Meta-Continual Mobility Forecasting for Proactive Handover Prediction

Sasi Vardhan Reddy Mandapati
Independent Researcher
Email: mandapativardhan135@gmail.com

Abstract—Short-term mobility forecasting is a core building block for proactive handover (HO) decisions in modern cellular systems. Real-world mobility is non-stationary: abrupt turns and rapid speed changes cause conventional predictors to drift and produce mistimed handovers, leading to ping-pongs and missed-HO events. We propose a lightweight meta-continual forecasting framework that combines a GRU-based base predictor, Reptile meta-initialization for fast few-shot adaptation, and an EWMA residual-based drift detector to trigger compact online updates. Evaluated on a reproducible GeoLife + DeepMIMO pipeline, our approach achieves an ADE of 4.46 m and FDE of 7.79 m in zero-shot settings, improves few-shot ADE to 3.71 m at 10-shot, recovers from abrupt drift roughly 2–3× faster than an offline GRU, and yields downstream HO improvements (F1 = 0.83, AUROC = 0.90) with substantial reductions in missed-HO and ping-pong rates. The model is lightweight (128k parameters) and suitable for edge deployment.

Index Terms—mobility forecasting, handover prediction, meta-learning, continual learning, 5G, online adaptation

I. INTRODUCTION

Real-world mobility is irregular, abrupt, and difficult to model. Pedestrians change routes without warning, vehicles accelerate unpredictably, and these behaviors translate into rapid shifts in radio conditions experienced by user equipment (UE). Such non-stationarities cause trajectory predictors to drift and lead to mistimed handover (HO) decisions, ping-pong transitions, or late HOs that degrade quality of service. As cellular networks densify and 5G/6G architectures move toward finer spatial reuse, the cost of unreliable mobility and HO prediction grows significantly [1]–[6].

Recent ML-based mobility and HO forecasting approaches perform well under stationary or slowly evolving conditions but degrade sharply when motion patterns change abruptly [1], [3], [4], [8]. Many existing sequence models require long support windows to adapt, rely on computationally heavy architectures, or only support offline retraining rather than real-time updates [2], [4], [7]. Although meta-learning has shown promise for fast adaptation in wireless tasks such as beam or channel prediction [9]–[11], it has rarely been applied to mobility forecasting, and typically does not incorporate a drift detector to decide when adaptation is needed. Likewise, online and continual learning methods are well studied in general ML [12]–[14], but have not yet been combined with mobility forecasting and handover prediction in a way that supports lightweight, streaming deployment.

We address these challenges by designing a lightweight forecasting pipeline that adapts quickly using only a handful of new samples. The method combines a GRU-based base model for stable sequence processing, a Reptile meta-initialization for rapid few-shot adaptation across diverse mobility patterns, and an EWMA-based residual statistic to detect drift and trigger compact online updates. Our goal is not to build the most expressive model, but a model that stays reliable when user behavior changes sharply—an important requirement for resource-constrained, edge-deployable systems.

II. RELATED WORK

Machine learning for mobility forecasting has advanced rapidly in recent years. Deep recurrent and attention-based architectures have been applied to predict both pedestrian and vehicular trajectories, demonstrating strong performance under structured or slowly varying motion [1]–[3]. For instance, Zhao et al. use spatio-temporal attention to capture contextual dependencies in urban trajectories [1], while Sun et al. provide a comprehensive overview of deep trajectory models and their limitations in dynamic environments [2]. Convolutional–recurrent hybrids also model short-term mobility patterns effectively [3]. However, these models degrade significantly under sudden non-stationarities such as sharp turns or rapid accelerations. Li et al. show that even adaptive vehicle trajectory methods struggle when user behavior deviates abruptly from past patterns [4]. This sensitivity motivates real-time drift-aware forecasting.

A parallel body of work addresses handover (HO) prediction in cellular systems. Surveys such as Zhang et al. [5] outline classical and ML-based HO strategies, while recent studies apply LSTMs, MLPs, or CNNs for HO estimation using RSRP, RSRQ, or signal trends [6]–[8]. Pang et al. use deep models for proactive HO triggering in dense networks [7], and Wang et al. predict HO failures via LSTM-based sequence models [8]. Yet most HO predictors assume stable mobility within the prediction window and require long observation sequences or offline retraining, limiting their robustness under abrupt motion changes or sparse updates.

