43 ± 0.30	0.680 ± 0.022	0.91 ± 0.09	0.47 ± 0.10	101.9 ± 4.2
5	67.8 ± 1.60	3.45 ± 0.30	0.679 ± 0.022	0.91 ± 0.09	0.47 ± 0.10	47.4 ± 3.1
10	66.4 ± 1.60	3.47 ± 0.30	0.678 ± 0.022	0.91 ± 0.09	0.47 ± 0.10	39.2 ± 2.9
50	65.5 ± 1.70	3.91 ± 0.34	0.657 ± 0.024	0.91 ± 0.09	0.47 ± 0.10	34.5 ± 2.5
100	63.1 ± 1.82	4.28 ± 0.37	0.640 ± 0.025	0.91 ± 0.09	0.47 ± 0.10	30.4 ± 2.5
No refresh (AV5)	62.3 ± 1.85	4.62 ± 0.37	0.624 ± 0.026	–	–	–

The results reveal a clear trade-off between adaptation frequency, handover performance, and computational overhead. The measured wall-clock costs of a single encoder and DRL refresh are 0.91 ± 0.09 s and 0.47 ± 0.10 s, respectively. Although $K_1/K_2 = 1$ achieves the highest throughput (68.5 ± 1.58 Mbps) and the lowest HFR ($3.43 \pm 0.30\%$), its frequent model updates result in the largest amortized wall-clock refresh overhead of 101.9 ± 4.2 ms/epoch. Increasing the ratio to 10 reduces this overhead by approximately 61.5% to 39.2 ± 2.9 ms/epoch, while maintaining competitive throughput (66.4 ± 1.60 Mbps), HFR ($3.47 \pm 0.30\%$), and average utility (0.678 ± 0.022). In contrast, ratios of 50–100 further reduce the wall-clock overhead but degrade throughput, HFR, and utility because of less frequent adaptation. The no-refresh configuration (AV5) exhibits the poorest overall performance, confirming the benefit of closed-loop model adaptation. Overall, $K_1/K_2 = 10$ provides a favorable performance–wall-clock overhead trade-off, supporting its selection as a practical operating point for the proposed dual-frequency refresh strategy.

5.6. Generalization Evaluation

The experiments presented in Sections 5.3–5.5 evaluate the proposed framework under test realizations generated from the same general operating regimes represented during model development. Although these experiments establish prediction accuracy, representation quality, mobility robustness, and scalability, they do not by themselves demonstrate whether the learned predictive representation and handover policy remain effective when the deployment conditions differ from those encountered during training. This distinction is particularly important for AI-native mobility management because practical 5G/6G networks are inherently non-stationary: UE mobility patterns, cell density, propagation characteristics, and traffic conditions may vary considerably across smart city, ITS, and connected vehicle environments. Accordingly, this subsection evaluates the ability of the proposed framework to generalize across previously unseen mobility and deployment conditions.

5.6.1. Cross-Scenario Generalization Protocol

To evaluate mobility-domain generalization, the proposed framework was trained using trajectories generated from one mobility scenario and subsequently evaluated on a

different scenario without retraining the Bi-LSTM encoder or the Dueling DDQN policy before initial deployment. The three scenarios defined in Sections 3.2 and 5.1 were considered: smart city (0–5 m/s), ITS (5–20 m/s), and connected vehicle (20–33 m/s). This evaluation therefore introduces distribution shifts in UE velocity, temporal mobility characteristics, and the corresponding rate of wireless-channel variation. In addition to cross-scenario testing, deployment-domain generalization was evaluated by varying the radio environment independently of the mobility scenario. Models trained under the urban deployment regime were evaluated under suburban and mixed regimes, while models trained under suburban conditions were evaluated under urban and mixed regimes. As defined in Section 5.1, the mixed regime represents a transition corridor between urban and suburban network conditions and therefore provides a particularly relevant test of generalization because the UE encounters changes in cell density, coverage radius, and propagation characteristics within the same trajectory.

Importantly, the initial cross-domain evaluation was performed without retraining or fine-tuning on the target domain, thereby separating true zero-shot generalization from subsequent online adaptation. After this initial evaluation, the closed-loop refresh mechanism described in Sections 4.1 and 4.3 was enabled to determine whether the proposed AI-native architecture could progressively recover performance when exposed to previously unseen conditions. This two-stage protocol distinguishes (i) the inherent transferability of the learned representation and policy from (ii) the additional benefit provided by continual closed-loop adaptation.

5.6.2. Training-Distribution Characterization and Mobility-Diversity Control

The asymmetric cross-scenario generalization evaluated in Section 5.6.3 should not be attributed solely to domain shift, because the mobility dynamics represented in the training datasets differ substantially across scenarios. To disentangle the effects of training-distribution coverage from scenario-specific domain differences, we first characterize the distributions of UE velocity, Doppler shift, and handover-event frequency for the smart city, ITS, and connected vehicle training datasets. We then conduct a controlled augmentation experiment in which the original smart city training set is supplemented with synthetic high-velocity trajectories while preserving the remaining smart city simulation configuration. This design enables us to examine whether broader mobility coverage alone can reduce the smart city-to-connected vehicle generalization gap.

