PerFL: A Personalized Privacy-Enhanced Federated Learning Approach for Vehicle Dynamics Prediction

PerFL: A Personalized Privacy-Enhanced Federated Learning Approach for Vehicle Dynamics Prediction

Wenjun Zhang, Boyu Wang, Rasolondrazanaka Ryzzyie Vickene, Chao Zhu*, Xiaoli Liu, Sasu Tarkoma

***Abstract*—Deep learning has significantly improved vehicular dynamics prediction such as travel speed, acceleration, and heading. Centralized training, however, raises concerns about data privacy and communication cost. Federated Learning (FL) addresses these by transmitting model updates instead of raw data, but the updates themselves can still leak sensitive personal information. Local Differential Privacy (LDP) offers a complementary defense through calibrated noise injection, yet applying a uniform LDP budget across drivers ignores their heterogeneous privacy needs and degrades prediction accuracy unnecessarily. In this paper, we propose PerFL, a personalized privacy-preserving vehicle dynamics prediction method that introduces LDP into FL with a per-driver budget tailored to individual privacy sensitivity. Motivated by a questionnaire study revealing that drivers' privacy preferences correlate with observable mobility attributes, PerFL derives each vehicle's sensitivity from passively observable mobility statistics. The framework pairs a Gated Recurrent Unit (GRU) for trajectory prediction with a Double Deep Q-Network (DDQN) for per-round privacy-budget allocation, and aggregates the resulting perturbed updates via an SNR-aware FedAvg that down-weights noisier contributions. Comprehensive experiments on a real-world public-transit trajectory dataset from Helsinki show that PerFL Pareto-dominates uniform-DP and random-noise FL baselines, attaining lower per-tier prediction error while consuming under half of the cumulative privacy budget.**

***Index Terms*—Vehicle Dynamics Prediction, Privacy Protection, Local Differential Privacy, Gated Recurrent Unit, Double Deep Q-Network**

I. Introduction

VEHICLE dynamics prediction is a core technique for developing intelligent transportation systems. Forecasting vehicle dynamics, such as remaining battery capacity and projected driving speed, enables various stakeholders (e.g., map providers, car owners, and intelligent automobile manufacturers) to make informed decisions and optimize their travel paths [1]. In recent years, the application of machine learning and deep learning methods has made significant strides in the domain of vehicle dynamics prediction. To obtain a precise model, the conventional deep learning approach gathers raw data from vehicles as much as possible and centrally processes it in cloud-based systems. However, this practice has raised valid concerns regarding the huge communication costs. In addition, gathering comprehensive data from individual vehicles commonly involves sensitive information, e.g., speed, location, and driving mannerisms, which raises privacy concerns from the driver's side.

In response to these challenges, Federated Learning (FL) was introduced first in 2016 by Google [2]. Instead of sharing the raw data to a central server, FL enables each client to retain their data locally, conduct individual model training using their local data, and transmit only the trained model or updates to the server. The advent of FL offers a compelling alternative to conventional machine learning methods in vehicle dynamics prediction. By allowing raw data from onboard sensors to be processed directly on individual vehicles, FL could notably reduce communication costs and enhance data privacy [3].

Nevertheless, FL still faces two data privacy challenges. The first is model update leakage, and the second is Membership Inference Attacks (MIA) [4]. Model update leakage jeopardizes FL security by exploiting flaws in transferring model gradients or weights between the client and the central server [5]. For example, the attacker can recover the private local data (which are images in the reference work) of each client in the FL system using model weights and loss functions [6]. Furthermore, in Natural Language Processing (NLP), by inverting the real-valued vector representations (words, sentences, user activities), an attacker can reveal phrases that have been used in other participants' training sets [7], [8]. In MIA, the attacker can infer the presence of specific data records within the training data used by the models [4]. Previous work has shown that an attacker can determine whether a specific genome is present in a dataset by using publicly available statistics about the genomics dataset [9]. Note that, in vehicle dynamics prediction, the aggregate location time series has the potential to facilitate the extraction of supplementary driver data, encompassing details like trajectories and mobility profiles [10]. Therefore, the aforementioned attacks directly impacted vehicle dynamics since an attacker can extract driving patterns from model updates and identify if a particular vehicle's data was used in the model's training.

