Data-Driven Cellular Mobility Management via Bayesian Optimization and Reinforcement Learning

Mohamed Benzaghta, Sahar Ammar, David López-Pérez, Basem Shihada, and Giovanni Geraci

***Abstract*—Mobility management in cellular networks faces increasing complexity due to network densification and heterogeneous user mobility characteristics. Traditional handover (HO) mechanisms, which rely on predefined parameters such as A3-offset and time-to-trigger (TTT), often fail to optimize mobility performance across varying speeds and deployment conditions. Fixed A3-offset and TTT configurations either delay HOs, increasing radio link failures (RLFs), or accelerate them, leading to excessive ping-pong effects. To address these challenges, we propose two distinct data-driven mobility management approaches leveraging high-dimensional Bayesian optimization (HD-BO) and deep reinforcement learning (DRL). While HD-BO optimizes predefined HO parameters such as A3-offset and TTT, DRL provides a parameter-free alternative by allowing an agent to select serving cells based on real-time network conditions. We systematically compare these two approaches in real-world site-specific deployment scenarios (employing Sionna ray tracing for site-specific channel propagation modeling), highlighting their complementary strengths. Results show that both HD-BO and DRL outperform 3GPP set-1 (TTT of 480 ms and A3-offset of 3 dB) and set-5 (TTT of 40 ms and A3-offset of -1 dB) benchmarks. We augment HD-BO with transfer learning so it can generalize across a range of user speeds. Applying the same transfer-learning strategy to the DRL method reduces its training time by a factor of 2.5 while preserving optimal HO performance, showing that it adapts efficiently to the mobility of aerial users such as UAVs. Simulations further reveal that HD-BO remains more sample-efficient than DRL, making it more suitable for scenarios with limited training data.**

***Index Terms*—Mobility management, cellular networks, Bayesian optimization, reinforcement learning, data-driven optimization.**

M. Benzaghta is with Universitat Pompeu Fabra, Spain.

S. Ammar and B. Shihada are with King Abdullah University of Science and Technology, Saudi Arabia.

D. López-Pérez is with Universitat Politècnica de València, Spain.

G. Geraci is with Nokia Standards and Universitat Pompeu Fabra, Spain. He was with Telefónica Research, Spain, when the work was carried out.

This work was supported by *a)* HORIZON-SESAR-2023-DES-ER-02 project ANTENNAE (101167288), *b)* the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union NextGenerationEU through actions CNS2023-145384 and CNS2023-144333, *c)* the Spanish State Research Agency through grants PID2021-123999OB-I00, PID2024-156488OB-I00, and CEX2021-001195-M, *d)* the Generalitat Valenciana, Spain, through the CIDEGENT PlaGenT, Grant CIDEXG/2022/17, Project iTENTE, and *e)* the UPF-Fractus Chair.

Some of the results in this paper have been presented at IEEE PIMRC 2025 [1].

I. Introduction

A. Background and Motivation

To accommodate growing mobile data traffic and new use cases, network operators are deploying additional infrastructure to enhance coverage, improve spectrum efficiency, and increase spatial reuse [2], [3]. Managing user mobility in cellular networks remains a critical challenge, particularly with the increasing densification of cells in next-generation networks [4], [5].

Handover (HO) mechanisms allow a moving user equipment (UE) to seamlessly transition from a serving cell to a target cell while maintaining quality of service (QoS). However, as cellular networks become more dense, frequent HOs can significantly increase the complexity of mobility management [6]. The effectiveness of mobility management is heavily influenced by the configuration of the hysteresis margin (A3-offset) and time-to-trigger (TTT)—two parameters playing a crucial role in minimizing ping-pongs and HO failures (HOFs). In conventional cellular networks, UEs typically operate with a finite set of such parameters, which may be adjusted on a per-cell basis. However, due to the interplay between cells, optimizing HO settings individually does not guarantee optimal mobility performance at the network level. Achieving an efficient configuration requires a joint optimization approach, which becomes increasingly complex in large-scale deployments.

Moreover, mobility management must account for UE speed variations, as a parameter set optimized for one speed may not be suitable for another. For instance, high-mobility UEs may travel deep inside a target cell before the TTT expires, increasing the likelihood of HOF due to degraded signal-to-interference-plus-noise ratio (SINR). Conversely, these UEs may also experience unnecessary HOs (ping-pongs) when passing through small cells too quickly [7]. These challenges are intensified for aerial UEs, such as uncrewed aerial vehicles (UAVs) or drones, as they experience rapid fluctuations in received signal strength and strong interference from neighboring cells, further worsening connectivity issues [8], [9]. These factors collectively contribute to frequent and unnecessary HOs (ping-pongs), increasing both signaling overhead and the likelihood of radio link failures (RLFs).

These challenges highlight the need for UE-specific and site-specific solutions to mobility management, beyond a traditional

3GPP approach that relies on a specific set of HO parameters across all network cells. To address this, we leverage recent advancements in data-driven models and study two alternative strategies for mobility management optimization in real-world deployments: (i) high-dimensional Bayesian optimization (HD-BO), which operates by optimizing HO control parameters, and (ii) deep reinforcement learning (DRL), which follows a parameter-free paradigm by directly making HO decisions based on observed network states. Rather than merging these two methods into a single approach, our work positions them as complementary yet independent solutions to the same problem, enabling a systematic performance comparison.

B. Related Work

Mobility management in cellular networks has attracted sustained interest from industry, academia, and standards bodies. Below we organize prior work into two major classes that considers ground UEs (GUEs) and UAVs: (i) *parameter-based mobility optimization* (including analytical models), and (ii) *reinforcement-learning–based mobility management*.

Parameter-based (and analytical) mobility optimization: A large body of work tunes HO control parameters—typically A3-offset and time-to-trigger (TTT)—sometimes aided by data-driven or federated mechanisms. For example, [4] exploits UE mobility direction and RSRP patterns during TTT to reduce frequent HOs, while [10] proposes an online learning mechanism using posterior RSRP probabilities to identify the optimal target cell. Federated learning has also been used to dynamically adjust HO thresholds based on predicted RSRPs and historical HO outcomes [11]. Analytical frameworks provide complementary insight into HO behavior. Stochastic-geometry–based analyses and closed-form triggering models quantify association/HO rates and triggering conditions under specific assumptions [12], [13]. While valuable for intuition and design guidelines, these models typically abstract away site-specific 3D geometry, material-dependent propagation, heterogeneous speed mixes, and controller-timer details (e.g., L1/L3 filtering, T310/Qin/Qout), which can limit deployability at city scale and in dense, interference-limited scenarios.

Reinforcement-learning–based mobility management: A second line of work formulates HO as a sequential decision problem and applies RL to select the serving cell (or beam) without relying on fixed thresholds. Prior studies span traditional and deep RL methods, across both GUE and UAV settings, with reward functions typically involving throughput/RSRP maximization and HO rate minimization. Examples include Q-learning for high-mobility vehicular beam association [14], multi-agent Q-learning for dense mmWave networks with load balancing [15], Deep Q-Network (DQN)-based next-BS selection with proportional fairness [16], and Proximal Policy Optimization (PPO) within Open-RAN architectures optimizing weighted RSRQ/throughput/spectral-efficiency rewards [17]. UAV-oriented works have likewise applied Q-learning, DQN and their variants to reduce HO rate while preserving signal quality and availability [18]–[21]. Across the RL literature, feasibility hinges on what the agent can realistically observe and act upon at run-time. State designs often include some combination of RSRP/RSRQ/SINR traces, serving and candidate BS IDs, position and direction of the user, and sometimes load/interference indicators. Some works assume instantaneous knowledge of interference or global network states, which may be difficult to satisfy in operational RANs without added signaling overhead or latency.

Despite notable progress, several recurring limitations remain: (i) parameter-based methods can require expert tuning and may not scale under strong inter-cell coupling; (ii) analytical models offer insight but may not capture site-specific propagation and mixed-mobility conditions at scale; and (iii) many RL studies adopt assumptions about observability or training/sample budgets that are challenging in practice, or they target a single mobility profile, hindering generalization across speeds and routes. These considerations motivate evaluations that emphasize realistic, site-specific deployments, standardized HO KPIs (ping-pongs, RLF/HOF), and explicit discussion of sample efficiency. In this context, our PPO-based DRL study differs from prior works by being conducted in a site-specific ray-tracing–based urban deployment, with standard-oriented reward design and novel state space including a history of previous serving BSs and the time of stay at current BS. Additionally, for the first time to our knowledge, we benchmark DRL directly against high-dimensional Bayesian optimization, thereby quantifying the relative merits of parameter-free and parameter-based mobility optimization approaches.

C. Approach and Contribution

We provide two methodologies based on HD-BO and DRL for scalable cellular mobility management, optimizing practical HO metrics for diverse speeds across designated streets.

Although BO [22] has proven effective in addressing coverage-capacity tradeoffs and optimizing radio resource allocation [23]–[30], it is inherently limited by the number of decision variables it can efficiently handle—typically around twenty or fewer in continuous domains [31]. This constraint restricts the scalability of BO for optimizing mobility parameters in large-scale cellular networks. To overcome these limitations, this paper takes the first step in employing high-dimensional Bayesian optimization (HD-BO) to optimize mobility-related HO KPIs. To the best of our knowledge, this paper is the first to (i) apply HD-BO tools to address practical mobility management challenges in large-scale cellular networks using real-world scenarios, and (ii) explore model generalization to diverse UE speeds within the context of transfer learning through HD-BO.

In addition, we introduce a non-parameter-based model-free mobility management approach leveraging DRL. Unlike the HD-BO method, which requires predefined parameters for HO decisions, the DRL-based solution enables an agent to dynamically select the next serving cell for a certain UE based on network state information, eliminating the need for static thresholds.

Our main contributions can be summarized as follows:

- *High-dimensional BO for HO parameter-based mobility management:* We apply a state-of-the-art HD-BO technique to mobility management, demonstrating its effectiveness in optimizing HO decisions for both GUEs and UAVs. In dense urban deployments, mobility management involves the joint tuning of multiple handover control parameters (e.g., A3-offset and TTT) across many cells. This creates a high-dimensional search space, where the performance of one cell is strongly coupled with the configurations of its neighbors. Traditional analytical models and manual tuning approaches cannot efficiently capture these interactions, and they quickly become intractable at scale. HD-BO provides a sample-efficient way to explore such large parameter spaces by building surrogate models that guide the search towards promising configurations. Specifically, we identify optimal A3-offset and TTT configurations for real-world cellular network deployments, that balance conflicting HO KPIs, such as ping-pongs vs. HOF and ping-pongs vs. RLF. By extending BO to high-dimensional settings, we enable large-scale, data-driven mobility optimization beyond traditional BO constraints. Our case studies consider both GUEs and UAVs moving at varying speeds. Our extensive evaluations on a real-world cellular deployment scenario in London show that, for per-cell optimization, HD-BO reduces ping-pongs by 73% for GUEs moving at 60 km/h compared to 3GPP benchmarks. Also, for UAVs at a 150 m altitude, HD-BO outperforms 3GPP set-1 and set-5 benchmarks, achieving a 3% ping-pong rate (vs. 15% and 11%) and 0% RLF similar to the upper-bound set-5, and performing better than set-1 of 9% RLF.
- *DRL for parameter-free mobility management:* We compare the performance of the HD-BO approach to a non-threshold-based mobility management method utilizing DRL. Unlike the HD-BO method, which optimizes predefined HO parameters such as A3-offset and TTT, the DRL-based solution eliminates the reliance on fixed HO thresholds. Instead, an agent autonomously learns and selects the optimal serving cell in real-time based on the network state by directly interacting with the environment. Our results show that the performance achieved through DRL is comparable to HD-BO in both ping-pong reduction and RLF minimization, confirming that DRL is a viable alternative to parameter-based mobility management, without predefined A3-offset and TTT thresholds.
- *Transfer learning:* Aiming at faster convergence to optimal solutions, and aligning with the 3GPP vision on the need for data-driven model generalization [32], we explore the *transfer learning* capabilities of the HD-BO and DRL approaches. We use transfer learning to leverage measurement outcomes from a previously performed optimization process, denoted as the *scenario source*, to predict the best solution for a new optimization, termed the *scenario target*. Our experiments reveal that the HD-BO approach is capable of generalization to diverse UE speeds; furthermore, transfer learning applied through DRL reduces training time by 2.5× while maintaining optimal HO performance.