Figure 12 and Table 17 reveal a progressive increase in mobility diversity from smart city to ITS and connected vehicle scenarios. The smart city training set is concentrated in the low-mobility regime (0–5 m/s), with a mean velocity of 2.19 m/s, a mean absolute Doppler shift of 16.13 Hz, and 2.21 handover events per episode. In comparison, the ITS dataset increases these values to 12.86 m/s, 95.59 Hz, and 5.86 events/episode, while the connected vehicle dataset reaches 26.79 m/s, 196.89 Hz, and 9.28 events/episode. The broader velocity and Doppler distributions and more frequent handover transitions in the higher-mobility datasets indicate that models trained on ITS or connected vehicle data are exposed to substantially richer temporal dynamics than those trained exclusively on smart city data. This distributional imbalance therefore provides a plausible explanation for part of the asymmetric generalization behavior observed in Section 5.6.3, but does not by itself establish that mobility diversity is the sole cause of the generalization gap.

To determine whether the limited mobility coverage of the smart city training set contributes causally to this degradation, we next perform a controlled augmentation experiment. Synthetic high-velocity trajectories are added to the original smart city training set to extend its velocity coverage from 0–5 m/s to 0–33 m/s, while the smart city deployment and the remaining simulation and learning settings are kept unchanged. Importantly, no connected vehicle test samples are included in the augmented training data.

The original smart city, augmented smart city, and matched connected vehicle training configurations are then evaluated on the same connected vehicle test scenario. Each configuration is independently repeated 10 times, and the results are reported as the mean $\pm$ 95% confidence interval.

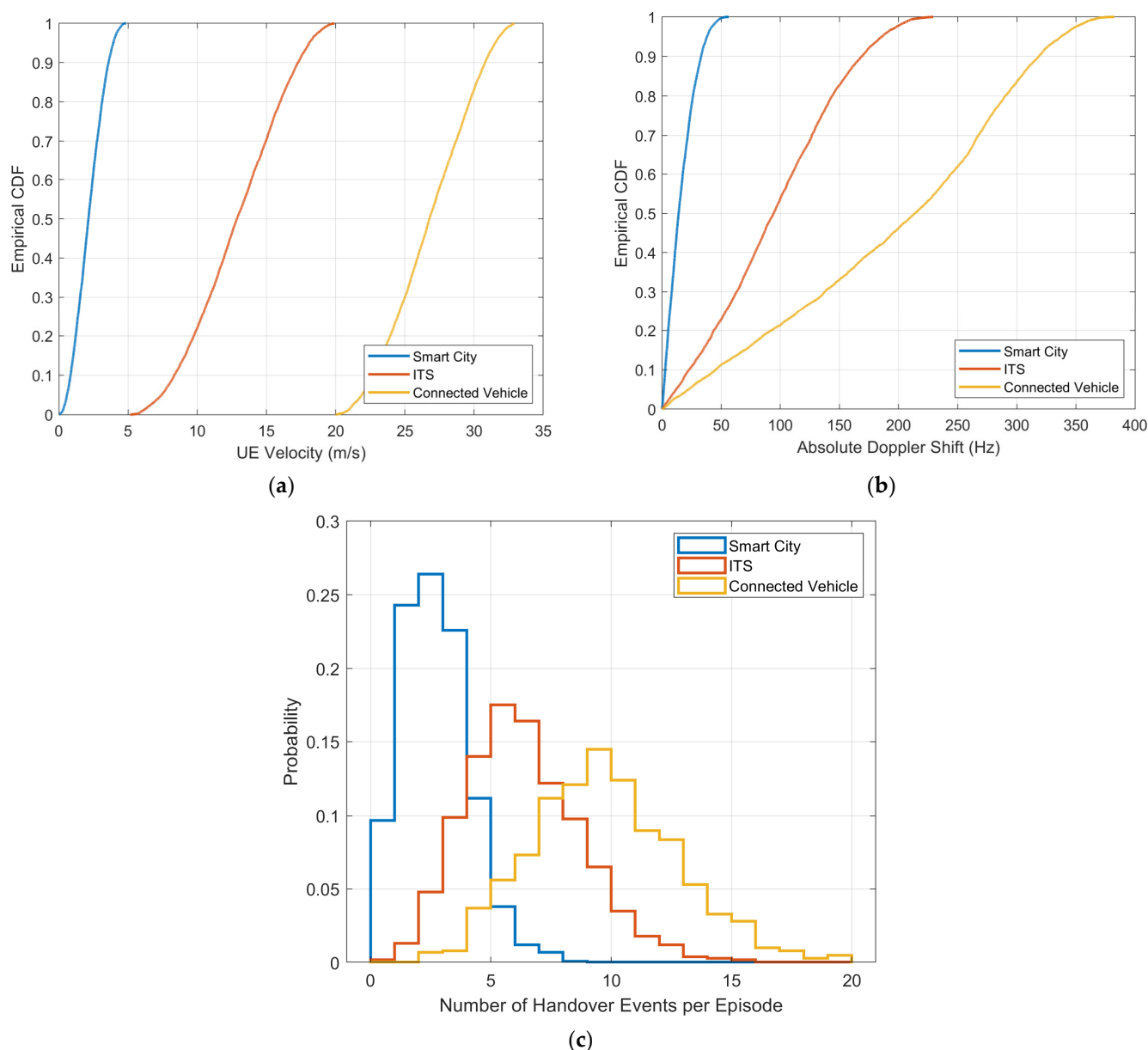


Figure 12. Empirical distributions characterizing mobility diversity across the training scenarios: UE velocity (a), absolute Doppler shift (b), and number of handover events per episode (c) for the smart city, ITS, and connected vehicle datasets.