In response to the privacy concerns mentioned above, researchers have suggested enhancing the security of FL with complementary techniques such as Secure Multi-Party Computation (SMC) [11], Homomorphic Encryption (HE) [12], and Differential Privacy (DP) [13]. SMC and HE are cryptography-based approaches that typically necessitate the development of advanced encryption computing protocols in the raw data, such as encrypting the data with sophisticated algorithms and protocols. However, the mechanism of iterative communication and transmission of encrypted data between nodes results in significant communication and computation costs [14], [15].

This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFC3306900, and in part by NordForsk Nordic University Cooperation on Edge Intelligence (Grant No. 168043). (Chao Zhu* is the corresponding author: chao@bit.edu.cn)

W. Zhang, X. Liu and S. Tarkoma are with the Department of Computer Science, University of Helsinki, Finland. X. Liu is also with the School of Technology, University of Vaasa, Finland.

C. Zhu*, B. Wang, and R. Rasolondrazanaka are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, China.

In comparison, DP is a more cost-effective privacy protection method that adds random noise to the data, ensuring individual privacy in scenarios like location-based services. To further prioritize local privacy, a more decentralized DP method called Local Differential Privacy (LDP) [16], [17] has been proposed. LDP ensures that each data contributor locally adds a layer of noise to its model update before uploading it. The absence of intricate encryption and decryption operations and centralized computations in LDP leads to a more lightweight computational process. Meanwhile, considering FL and LDP both possess inherent local awareness, combining them into distributed vehicular scenarios is a natural decision [18].

Although LDP offers a degree of user privacy protection, its strategy of adding noise to counteract attackers can adversely affect the accuracy of predictive models. Furthermore, in real life, the sensitivity toward geographic location privacy varies among different vehicle types and drivers. For example, private car owners typically exhibit greater concern for their location privacy compared to drivers of transport trucks or taxis.

Given the diverse privacy sensitivities among drivers, applying a uniform noise level would lead to excessive degradation of the aggregated model's performance. To safeguard driver privacy while preserving the precision of predictive models, we propose PerFL, a personalized privacy-preserving vehicle dynamics prediction method, aiming at enhancing drivers' trajectory-level privacy protection by introducing LDP into FL. Specifically, PerFL applies a per-driver privacy budget tailoring strategy within an FL+LDP framework, where drivers with varying levels of privacy sensitivity have different noise levels added to their local model updates prior to aggregation. This approach aims to meet the personalized privacy protection needs of different drivers while minimizing the negative impact of privacy protection on the model's performance. To address the trade-off between privacy assurance and model accuracy, we develop a novel FL framework integrating a Gated Recurrent Unit (GRU) for vehicle dynamics prediction with a Double Deep Q-Network (DDQN) for privacy-budget allocation, which could judiciously tailor an appropriate privacy budget to align with individual drivers' specific privacy preferences. To assess the resilience and performance of PerFL, we construct a series of comprehensive simulations on a real-world trajectory dataset in Helsinki. The results demonstrate that PerFL surpasses benchmark methods in privacy assurance and prediction accuracy across diverse settings.