II. System Model

In this section, we describe the network deployment, channel model, and mobility management performance metrics used in our study.

A. Cellular Topology and Site-specific Propagation Channel

We consider a site-specific scenario based on a real-world production radio network operated by a leading commercial mobile provider in the UK.

Cellular network deployment: The deployment under study consists of 10 cell sites, with antenna heights ranging from 22 m to 56 m. Each site comprises three sector antennas, resulting in a total of 30 cells across the network. The selected geographical area spans 1400×1275 m and is located in London, between latitudes $[51.5087, 51.5215]$ and longitudes $[-0.1483, -0.1296]$. A 3D representation of the selected area is constructed using OpenStreetMap and Blender, incorporating both terrain and building information. In our first case study, focusing on ground UE mobility management, the BSs antennas are configured according to the actual cellular network. In our second case study, focusing on UAV mobility management, BS antennas are optimized to achieve a trade-off between ground and aerial coverage. More details are provided in Section III-D.

Propagation channel: The channel between BS b and UE k is computed using Sionna RT [33], a widely used 3D ray-tracing tool for site-specific radio wave propagation analysis. Simulations are performed at a carrier frequency of 2 GHz. The material `itu_concrete` is used to model the permittivity and conductivity of all buildings. The maximum number of reflections and diffractions is set to 5 and 1, respectively.

SINR formulation: We compute the downlink wideband SINR in dB experienced by UE k from its serving BS b_k as

$$\texttt{SINR}_{\text{dB},k} = 10 \log_{10} \left(\frac{p_{b_k} \cdot G_{b_k,k}}{\sum\limits_{b \in \mathcal{B} \setminus b_k} p_b \cdot G_{b,k} + \sigma_{\text{T}}^2} \right), \qquad (1)$$

where $G_{b,k}$ is the square magnitude of the channel gain, incorporating both small-scale and large-scale fading, averaged over 50 physical resource blocks (PRBs), each with a bandwidth of 180 kHz. The thermal noise power σ_{T}^2 over 10 MHz is obtained

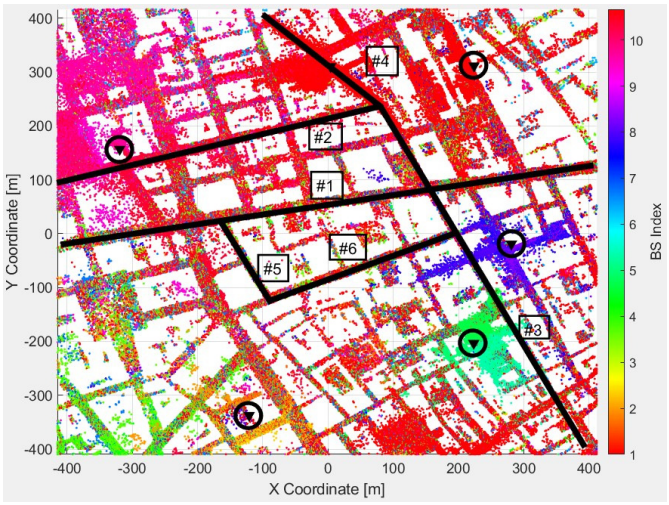

Fig. 1: 2D representation of the selected urban area, illustrating UE positions along streets (colored dots) and cell deployment sites (triangles). The five chosen streets are marked with black lines, and colors denote the index of the cell site providing the strongest average received power.

from a power spectral density of -174 dBm/Hz, while the transmit power of BS b across the entire bandwidth is $p_b = 46$ dBm [34].

Cellular mobility management problem: For our mobility study, we choose five main streets within the selected geographical area of London for experimentation. Fig. 1 provides a 2D representation of the selected urban area, where colored dots indicate outdoor locations of UEs along the streets. The five selected streets are marked with black lines, with their corresponding IDs labeled beside them. Some of the BS deployment sites are represented as circled triangles, where each site consists of three sector antennas. The color of each dot indicates the cell site (one out of ten) providing the strongest average received power, as shown by the heatmap bar on the side. This visualization highlights the challenges of mobility management in real-world scenarios, since rapid signal fluctuations may cause frequent changes in serving cells (handovers) over short distances. Consequently, an effective data-driven mobility management framework is required, one that is robust and capable of targeting site-specific scenarios.

B. Handover Mechanism in Cellular Networks

In the following, we further detail the cellular mobility management problem and introduce the concepts of HO, HOF, RLF, and ping-pongs.

Handover process: In cellular networks, handovers can occur between different radio access technologies (RATs), carriers, or cells [35]. In this study, we focus on intra-RAT, intra-carrier HOs and on hard HOs, where a UE disconnects from the source cell before establishing a new connection with the target cell.[1]

Handover measurements: The UE performs HO measurements and processing at both Layer 1 (physical layer) and Layer 3 (network layer). For HO measurements, the UE typically estimates the RSRP for the cells listed in its neighboring cell list. To mitigate the effects of small-scale fading in RSRP estimations, the UE computes each RSRP sample as the linear average of power contributions from all resource elements carrying reference symbols within a single subframe and the designated measurement bandwidth (e.g., six PRBs). These averaged RSRP samples are then further smoothed over multiple samples. This linear averaging process, performed at Layer 1, is known as *L1 filtering*. In a typical setup, downlink RSRP samples may be collected every 40 ms and then averaged over five successive samples to obtain an L1-filtered HO measurement [36]. The L1-filtered HO measurements are further averaged using a first-order infinite impulse response filter to mitigate the effects of fading and estimation imperfections. This moving averaging process is performed at Layer 3 and is referred to as *L3 filtering*. A typical L3 filtering period is 200 ms. For further details on the L1 and L3 filtering procedures, we refer the reader to Fig. 1 in [36].

Handover trigger: A HO is triggered when the L3-filtered HO measurement satisfies a HO event entry condition. While there are eight types of HO event entry conditions [37, Section 5.5.4], we focus on *Event A3: 'neighbor becomes offset better than server'*, since it is typically used to trigger intra-RAT intra-carrier HOs [36]. Once the A3 condition is met—i.e., the L3-filtered RSRP of the target cell exceeds that of the serving cell by a hysteresis margin (also known as the event *A3 offset*)—the UE initiates a *TTT* timer. The UE initiates the HO preparation process only if the event A3 condition remains satisfied throughout the TTT period. If that is the case, the UE notifies the serving cell and reports the event A3 condition via a measurement report. Then, the HO preparation phase begins.

Handover preparation and execution: The source cell issues a HO request message to the target cell, which performs admission control procedures based on the quality of service requirements of the UE. The target cell prepares for the HO process and sends a HO request acknowledgment to the source cell. Upon receiving the HO request acknowledgment, the source cell initiates data forwarding to the target cell and sends a HO command to the UE. Finally, in the HO execution phase, the UE synchronizes with the target cell and establishes access. Once the HO procedure is completed, the UE sends a HO complete message to the target cell, allowing the target cell to begin data transmission to the UE [35], [36].

Radio link failure and handover failure: RLF occurs when a UE is unable to maintain a reliable connection with the serving

[1] Handovers can be governed by both signal strength and signal quality. In this study, we follow the 3GPP assumptions in [35], where the handover procedure is based on Reference Signal Received Power (RSRP) measurements.

cell due to sustained poor signal quality. Specifically, a UE is considered out of synchronization when its wideband SINR, denoted by $\texttt{SINR}_{\text{dB},k}$, falls below a threshold Q_{out}. The UE regains synchronization when this SINR exceeds a threshold Q_{in}. Once $\texttt{SINR}_{\text{dB},k}$ drops below Q_{out}, a timer T_{310} is triggered. If the SINR does not recover above Q_{in} before T_{310} expires, the UE declares a RLF [35], [38]. HOF is then defined as a specific instance of RLF occurring during the HO process. Based on [35], [38], a HOF is declared if any of the following conditions are met: (1) RLF occurs after the HO is triggered (e.g., event A3) but before the HO command is received; (2) the T_{310} timer is already running when the handover command is sent; or (3) $\texttt{SINR}_{\text{dB},k}$ remains below Q_{out} at the time the handover complete message is sent. In these cases, the HO cannot be successfully completed due to poor radio conditions. For the sake of tractability, in our model, we adopt the definition of HOF as follow: a HO is considered to have failed if the UE's SINR is below Q_{out} at the moment the HO complete message is sent. This simplification allows us to assess HOF events directly based on instantaneous SINR measurements during the handover execution phase.

Handover ping-pongs: The occurrence of a HO ping-pong is determined by the duration for which a UE remains connected to a cell immediately after a handover, referred to as the *time-of-stay*. This duration begins when the UE sends a handover complete message to the target cell and ends when the UE sends another HO complete message to a new cell. A HO is classified as a ping-pong if the UE's time-of-stay is shorter than a predefined threshold, T_p (e.g., 1 s), and if the new target cell is the same as the original source cell prior to the previous HO, leading to increased signaling overhead and reduced network efficiency.

Performance trade-off: Small A3 offsets and TTT values can trigger premature handovers, increasing the ping-pong effect, while larger values may excessively delay handovers, leading to a higher risk of HOF or RLF. Therefore, optimizing the A3 offsets and TTT based on UE velocity and site-specific radio propagation conditions is crucial for effective mobility management. In 3GPP, mobility management relies on predefined threshold sets to regulate handover decisions. Among the benchmark configuration suggested by the 3GPP in the simulation recommendations, we specifically focus on *set-1* and *set-5* [35], as they represent two extreme cases in handover optimization:

- Set-1 is designed to reduce ping-pongs by delaying handovers, applying a uniform configuration across all cells with a TTT of 480 ms and an A3-offset of 3 dB.
- Set-5, on the other hand, aims to minimize HOF by accelerating handovers, setting TTT to 40 ms and A3-offset to -1 dB, again uniformly across all cells. While this configuration allows the UE to switch to a stronger serving cell more quickly, it increases the likelihood of ping-pongs, as rapid transitions may cause frequent unnecessary handovers.

We employ 3GPP set-1 and set-5 as performance benchmarks, as they define the two extremes in handover performance trade-offs, serving as a reference for evaluating the performance of our data-driven methods. Set-1 designed to reduce ping-pongs, and Set-5, on the other hand, aims to minimize HOF. It should be noted that 3GPP evaluations apply these settings uniformly across all cells in their performance evaluations rather than adapting them per-cell, primarily to simplify network-wide mobility management and reduce optimization complexity. However, this approach is often suboptimal, as it lacks adaptability to site-specific radio conditions and UE mobility patterns, motivating the need for adaptive, data-driven optimization techniques.

In the remainder of this paper, we demonstrate how data-driven machine learning methods can optimize mobility management by addressing two key optimization problems: i) balancing ping-pongs and HOF, ii) balancing ping-pongs and RLF. The exact problem formulations will be detailed in the following two sections, where we introduce two different optimization methodologies based on HD-BO and DRL, respectively.

III. Cellular Mobility Management via High-dimensional Bayesian Optimization

In this section, we formulate the HO parameter-based mobility management optimization problem and propose a solution based on high-dimensional Bayesian optimization (HD-BO).

A. Problem Formulation

Our objective is to identify the joint optimal sets of A3-offset and TTT parameters for all cells under consideration that minimize conflicting HO KPIs. We consider two practical trade-offs defined as follows.