Table 18 further isolates the effect of training-set mobility diversity. Augmenting the smart city training set with synthetic high-velocity trajectories reduces the prediction RMSE from 0.228 ± 0.014 to 0.164 ± 0.011 and increases the average utility from 0.571 ± 0.028 to 0.618 ± 0.026 , while also improving throughput, HFR, and handover latency. These results confirm that limited high-mobility coverage contributes substantially to the original generalization gap. However, the augmented model still underperforms the matched connected vehicle-trained model, indicating that mobility augmentation alone does not fully eliminate the domain shift. Thus, cross-scenario generalization is influenced by both training-distribution coverage and scenario-specific domain differences.

Table 17. Statistical characterization of mobility diversity across the smart city, ITS, and connected vehicle training datasets.

Training Scenario	Velocity Range (m/s)	Mean Velocity (m/s)	Mean Doppler Shift (Hz)	Mean HO Events/Episode
Smart city	0–5	2.19	16.13	2.21
ITS	5–20	12.86	95.59	5.86
Connected vehicle	20–33	26.79	196.89	9.28

Table 18. Controlled evaluation of high-velocity trajectory augmentation for cross-scenario generalization from smart city to connected vehicle scenarios.

Training Configuration	Training Velocity Range (m/s)	Test Scenario	Prediction RMSE	Throughput (Mbps)	HFR (%)	HO Latency (ms)	Average Utility
Original smart city	0–5	Connected vehicle	0.228 ± 0.014	54.8 ± 1.92	6.31 ± 0.45	63.7 ± 2.31	0.571 ± 0.028
Augmented Smart city	0–33	Connected vehicle	0.164 ± 0.011	59.6 ± 1.81	5.18 ± 0.40	58.2 ± 2.14	0.618 ± 0.026
Connected vehicle	20–33	Connected vehicle	0.121 ± 0.009	63.9 ± 1.76	4.29 ± 0.36	52.6 ± 1.95	0.646 ± 0.025

5.6.3. Generalization Across Mobility Scenarios

The zero-shot cross-scenario generalization performance of the proposed framework is summarized in Table 19, where each model trained under one mobility scenario is evaluated across smart city, ITS, and connected vehicle environments without target-domain retraining or fine-tuning. The results are reported in terms of prediction RMSE, throughput, HFR, PPR, and average utility, with all performance values obtained from 10 independent simulation runs.

As expected, the strongest performance for each test scenario is obtained when the training and test domains are matched: The diagonal entries of Table 19 yield the lowest prediction RMSE and the highest average utility in every column (e.g., 0.086 and 0.795 for the smart city scenario, 0.111 and 0.743 for the ITS scenario, 0.155 and 0.678 for the connected vehicle scenario). This occurs because the temporal and channel statistics observed during deployment remain consistent with those represented in the training distribution. It is worth noting that this advantage holds column-wise rather than row-wise; for instance, the model trained on ITS achieves a higher average utility when tested on smart city (0.772) than on its own training scenario (0.743), since smart city corresponds to the lowest-mobility and therefore least demanding, test condition regardless of the source domain. Nevertheless, the proposed framework retains comparatively stable performance when transferred between mobility scenarios, indicating that the learned representation captures temporal network characteristics that are not restricted to a single velocity regime.

The most challenging distribution shift occurs when a model trained under smart city conditions is directly evaluated in the connected vehicle scenario, where the prediction RMSE rises from 0.086 to 0.228, an increase of approximately 165%, and average utility falls from 0.795 to 0.571, a decrease of approximately 28%. In this case, the test environment introduces substantially higher UE velocities and faster Doppler-induced channel variation than those encountered during training. By comparison, the model trained on the ITS condition and evaluated on the connected vehicle condition shows a smaller relative degradation, with the RMSE increasing by approximately 59% (0.111 to 0.176) and utility decreasing by approximately 15% (0.743 to 0.631). This degradation is theoretically consistent with the system model of Sections 3.2 and 3.3, since as velocity increases, the

UE-cell geometry and the instantaneous channel conditions evolve more rapidly, reducing the temporal validity of patterns learned exclusively from low-mobility trajectories.

Conversely, models trained using ITS or connected vehicle trajectories generalize more effectively toward lower-mobility scenarios, with only a marginal RMSE increase and utility loss when tested on the smart city scenario (ITS-to-smart city trajectory: RMSE of 0.094 versus 0.086 for the smart-city-matched configuration, utility of 0.772 versus 0.795; connected vehicle-to-smart city trajectory: RMSE of 0.101, utility of 0.759). Exposure to faster and more diverse temporal variations during training provides the Bi-LSTM encoder, whose state representation is defined in Section 4, with a broader range of mobility transitions, allowing the resulting representation to remain informative when the rate of network-state evolution decreases. This asymmetry suggests that training-domain diversity is an important determinant of generalization performance and provides practical motivation for incorporating heterogeneous mobility regimes into the historical trajectory dataset used during initial Bi-LSTM pretraining. Most importantly, although performance decreases under cross-scenario transfer, the degradation remains moderate across all transfer pairs, and the framework preserves a stable ranking among the mobility scenarios rather than failing catastrophically under domain shift. This suggests that the performance advantage of the proposed framework is unlikely to be restricted to the particular mobility distribution from which its training samples were generated.