Our contributions are outlined below:

- We conduct a questionnaire showing that drivers' privacy sensitivity is heterogeneous and systematically correlated with observable attributes such as gender, occupation, and driving frequency, motivating the use of mobility statistics as a deployment-time proxy for sensitivity.
- We propose PerFL, a method that melds LDP with FL for privacy-conscious vehicle dynamics prediction. Specifically, we craft a unique architecture that pairs a GRU predictor with a DDQN-based privacy-budget allocator, enabling the system to discern and apply the optimal privacy budget tailor strategy while respecting the varying privacy sensitivities of vehicles.
- Comprehensive simulations on a real-world public-transit trajectory dataset from Helsinki show that PerFL Pareto-dominates uniform-DP and random-noise FL baselines. It attains lower per-tier prediction error while consuming under half of the cumulative privacy budget, and yields a 3–4$\times$ tighter formal (ϵ, δ)-LDP composition bound.

The rest of the paper is organized as follows. The background and related work on vehicle dynamics prediction and its privacy preservation are introduced in Section II. A questionnaire on drivers' privacy sensitivity toward both location and speed sharing is presented in Section III. We then present PerFL's system overview in Section IV and state the problem formulation in Section V, before describing PerFL's methodology and learning algorithms in Section VI. In Section VII, we carry out extensive empirical evaluations to validate PerFL and compare it against existing reference methods. Finally, Section VIII draws our conclusions and discusses future work.

II. Related Work

A. Vehicle Dynamics Prediction

In recent years, extensive research has focused on vehicle dynamics prediction, particularly with the advent of real-time data processing. These advancements have necessitated the development of sophisticated approaches to improve prediction accuracy for real-time vehicle dynamics.

Yim*et al.* [19] proposed a Recurrent Neural Network (RNN) model using real-time measurements for vehicle dynamics prediction. Their two-stage hybrid learning approach, combining open-loop and closed-loop training, enhances long-term prediction accuracy. Jiang *et al.* [20] developed a two-level data-driven vehicle speed prediction model for vehicular networks. First, they use neural networks to predict traffic speed on road segments based on historical data. Then, hidden Markov models (HMMs) are employed to capture the relationship between individual vehicle speeds and predicted traffic speeds, incorporating stochastic driver behaviors. Zhao *et al.* [21] proposed a Deep Belief Network (DBN) model for predicting vehicle speed and steering angle using real-world driving data, leveraging surrounding vehicle information and previous states. Cheng *et al.* [22] explored a big data-driven deep learning model for vehicle speed prediction using an adaptive neuro-fuzzy inference system. By considering factors like driver behavior, route, traffic, and weather conditions, their model effectively predicts vehicle speed in both urban and freeway scenarios, outperforming traditional methods. Shin *et al.* [23] developed a vehicle speed prediction model using a Markov chain with speed constraints. Their approach integrates road geometry and speed limits into the prediction algorithm, enhancing accuracy in real-time driving scenarios. Qian *et al.* [24] introduced a two-level adaptive Kalman filtering algorithm that integrates an auto-regressive moving average model to predict vehicle acceleration, speed, and location, quantifying vehicle location into direction and lane and improving traffic safety at intersections.

Collectively, these studies highlight the essential role of advanced vehicle dynamics prediction models in enhancing

the accuracy and reliability of vehicular network performance. As algorithms drive innovations in connected vehicles, the demand for predictive models that can adapt to dynamic driving environments becomes increasingly vital. These models, however, follow a centralized training paradigm in which vehicle data is aggregated at a central server. As IoVs scale up and on-board data becomes increasingly sensitive, such paradigms raise both communication and privacy concerns. This motivates the recent shift toward federated and privacy-preserving learning paradigms in vehicular environments.

B. Privacy Preservation in Vehicular Environment

The demand for privacy preservation in vehicular environments is growing rapidly as the exchange of sensitive data in the Internet of Vehicles (IoVs) is increasing. Efficient, reliable, and secure methods for protecting user privacy are essential in maintaining trust and participation within these networks.