KPI study #1: Ping-pongs vs. HOF. We examine the trade-off between reducing the number of ping-pongs and reducing the number of HOF. The problem is formally defined as follows:

$$\min_{\mathbf{A3},\mathbf{TTT}} \quad w_{\text{PP}} \cdot \frac{\sum_t \mathbb{1}_{\text{PP}_t}}{\sum_t \mathbb{1}_{\text{HO}_t} - \mathbb{1}_{\text{HOF}_t}} + w_{\text{HOF}} \cdot \frac{\sum_t \mathbb{1}_{\text{HOF}_t}}{\sum_t \mathbb{1}_{\text{HO}_t}}, \tag{2}$$

$$\text{s.t.} \quad \text{A3}_b \in \left(\underline{\text{A3}}, \overline{\text{A3}}\right), \ \ b = 1, \ldots, \mathcal{B} \tag{2a}$$

$$\text{TTT}_b \in \left(\underline{\text{TTT}}, \overline{\text{TTT}}\right), \ \ b = 1, \ldots, \mathcal{B} \tag{2b}$$

where $\mathbb{1}_{(\cdot)}$ is an indicator function that evaluates whether a HO, a ping-pong, or a HOF event occurs at time-step t. This function takes a value of 1 if the respective event is observed and 0 otherwise. The first term captures the percentage of ping-pongs with respect to the total number of successful handovers, excluding handover failures. The objective function (2) minimizes a weighted sum, where w_{PP} and w_{HOF} are positive real numbers that determine the relative importance of reducing ping-pongs versus HOF. The vectors **A3** and **TTT** contain the A3-offset A3_b and time-to-trigger TTT_b of all BSs $b \in \mathcal{B}$, respectively. The

smallest allowed values are $\underline{\text{A3}}$ and $\underline{\text{TTT}}$, while $\overline{\text{A3}}$, $\overline{\text{TTT}}$ are the largest allowed values. Once selected by the optimizer, the values of A3-offset and TTT remain fixed throughout the evaluation. They are not adapted dynamically during a simulation run, but instead are assessed over all UEs and trajectories under the chosen configuration. This ensures a fair comparison with baseline 3GPP settings.

KPI study #2: Ping-pongs vs. RLF. In this experiment, we focus on RLF instead of HOF, i.e., on reducing link outages while minimizing the number of ping-pongs. Similarly, the problem is formally defined as follows:

$$\min_{\mathbf{A3},\mathbf{TTT}} \quad w_{\text{PP}} \cdot \frac{\sum_t \mathbb{1}_{\text{PP}_t}}{\sum_t \mathbb{1}_{\text{HO}_t} - \mathbb{1}_{\text{HOF}_t}} + w_{\text{RLF}} \cdot \frac{\sum_t \mathbb{1}_{\text{RLF}_t}}{\sum_t \mathbb{1}_{\text{HO}_t}}, \tag{3}$$

$$\text{s.t.} \quad \text{A3}_b \in \left(\underline{\text{A3}}, \overline{\text{A3}}\right), \ \ b = 1, \ldots, \mathcal{B} \tag{3a}$$

$$\text{TTT}_b \in \left(\underline{\text{TTT}}, \overline{\text{TTT}}\right), \ \ b = 1, \ldots, \mathcal{B} \tag{3b}$$

The selection of weights for ping-pongs, HOFs, and RLFs are not unique, but rather depends on operator preferences and deployment objectives. In our experiments, we tested several combinations and reported those that yielded the most balanced performance across the considered KPIs. For instance, higher weights on RLFs naturally drive the optimizer to prioritize connection stability, whereas higher weights on ping-pongs lead to more conservative handover triggering. The results presented in this paper correspond to the combinations that achieved the best trade-off in our scenario under consideration. Importantly, the proposed framework is flexible and scalable, allowing operators to adapt the weight configuration to their own performance requirements.

The optimization problems (2) and (3) are nonconvex, intractable, black-box optimization problems. Furthermore, for our practical scenario with 30 cells, they involve 60 optimization variables, requiring efficient methods to handle the large search space. In the following, we introduce high-dimensional Bayesian optimization as a potential solution for optimizing HO parameter-based mobility management.

B. High-dimensional Bayesian Optimization

Bayesian optimization (BO) is a powerful method for optimizing black-box functions that are expensive to evaluate. Its practical value lies in the ability to reduce the number of costly evaluations by learning a surrogate model (typically a Gaussian Process or related variant) of the objective function. Each new observation, obtained by simulating or measuring the mobility performance of the network under a given configuration, updates the surrogate model, which in turn guides the selection of the next most informative configuration to test [22]. This process allows BO to focus evaluations on the most promising regions of the search space, achieving near-optimal solutions with far fewer samples than exhaustive search or random exploration.

Thus, BO operates by iteratively constructing a probabilistic *surrogate model* of the objective function $f(\cdot)$ based on prior evaluations at selected points [22]. This surrogate model, which is computationally easier to evaluate than $f(\cdot)$, is continuously updated as new points are assessed. To determine the next point to evaluate, an acquisition function $\alpha(\cdot)$ scores the surrogate model's response, guiding the search process. The acquisition function strategically balances exploration (searching for new, potentially better solutions) and exploitation (refining the current best solutions).

Objective function evaluation: We define a query point $\mathbf{x} = [\mathbf{A3}, \mathbf{TTT}]$ as a configuration for both the A3-offset and TTT for each BS $b \in \mathcal{B}$, and obtain the corresponding objective function value $f(\mathbf{x})$ as in (2) or (3). In both cases, the objective function $f(\cdot)$ being optimized is a mathematically intractable stochastic function that captures the model detailed in Section II, along with the inherent randomness of UE locations and the wireless channel. As a result, we do not directly observe $f(\mathbf{x})$; instead, we obtain a noisy realization or observation of the function, denoted by $\tilde{f}(\mathbf{x})$, through system-level simulations. Importantly, repeated evaluations at the same query point $\mathbf{x}$ may yield different outcomes due to the intrinsic stochasticity of the environment. For convenience, we define a set of N query points as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and the corresponding set of observations as $\tilde{\mathbf{f}}(\mathbf{X}) = [\tilde{f}_1, \ldots, \tilde{f}_N]^\top$, where each $\tilde{f}_i = \tilde{f}(\mathbf{x}_i)$ represents a stochastic observation of $f(\mathbf{x}_i)$ for $i = 1, \ldots, N$. In practical deployments, these observations could also be obtained through real-world network measurements.

Gaussian Process prior distribution: We use a Gaussian Process (GP) prior, $\widehat{f}(\cdot)$, to construct a surrogate model (i.e., the posterior) that approximates the objective function $f(\cdot)$ [22]. The resulting GP model enables the prediction of $\widehat{f}(\mathbf{x})$ at a query point $\mathbf{x}$ based on previously observed values, $\tilde{\mathbf{f}}(\mathbf{X}) = \tilde{\mathbf{f}}$, over which the model is trained. Formally, the GP prior on the objective function $\tilde{f}(\mathbf{x})$ assumes that for any set of input points $\mathbf{X}$, the corresponding function values $\tilde{\mathbf{f}}$ are jointly distributed as

$$p(\tilde{\mathbf{f}}) = \mathcal{N}(\tilde{\mathbf{f}} \mid \boldsymbol{\mu}(\mathbf{X}), \mathbf{K}(\mathbf{X})), \tag{4}$$

where $\boldsymbol{\mu}(\mathbf{X}) = [\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_N)]^\top$ is the $N \times 1$ mean vector, and $\mathbf{K}(\mathbf{X})$ is the $N \times N$ covariance matrix, with each entry (i, j) given by the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. For a given point $\mathbf{x}$, the mean function $\mu(\mathbf{x})$ provides prior knowledge about $f(\mathbf{x})$, while the kernel function $\mathbf{K}(\mathbf{X})$ captures the uncertainty between different input values $\mathbf{x}$.

Gaussian Process posterior distribution: Given a set of observed noisy samples $\tilde{\mathbf{f}}$ at previously sampled points $\mathbf{X}$, the posterior distribution of $\widehat{f}(\mathbf{x})$ at a new query point $\mathbf{x}$ can be expressed as [31]:

$$p(\widehat{f}(\mathbf{x}) = \widehat{f} \mid \mathbf{X}, \tilde{\mathbf{f}}) = \mathcal{N}(\widehat{f} \mid \mu(\mathbf{x} \mid \mathbf{X}, \tilde{\mathbf{f}}), \sigma^2(\mathbf{x} \mid \mathbf{X}, \tilde{\mathbf{f}})), \quad (5)$$

where the posterior mean and variance are given by:

$$\mu(\mathbf{x} \mid \mathbf{X}, \tilde{\mathbf{f}}) = \mu(\mathbf{x}) + \tilde{\mathbf{k}}(\mathbf{x})^\top (\tilde{\mathbf{K}}(\mathbf{X}))^{-1} (\tilde{\mathbf{f}} - \boldsymbol{\mu}(\mathbf{X})), \quad (6)$$

$$\sigma^2(\mathbf{x} \mid \mathbf{X}, \tilde{\mathbf{f}}) = k(\mathbf{x}, \mathbf{x}) - \tilde{\mathbf{k}}(\mathbf{x})^\top (\tilde{\mathbf{K}}(\mathbf{X}))^{-1} \tilde{\mathbf{k}}(\mathbf{x}), \quad (7)$$

where $\tilde{\mathbf{k}}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_N)]^\top$ is the $N \times 1$ covariance vector, and $\tilde{\mathbf{K}}(\mathbf{X}) = \mathbf{K}(\mathbf{X}) + \sigma^2 \mathbf{I}_N$, with σ^2 representing the observation noise (i.e., the variance of the Gaussian distribution), and $\mathbf{I}_N$ denoting the $N \times N$ identity matrix. Note that (6) and (7) define the mean and variance of the estimated function $\widehat{f}(\mathbf{x})$, where the variance quantifies the uncertainty in the prediction.

Initial dataset creation and acquisition function: The BO algorithm begins by constructing a Gaussian Process (GP) prior $\{\mu(\cdot), k(\cdot, \cdot)\}$ using an initial dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_\text{o}}, \tilde{f}_1, \ldots, \tilde{f}_{N_\text{o}}\}$, which consists of N_o initial observations. The dataset is generated through system-level simulations based on the objective function defined in (2) or (3) and the model described in Section II. For each observation point $\mathbf{x}_i \in \mathcal{D}$, the A3-offset and TTT values are randomly selected from the ranges $[-1 \text{ dB}, 3 \text{ dB}]$ and $[40 \text{ ms}, 480 \text{ ms}]$, respectively. The algorithm then leverages the observations in $\mathcal{D}$ to choose $\mathbf{x}_n$. This is performed via an acquisition function $\alpha(\cdot)$, which is designed to trade off the exploration of new points in less favorable regions of the search space with the exploitation of well-performing ones. The former prevents getting caught in local minimum, whereas the latter minimizes the risk of testing points with excessively degrading performance. We adopt Thompson sampling as the acquisition function, which has shown to perform well in terms of balancing the trade-off between exploration and exploitation [39].