Table 19. Cross-scenario generalization performance of the proposed framework across smart city, ITS, and connected vehicle mobility environments.

Training Scenario	Test Scenario	Prediction RMSE	Throughput (Mbps)	HFR (%)	PPR (%)	Average Utility
Smart city	Smart city	0.086 ± 0.006	77.3 ± 1.15	1.20 ± 0.14	1.90 ± 0.19	0.795 ± 0.015
Smart city	ITS	0.148 ± 0.010	69.2 ± 1.45	3.10 ± 0.27	4.10 ± 0.34	0.704 ± 0.019
Smart city	Connected vehicle	0.228 ± 0.015	58.4 ± 1.90	6.20 ± 0.46	7.80 ± 0.57	0.571 ± 0.026
ITS	Smart city	0.094 ± 0.007	74.5 ± 1.25	1.60 ± 0.17	2.40 ± 0.22	0.772 ± 0.016
ITS	ITS	0.111 ± 0.010	73.1 ± 1.30	2.20 ± 0.20	3.00 ± 0.27	0.743 ± 0.017
ITS	Connected vehicle	0.176 ± 0.012	63.4 ± 1.65	4.50 ± 0.36	5.70 ± 0.44	0.631 ± 0.023
Connected vehicle	Smart city	0.101 ± 0.007	73.1 ± 1.30	1.80 ± 0.18	2.60 ± 0.24	0.759 ± 0.017
Connected vehicle	ITS	0.129 ± 0.009	70.8 ± 1.40	2.80 ± 0.25	3.80 ± 0.32	0.716 ± 0.019
Connected vehicle	Connected vehicle	0.155 ± 0.010	67.0 ± 1.60	3.47 ± 0.30	4.37 ± 0.35	0.678 ± 0.022

5.6.4. Generalization Across Deployment Environments

As with the cross-scenario results of Section 5.6.4, the strongest performance is obtained when the training and deployment environments are matched, as shown along the diagonal of Table 20 (urban: RMSE 0.096, utility: 0.681; suburban: RMSE: 0.091, utility: 0.689; mixed: RMSE: 0.089, utility: 0.695). A second experiment evaluates whether the proposed framework remains effective when the radio-access environment changes independently of UE mobility. This experiment is particularly relevant to the heterogeneous network architecture of Section 3.1, since a UE traveling through a realistic smart city or

ITS environment may move continuously between dense urban small-cell deployments and lower-density suburban coverage. When a model trained in the urban regime is evaluated directly in the suburban regime, a moderate performance reduction is observed, with the RMSE increasing by approximately 23% (0.096 to 0.118), the HFR increasing by approximately 22% (3.42% to 4.18%), and the average utility decreasing by approximately 6% (0.681 to 0.642), because the distribution of serving-cell residence time, neighbor-cell availability, path loss, and handover opportunities differs from that encountered during training. The reverse transfer, from the suburban to urban regimes, produces a larger degradation, with the RMSE increasing by approximately 39% (0.091 to 0.126), the HFR increasing by approximately 37% (3.31% to 4.53%), and the utility decreasing by approximately 9% (0.689 to 0.624), since the higher frequency of candidate-cell transitions and the more densely overlapping coverage regions of the urban environment increase the difficulty of target-cell selection for a model trained under comparatively sparser suburban conditions. The mixed regime provides the most representative generalization test, because network conditions evolve within the same UE trajectory rather than remaining statistically homogeneous throughout an episode; models trained on the mixed regime show the smallest cross-environment degradation among all training sources, with the utility decreasing by only 3–5% and the RMSE increasing by 14–18% when evaluated on urban or suburban conditions. Across all nine training-test combinations reported in Table 20, the average utility does not fall below 0.624 and the HFR does not exceed 4.53%, indicating that generalization loss remains bounded rather than producing a severe performance collapse under any deployment-domain transfer. This behavior suggests that the temporal embedding does not merely encode scenario-specific signal levels; instead, it captures temporal relationships among mobility, propagation, and network-load measurements that remain useful when the absolute operating conditions change. This result is consistent with the knowledge-representation analysis of Section 5.4. Because the Bi-LSTM encoder defined in Section 4 compresses sequences of heterogeneous measurements rather than relying on instantaneous RSRP or SINR alone, changes in individual signal values do not necessarily invalidate the resulting state representation, allowing the Dueling DDQN policy to remain robust to deployment-domain shifts.

Table 20. Cross-deployment generalization performance of the proposed framework across urban, suburban, and mixed network environments without target-domain retraining.