Meta-learning and few-shot adaptation have recently emerged in wireless communication applications. Alrabeiah and Alkhateeb [9] use deep learning for mmWave beam prediction, while later works show the value of meta-learning for fast beam selection and adaptation under shifting propagation conditions [10], [11]. However, meta-learning for mobility forecasting remains largely unexplored, and existing meta-learning approaches for wireless systems rarely incorporate

online drift detection or streaming adaptation. Meanwhile, the broader ML literature emphasizes continual adaptation under distribution shift, as covered in surveys on concept drift and online continual learning [12]–[14]. These insights suggest that unifying mobility forecasting, HO prediction, and meta-continual adaptation may provide robust performance under real-world dynamics.

Overall, prior research provides strong foundations in mobility prediction, HO forecasting, and meta-learning, but these efforts remain siloed. Mobility models rarely integrate with HO decisions, HO models often ignore mobility dynamics, and meta-learning approaches lack online drift triggers. This motivates our unified framework integrating mobility forecasting, proactive HO prediction, and drift-aware meta-continual learning.

III. PROBLEM FORMULATION

We consider short-term mobility forecasting and proactive handover prediction for a single UE moving within a cellular network. At time t, the UE reports its position and a set of radio and mobility-derived measurements. The goal is to forecast the UE's future positions over horizon H and use those predictions to infer future serving cells and HO events.

A. Inputs

Let $p_t \in \mathbb{R}^2$ denote the UE position at time t. The measurement vector is

$$m_t = [\text{RSRP}_t, \text{ BeamIndex}_t, \text{ Speed}_t, \text{ Heading}_t] \in \mathbb{R}^d.$$

The model receives a sliding window of the past k observations:

$$X_t = \{(p_{t-k+1}, m_{t-k+1}), \ldots, (p_t, m_t)\}.$$

B. Trajectory Forecasting

Given X_t, the model outputs predicted positions

$$\hat{P}_{t+1:t+H} = \{\hat{p}_{t+1}, \ldots, \hat{p}_{t+H}\}.$$

C. Handover Prediction

Using predicted positions, serving cells are inferred via an RSRP proxy:

$$\hat{c}_{t+\tau} = \arg\max_{b \in \mathcal{B}} \text{RSRP}(b, \hat{p}_{t+\tau}).$$

Handover labels are $h_{t+\tau} = 1$ if $c_{t+\tau} \neq c_{t+\tau-1}$ (else 0). We use a 3-step HO horizon ($H = 3$) for proactive decisions.

D. Metrics

Trajectory accuracy: ADE and FDE:

$$\text{ADE} = \frac{1}{H} \sum_{\tau=1}^{H} \|\hat{p}_{t+\tau} - p_{t+\tau}\|_2, \quad \text{FDE} = \|\hat{p}_{t+H} - p_{t+H}\|_2.$$

HO metrics: Accuracy, Precision, Recall, F1, AUROC, ping-pong rate, missed-HO count. Adaptation metrics: Error vs. shots (1,5,10,20), drift recovery time.

IV. METHOD

We propose a meta-continual forecasting framework integrating: (i) GRU-based base predictor, (ii) Reptile meta-initialization, and (iii) EWMA residual detection with compact online adaptation. The architecture is shown in Fig. 1.

A. Base GRU Predictor

Inputs are concatenated $z_\tau = [p_\tau, m_\tau]$. The GRU updates:

$$h_\tau = \text{GRU}(z_\tau, h_{\tau-1}),$$

and the MLP heads produce $\hat{p}_{t+\tau} = f_\theta(h_t, \tau)$ and HO logits $\hat{s}_{t+\tau} = g_\theta(h_t, \tau)$.