Zhou *et al.* [25] provided an adaptive pseudonym-changing scheme that uses dynamic pseudonyms and a gamma distribution to model privacy leakage. Their approach adapts pseudonym changes based on real-time data, effectively protecting vehicle privacy without sacrificing location accuracy. Kong *et al.* [26] proposed a blockchain-based privacy-preserving data collection and sharing scheme for vehicular fog networks. Using homomorphic signcryption and a permissioned blockchain, their model ensures location privacy and verifiable data aggregation. Yu *et al.* [27] introduced a decentralized algorithm with DP for privacy preservation in vehicular networks, demonstrating effective privacy protection in decentralized environments. Wang *et al.* [28] proposed a blockchain-based privacy-preserving FL scheme for IoVs, using homomorphic encryption and secure model aggregation. Batool *et al.* [29] focused on a privacy-preserving infrastructure for IoVs using FL with LDP. Their model enhances privacy by applying LDP at the vehicle level, protecting against gradient leakage and inference attacks while improving data sharing efficiency.

Existing FL with LDP works in vehicular networks, exemplified by Batool *et al.* [29], apply a uniform privacy budget to every vehicle. This one-size-fits-all setting ignores the heterogeneity of driver privacy preferences, and results in either under-protection of sensitive drivers or unnecessary accuracy loss for the rest. To address this gap, we propose PerFL, the first FL with LDP framework in vehicular networks that personalizes per-driver privacy budgets. By tailoring the budget to each driver's sensitivity, PerFL balances prediction accuracy against the diversity of drivers' privacy concerns.

III. Drivers' Privacy Sensitivity Questionnaire

To explore the diversity in drivers' privacy sensitivity, we designed and conducted a questionnaire to gauge and understand individuals' privacy concerns and preferences regarding the sharing of their geographic location and speed. The questionnaire involved 56 participants from different countries, genders, and occupations. We gathered diverse data categories encompassing gender (male, female), country (China, other countries), occupations (student, other occupations), driving

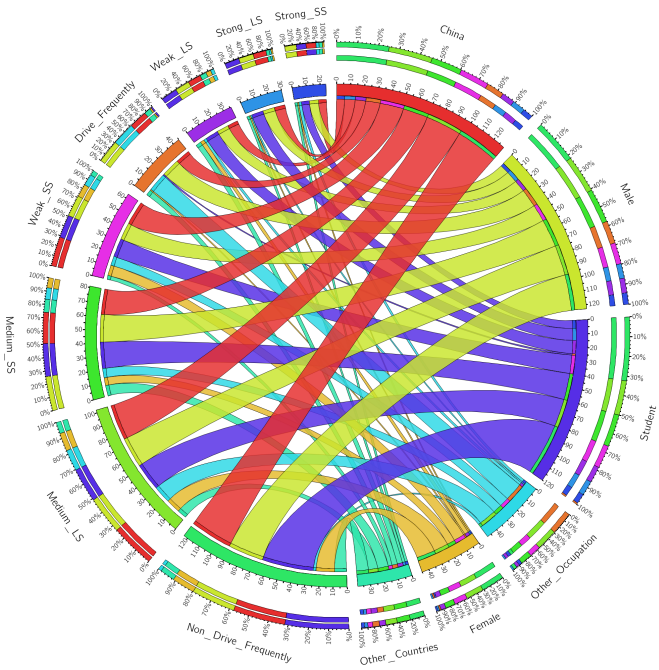

Fig. 1: Driver privacy sensitivity questionnaire results ($N = 56$). The outer ring labels each category's marginal share of all participants. Chord widths represent the joint count of respondents falling into two categories. For example, the chord connecting "Male" and "Strong_LS" has width 9, indicating that 9 of the 56 respondents are male and strongly location-sensitive. All percentages quoted in the main text are derived from these joint counts. Marginal percentages use 56 as the denominator, while conditional percentages use the relevant subgroup size, as indicated in context. Different colors distinguish the relationships between outer categories.

frequency, location sensitivity(LS), and speed sensitivity(SS), shown in Fig. 1. The latter two categories are pivotal in assessing drivers' privacy concerns, which we classify into three distinct levels. *Strong* means the driver is highly protective of their privacy and reluctant to share the data, *Medium* indicates a moderate, situation-dependent willingness to share, and *Weak* reflects little concern about disclosing the data.