Training Environment	Test Environment	Prediction RMSE	Throughput (Mbps)	Handover Latency (ms)	HFR (%)	PPR (%)	Average Utility
Urban	Urban	0.096	67.2	47.8	3.42	4.28	0.681
Urban	Suburban	0.118	63.5	52.4	4.18	5.06	0.642
Urban	Mixed	0.111	64.8	50.9	3.91	4.82	0.655
Suburban	Urban	0.126	62.1	54.6	4.53	5.41	0.624
Suburban	Suburban	0.091	68.0	46.9	3.31	4.12	0.689
Suburban	Mixed	0.113	64.3	51.7	4.02	4.95	0.649
Mixed	Urban	0.105	65.4	49.8	3.76	4.67	0.662
Mixed	Suburban	0.101	66.1	48.9	3.61	4.51	0.670
Mixed	Mixed	0.089	68.5	46.2	3.25	4.05	0.695

5.6.5. Effect of Closed-Loop Adaptation Under Distribution Shift

Zero-shot generalization alone does not fully characterize the intended AI-native operation of the proposed framework. As illustrated in Figure 2, the framework continuously stores newly observed transitions and periodically refreshes its prediction and decision modules. Therefore, an additional experiment evaluates performance after the model enters an unseen target environment and begins accumulating target-domain

experience. As shown in Figure 13, the average utility drops sharply immediately following the smart city-to-connected vehicle domain transition, from a matched-domain smart city level of approximately 0.784 to approximately 0.588 for the proposed framework and approximately 0.571 for AV5, since both pretrained models initially operate under an unresolved distribution shift. As new observations accumulate, periodic Bi-LSTM fine-tuning and Dueling DDQN refresh update the temporal representation and decision policy using transitions obtained from the new environment; consequently, the proposed framework recovers progressively, reaching an average utility of approximately 0.678 by epoch 100, closely matching the matched-domain connected vehicle level. This behavior directly distinguishes the proposed framework from AV5, in which closed-loop refresh is disabled: Over the same adaptation window, AV5 improves only marginally, from approximately 0.571 to approximately 0.578, remaining close to its initial post-shift value and roughly 0.10 (about 17%) below the proposed framework at epoch 100. The comparison therefore extends the ablation findings of Table 11: Closed-loop refresh not only improves performance under high mobility but also provides a mechanism for recovering from previously unseen operating conditions. This result provides empirical support for the AI-native closed-loop design introduced in Sections 4.1 and 4.3, since the observed recovery under distribution shift is achieved entirely through accumulated network experience, without the manual threshold re-tuning that reactive mechanisms would otherwise require.

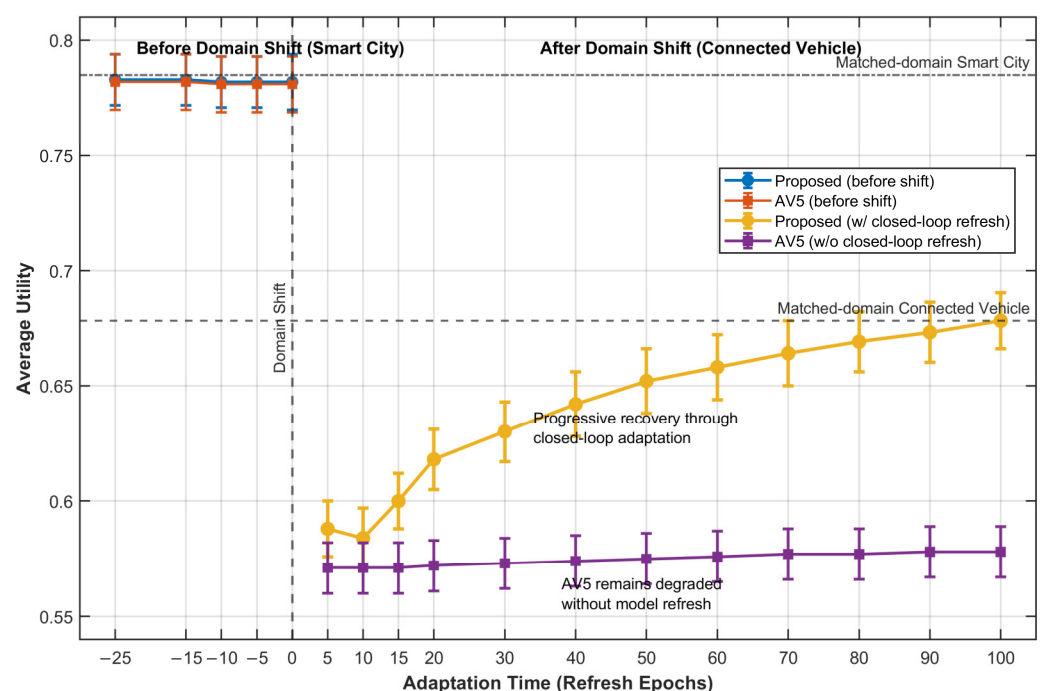


Figure 13. Average utility adaptation following an unseen smart city-to-connected vehicle domain shift. The proposed framework employs periodic closed-loop model refresh, whereas AV5 operates without closed-loop adaptation. Error bars indicate 95% confidence intervals over 10 independent simulation runs ($n = 10$).

5.6.6. Overall Generalization Analysis

Taken together, the cross-scenario and cross-deployment experiments demonstrate two distinct properties of the proposed framework. First, the learned Bi-LSTM representation and Dueling DDQN policy exhibit transferability, retaining useful predictive and decision-making capability when evaluated under mobility and deployment conditions different from those represented during training. Second, the closed-loop architecture

provides adaptability, allowing the framework to progressively recover from distribution shifts through periodic updates based on newly accumulated network experience, as demonstrated by the recovery from approximately 0.588 to approximately 0.678 in average utility over 100 refresh epochs in Section 5.6.5. The remaining performance degradation under the most severe domain shifts is nevertheless important and should not be interpreted as complete domain invariance. In particular, transferring directly from low-mobility smart city training conditions to high-speed connected vehicle operation introduces substantially different temporal and propagation statistics, resulting in a measurable generalization gap before adaptation, corresponding to the approximately 165% increase in prediction RMSE and the approximately 28% reduction in average utility reported for this transfer pair in Section 5.6.3. The proposed framework should therefore be regarded as generalizable and adaptive within the heterogeneous operating conditions considered in this study rather than universally invariant to arbitrary unseen environments.