B. Task definition and Joint Loss

Meta-tasks correspond to city tiles: $\mathcal{T} = \{T_1, \ldots, T_M\}$. Each example at time t has loss

$$\mathcal{L}_{\text{traj}}(t) = \frac{1}{H} \sum_{\tau=1}^{H} \|\hat{p}_{t+\tau} - p_{t+\tau}\|_2^2,$$

$$\mathcal{L}_{\text{HO}}(t) = -\frac{1}{H} \sum_{\tau=1}^{H} \left[h_{t+\tau} \log \sigma(\hat{s}_{t+\tau}) + (1-h_{t+\tau}) \log(1-\sigma(\hat{s}_{t+\tau}))\right],$$

$$\mathcal{L}(t) = \mathcal{L}_{\text{traj}}(t) + \lambda_{\text{ho}} \mathcal{L}_{\text{HO}}(t).$$

C. Meta-Learning via Reptile

For each task T_i perform S inner-loop steps to obtain $\theta^{(i)}$ and meta-update:

$$\theta \leftarrow \theta + \beta(\theta^{(i)} - \theta).$$

Reptile avoids second-order gradients and scales well for many tasks.

D. EWMA Drift Detection

Residual $r_t = \|p_t - \hat{p}_t\|_2$, EWMA statistic $s_t = \lambda r_t + (1-\lambda)s_{t-1}$. Trigger when $s_t > \mu + \gamma\sigma$ (rolling μ, σ).

E. Online Adaptation

Upon trigger, perform K adaptation steps on the last N samples:

$$\theta \leftarrow \theta - \eta \nabla_\theta \left(\mathcal{L}_{\text{adapt}} + \lambda_{\text{reg}} \|\theta - \theta^*\|_2^2\right),$$

where θ^* is meta-initialization.

F. Algorithms

V. DATASETS AND SIMULATION PIPELINE

To evaluate the proposed forecasting and handover prediction framework under realistic and reproducible conditions, we construct a hybrid dataset that combines real human and vehicular mobility traces with ray-traced radio measurements. This section describes (i) the mobility data source, (ii) the radio-layer simulation using DeepMIMO, (iii) the construction of meta-learning tasks, and (iv) the drift scenarios used for stress testing.
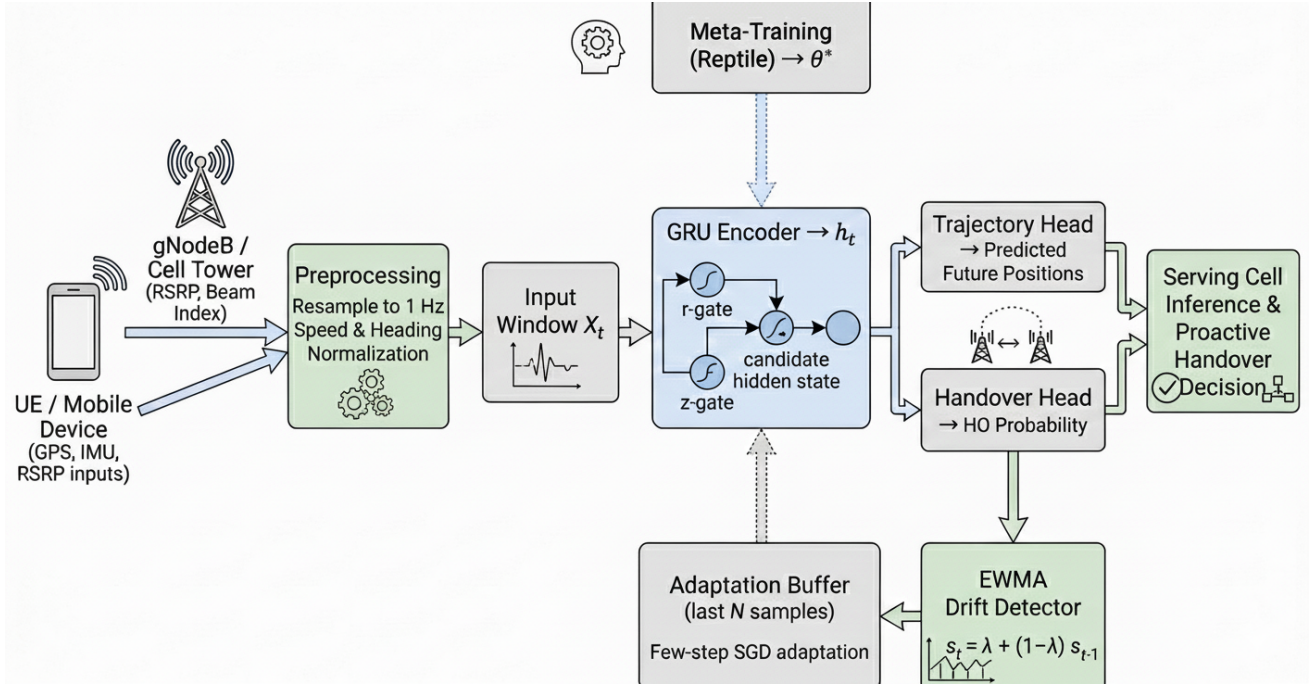
Fig. 1. System architecture: offline meta-training produces θ^* (top). Runtime uses a GRU encoder with trajectory and HO heads; EWMA residual monitor triggers compact adaptation using last-N samples with regularization toward θ^*. Predicted positions feed serving-cell inference and proactive HO decisions.