As illustrated in Fig. 1, individuals from different gender groups (i.e., 73.21% male and 26.79% female) have distinct preferences and sensitivities regarding location and speed.

For location sensitivity, our analysis delves deeper into how individuals perceive and approach location-sharing. The findings reveal that 17.86% of drivers express strict concerns regarding sharing their precise location. Among them, 90% are male and 10% are female, showing a much larger gender disparity compared to the overall gender distribution (i.e., 73.21% vs 26.79%) of all participants. This group values their privacy and is cautious about sharing location data. On the other hand, most drivers, approximately 60.71%, maintain a more relaxed stance regarding location privacy. They are willing to share location data to some extent. Lastly, an additional 21.43% of drivers have no issues giving their location information.

Regarding driving speed sensitivity, 14.29% of drivers were found to be extremely sensitive to sharing their speed data,

highlighting strict privacy requirements. Within this group, males accounted for 87.5%, while females represented only 12.5%, indicating a stronger gender imbalance compared with the overall participant distribution. Meanwhile, 48.21% of participants demonstrated a more flexible attitude toward sharing driving speed data, suggesting relatively lower privacy concerns. In addition, 37.5% of drivers reported no concern about disclosing their driving speed information, reflecting a general willingness to share this type of data.

Privacy preference is also reflected in driving behavior and occupation. Only 6.7% of frequent drivers express strong concern over location sharing, compared with 22.0% of infrequent drivers; conversely, among the respondents most concerned about location privacy, nearly all (90%) are infrequent drivers, well above the 73.2% infrequent share of the sample overall. A similar pattern appears across occupation: 22.5% of students express strong location concern, against 6.3% of other occupations, and students account for 90% of the most location-sensitive respondents, well above their 71.4% share.

Taken together, these results demonstrate that privacy sensitivity varies substantially across drivers and exhibits clear behavioral patterns rather than random variation. Consequently, applying a single uniform privacy budget to all clients, as commonly assumed in conventional DP-FL, is fundamentally mismatched to the population. Such a strategy injects unnecessary noise into the updates of drivers with low privacy concern, thereby degrading model utility, while still providing insufficient protection for highly privacy-sensitive individuals.

An effective privacy-preserving system should therefore adapt the privacy budget to each driver's individual sensitivity. Such personalization, however, cannot depend on collecting demographic attributes during deployment, as this would itself introduce additional privacy risks and would not scale beyond voluntary survey participants. PerFL therefore infers each vehicle's privacy sensitivity from passively observable mobility statistics generated during normal operation, whose construction is detailed in Section V.

IV. System Overview

A. Related Terms

Participant Vehicles: In PerFL, *Participant Vehicles* (PVs) refer to all the vehicles across the city that participate in vehicular dynamic prediction. Each PV is equipped with cellular connectivity for communication and with on-device compute sufficient to (i) train its local GRU on its private trajectory data and (ii) run inference of the lightweight DDQN policy that decides its per-round privacy expenditure.

Privacy Sensitivity and Privacy Budgets: Drivers have heterogeneous privacy needs. Findings from our questionnaire study (Section III) suggest that demographic and behavioral factors such as gender and driving habits correlate with these needs, motivating a per-PV *privacy sensitivity* score. In practice, however, no demographic information is collected at deployment time; instead, the sensitivity score is computed from observable mobility statistics (e.g., data volume, geographic span, line and operator diversity, speed variability) that act as data-driven proxies for the latent privacy need. The sensitivity score in turn determines two complementary quantities: a *total privacy budget* that caps each PV's cumulative differential-privacy expenditure across an entire FL session, and a *per-round privacy budget* chosen by the on-device DDQN policy at each round, constrained by what remains of the total. A larger per-round budget injects less noise (looser privacy, better accuracy) and vice versa. Decoupling these two levels lets PerFL allocate budget adaptively across rounds while still honouring the session-level privacy guarantee; the formal definitions and budget formulas are deferred to Section V.