Overall, these findings complement the mobility-robustness and load scalability results of Section 5.5. Whereas Figures 3–10 demonstrate that the proposed framework remains effective as mobility and network load increase within the evaluated operating distributions, the generalization experiments demonstrate that the learned predictive mobility management mechanism can also retain and subsequently recover performance when those distributions change. This capability is particularly important for practical smart city, ITS, and connected vehicle deployments, where UEs continuously encounter heterogeneous and non-stationary radio environments and where repeated offline retraining for every operating condition would be impractical.

6. Conclusions

This study developed an integrated predictive mobility-management architecture for high-mobility 5G/6G networks by coupling Bi-LSTM-based future-state prediction and learned temporal representation with Dueling-DDQN-based sequential handover decision-making and dual-frequency closed-loop adaptation. The contribution does not lie in introducing a new Bi-LSTM or reinforcement-learning algorithm individually; rather, it lies in the architectural integration of the current observation, predicted future network conditions, and a compact learned temporal representation within a unified predictive state, together with asynchronous adaptation of the predictive and decision components. This design enables the handover agent to anticipate future link evolution rather than relying solely on instantaneous measurements.

Extensive Monte Carlo simulation across smart city, ITS, and connected vehicle scenarios demonstrated that the proposed Bi-LSTM predictor consistently achieves the lowest prediction error among the evaluated architectures, reducing the RMSE by 51.9–68.6% relative to a persistence baseline and by 8.4–25.6% relative to a Transformer predictor, with this advantage growing under higher-mobility conditions. The knowledge-representation analysis further confirmed that the learned Bi-LSTM embedding forms are more discriminative, better-separated latent clusters than a unidirectional LSTM, particularly under ITS and connected vehicle conditions. End-to-end handover evaluation showed that the complete framework maintains the highest throughput and average utility and the lowest handover latency, HFR, and PPR across increasing UE velocity and network load, and the ablation study confirmed that each architectural component—explicit prediction, knowledge embedding, bidirectional encoding, and closed-loop refresh—contributes complementary and non-redundant benefits, with disabling closed-loop refresh producing the largest single degradation (a 33.1% increase in HFR relative to the complete framework).

The generalization experiments showed that the proposed framework retains useful predictive and decision-making capability when evaluated under previously unseen mobility scenarios and deployment environments, with performance degradation remaining bounded rather than catastrophic, and that the closed-loop architecture allows the framework to progressively recover most of the residual performance gap, average utility improving from approximately 0.588 to 0.678 within 100 refresh epochs following an unseen smart city-to-connected vehicle domain shift. Taken together, these results indicate that the proposed framework is both generalizable across heterogeneous mobility and deployment conditions and adaptive under non-stationary operation, consistent with the AI-native design principles motivating this study, rather than offering a form of universal invariance to arbitrary unseen environments. Accordingly, the term AI-native in this study is confined to the mobility-management function and does not imply AI-native operation across the complete end-to-end protocol stack.

This study has some limitations that suggest directions for future research. First, the evaluation relies on Monte Carlo simulation of the mobility, channel, and network models of Section 3 rather than measurements from a live or testbed 5G/6G deployment; validating the proposed framework using real network traces or a hardware-in-the-loop testbed would strengthen practical applicability. Second, the generalization study considered a limited set of mobility scenarios and deployment regimes; future work could extend the evaluation to additional environments, such as high-speed rail, unmanned aerial vehicles, or satellite-terrestrial integration, and to multi-agent settings in which several UEs are jointly optimized under shared radio resources. Third, the reward formulation of Equation (20) uses fixed weighting coefficients; incorporating adaptive or learned reward weighting, as well as energy-efficiency and signaling-overhead objectives, may further improve the practical utility of the framework. Finally, extending the closed-loop refresh mechanism with more sample-efficient continual-learning techniques could reduce the residual adaptation delay observed in Section 5.6.4 and further improve robustness under rapidly evolving non-stationary network conditions.

Author Contributions: Conceptualization, S.K. and N.P.; methodology, N.P. and S.K.; software, N.P.; validation, N.P. and S.K.; formal analysis, N.P. and S.K.; investigation, N.P.; resources, S.K.; data curation, N.P.; writing—original draft preparation, N.P.; writing—review and editing, S.K. and N.P.; visualization, N.P.; supervision, S.K.; project administration, S.K.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Faculty of Engineering, Srinakharinwirot University (grant number: 469/2563).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the author.

Acknowledgments: During the preparation of this manuscript, the author used MATLAB (version R2025b) for the purposes of data analysis and figure preparation. The author has reviewed and edited the output and takes full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cui, Q.; You, X.; Wei, N.; Nan, G.; Zhang, X.; Zhang, J.; Lyu, X.; Ai, M.; Tao, X.; Feng, Z.; et al. Overview of AI and communication for 6G network: Fundamentals, challenges, and future research opportunities. *Sci. China Inf. Sci.* **2025**, *68*, 171301. <https://doi.org/10.1007/s11432-024-4337-1>.