Batch evaluation of candidate points: We employ a batch evaluation strategy that enables efficient query space exploration while reducing the number of required physical experiments. At each iteration, a set of $N_\text{c} = 500$ candidate points is selected based on the posterior distribution (5) and evaluated in parallel across available computational resources. This approach leverages the capability of BO to learn from a limited number of samples, making it particularly suitable for scenarios where extensive real-world experimentation is impractical. We split the candidate points into 10 batches each consisting of 50 points. The query point $\mathbf{x}_n$ is then chosen as

$$\mathbf{x}_n = \arg\max_i \ \alpha\left(\mathbf{x}_{\text{cand}_i} \mid \mathcal{D}\right). \quad (8)$$

Once $\mathbf{x}_n$ is determined, a new observation of the objective function $\tilde{f}(\mathbf{x}_n)$ is then produced, and the dataset $\mathcal{D}$, the GP prior, and the best observed objective value $\tilde{f}^*$ are all updated. In practice, the maximization in (8) is not solved as a continuous optimization problem. Instead, the acquisition function is evaluated over a set of randomly drawn candidate points, and the point with the highest score is selected. This makes the selection step computationally efficient even in high dimensions. The BO loop terminates after a fixed budget of evaluations, which reflects the maximum number of system-level simulations (or real-world measurements) that can be afforded. This criterion is commonly adopted in Bayesian optimization, since exact convergence to the global optimum cannot be guaranteed for general black-box objectives.

For BO methods to achieve greater sample efficiency, it is essential to introduce a hierarchical significance structure for the dimensions of $\mathbf{x} \in D$. In high-dimensional problems, certain features, such as $\{\mathbf{x}_{22}, \mathbf{x}_{44}\}$, may play a critical role in capturing the primary variations of the objective function f, while others, such as $\{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_{60}\}$, may have moderate significance. The remaining features may contribute negligibly. HD-BO exploits these hierarchical relationships to improve optimization efficiency. In the following, we introduce the core features of a HD-BO method known as Trust Region BO (TuRBO) [39].[2]

Trust Region BO (TuRBO): To address the challenges of high dimensionality in BO, the authors in [39] proposed Trust Region BO (TuRBO), an approach that shifts from global surrogate modeling to managing multiple independent local models, each focusing on a distinct region of the search space. TuRBO achieves global optimization by simultaneously maintaining several local models and allocating samples using an implicit multi-armed bandit strategy. This enhances the acquisition strategy by directing samples toward promising local optimization efforts. TuRBO leverages trust region (TR) methods from stochastic optimization, which are gradient-free and employ a simple surrogate model within a defined TR—typically a sphere or polytope centered around the best solution found. However, simple surrogate models may require excessively small trust regions for accurate modeling. To mitigate this, TuRBO utilizes a GP surrogate model within the TR, preserving key BO features such as noise robustness and systematic uncertainty handling. In TuRBO, the TR is defined as a hyperrectangle centered at the current optimal solution, f^*. The initial side length of the TR is set to $L \leftarrow L_\text{init}$. Each dimension's side length is then adjusted according to its respective length scale λ_i in the GP model. The side length for each dimension is given by:

$$L_i = \lambda_i L \cdot \left(\prod_{j=1}^{d} \lambda_j \right)^{-1/d}. \quad (9)$$

where d is the total number of dimensions (i.e, optimization parameters under consideration). During each local optimization

[2]We implemented and tested three HD-BO methods: Sparse Axis-Aligned Subspaces (SAASBO) [40, Section 4], BO via Variable Selection (VSBO) [41, Section 3], and Trust Region BO (TuRBO) [39, Section 2]. TuRBO demonstrated superior performance and higher suitability for the problem under consideration. For a detailed analysis of the limitations of SAASBO and VSBO in related cellular optimization problems, see [42].

run, an acquisition function selects a batch of q candidates at each iteration, ensuring they remain within the designated TR. If the TR's side length L was large enough to cover the entire search space, this method would be equivalent to standard vanilla-BO. Thus, adjusting L is crucial: the TR must be large enough to encompass promising solutions while remaining compact enough to ensure the local model's accuracy. The TR is dynamically resized based on optimization progress: it is doubled ($L \leftarrow \min\{L_{\max}, 2L\}$) after τ_{succ} consecutive successes and halved ($L \leftarrow L/2$) after τ_{fail} consecutive failures. A success is defined as an iteration where the objective function value improves compared to the previous one, whereas a failure corresponds to an iteration with no improvement. Success and failure counters are reset after each adjustment. If L falls below $L_{\min}$, the TR is discarded, and a new one is initialized at L_{init}. The TR's side length is capped at $L_{\max}$. TuRBO maintains m trust regions simultaneously, denoted as TR_l, where $l \in \{1, \ldots, m\}$, each defined as a hyperrectangle with a base side length $L_l \leq L_{\max}$. Candidate selection involves choosing a batch of q candidates from the union of all TRs. Thompson sampling is used for selecting candidates both within and across trust regions.

In this study, TuRBO is run using an open-source repository [39] with the following hyperparameters: $\tau_{\text{succ}} = 3, \tau_{\text{fail}} = 15, L_{\text{init}} = 0.8, L_{\min} = 2^{-7}, L_{\max} = 1.6$.

Motivation for High Dimensionality: In our setting, the handover control parameters (A3-offset and TTT) must be tuned across multiple cells in a dense urban deployment. We formulate this as a high-dimensional optimization problem, where each cell contributes two optimization variables, resulting in a search space with tens of dimensions (e.g., 60 variables for 30 cells). While this formulation may appear conservative compared to optimizing only the neighboring cells of each UE, it is motivated by the strong inter-cell coupling that characterizes dense urban networks. Handover dynamics at one cell boundary are often influenced by the configuration of adjacent and even non-adjacent cells, due to overlapping coverage areas, LoS interference, and the fact that ping-pongs and RLF events can cascade across multiple sectors. Optimizing all cells jointly allows us to explicitly capture these dependencies and avoid network-wide imbalances.[3]

C. Case Study #1: Ground UE Mobility Management

In our first case study, we analyze GUE mobility across three different speed categories: i) pedestrian speed of 3 km/h, ii) moderate speed of 30 km/h, iii) high speed of 60 km/h. Our study aims to understand the cross-impact of speed-specific optimizations, examining how optimizing HO parameters for one speed affects the performance of others. Additionally, we evaluate the capability of HD-BO to handle heterogeneous speed scenarios, where GUEs move along a street portfolio consisting of five main roads in London. Finally, we compare the benefits of per-cell optimization versus applying a uniform TTT and A3-offset across the entire network, highlighting the advantages of cell-specific mobility management.

Minimizing HOF vs. RLF: Table I compares the performance of GUEs for a speed of 3km/h for the two KPI trade-offs:

- Ping-pongs vs. HOF ('PP-HOF', KPI study #1), i.e., problem (2) with $w_{\text{PP}} = 1$ and $w_{\text{HOF}} = 9$.
- Ping-pongs vs. RLF ('PP-RLF', KPI study #2), i.e., problem (3) with $w_{\text{PP}} = 1$ and $w_{\text{RLF}} = 9$.

These optimizations are achieved through the per-cell joint tuning of the A3-offset and TTT parameters using HD-BO and are compared to the reference performances under 3GPP benchmark configurations set-1 and set-5. For our evaluations, we set $Q_{\text{out}} = -8$ dB, T_{310} = 1 s, and T_p = 1 s, as per [35].

Fig. 2 illustrates the cumulative distribution function (CDF) of the SINR for UEs of speed 3 km/h. The solid and dashed black lines represent the SINR performance under the 3GPP benchmark configurations, set-5 and set-1, respectively. The orange curve in Fig. 2 represents the performance after the data-driven optimization for KPI study #1, while the blue curves depict the performance for KPI study #2. Based on the results in Table I and Fig. 2, the following key observations can be drawn:

- The HD-BO approach successfully reduces HOF to 0% in KPI study #1 and RLF to 0% in KPI study #2, achieving the upper bound set by 3GPP set-5 while also reducing ping-pongs by 25%.
- Minimizing RLF leads to better outage performance than minimizing HOF. Specifically, when focusing on HOF reduction, the optimization framework may delay handovers to avoid reporting HOF at the measurement time. While this results in a 0% HOF metric, it can also cause UEs to remain connected to a weak-serving BS for too long, leading to outages (RLF) before the next handover occurs (in 4.8% of the cases, as seen in Fig. 2).
- In contrast, when optimizing for RLF, the framework accounts for UE conditions both at the time of handover and afterward. This ensures that UEs do not experience outages due to delayed handovers. As a result, optimizing for RLF not only eliminates outages (0% RLF) but also naturally leads to 0% HOF, as seen in Fig. 2.

Remark: The proposed framework is flexible and allows tuning the weights of the objective function to achieve KPI trade-offs aligned with operator preferences. For instance, one can select weights such that the resulting ping-pong (or RLF) percentages match those of a 3GPP baseline (e.g., set-1), while still improving the remaining KPIs. This flexibility is not exploited in

[3]We note that the proposed HD-BO framework can be iteratively applied to optimize handover parameters over clusters of neighboring cells, taking into account the inter-cell coupling. The size of these clusters could be chosen based on a trade-off between algorithm complexity and network-wide performance. This trade-off could be site-specific, depending on cell isolation, buildings topology, and type of user (GUE or UAV).

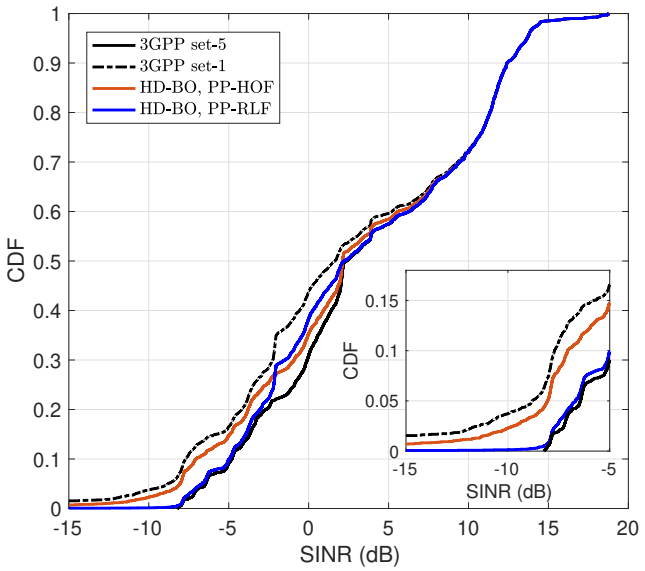

Fig. 2: SINR for GUEs at 3 km/h: the 3GPP benchmark configurations (set-1 and set-5), optimizing ping-pongs vs. HOF ('PP-HOF'), and optimizing ping-pongs vs. RLF ('PP-RLF').

Table I, which focuses on comparing PP–HOF and PP–RLF directly. Broader comparisons against 3GPP baselines under different weight configurations are presented in the following sections.

Since minimizing RLF not only reduces outages but also naturally minimizes HOF, in the following, we focus on KPI study #2, i.e., ping-pongs vs. RLF.

TABLE I: Mobility performance for GUEs at 3 km/h: 3GPP benchmark configurations (set-1 and set-5), optimizing ping-pongs vs. HOF ('PP-HOF'), and optimizing ping-pongs vs. RLF ('PP-RLF').

	3GPP set-1	3GPP set-5	PP-HOF	PP-RLF
Ping-pongs (%)	**48.8**	75.3	56.4	56.4
HOF (%)	6.5	**0.0**	**0.0**	**0.0**
RLF (%)	7.4	**0.0**	4.8	**0.0**

In addition to the extreme 3GPP configurations (sets 1 and 5), we evaluated a representative non-extreme 3GPP setting (set 4, TTT = 80 ms, A3-offset = 1 dB). While set 4 achieves zero RLFs, it results in a ping-pong rate of 73.5%, highlighting the limitations of globally applied handover parameters. In contrast, the proposed HD-BO framework preserves the same RLF performance (0.0%) while reducing ping-pongs to 56.4%, owing to its ability to optimize handover thresholds to local radio and mobility conditions.