Service Zones: Given the local relevance of the city traffic, it is undesirable to train and use a single model for predicting the vehicular dynamics for the entire city. On the other hand, modern cities are fully covered by cellular networks and divided into multiple zones based on the coverage of base station towers. Therefore, we divide the city into multiple service zones and execute FL independently within each zone.

Base Transceiver Stations: Within a specific service zone, the *Base Transceiver Station* (BTS) located in the center of the area coordinates all in-zone PVs. Unlike centralized learning, where each vehicle's raw trajectory would be transmitted to a server, in PerFL only perturbed model updates leave the PV: the BTS itself does not access raw trajectories or unperturbed local gradients. Its responsibilities are limited to (i) broadcasting the initial global model together with the offline-trained DDQN policy and the DP/SNR hyperparameters at session start, (ii) collecting each round's perturbed local updates from active PVs and aggregating them via SNR-aware FedAvg over the active subset, and (iii) broadcasting the refreshed global model and a scalar global-accuracy summary used by PVs in the next round's state. Per-round budget tailoring itself happens on each PV via local DDQN inference, so the BTS never inspects any PV's private state.

B. Workflow of PerFL

Fig. 2 illustrates the workflow of PerFL in a specific service zone. The DDQN policy that controls each PV's per-round privacy budget is trained offline (Section VI) and deployed in a frozen form. The online workflow proceeds in five steps:

1) *Model Distribution.* The BTS randomly initializes a global model and broadcasts a query to ascertain the presence of all vehicles within its service zone. Each willing vehicle (PV) acknowledges, and the BTS distributes the initial global model, the offline-trained DDQN policy, and the DP and SNR-aggregation hyperparameters to all participating PVs.
2) *Local Model Training.* When PVs receive the model from the BTS, they start training it locally on their own private dataset. The dataset comprises the temporal trajectory parameters of the onboard units, such as travelling speed, heading, geographic increments, and inter-observation intervals. Before each round's training, each PV runs the DDQN policy to choose a per-round privacy budget that balances expected accuracy gain against privacy spend. A PV whose remaining budget has been exhausted abstains from the round and waits for the next refresh.
3) *Noise Addition.* To safeguard data privacy, each active PV perturbs its local model update by first clipping it

to a bounded sensitivity and then adding Gaussian noise whose scale is inversely proportional to the per-round budget chosen by the DDQN. The perturbed update, together with a single scalar summarizing the PV's local accuracy, is uploaded to the BTS. Importantly, no raw trajectory data, no unperturbed model updates, and no demographic attributes leave the device.

4) *Model Aggregation.* Once perturbed updates have been collected from all active PVs, the BTS aggregates them into a refreshed global model. PerFL replaces the standard equal-weight FedAvg [2] with an SNR-aware weighted aggregation that down-weights noisier (lower-budget) contributions, so that a few heavily-noised updates from privacy-sensitive PVs do not derail the global signal-to-noise ratio. The personalized per-round budgets themselves are not adjusted by the BTS during deployment; they are decided by each PV's on-device DDQN policy, which delivers the personalization along the privacy and accuracy trade-off without ever exposing private state to the server.
5) *Local Model and Privacy Budget Update.* After aggregation, the refreshed global model and a single scalar summarizing global accuracy are broadcast back to all PVs (active and inactive), so that each PV, including those whose budget is already exhausted, can refresh its locally cached accuracy on the latest global model. Each PV evaluates the new global model on its local test split, updates its cached accuracy and remaining-budget signals, and proceeds to the next round.