2. Ogenyi, F.C.; Ugwu, C.N.; Ugwu, O.P.-C. A comprehensive review of AI-native 6G: Integrating semantic communications, reconfigurable intelligent surfaces, and edge intelligence for next-generation connectivity. *Front. Commun. Netw.* **2025**, *6*, 1655410. <https://doi.org/10.3389/frcmn.2025.1655410>.
3. Li, Z.; Wang, J.; Zhao, S.; Wang, Q.; Wang, Y. Evolving towards artificial-intelligence-driven sixth-generation mobile networks: An end-to-end framework, key technologies, and opportunities. *Appl. Sci.* **2025**, *15*, 2920. <https://doi.org/10.3390/app15062920>.
4. Mollel, M.S.; Abubakar, A.I.; Ozturk, M.; Kaijage, S.F.; Kisangiri, M.; Hussain, S.; Imran, M.A.; Abbasi, Q.H. A survey of machine learning applications to handover management in 5G and beyond. *IEEE Access* **2021**, *9*, 45770–45802. <https://doi.org/10.1109/ACCESS.2021.3067503>.
5. Amirova, A.; Shayea, I.; Yedilkhan, D.; Aldasheva, L.; Zakirova, A. Handover decisions for ultra-dense networks in smart cities: A survey. *Technologies* **2025**, *13*, 313. <https://doi.org/10.3390/technologies13080313>.
6. Safiullin, I.A.; Ashaev, I.P.; Korobkov, A.A.; Gaysin, A.K.; Nadeev, A.F. Reinforcement learning-based handover algorithm for 5G/6G AI-RAN. *Inventions* **2026**, *11*, 8. <https://doi.org/10.3390/inventions11010008>.
7. Junejo, Y.S.; Shaikh, F.K.; Chowdhry, B.S.; Ejaz, W. Adaptive handover management in high-mobility networks for smart cities. *Computers* **2025**, *14*, 23. <https://doi.org/10.3390/computers14010023>.
8. Feng, C.; Yang, H.H.; Guo, K.; Xia, W.; Liu, C.; Quek, T.Q.S. AI-RAN: The pathway to future wireless networks. *J. Inf. Intell.* **2026**, *4*, 5–22. <https://doi.org/10.1016/j.jiixd.2025.12.013>.
9. Baz, A.; Logeshwaran, J.; Natarajan, Y.; Patel, S.K. Enhancing mobility management in 5G networks using deep residual LSTM model. *Appl. Soft Comput.* **2024**, *165*, 112103. <https://doi.org/10.1016/j.asoc.2024.112103>.
10. Sulaiman, T.H.; Al-Raweshidy, H.S. Predictive handover mechanism for seamless mobility in 5G and beyond networks. *IET Commun.* **2025**, *19*, e12878. <https://doi.org/10.1049/cmu2.12878>.
11. Dzaferagic, M.; Missi Xavier, B.; Collins, D.; D'Onofrio, V.; Martinello, M.; Ruffini, M. ML-based handover prediction over a real O-RAN deployment using RAN intelligent controller. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 635–647. <https://doi.org/10.1109/TNSM.2024.3468910>.
12. Park, H.-S.; Kim, H.; Lee, C.; Lee, H. Mobility management paradigm shift: From reactive to proactive handover using AI/ML. *IEEE Netw.* **2024**, *38*, 18–25. <https://doi.org/10.1109/MNET.2024.3357108>.
13. Prado, A.; Stöckeler, F.; Mehmeti, F.; Krämer, P.; Kellerer, W. Enabling proportionally-fair mobility management with reinforcement learning in 5G networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1845–1858. <https://doi.org/10.1109/JSAC.2023.3273705>.
14. Park, H.; Yoon, S.H. Deep reinforcement learning for base station switching scheme with federated LSTM-based traffic predictions. *ETRI J.* **2024**, *46*, 379–391. <https://doi.org/10.4218/etrij.2023-0065>.
15. Shahid, S.M.; Na, J.-H.; Kwon, S. Incorporating mobility prediction in handover procedure for frequent-handover mitigation in small-cell networks. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 186–197. <https://doi.org/10.1109/TNSE.2024.3487415>.
16. Alraih, S.; Nordin, R.; Abu-Samah, A.; Shayea, I.; Abdullah, N.F. A survey on handover optimization in beyond 5G mobile networks: Challenges and solutions. *IEEE Access* **2023**, *11*, 59317–59345. <https://doi.org/10.1109/ACCESS.2023.3284905>.
17. Lyu, Z.; Gao, Y.; Chen, J.; Du, H.; Xu, J.; Huang, K.; Kim, D.I. Empowering intelligent low-altitude economy with large AI model deployment. *IEEE Wirel. Commun.* **2026**, *33*, 64–72. <https://doi.org/10.1109/MWC.2025.3631765>.
18. International Telecommunication Union. *Framework and Overall Objectives of the Future Development of IMT for 2030 and Beyond*; Recommendation ITU-R M.2160-0; ITU: Geneva, Switzerland, 2023.
19. Karmakar, R.; Kaddoum, G.; Chattopadhyay, S. Mobility management in 5G and beyond: A novel smart handover with adaptive time-to-trigger and hysteresis margin. *IEEE Trans. Mob. Comput.* **2023**, *22*, 5995–6010. <https://doi.org/10.1109/TMC.2022.3188212>.
20. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. <https://doi.org/10.1109/COMST.2017.2745201>.
21. Chen, X.; Guo, Z.; Wang, X.; Feng, C.; Yang, H.H.; Han, S.; Wang, X.; Quek, T.Q.S. Toward 6G native-AI network: Foundation model-based cloud-edge-end collaboration framework. *IEEE Commun. Mag.* **2025**, *63*, 23–30. <https://doi.org/10.1109/MCOM.001.2400582>.
22. Kholmatov, S.; Cho, S.; Chong, S.; Lee, K. Toward AI-on-RAN: Enabling on-RAN edge intelligence for AI-native 6G networks. *IEEE Netw.* **2026**, *in press*. <https://doi.org/10.1109/MNET.2026.3679373>.
23. Chabira, C.; Shayea, I.; Nurzhaubayeva, G.; Aldasheva, L.; Yedilkhan, D.; Amanzholova, S. AI-driven handover management and load balancing optimization in ultra-dense 5G/6G cellular networks. *Technologies* **2025**, *13*, 276. <https://doi.org/10.3390/technologies13070276>.