All UEs at the same speed: Fig. 3 illustrates the mobility performance when all UEs move at the same speed, for different values of such speed (3 km/h, 30 km/h, and 60 km/h). For each value of the speed, the A3-offset and TTT is optimized for ping-pongs and RLF via HD-BO and the objective function weights are set to $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$. The comparison includes a one-threshold optimization approach (where all cells share the same configuration) and a per-cell configuration. Compared to the 3GPP set-1 benchmark (optimized for reducing ping-pongs), HD-BO with one-threshold optimization (uniform configuration for all cells) achieves a 20% reduction in ping-pongs. Per-cell optimization further improves performance, reducing ping-pongs by 35% for GUEs moving at 3 km/h. At higher speeds, such as 60 km/h, the improvement is even more significant, with ping-pong reductions of 57% and 73% for one-threshold and per-cell optimization, respectively, compared to the 3GPP benchmark. The relatively high ping-pong ratio observed for 3GPP set-1 at pedestrian speeds arises from its uniform configuration across all cells. Slow-moving UEs spend more time in cell-edge regions where RSRP levels of neighboring cells fluctuate around similar values. With a fixed, non-adaptive configuration, these fluctuations frequently cross the handover threshold and trigger ping-pongs. By contrast, per-cell optimization adapts thresholds locally to the propagation conditions, thereby reducing unnecessary handovers. Thus, adaptive thresholding—especially when done on a per-cell basis—helps mitigate unnecessary handovers. These results highlight the advantages of per-cell optimization over one-threshold optimization. It should be noted that slower UEs (3 km/h) exhibit a higher ping-pong ratio compared to faster UEs. While one might intuitively expect the opposite, this behavior arises from the interaction between user speed and the TTT parameter. At slower speeds, UEs remain longer in cell-edge regions where neighboring cells provide similar RSRP values. Even after L1/L3 filtering, small fluctuations can repeatedly cross the handover threshold, causing unnecessary handovers and thus more ping-pongs. In contrast, faster UEs traverse cell boundaries more quickly, reducing the likelihood of returning to the previous serving cell after a handover and therefore experiencing fewer ping-pongs.

Impact of speed-mismatched optimization: Fig. 4 illustrates the performance of UEs when the handover parameters are optimized for a *different* mobility speed. The results show that speed-specific configurations obtained via HD-BO may significantly degrade performance when applied to UEs moving at other speeds. For instance, a configuration optimized for pedestrian mobility (3 km/h) increases ping-pongs by nearly $8\times$ when applied to higher speeds such as 30 km/h. This effect arises because the TTT parameter plays a critical role at higher velocities, where handovers must occur promptly before the UE traverses the cell and moves into a new coverage area. Conversely, a configuration optimized for high-speed mobility (60 km/h) yields a $2\times$ increase in ping-pongs when applied to pedestrian users at 3 km/h. While one might expect that configurations optimized for slower speeds, which generally favor longer TTT values, would reduce ping-pongs for faster UEs, the results show the opposite. This behavior arises because both TTT and A3-offset are optimized jointly in our framework. Configurations tailored

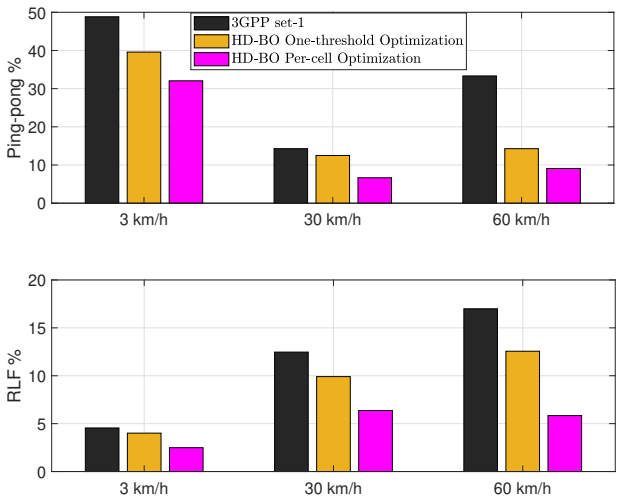

Fig. 3: GUE ping-pongs and RLF performance at a certain speed when the network is optimized for that speed via HD-BO using one-threshold optimization (uniform configuration for all cells), per-cell configuration, and with the 3GPP set-1 configuration. Lower ping-pong and RLF percentages are better, as they reflect more stable connectivity.

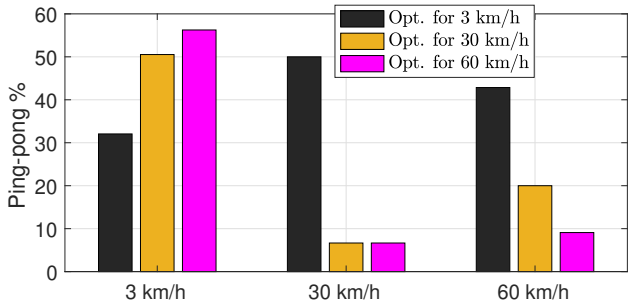

Fig. 4: GUE ping-pongs at a certain speed when the network is optimized via HD-BO for a specific (not necessarily the same) speed.

for pedestrian mobility tend to include A3-offsets configuration, which are effective at capturing RSRP differences at low speeds. However, when applied to fast-moving UEs, these offsets cause the handover threshold to be crossed more frequently as users traverse overlapping coverage areas, leading to an increase in ping-pongs. These findings emphasize that handover parameters optimized for a single mobility profile are not transferable across heterogeneous speed scenarios. They highlight the necessity of accounting for *all* speed categories simultaneously when designing mobility management schemes, a point that motivates the all-speed optimization strategy presented in the following section.

Joint optimization across all speeds: Fig. 5 shows the performance of UEs at different speed categories when the network is optimized simultaneously for all speeds, using per-cell optimization with $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$. Compared to speed-specific optimization, this strategy naturally introduces trade-offs across mobility profiles. For example, GUEs at 60 km/h see ping-pongs increase moderately from 9% to 14%, while for pedestrian users at 3 km/h the degradation is significantly constrained: instead of doubling as in the speed-mismatched scenario, ping-pongs rise only slightly from 32% to 36%. Similar patterns hold across the other speed categories, with HD-BO per-cell optimization consistently mitigating the most severe mismatches. These results confirm that optimizing for all speeds jointly cannot achieve the absolute optimum for any single mobility profile. Nevertheless, it provides a balanced solution that substantially reduces the performance degradations observed under mismatched configurations, thereby supporting more robust mobility management across heterogeneous UE populations.

To further understand the impact of HD-BO on radio link quality, Fig. 6 illustrates the CDF of the SINR for UEs at each speed category when the network is optimized for all speeds simultaneously. This figure complements the previous analysis by showing that, despite optimizing handover parameters for reduced RLF and ping-pongs, HD-BO can maintain SINR performance close to the 3GPP set-5 benchmark. The black lines represent the SINR under 3GPP set-5, which serves as an upper bound by enforcing fast handovers to the best candidate BS while ignoring ping-pong constraints. The blue curves show the performance after HD-BO optimization with $w_{\text{PP}} = 1$ and $w_{\text{RLF}} = 9$, prioritizing RLF minimization across a street portfolio with diverse mobility patterns. The similarity in SINR distributions between HD-BO and 3GPP set-5 explains why RLF performance remains nearly identical in both cases: in both strategies, UEs are handed over to strong-signal BSs in a timely manner. Moreover, the reduction in ping-pongs relative to 3GPP set-5 is annotated in boxed text for each speed category—8% for 3 km/h, 28% for 60 km/h, and 100% for 30 km/h—demonstrating that HD-BO successfully mitigates unnecessary handovers without compromising signal quality. It should be noted that the hump observed in the CDF of the SINRs is due to site-specific propagation conditions. The urban effects on the channel such as LoS dominance and diffraction, results in the concentration of users around these SINR levels, which manifests as a hump in the CDF.

D. Case study #2: Aerial UE Mobility Management

In our second case study, we focus on mobility management for aerial UEs (i.e., UAVs) along predefined 3D aerial corridors at altitudes of 140–160 m.

Optimal cellular antenna configurations: As cellular BSs are traditionally designed to optimize 2D ground-level connectivity, aerial UEs are often limited to receiving signals through the weaker upper antenna sidelobes, resulting in significant signal in-

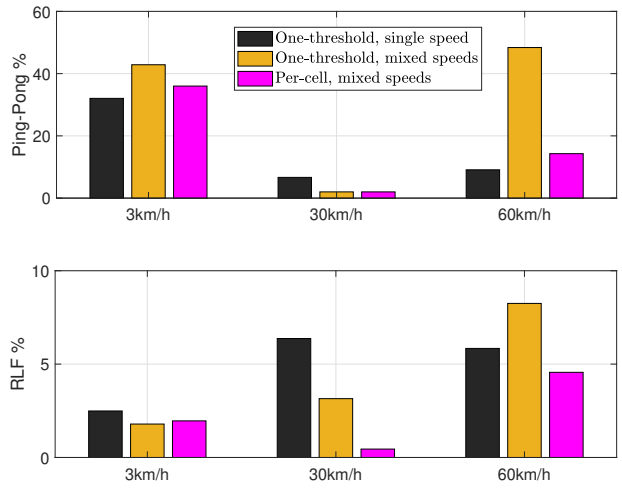

Fig. 5: Ping-pongs and RLF experienced by GUEs at different speeds when the network is optimized via HD-BO for a mix of all speeds.

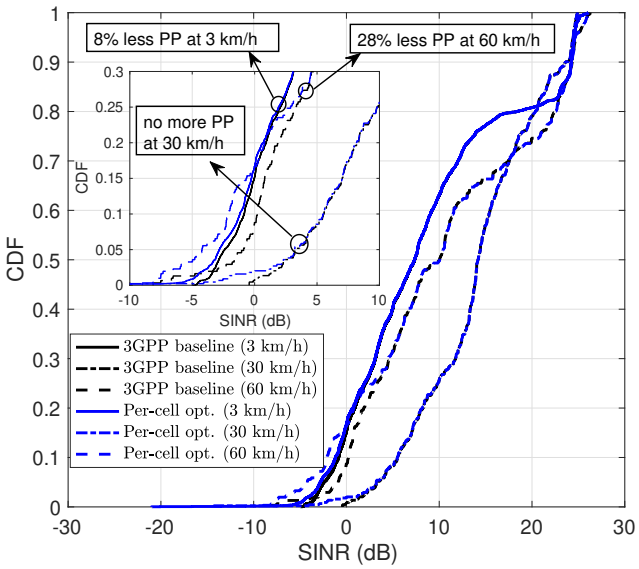

Fig. 6: SINR achieved by GUEs of different speeds when the network is optimized for all speeds with $w_{\text{PP}} = 1$ and $w_{\text{RLF}} = 9$, compared to a 3GPP set-5 configuration. Ping-pongs reduction with respect to 3GPP set-5 is indicated in boxed text for each speed.

stability during flight. Additionally, when UAVs fly above buildings, they frequently face interference from line-of-sight (LoS) signals from nearby BSs [8], [43], which degrades their signal-to-interference-plus-noise ratio (SINR) [44], [45]. To address this limitation, next-generation mobile networks are expected to provide reliable UAV connectivity by re-engineering existing ground-focused deployments [46]–[48]. In our previous study, we proposed a framework for designing a cellular deployment that accommodates both GUEs and UAVs flying along specific streets (i.e., corridors) by optimizing the electrical antenna tilts of each BS. Our findings indicate that, unlike conventional cellular networks where all BSs are downtilted, balancing coverage between ground UEs and UAV corridors necessitates uptilting a subset of BSs [42]. Therefore, for this Case Study #2 on aerial UE mobility management, before optimizing mobility, we first optimize the electrical antenna tilts of all BSs following the approach in [9]. The goal is to determine the set of antenna tilts that maximize the rates of GUEs and UAVs with equal weights. This ensures an optimized cellular deployment configuration that enhances coverage and capacity for UAVs, serving as a foundation for the subsequent aerial UE mobility optimization.