Steps 2–5 are repeated until the FL session ends or the PV leaves the current service zone. For identity verification and dynamic membership, each PV transmits an identification beacon alongside its model upload; an unrecognized beacon flags a newcomer, who is initialized as in Step 1 and is excluded from the current round's aggregation but joins from the next round onward.

TABLE I: Summary of symbol notations.

Notation	Description
Participants, data, and models	
v_i, i	Individual PV; index
N	Number of PVs in the service zone
d_i	Local dataset of vehicular dynamics of v_i
$gW^{(t)}$	Global model at FL round t
$oW_i^{(t)}$	Local model of v_i after round-t training
$\Delta_i^{(t)}$	Local update $oW_i^{(t)} - gW^{(t)}$
$\tilde{\Delta}_i^{(t)}$	Perturbed local update after clipping + noise
Vehicle dynamics task	
$x_{t,i}$	Vehicle dynamics of v_i at timestamp t
$X_{t,i}^H$	History of v_i over the last H timestamps
$\tilde{X}_{t,i}^U$	Predicted future of v_i over the next U timestamps
$\mathcal{A}_i$	Accuracy score, $\mathcal{A}_i = \exp(-\mathrm{MAE}_i/M)$
M	MAE normalisation cap
Privacy machinery	
s_i	Privacy sensitivity score of v_i, $s_i \in [1,3]$
C	Privacy-budget calibration constant ($E_i = C/s_i$)
E_i	Total privacy budget of v_i across T rounds
$\epsilon_i^{(t)}$	Per-round privacy budget at round t, $\sum_\tau \epsilon_i^{(\tau)} \le E_i$
U_i	Accumulated privacy spend of v_i, $U_i = \sum_{\tau<t} \epsilon_i^{(\tau)}$
p_i, $\mathcal{P}_i$	Privacy loss $s_i\epsilon_i/C$; privacy score e^{-p_i}
$\mathcal{N}(0,\sigma_i^2)$	Gaussian noise injected by the LDP mechanism
DP-FedAvg	
T, J	FL rounds per session; local SGD epochs per round
S, ζ	DP clip norm; DP noise multiplier
α_w, $w_i^{(t)}$	SNR exponent; aggregation weight
n_i	Number of training samples on v_i
$\mathcal{S}^{(t)}$	Active (non-abstaining) subset of PVs at round t
DDQN policy	
$\mathbf{s}_i^{(t)}$	7-dim DDQN state vector (Eq. 19)
$\mathcal{A}_i^{\mathrm{loc}}$, $\mathcal{A}^{\mathrm{glb}}$	Local / global accuracy summaries used in the state
$a_i^{(t)}$, K	Action index; size of the discrete multiplier set
θ, θ^-	Online / target Q-network weights
C_{tgt}	Target-network update period (in episodes)
$r^{(t)}$, α	Per-round reward; reward weight in $\alpha\mathcal{A}_i + (1-\alpha)\mathcal{P}_i$

V. Problem Formulation

In this section, we introduce the problem formulation of PerFL. As PerFL follows a uniform workflow across all service zones, we illustrate the process using one service zone as an example. Table I summarizes the key notations used throughout the paper, grouped by role.

A. Vehicle Dynamic Prediction

We assume that there are N PVs active in a specific service zone. Let v_i denote an individual PV, and $V = \{v_1, v_2, \ldots, v_N\}$ represent the set of all PVs. We use d_i to denote the dataset of vehicular dynamics (e.g. travel speed, direction, and acceleration) of PV v_i.

Each PV holds a personalized privacy budget that controls how much noise it injects before sharing its model contribution with the BTS; this budget is derived from the PV's privacy sensitivity and is formalized in V-B. We denote the local model of v_i by oW_i and the global model by gW.