Algorithm 1 Reptile meta-training (offline)

1: Input: Tasks $\mathcal{T}$, model f_θ, inner LR α, inner steps S, meta LR β, iterations I
2: **for** iter $= 1$ to I **do**
3:     Sample batch of tasks $B \subset \mathcal{T}$
4:     **for** each task $\tau \in B$ **do**
5:         $\theta_\tau \leftarrow \theta$
6:         **for** $s = 1$ to S **do**
7:             Compute $\mathcal{L}_{\text{support}}$ on support set
8:             $\theta_\tau \leftarrow \theta_\tau - \alpha \nabla_{\theta_\tau} \mathcal{L}_{\text{support}}$
9:         **end for**
10:     **end for**
11:     $\theta \leftarrow \theta + \beta \frac{1}{|B|} \sum_{\tau \in B} (\theta_\tau - \theta)$
12: **end for**
13: **return** θ^*

Algorithm 2 Online deployment with EWMA-triggered adaptation

1: Input: θ^*, stream (X_t, p_t, m_t), EWMA λ, threshold γ, N_{adapt}, K, η, λ_{reg}
2: $\theta \leftarrow \theta^*$; $s \leftarrow 0$; residual window $R \leftarrow []$
3: **for** each time t **do**
4:     Predict $(\hat{P}_{t+1:t+H}, \hat{S}_{t+1:t+H}) = f_\theta(X_t)$
5:     Observe p_t, compute $r_t = \|p_t - \hat{p}_t\|_2$
6:     Update $s \leftarrow \lambda r_t + (1 - \lambda)s$
7:     Append r_t to R, estimate μ, σ
8:     **if** $s > \mu + \gamma\sigma$ **then**
9:         $D_{\text{adapt}} \leftarrow$ last N_{adapt} samples
10:         **for** $k = 1$ to K **do**
11:             Compute $\mathcal{L}_{\text{adapt}}$ on D_{adapt}
12:             $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_{\text{adapt}} + \lambda_{\text{reg}} \|\theta - \theta^*\|_2^2)$
13:         **end for**
14:     **end if**
15: **end for**

A. Mobility Traces: GeoLife Dataset

We obtain real mobility trajectories from the GeoLife GPS Trajectory Dataset [15], which contains long-duration recordings of pedestrian, bicycle, and vehicle movement collected across Beijing. The dataset offers high temporal resolution (1–5 seconds) and diverse mobility patterns, making it suitable for learning heterogeneous motion behaviors.

Each raw GPS sequence is resampled to a fixed rate of 1 Hz and converted to local planar (UTM) coordinates. We segment long trajectories into windows using a sliding or overlapping time index, and discard segments with missing or noisy GPS values. Speed and heading are computed using finite differences, and used as part of the measurement vector m_t.

GeoLife trajectories are spatially clustered using k-means on coordinates or via fixed geographic tiles (e.g., 500 m × 500 m). Each cluster or tile is treated as a distinct meta-task for the Reptile training framework.

B. Radio Feature Generation Using DeepMIMO

To augment mobility traces with radio-layer information, we use DeepMIMO [16], a ray-traced dataset generator based on Remcom Wireless InSite. DeepMIMO provides realistic radio

features for arbitrary transmitter–receiver geometries in urban scenarios.

For each position sample p_t, we query the following DeepMIMO-generated values relative to all candidate base stations in the scenario:

- Reference Signal Received Power (RSRP).
- Serving beam index (highest-gain beam).
- Optional additional features (path loss, angle of arrival), though not used in this study.