Mobility performance: Table II presents the performance of UAVs across all speeds (i.e., 3 km/h, 30 km/h, and 60 km/h), considering an equal weight distribution over the five-street portfolio. The evaluation compares the performance of HD-BO with $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$ to the two 3GPP benchmark configurations, set-1 and set-5. HD-BO outperforms both 3GPP set-1 and set-5 in reducing ping-pongs and RLF. It achieves a 3% ping-pong rate (compared to 15% for set-1 and 11% for set-5) and 0% RLF (vs. 9% for set-1). Even though the objective function prioritizes ping-pong reduction ($w_{\text{PP}} = 9$, $w_{\text{RLF}} = 1$), HD-BO is still able to completely eliminate RLF (0%), while achieving the best ping-pong performance.

TABLE II: Ping-pong (PP) and RLF performance for UAVs across all speeds with equal weight distribution over the five-street portfolio.

3GPP set-1		3GPP set-5		HD-BO	
PP (%)	RLF (%)	PP (%)	RLF (%)	PP (%)	RLF (%)
15	9	11	**0**	**3**	**0**

E. Transfer Learning Experiments

Since it is desirable for a machine learning model to deliver consistent performance across diverse scenarios [49], we now examine the generalization capabilities of the HD-BO framework across different UE speeds in the context of transfer learning.

Scenario source vs. scenario target: Transfer learning leverages knowledge or data from a previously solved problem (source) to accelerate the solution of a new but related problem (target). This approach is particularly beneficial when generating the initial dataset $\mathcal{D}$ for the BO posterior is costly or time-consuming, such as when real-world measurements are required. Let $\mathcal{D}_{\text{sr}}$ and $\mathcal{D}_{\text{tg}}$ represent the initial datasets obtained for the source and target scenarios, respectively. We conduct three evaluations by varying the proportion of the initial dataset $\mathcal{D}$ that originates from the target scenario, as follows:

- 100% ($\mathcal{D} = \mathcal{D}_{\text{tg}}$, prior knowledge based on scenario target).
- 50% (half of $\mathcal{D}$ is drawn from $\mathcal{D}_{\text{sr}}$, half is from $\mathcal{D}_{\text{tg}}$).
- 0% ($\mathcal{D} = \mathcal{D}_{\text{sr}}$, prior knowledge based on scenario source).

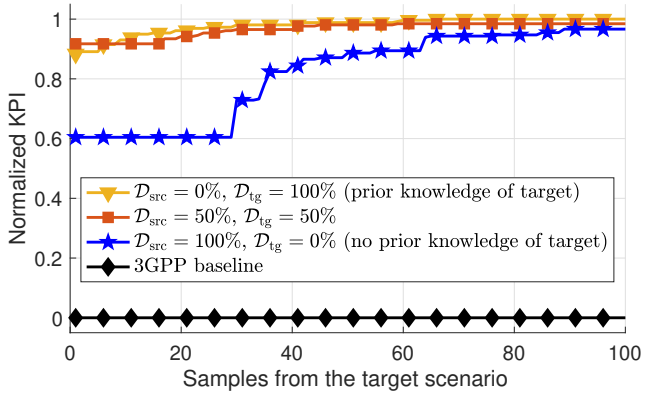

Fig. 7: Convergence of transfer learning applied on Case Study #1. Source: All GUEs at 60 km/h. Target: All GUEs at 30 km/h. The y-axis shows the min–max normalized KPI from (3), where 0 corresponds to the 3GPP baseline (worst performance) and 1 corresponds to the best achievable performance when the HD-BO posterior is trained entirely on the target scenario.

We apply scenario-specific transfer learning to Case Study #1, where the objective is to leverage data from a scenario with a certain GUE speed to optimize a new scenario with a different speed.

Convergence of transfer learning: Fig. 7 is obtained considering a source scenario based on the previously described Case Study #1, where GUEs move along a portfolio of five streets at a speed of 60 km/h. In the target scenario, we modify the GUE speed to 30 km/h. The figure illustrates the convergence of transfer learning using HD-BO by showing the best observed objective at each iteration n. To present a practically relevant measure, we plot the min-max normalized KPI from (3), where 0 corresponds to the 3GPP baseline (worst performance) and 1 represents the best achievable KPI when the HD-BO posterior contains 100% prior knowledge of the target scenario. The initial dataset $\mathcal{D}$ consists of $N_{\text{o}} = 60$ observations, following the recommendation in [39], which suggests building the prior on an initial dataset of size twice the number of optimization parameters. $\mathcal{D}$ is drawn from $\mathcal{D}_{\text{sr}}$ (blue), from $\mathcal{D}_{\text{tg}}$ (green), or half each (red). Fig. 7 shows that with a 50%/50% reliance on $\mathcal{D}_{\text{tg}}/\mathcal{D}_{\text{sr}}$, convergence occurs in a comparable number of iterations to that observed with 100% reliance on $\mathcal{D}_{\text{tg}}$ (i.e., without transfer learning). This demonstrates the HD-BO posterior's ability to generalize after optimizing a related task. Even in the absence of prior knowledge of the target scenario ($\mathcal{D} = \mathcal{D}_{\text{sr}}$), performance declines by only 3%, indicating that the posterior trained at one speed retains useful structural information that can be reused at another speed.

It is important to note that the KPI defined in (3) serves as a scalar optimization objective for the learning algorithm rather than a directly interpretable performance metric. Due to the weighted combination of multiple mobility indicators, temporal averaging, and evaluation across heterogeneous user speeds, the resulting value has no intrinsic physical meaning and cannot be interpreted in isolation (e.g., as a percentage). In our experiments, the raw KPI values range between 4.22 and 6.40; however, these magnitudes alone do not convey actionable insight. Therefore, a min–max normalization is applied to contextualize the learning process between two meaningful reference points: the current network deployment based on a 3GPP configuration (normalized to 0) and the achievable upper bound corresponding to full posterior knowledge of the target scenario (normalized to 1). This representation enables a clear visualization of transfer learning efficiency and convergence behavior without altering the relative performance trends.

Performance of successful transfer learning: Fig. 8 compares ping-pong and RLF percentages under different source–target dataset compositions. In the successful case (top), transferring from 60 km/h to 30 km/h achieves performance comparable to a fully target-trained model, confirming that prior knowledge at higher speeds can generalize well to slower mobility. These results illustrate the potential of data-driven transfer mechanisms and motivating future extensions to more complex scenarios.

Example of unsuccessful transfer learning: However, transfer learning proves ineffective for pedestrian speeds (3 km/h). The figure highlights this limitation, as scenario-specific transfer learning fails in both initial dataset variations (50%-50% and 100% scenario source). In both cases, the achieved ping-pongs and RLF performance do not match the levels obtained when the optimization is conducted with an initial dataset composed entirely of the target scenario (100% scenario target). This occurs because the HD-BO method establishes trust regions based on presumed solution locations, primarily favoring lower TTT values. Without data specific to pedestrian performance, it lacks awareness of the importance of higher TTT values and consequently defaults to optimizing only for lower ones. As a result, this approach fails to yield improvements for pedestrian-based scenarios.

IV. Cellular Mobility Management via Deep Reinforcement Learning

Having examined mobility management from a parameter-based perspective using HD-BO, we now turn to a complementary paradigm: parameter-free optimization via deep reinforcement learning (DRL). While HD-BO focuses on tuning predefined control parameters such as A3-offset and TTT, DRL eliminates these thresholds altogether by allowing an agent to directly select serving cells based on network state information. This transition from parameter optimization to parameter-free decision-making highlights the two distinct ways in which data-driven approaches can be applied to the same mobility management problem, setting the stage for a systematic comparison between them. In our case studies, we compare the KPI

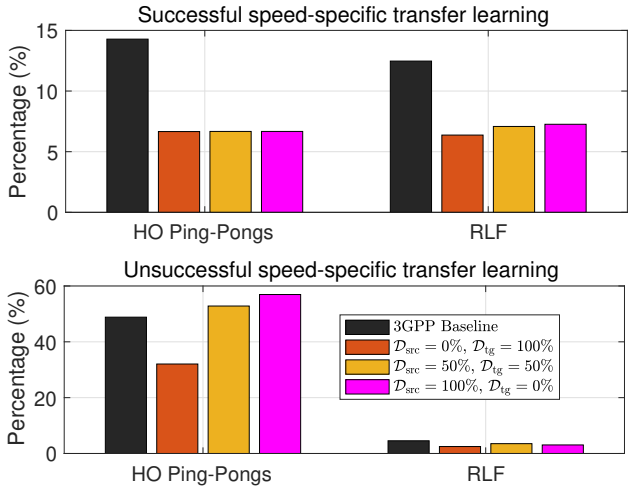

Fig. 8: Performance of transfer learning applied on Case Study #1.

performance and convergence of the DRL approach to the one based on HD-BO

A. Overview of Deep Reinforcement Learning

RL is a type of machine learning in which an agent learns by interacting with the environment. Using the data collected from these interactions, the agent learns a policy allowing it to select the actions maximizing cumulative rewards [50]. RL problems are typically modeled as Markov decision processes (MDPs), characterized by states, actions, rewards, state transition dynamics, and a discount factor balancing between immediate and future rewards. At each step, the agent observes a state describing the environment, chooses an action based on its policy (which defines the agent's behavior), and receives an immediate reward. RL algorithms can be classified into two main categories: value-based methods and policy gradient algorithms. Value-based methods, such as Q-learning, rely on value functions which estimates the expected cumulative reward that the agent can achieve, under a specific policy, from a particular state or a state-action pair. These algorithms optimize policies in an implicit fashion by selecting the actions maximizing the estimated value function. Meanwhile, policy gradient methods, including REINFORCE, directly optimize the policy parameters using gradient ascent on the expected reward.

To expand the applicability of RL, deep RL was introduced combining RL with deep neural networks (DNNs) [51]. This enables agents to handle more complex environments. Examples of DRL algorithms include Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), and Advantage Actor-Critic (A2C). While DQN is designed to deal with discrete actions using DNNs for Q-value function approximation, DDPG is developed to handle continuous actions through an actor-critic architecture. Meanwhile, PPO and A2C use stochastic policies, which not only allow them to make both discrete and continuous actions, but also improve their state space exploration compared to DQN and DDPG. Moreover, as policy-gradient algorithms, PPO and A2C can address the limitations of value-based methods like DQN which suffer from slow convergence and high approximation errors.

B. DRL-based Mobility Management

In our DRL-based mobility management framework, the state space, action space, and reward function are defined as follows:

State space: The state s_t of the environment at time step t includes the ID Γ_t of the street where the UE is located, the ID α_t of the current serving BS, the time of stay at current BS ToS_{α_t}, the ID β_t and RSRP Ξ_t of the candidate BS. Additionally, we consider a history H_t^n of n previous serving BSs, defined by the ID β_t^i and the RSRP value Ξ_t^i $i = 1 \dots n$, in the state representation. Hence, the state is given by $s_t = \{\Gamma_t, \alpha_t, \text{ToS}_{\alpha_t}, \beta_t, \Xi_t, H_t^n\}$. The agent is at the network-side controller, which observes these standard 3GPP measurements and makes mobility decisions on behalf of the UEs.

Action space: The actions of the agent at time step t include a binary action $a_t = \{0, 1\}$, representing the decision to stay connected to the current BS or make a handover to the candidate BS.

Reward function: The reward r_t at time step t is a weighted sum of the ping pong event and the RLF event, given by:

$$r_t = -w_{\text{PP}} \cdot \mathbb{1}_{\text{PP}_t} - w_{\text{RLF}} \cdot \mathbb{1}_{\text{RLF}_t}. \tag{10}$$

which is consistent with the multi-objective formulations in (2)–(3). Those equations aggregate PP and RLF rates across the whole trajectory, whereas (10) expresses the same cost in an event-driven form, suitable for RL training.