24. Yalın, B.; Mumcu, T.V. A survey on mobility management in beyond 5G ultra-dense networks. *Ann. Telecommun.* **2026**, *81*, 451–471. <https://doi.org/10.1007/s12243-025-01137-6>.
25. Lee, H.; Kang, Y.; Gwak, M.; An, D. Bi-LSTM model with time distribution for bandwidth prediction in mobile networks. *ETRI J.* **2024**, *46*, 205–217. <https://doi.org/10.4218/etrij.2022-0459>.
26. Mehregan, N.; De Grande, R.E. GCN-based throughput-oriented handover management in dense 5G vehicular networks. In Proceedings of the 2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Lucca, Italy, 9–11 June 2025; pp. 895–902. <https://doi.org/10.1109/DCOSS-IoT65416.2025.00136>.
27. He, X.; Ying, Z.; You, C.; Gao, Y. PreHO: Predictive handover for LEO satellite networks. *arXiv* **2026**, arXiv:2603.07987. <https://doi.org/10.48550/arXiv.2603.07987>.
28. Wang, H.; Li, B. Double-deep Q-learning-based handover management in mmWave heterogeneous networks with dual connectivity. *Trans. Emerg. Telecommun. Technol.* **2024**, *35*, e4907. <https://doi.org/10.1002/ett.4907>.
29. Amaira, A.; Koubaa, H.; Zarai, F. Handover management in a sliced 5G network using deep reinforcement learning. In Proceedings of the 2024 International Wireless Communications and Mobile Computing Conference (IWCMC), Ayia Napa, Cyprus, 27–31 May 2024; pp. 1394–1399. <https://doi.org/10.1109/IWCMC61514.2024.10592506>.
30. Amaira, A.; Zarai, F. Deep reinforcement learning-based handover management for an advanced sliced vehicular network. *Procedia Comput. Sci.* **2025**, *270*, 4233–4242. <https://doi.org/10.1016/j.procs.2025.09.548>.
31. Amponis, G.; Lagkas, T.; Argyriou, V.; Moscholios, I.; Zevgara, M.; Ouzounidis, S.; Sarigiannidis, P. Anchored self-similar 3D Gauss-Markov mobility model for ad hoc routing scenarios. *IET Netw.* **2023**, *12*, 250–259. <https://doi.org/10.1049/ntw2.12089>.
32. Pang, L.; Zhang, J.; Zhang, Y.; Huang, X.; Chen, Y.; Li, J. Investigation and comparison of 5G channel models: From QuaDRiGa, NYUSIM, and MG5G perspectives. *Chin. J. Electron.* **2022**, *31*, 1–17. <https://doi.org/10.1049/cje.2021.00.103>.
33. Yan, Z.; Zhang, W.; Li, F. Doppler spread analysis for suppressing channel time variation in high-mobility massive MIMO V2V communications. *IET Commun.* **2024**, *18*, 655–670. <https://doi.org/10.1049/cmu2.12696>.
34. Wu, D.-F.; Huang, C.; Yin, Y.; Huang, S.; Guo, Q.; Zhang, L. State aware-based prioritized experience replay for handover decision in 5G ultradense networks. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5006770. <https://doi.org/10.1155/2022/5006770>.
35. Xie, J.; Zhu, B.; Li, C. Research of 5G HUDN network selection algorithm based on Dueling-DDQN. *EURASIP J. Wirel. Commun. Netw.* **2023**, *2023*, 113. <https://doi.org/10.1186/s13638-023-02323-7>.
36. Pang, H.; Wang, Z. Dueling double deep Q network strategy in MEC for smart Internet of Vehicles edge computing networks. *J. Grid Comput.* **2024**, *22*, 37. <https://doi.org/10.1007/s10723-024-09752-8>.
37. Li, H.; Qian, X.; Song, W. Prioritized experience replay based on dynamics priority. *Sci. Rep.* **2024**, *14*, 6014. <https://doi.org/10.1038/s41598-024-56673-3>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.