The measurement vector is then defined as

$$m_t = [\text{RSRP}_t, \ \text{BeamIndex}_t, \ \text{Speed}_t, \ \text{Heading}_t].$$

The serving cell ID at time t is set as the base station with maximum predicted RSRP. We compute handover labels by observing cell transitions over time.

C. Task Construction for Meta-Learning

To support meta-learning across heterogeneous environments, we partition the training region into spatial tasks. Each task T_i corresponds to one city zone, defined either by:

- **Fixed geo-tiles:** Equal-sized spatial grids (e.g., 0.5–1 km tiles), or
- **Density-aware clusters:** K-means clustering in latitude–longitude space.

For each task, we aggregate all trajectory segments whose positions fall predominantly within the corresponding zone. This yields tasks with distinct mobility statistics, radio conditions, and movement patterns.

During meta-training, tasks are split into:

- **Meta-train:** Used for Reptile updates.
- **Meta-validation:** Used for tuning inner-loop and meta-parameters.
- **Meta-test:** Unseen zones used exclusively for zero-shot and few-shot evaluation.

This setup ensures that deployment conditions differ from training environments, making the problem setting closer to real cellular networks.

D. Drift Scenarios for Stress Testing

To evaluate robustness under non-stationary mobility, we construct two drift scenarios:

a) Sudden Turn:: Trajectory segments containing sharp direction changes (greater than $45°$ within 1–2 seconds). These represent road intersections or abrupt user behavior.

b) Speed Shift:: Periods where speed changes rapidly (e.g., walking $\rightarrow$ jogging or braking $\rightarrow$ acceleration). These introduce abrupt variations in future motion distributions.

Drift labels are not used during training; they serve only to define evaluation intervals where the model's robustness and adaptation speed are analyzed.

E. Data Splits and Preprocessing

The combined dataset is split into:

- **Training tasks:** 60% of zones
- **Validation tasks:** 20%
- **Unseen test tasks:** 20%

For each sample:

- Inputs and outputs are normalized (mean and variance computed per task).
- Position coordinates are standardized relative to the tile center.
- Speed and heading are smoothed using a small temporal median filter.
- RSRP values are normalized per scenario to reduce hardware-specific artifacts.

All data preparation pipelines are made deterministic and reproducible using fixed seeds.

VI. EXPERIMENTAL SETUP

We evaluate the proposed framework on the hybrid mobility–radio dataset described in Section V. Experiments are designed to assess (i) zero-shot generalization to unseen city zones, (ii) few-shot adaptation performance, (iii) robustness under drift, and (iv) downstream improvements in proactive handover prediction. This section details the baselines, training configuration, evaluation protocol, and implementation specifics.

A. Baselines

We compare our approach against a set of classical, non-adaptive, and adaptive baselines commonly used in trajectory and mobility forecasting:

a) Constant Velocity (CV):: A kinematic model that extrapolates future positions assuming constant velocity.

b) Kalman Filter (KF):: A first-order constant-velocity KF using position and inferred velocity. This serves as a lightweight classical baseline.

c) Offline GRU:: A GRU-based predictor trained on pooled data without meta-learning or online adaptation.

d) Sliding-Window Fine-Tuning:: The offline GRU updated using the most recent N samples at every timestep without drift detection. This baseline measures the effect of naïve adaptation.

e) MAML-like Meta-Learning (First-Order):: A first-order MAML baseline to compare with Reptile under similar adaptation budgets.

For handover prediction, we include a threshold-based RSRP heuristic that triggers HO based on serving-cell RSRP crossings.

B. Hyperparameters

Model

- GRU hidden size: 64
- MLP heads: 2 layers $\times$ 64 units
- Prediction horizon $H = 3$

Meta-Learning (Reptile)

- Inner-loop steps $S = 5$
- Inner-loop learning rate $\alpha_{\text{inner}} = 1 \times 10^{-2}$
- Meta step size $\beta = 0.1$
- Support set size: 10 samples
- Query set size: 20 samples

Online Adaptation

- Adaptation buffer: $N_{\text{adapt}} = 10$
- Adaptation steps: $K = 5$
- Learning rate $\alpha_{\text{adapt}} = 1 \times 10^{-3}$
- Regularizer $\lambda_{\text{reg}} = 1 \times 10^{-4}$

Drift Detection (EWMA)

- EWMA decay $\lambda = 0.2$
- Residual window length: 100 samples
- Drift threshold: $\gamma = 2$

C. Evaluation Protocol

We perform:

- Zero-shot evaluation (no adaptation).
- Few-shot adaptation (1, 5, 10, 20 samples).
- Drift stress tests and recovery measurement.
- Downstream HO decision evaluation using predicted positions.