Proximal Policy Optimization: To solve the formulated RL problem, we adopt PPO [52], a DRL algorithm widely used in wireless communication applications, thanks to its robustness, stability, and sample efficiency. PPO optimizes its policy through a clipped objective function, which prevents excessively large policy updates. This offers more stable and reliable training compared to other actor-critic methods. PPO includes two neural networks: the policy network (actor) and the value network (critic). The former is responsible for action selection where the actions are sampled from a probability distribution based on the actor's stochastic policy π_ψ. Meanwhile, the latter evaluates the actions taken by the policy network through value function estimation. Through interactions between the agent and the environment, batches of trajectories (i.e., sequences of states, actions, rewards, and next states) are collected to update both networks. Unlike HD-BO, the DRL approach does not rely on fixed

HO parameters (A3/TTT). Instead, the policy learns serving-cell decisions directly from experience, making it effectively "parameter-free" with respect to handover thresholds.

Value loss: The value network parameters β are updated by minimizing the value loss, defined as

$$\mathcal{L}_{\text{value}}(\beta) = \mathbb{E}_t \left[\left(V_\beta \left(s_t \right) - G_t \right)^2 \right] \tag{11}$$

where $V_\beta \left(s_t \right)$ is the estimated state-value function and G_t denotes the observed return.

Policy objective: Simultaneously, the policy network π_ψ is updated by maximizing the policy objective, given by

$$\begin{aligned} \mathcal{L}_{\text{policy}}(\psi) = \mathbb{E}_t[\min(\rho_t A_{\pi_\psi}, \text{clip}(\rho_t, 1-\tau, 1+\tau) A_{\pi_\psi}] \\ + \epsilon_{\text{exp}} \mathbb{E}_{\pi_\psi}[\log \pi_\psi(a|s_t)] \end{aligned} \tag{12}$$

where the first term refers to the clipped surrogate objective and the second terms is the entropy loss which encourages exploration. Also, $\rho_t = \frac{\pi_\psi(a_t|s_t)}{\pi_{\psi_{\text{old}}}(a_t|s_t)}$ represents the probability ratio between the new and old policies and A_{π_ψ} denotes the advantage estimate, which measures the quality of an action given a state based on the critic's estimated value function. In addition, ϵ_{exp} and τ are the entropy coefficient and the clipping hyperparameter, respectively.

Training process: The training process is repeated over multiple episodes, allowing the critic to improve its value function estimation and the actor to enhance its action selection based on the critic's feedback. We fine-tune the different hyperparameters of the PPO algorithm (including both the actor and the critic networks architectures) through extensive experimentation. The policy network is designed using three layers with 256, 128, and 256 neurons, respectively, while the value network includes three layers with 64 neurons each. To optimize the losses, an Adam optimizer is used with a learning rate of 0.0001. Additionally, we select a discount factor $\gamma = 0.95$, an entropy coefficient $\epsilon_{\text{exp}} = 0.002$, and a clipping parameter $\tau = 0.2$.

C. Performance and Convergence Assessment

We now compare the performance of the PPO agent to the HD-BO approach and to the 3GPP baseline configurations (set-1 and set-5) for Case Study #1, i.e., GUE mobility management. The PPO agent's objective is to maximize the reward function defined in (10). Unlike traditional approaches, the RL-PPO agent selects the next serving BS without relying on predefined network parameters such as the A3-offset and TTT.

Fig. 9 illustrates the mobility performance at different speeds (3 km/h, 30 km/h, and 60 km/h) in terms of ping-pongs and RLF. The objective function weights are set to $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$. The following key observations can be drawn.

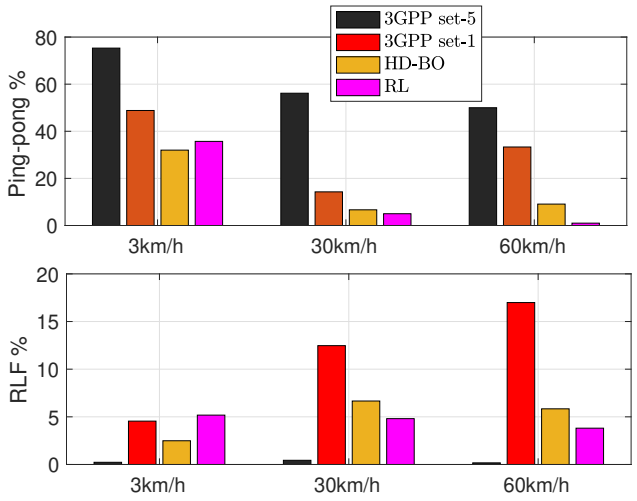

Fig. 9: Ping-pongs and RLF performance for GUEs at different speeds when the network is optimized via RL-PPO, HD-BO, and the 3GPP baseline configurations. Lower ping-pong and RLF percentages are better, indicating more efficient mobility management.

RL-PPO vs. 3GPP baselines: The RL-PPO framework significantly outperforms both 3GPP set-1 and set-5 configurations in terms of reducing ping-pongs. For example, at 30 km/h, RL-PPO achieves a ping-pong rate of 5%, compared to 56.16% for set-5 and 14.2% for set-1. Additionally, RL-PPO maintains a lower RLF rate across all speeds, despite the objective function prioritizing ping-pong reduction by assigning it a higher weight. This demonstrates RL-PPO's ability of providing robust mobility management.[4]

RL-PPO vs. HD-BO: The performance achieved through RL-PPO is comparable to HD-BO in both ping-pong reduction and RLF minimization. For instance, at 3 km/h, RL-PPO maintains a ping-pong rate of 35%, similar to HD-BO's 32%. At 30 km/h RL-PPO performs slightly better achieving a ping-pong rate of 5%, compared with HD-BO's 7%. At 60 km/h, RL-PPO achieves a ping-pong rate of nearly 0%. Performance-wise, this confirms RL-PPO as a viable alternative to parameter-based mobility management, without predefined A3-offset and TTT thresholds.

Sample efficiency: Table III compares the convergence behavior of HD-BO and the RL-PPO framework in terms of the total number of iterations required. This comparison is conducted for both objective function weight configurations: $\{w_{\text{PP}} = 9,\ w_{\text{RLF}} = 1\}$ and $\{w_{\text{PP}} = 1,\ w_{\text{RLF}} = 9\}$. While RL-PPO achieves comparable performance to HD-BO, a key drawback is its significantly higher sample complexity. Each

[4]3GPP set-5 can be regarded as a practical upper bound in terms of RLF performance, as it performs fast handovers without considering the ping-pong effect, focusing solely on minimizing RLF. Therefore, the primary comparison should be made with 3GPP set-1, which is specifically designed to reduce ping-pongs while also accounting for RLF.

iteration in both methods involves running a simulation or taking a measurement, making sample efficiency critical. In our setting, each optimization iteration requires one system-level evaluation (i.e., one sample) obtained through simulation or real-world measurements. Hence, the number of iterations directly corresponds to the sample complexity of the framework. We therefore report sample complexity in terms of iterations, since both are equivalent in this context. RL-PPO requires 10 to 250 times more iterations before convergence compared to HD-BO, depending on the KPI weight configuration and GUE speed. For instance, at 30 km/h with $w_{\text{PP}} = 9, w_{\text{RLF}} = 1$, RL-PPO requires 14,000 iterations—each corresponding to a costly simulation—whereas HD-BO converges in just 60 iterations.

Although RL-PPO is well-suited for digital twin environments, where large-scale simulations can efficiently generate training data, in real-world measurement-based scenarios, where data collection is mission-critical, costly, or labor-intensive, its high sample complexity makes RL-PPO less viable compared to HD-BO's data-efficient optimization approach.

TABLE III: Convergence comparison based on number of iterations at different speeds with varying KPI weight parameters.

	3 km/h		30 km/h		60 km/h	
	HD-BO	RL-PPO	HD-BO	RL-PPO	HD-BO	RL-PPO
$w_{\text{PP}} = 9$, $w_{\text{RLF}} = 1$	**125**	3200	**60**	14000	**30**	4300
$w_{\text{PP}} = 1$, $w_{\text{RLF}} = 9$	**140**	320	**75**	300	**45**	270

D. Transfer Learning Experiments

We now evaluate the generalization capability of the RL-PPO-based mobility management framework in adapting to aerial UEs at a altitude of 150 m, using an agent trained exclusively on GUEs at 1.5 m. The study focuses on a mobility scenario where both categories of UEs move at a speed of 30 km/h. As previously discussed, RL-PPO-based mobility management requires large-scale simulations to generate extensive training datasets. The objective of transfer learning in this context is to minimize the need for collecting new data when UE altitude changes (e.g., when an aerial highway is relocated due to regulations [53]).

Table IV presents the PP and RLF performance for $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$. The results highlight the effectiveness of transfer learning in reducing training overhead while maintaining optimal mobility performance. With transfer learning, both PP and RLF rates remain at 0%, while significantly reducing the number of iterations required for training. Without transfer learning, the RL-PPO agent requires 6,200 iterations to converge. Transfer learning cuts this down to 2,400 iterations—a 2.5× reduction in training effort—successfully generalizing to aerial UEs without compromising handover efficiency. These results also underscore the data efficiency of the proposed DRL framework. By leveraging transfer learning, the agent requires significantly fewer training iterations (2,400 vs. 6,200) to achieve the same optimal performance, demonstrating how prior knowledge can reduce the amount of data and training effort needed for convergence.

TABLE IV: PP and RLF performance for UAVs at 150 m and 30 km/h, w/ and w/o RL-PPO transfer learning, for $w_{\text{PP}} = 9$ and $w_{\text{RLF}} = 1$.

	Set-1	Set-5	w/o Transfer Learning	w/ Transfer Learning
PP (%)	0.0	20.0	0.0	0.0
RLF (%)	5.18	0.0	0.0	0.0
Iterations	-	-	6200	2400

V. Conclusion and Discussion

This paper explored two distinct data-driven approaches for mobility management in cellular networks: HD-BO and DRL. While both aim to optimize HO performance, they operate under fundamentally different paradigms—HD-BO as a parameter-based optimization method and DRL as a parameter-free learning framework. Our study highlights the complementary strengths of these two methods and provides comparisons of their applicability in real-world site-specific mobility management scenarios. Both approaches aim to optimize HO performance, targeting trade-offs between ping-pongs and RLF. The HD-BO method enables scalable, sample-efficient optimization over large cellular deployments by constructing local surrogate models and leveraging trust-region strategies. It demonstrates strong performance across diverse UE mobility profiles, including both GUEs and UAVs, without requiring extensive training data. In contrast, DRL provides a model-free solution that bypasses the need for predefined HO thresholds, directly learning optimal mobility policies from interaction with the environment. While DRL offers flexibility, it requires significantly more training iterations, which can limit its feasibility in real-world deployments where data collection is costly or constrained. To address this, we apply transfer learning to both approaches, demonstrating its ability to accelerate convergence and improve generalization across UE speeds and altitudes. These results suggest that transfer learning is particularly effective in reducing training demands, especially for DRL, and highlight the broader potential of adaptive, data-driven mobility management in dense and heterogeneous network scenarios. The transfer learning results presented in this paper can be regarded as a proof of concept illustrating the potential of data-driven mobility management. While our case study focuses on speed variation, which already poses non-trivial challenges due to its strong impact on signal behavior and handover dynamics, extending this framework to more complex scenarios; such as different urban layouts, remains an important direction for future work.

Deployment Practicality: A key aspect of data-driven mobility management is the feasibility of deployment in operational networks. In our framework, the optimization process is envisioned to run at the network controller, which already has access to the necessary performance measurements collected in

current systems. For the HD-BO approach, these measurements include for example L1/L3-filtered RSRP. Optimized handover thresholds (A3-offsets and TTTs) are then configured at the cell level via standard control signaling. Since these updates occur at large time scales (e.g., hours or days), the resulting signaling overhead is negligible. For the DRL framework, we explicitly restrict the state space to features that are realistically observable at the controller: serving and candidate cell IDs, and filtered RSRP values. These are all quantities already reported by UEs as part of the measurement reporting framework, and no new feedback channels are required. Overall, both approaches leverage existing 3GPP-compliant measurement procedures and rely on site-specific configurations. This makes the proposed methods compatible with ongoing standardization efforts, and we believe they can be integrated into future cellular deployments without introducing prohibitive communication or computational overhead.