All experiments report mean ± standard deviation across 5 seeds where applicable.

D. Implementation Details

The model is implemented in PyTorch and trained on an NVIDIA RTX 3090 GPU. Meta-training takes approximately 2–3 hours for 1000 meta-iterations. Inference runs at $>200\,\text{Hz}$ on a CPU-only setting, making the method suitable for edge deployment. Drift detection operations are constant-time and lightweight. All experiments use fixed random seeds for reproducibility.

VII. RESULTS

We present zero-shot trajectory results, few-shot adaptation, drift recovery under two stress events, and downstream handover metrics. All reported values are averages over 5 random seeds.

A. Zero-shot trajectory prediction

Table I shows ADE and FDE for the zero-shot (no adaptation) setting.

TABLE I
ZERO-SHOT TRAJECTORY PREDICTION (ADE / FDE IN METERS).

Method	ADE (m)	FDE (m)
Constant Velocity	7.42	12.20
Kalman Filter	6.85	11.10
Offline GRU	5.38	8.92
First-Order MAML	4.81	8.21
Ours (Reptile + GRU)	**4.46**	**7.79**

TABLE II
FEW-SHOT ADE (M) VS. NUMBER OF ADAPTATION SAMPLES.

Method	0-shot	1-shot	5-shot	10-shot	20-shot
Offline GRU	5.38	5.12	4.85	4.67	4.59
Ours (Reptile + EWMA)	**4.46**	**4.12**	**3.82**	**3.71**	**3.65**

B. Few-shot adaptation

Table II reports ADE after a small number of adaptation samples. The proposed method consistently outperforms the offline GRU and shows rapid gains with few shots.

C. Drift recovery

We measured recovery after two types of abrupt drift: sudden turn (greater than 45°) and speed shift (greater than 50% change). Tables III and IV report ADE measured at various steps after the drift event and the number of steps required to recover to within 80% of pre-drift accuracy.

TABLE III
DRIFT RECOVERY — SUDDEN TURN (ADE IN METERS).

Step After Drift	Offline GRU	Ours
Pre-drift	4.90	**4.30**
Immediately after	7.85	**6.22**
After 5 steps	7.10	**5.11**
After 10 steps	6.25	**4.55**
After 20 steps	5.48	**4.38**
Recovery time (to 80%)	28 steps	**11 steps**

TABLE IV
DRIFT RECOVERY — SPEED SHIFT (ADE IN METERS).

Step After Drift	Offline GRU	Ours
Pre-drift	4.75	**4.20**
Immediately after	6.92	**5.82**
After 5 steps	6.20	**5.10**
After 10 steps	5.72	**4.61**
After 20 steps	5.31	**4.39**
Recovery time (to 80%)	22 steps	**9 steps**

D. Handover prediction

Using predicted trajectories to infer serving cells and predict HO events yields improved HO metrics compared to RSRP-only heuristics and an offline GRU HO classifier. Table V summarizes these results.

TABLE V
HANDOVER PREDICTION PERFORMANCE.

Model	F1	AUROC	Missed-HO (%)	Ping-pong (%)
RSRP Threshold	0.71	0.80	19.8	11.4
Offline GRU HO Classifier	0.76	0.86	14.1	9.7
Ours (Trajectory $\rightarrow$ HO)	**0.83**	**0.90**	**9.4**	**6.8**

E. Efficiency and complexity

Table VI reports model size and inference cost. The proposed GRU-based model remains lightweight and suitable for edge deployment.

TABLE VI
MODEL EFFICIENCY AND ADAPTATION COST.

Model	Params	CPU latency	Adaptation cost
Offline GRU	128k	2.8 ms	—
First-Order MAML	140k	3.0 ms	High
Ours	**128k**	**2.9 ms**	**5e-3 s / adapt**

VIII. DISCUSSION AND LIMITATIONS

The experimental results show that a lightweight GRU combined with Reptile meta-initialization and EWMA-triggered compact updates provides meaningful improvements in short-term trajectory forecasting and downstream HO prediction. Few-shot adaptation yields large gains with only a handful of samples (notably between 1–10 shots), and EWMA-triggered updates allow rapid recovery under abrupt drift events while avoiding unnecessary adaptation during steady-state operation.

Limitations remain. The hybrid GeoLife + DeepMIMO pipeline abstracts away several real RAN effects (GNSS outages, vendor hysteresis, scheduling delays) and therefore may not capture all field idiosyncrasies. The EWMA detector, while simple and effective, can be improved with more sophisticated drift tests (CUSUM, GLR) or model-based uncertainty estimates. Finally, richer model families (graph or attention-based) may further improve accuracy in dense urban scenes but at the cost of increased latency and adaptation complexity.

IX. CONCLUSION

We presented a meta-continual mobility forecasting framework that combines a GRU base predictor, Reptile meta-initialization for fast few-shot adaptation, and an EWMA residual detector to trigger compact online updates. Evaluated on a reproducible GeoLife + DeepMIMO pipeline, our method achieves an ADE of 4.46 m and FDE of 7.79 m in zero-shot settings, improves few-shot ADE to 3.71 m at 10-shot, recovers from abrupt drift 2–3× faster than an offline GRU, and delivers downstream handover gains (F1 = 0.83, AUROC = 0.90) while remaining lightweight (128k parameters). These results suggest that meta-continual adaptation is a practical path toward robust, sample-efficient mobility forecasting for proactive handover in next-generation cellular networks.

REFERENCES

[1] X. Zhao, A. Zhang, and L. Liu, "T-STAN: Transformer-Based Spatio-Temporal Attention Network for Trajectory Prediction," *IEEE Trans. Intelligent Transportation Systems*, 2022.

[2] Y. Sun, Z. Deng, X. Ding, and Y. Hu, "Deep Learning-Based Vehicle Trajectory Prediction: A Survey," *IEEE Trans. Intelligent Transportation Systems*, 2022.

[3] K. Messaoud, C. Frémont, and S. Lecoeuche, "Trajectory Prediction Using Deep Recurrent and Convolutional Networks," *Neurocomputing*, vol. 423, 2021.

[4] Y. Li, C. Guo, and F. Yang, "Adaptive Vehicle Trajectory Prediction under Uncertain Motion Patterns," *IEEE Trans. Intelligent Transportation Systems*, 2023.

[5] S. Zhang, H. Liu, and X. Chen, "Machine Learning for Handover Management in 5G and Beyond Networks: A Survey," *IEEE Commun. Surveys & Tutorials*, 2023.

[6] O. Ghazal, M. Elhattab, and A. Yassine, "Deep Learning-Based Handover Prediction in 5G Networks," *IEEE Access*, 2021.

[7] J. Pang, X. Zhao, and Y. Lu, "Proactive Handover in Ultra-Dense Networks via Deep Learning," *IEEE Trans. Wireless Commun.*, 2020.

[8] R. Wang, T. Zhang, and L. Huang, "Handover Failure Prediction Using LSTM Networks," in *Proc. IEEE ICC Workshops*, 2022.

[9] M. Alrabeiah and A. Alkhateeb, "Deep Learning for mmWave Beam Prediction Using Channel Data," *IEEE Trans. Wireless Commun.*, 2020.

[10] H. Jiang, G. Gui, and J. Zhang, "Meta-Learning for Fast Beam Prediction in Mobile mmWave Networks," *IEEE Trans. Wireless Commun.*, 2022.

[11] K. Lam, J. S. Ng, and B. C. Ooi, "Few-Shot Learning for Wireless Communications via Meta-Learning," in *Proc. IEEE GLOBECOM*, 2021.

[12] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Comput. Surveys*, 2021.

[13] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning Under Concept Drift: A Review," *IEEE Trans. Knowledge Data Eng.*, 2019.

[14] J. Klink, S. L. Smith, and S. Arora, "Online Continual Learning in Neural Networks: A Survey," arXiv:2307.13219, 2023.

[15] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma, "GeoLife GPS Trajectory Dataset: Understanding Human Mobility Patterns," in *Proc. ACM UbiComp*, 2010.

[16] A. Alkhateeb, "DeepMIMO: A Generic Dataset for mmWave and Massive MIMO Applications," *IEEE Trans. Wireless Commun.*, 2019.