Future work: Several areas remain open for further exploration, including the following ones:

- *Beam-based mobility management:* Future 6G deployments may operate in the FR3 spectrum and rely on highly directional beams. Investigating beam-based HO strategies that integrate beam selection and mobility optimization is crucial for ensuring seamless connectivity.
- *Multi-RAT handovers:* In integrated terrestrial and non-terrestrial networks (NTN), UEs may switch between cellular and satellite network segments [54]. Extending our framework to multi-RAT handovers would enable load balancing and mobility robustness across a heterogeneous infrastructure.
- *Multi-agent RL:* Extending the DRL-based HO management to multi-agent systems could improve performance in large-scale networks with load balancing-aware mobility management. Specifically, in such scenario the state and action spaces would be extremely large for one agent to handle. Thus, multiple RL agents could be deployed where each agent has partial knowledge of the network. Then, using a distributed learning approach, HO management could be optimized with reduced overhead.
- *Robustness to stochastic RL training:* Training DRL agents can exhibit performance variability due to random initialization and stochastic optimization, even when using the same dataset. While the proposed framework is evaluated under a fixed training setup, a systematic multi-seed training and evaluation analysis is an important direction for future work to assess robustness and statistical stability.

REFERENCES

[1] M. Benzaghta, S. Ammar, D. López-Pérez, B. Shihada, and G. Geraci, "Data-driven design of 3GPP handover parameters with Bayesian optimization and transfer learning," *arXiv:2504.02633*, 2025.

[2] D. Lopez-Perez, N. Piovesan, and G. Geraci, "Capacity and power consumption of multi-layer 6G networks using the upper mid-band," in *Proc. IEEE ICC*, 2025.

[3] E. Oughton, G. Geraci, M. Polese, M. Ghosh, W. Webb, and D. Bubley, "The future of wireless broadband in the peak smartphone era: 6G, Wi-Fi 7, and Wi-Fi 8," *IEEE Wireless Commun. Mag.*, 2025.

[4] S. M. Shahid, J.-H. Na, and S. Kwon, "Incorporating mobility prediction in handover procedure for frequent-handover mitigation in small-cell networks," *IEEE Trans. Network Sci. and Eng.*, 2024.

[5] X. Ge, J. Ye, Y. Yang, and Q. Li, "User mobility evaluation for 5G small cell networks based on individual mobility model," *IEEE J. on Sel. Areas Commun.*, 2016.

[6] M.-T. Nguyen and S. Kwon, "Geometry-based analysis of optimal handover parameters for self-organizing networks," *IEEE Trans. Wireless Commun.*, 2020.

[7] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, "A comprehensive survey on mobility management in 5G heterogeneous networks: Architectures, challenges and solutions," *IEEE Access*, 2020.

[8] G. Geraci, A. Garcia-Rodriguez, M. M. Azari, A. Lozano, M. Mezzavilla, S. Chatzinotas, Y. Chen, S. Rangan, and M. Di Renzo, "What will the future of UAV cellular communications be? A flight from 5G to 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1304–1335, 2022.

[9] M. Benzaghta, G. Geraci, D. López-Pérez, and A. Valcarce, "Designing cellular networks for UAV corridors via Bayesian optimization," in *Proc. IEEE Globecom*, 2023, pp. 4552–4557.

[10] R. Karmakar, G. Kaddoum, and S. Chattopadhyay, "Mobility management in 5G and beyond: A novel smart handover with adaptive time-to-trigger and hysteresis margin," *IEEE Trans. Mobile Comput.*, 2022.

[11] W.-C. Chien, Y. Huang, B.-Y. Chang, and W.-Y. Hwang, "Privacy-preserving handover optimization using federated learning and LSTM networks," *Sensors*, 2024.

[12] R. Arshad and L. Lampe, "Stochastic geometry analysis of user mobility in RF/VLC hybrid networks," *IEEE Trans. Wireless Commun.*, 2021.

[13] A. A. Malik, M. A. Jamshed, A. Nauman, A. Iqbal, A. Shakeel, and R. Hussain, "Performance evaluation of handover triggering condition estimation using mobility models in heterogeneous mobile networks," *IET Networks*, 2024.

[14] N. V. Huynh, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Optimal beam association for high mobility mmwave vehicular networks: Lightweight parallel reinforcement learning approach," *IEEE Trans. Commun.*, 2021.

[15] A. Alizadeh, B. Lim, and M. Vu, "Multi-agent Q-learning for real-time load balancing user association and handover in mobile networks," *IEEE Trans. Wireless Commun.*, 2024.

[16] A. Prado, F. Stöckeler, F. Mehmeti, P. Krämer, and W. Kellerer, "Enabling proportionally-fair mobility management with reinforcement learning in 5G networks," *IEEE J. on Sel. Areas in Commun.*, 2023.

[17] J. Dai, S. Mahboob, H. Wang, and L. Liu, "Intelligent handover management enabled by O-RAN and deep reinforcement learning," in *Proc. IEEE VTC*, 2024, pp. 1–6.

[18] Y. Chen, X. Lin, T. Khan, and M. Mozaffari, "Efficient drone mobility support using reinforcement learning," in *Proc. IEEE WCNC*, 2020.

[19] Y. Chen, X. Lin, T. Khan, and M. Mozaffari, "A deep learning approach to efficient drone mobility support," in *Proc. ACM MobiCom Workshop on Drone Assisted Wireless Commun. for 5G and Beyond*, 2020, pp. 67–72.

[20] I. A. Meer, M. Ozger, D. A. Schupke, and C. Cavdar, "Mobility management for cellular-connected UAVs: Model-based versus learning-based approaches for service availability," *IEEE Trans. on Network and Ser. Man.*, 2024.

[21] B. Galkin, E. Fonseca, R. Amer, L. A. Dasilva, and I. Dusparic, "REQIBA: Regression and deep Q-learning for intelligent UAV cellular user to base station association," *IEEE Trans. Vehicular Tech.*, 2022.

[22] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[23] R. M. Dreifuerst, S. Daulton, Y. Qian, P. Varkey, M. Balandat, S. Kasturia, A. Tomar, A. Yazdan, V. Ponnampalam, and R. W. Heath, "Optimizing

coverage and capacity in cellular networks using machine learning," in *Proc. IEEE ICASSP*, 2021, pp. 8138–8142.

[24] L. Eller, P. Svoboda, and M. Rupp, "A differentiable throughput model for load-aware cellular network optimization through gradient descent," *IEEE Access*, 2024.

[25] Y. Zhang, O. Simeone, S. T. Jose, L. Maggi, and A. Valcarce, "Bayesian and multi-armed contextual meta-optimization for efficient wireless radio resource management," *IEEE Trans. on Cognitive Communications and Networking*, 2023.

[26] L. Maggi, A. Valcarce, and J. Hoydis, "Bayesian optimization for radio resource management: Open loop power control," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1858–1871, 2021.

[27] S. S. Tambovskiy, G. Fodor, and H. Tullberg, "Cell-free data power control via scalable multi-objective Bayesian optimisation," in *Proc. IEEE PIMRC*, 2022, pp. 1–6.

[28] L. Maggi, C. Mihailescu, Q. Cao, A. Tetich, S. Khan, S. Aaltonen, R. Koblitz, M. Holma, S. Macchi, M. E. Ruggieri *et al.*, "Energy savings under performance constraints via carrier shutdown with Bayesian learning," in *Proc. EuCNC*, 2023, pp. 1–6.

[29] E. Tekgul, T. Novlan, S. Akoum, and J. G. Andrews, "Joint uplink-downlink capacity and coverage optimization via site-specific learning of antenna settings," *IEEE Trans. Wireless Commun.*, 2024.

[30] E. de Carvalho, A. V. Rial, and G. Geraci, "Towards mobility management with multi-objective Bayesian optimization," in *Proc. IEEE WCNC*, 2023, pp. 1–6.

[31] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv:1807.02811*, 2018.

[32] 3GPP TR 38.843, "Study on artificial intelligence (AI)/machine learning (ML) for NR air interface (Release 18)," Jun. 2023.

[33] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna RT: Differentiable ray tracing for radio propagation modeling," *arXiv:2303.11103*, 2023.

[34] 3GPP Technical Report 36.814, "Evolved Universal Terrestrial Radio Access (EUTRA); Further advancements for E-UTRA physical layer aspects," Mar. 2017.

[35] 3GPP TR 36.839, "Mobility enhancements in heterogeneous networks," Aug. 2012.

[36] D. Lopez-Perez, I. Guvenc, and X. Chu, "Mobility management challenges in 3GPP heterogeneous networks," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 70–78, 2012.

[37] 3GPP TS 36.331, "Radio resource control; protocol specification," Dec. 2011.

[38] 3GPP TS 36.300, ""evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran)," Oct. 2011.

[39] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local Bayesian optimization," *NeurIPS*, vol. 32, 2019.

[40] D. Eriksson and M. Jankowiak, "High-dimensional Bayesian optimization with sparse axis-aligned subspaces," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 493–503.

[41] Y. Shen and C. Kingsford, "Computationally efficient high-dimensional Bayesian optimization via variable selection," *arXiv:2109.09264*, 2021.

[42] M. Benzaghta, G. Geraci, D. López-Pérez, and A. Valcarce, "Cellular network design for UAV corridors via data-driven high-dimensional Bayesian optimization," *arXiv:2504.05176*, 2025.

[43] A. Giuliani, R. Nikbakht, G. Geraci, S. Kang, A. Lozano, and S. Rangan, "Spatially consistent air-to-ground channel modeling via generative neural networks," *IEEE Commun. Lett.*, vol. 13, no. 4, pp. 1158–1162, 2024.

[44] G. Geraci, A. Garcia-Rodriguez, L. Galati-Giordano, D. López-Pérez, and E. Björnson, "Understanding UAV cellular communications: From existing networks to massive MIMO," *IEEE Access*, 2018.

[45] Y. Zeng, I. Guvenc, R. Zhang, G. Geraci, and D. W. Matolak, *UAV Communications for 5G and Beyond*. John Wiley & Sons, 2020.

[46] G. Geraci, D. López-Pérez, M. Benzaghta, and S. Chatzinotas, "Integrating terrestrial and non-terrestrial networks: 3D opportunities and challenges," *IEEE Commun. Mag.*, 2023.

[47] S. Karimi-Bidhendi, G. Geraci, and H. Jafarkhani, "Optimizing cellular networks for UAV corridors via quantization theory," *IEEE Trans. Wireless Commun.*, 2024.

[48] S. Karimi-Bidhendi, G. Geraci, and H. Jafarkhani, "Mathematical cell deployment optimization for capacity and coverage of ground and UAV users," *arXiv:2502.00928*, 2025.

[49] X. Lin, "An overview of the 3GPP study on artificial intelligence for 5G new radio," *arXiv:2308.05315*, 2023.

[50] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction, 2nd edition*. MIT press Cambridge, 2018.

[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[53] M. Bernabè, D. López-Pèrez, N. Piovesan, G. Geraci, and D. Gesbert, "Massive MIMO for aerial highways: Enhancing cell selection via SSB beams optimization," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3975–3996, 2024.

[54] M. Benzaghta, G. Geraci, R. Nikbakht, and D. López-Pérez, "UAV communications in integrated terrestrial and non-terrestrial networks," in *Proc. IEEE Globecom*, 2022, pp. 3706–3711.