We discretize a day (24 h) into timestamps, each of which is of equal duration (in seconds) and is indexed by an integer $t \in \mathbb{Z}^+$. In this context, the real-time vehicular dynamics of v_i at the t timestamp is denoted as $x_{t,i}$. Given the current global model gW broadcast by the BTS, v_i predicts its vehicle dynamics for the next U timestamps from the most recent H timestamps, which we formulate as:

$$\tilde{X}_{t,i}^U = f(X_{t,i}^H; gW), \tag{1}$$

where $X_{t,i}^H = (x_{t-H+1,i}, x_{t-H+2,i}, \ldots, x_{t,i})$ denotes the sequence of vehicle dynamics data from $t-H+1$ to t observed by v_i and $f(\cdot)$ represents the chosen prediction model.

Let $X_{t,i}^U = (x_{t+1,i}, x_{t+2,i}, \ldots, x_{t+U,i})$ denote the true vehicle dynamics of the future time stamps observed by v_i. Then, the prediction performance of the selected model in the task is expressed as:

$$\mathcal{A}_i = f_{\mathcal{A}}(X_{t,i}^U, \tilde{X}_{t,i}^U), \tag{2}$$

where $f_{\mathcal{A}}(\cdot)$ denotes a function that evaluates the similarity between the predicted and true values. Mirroring the smooth

exponential form adopted for the privacy score, we instantiate $f_{\mathcal{A}}(\cdot)$ as

$$\mathcal{A}_i = \exp\big(-\mathrm{MAE}(X^U_{t,i}, \tilde{X}^U_{t,i})/M\big), \tag{3}$$

where $\mathrm{MAE}(\cdot,\cdot)$ is the mean absolute error between the predicted and true sequences and M is a normalisation constant chosen to keep $\mathcal{A}_i \in (0,1]$ across the operating range of the task. This continuous mapping mirrors the privacy score and provides a smooth gradient that the DDQN agent in Section VI relies on for stable learning.

B. Privacy Preservation

1) Privacy Budget: In PerFL, each PV applies an LDP mechanism derived from DP to perturb its local update before sending it to the BTS, so that unperturbed model information never leaves the device. Here are the relevant definitions of DP and its derived LDP:

Definition 1. *(ϵ,δ)-DP [30]: A randomized mechanism $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ,δ)-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$, it holds that*

$$Pr[\mathcal{M}(d) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(d') \in \mathcal{S}] + \delta. \tag{4}$$

However, a more localized approach to privacy is required in scenarios like FL, where participants' data remains decentralized. LDP extends the principles of DP by enabling each participant to independently perturb their data before sharing it, ensuring that privacy is preserved even before data leaves the local device.

Definition 2. *(ϵ)-LDP [16]: A randomized mechanism $\mathcal{A}: \mathcal{X} \to \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ϵ)-local differential privacy if for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and for any two arbitrary input values $v_1, v_2 \in \mathcal{X}$,*

$$Pr[\mathcal{A}(v_1) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{A}(v_2) \in \mathcal{S}] \tag{5}$$

A mechanism $\mathcal{A}$ that satisfies this definition alleviates concerns regarding the potential disclosure of an individual's personal information. In the context of LDP, the Gaussian Mechanism (GM) is a commonly used approach to protect privacy by adding noise to the local data.

Although the classical LDP definition assumes $\delta = 0$, federated-learning deployments commonly adopt the relaxed (ϵ,δ)-LDP form obtained by applying the GM to the shared model updates. This relaxation permits a small failure probability δ in exchange for a better trade-off between privacy and utility, and is the notion used throughout PerFL.

Specifically, given a function $f : \mathcal{X} \to \mathbb{R}$ that operates on local input data $\mathcal{X}$, the GM $\mathcal{A}$ adds noise to the true output $f(\mathcal{X})$, where the noise follows a normal distribution with standard deviation σ, chosen to satisfy the privacy budget ϵ. Mathematically, the GM is defined as:

$$\mathcal{A}(\mathcal{X}) = f(\mathcal{X}) + \mathcal{N}(0, \sigma^2), \tag{6}$$

where the standard deviation σ of the noise is calculated as:

$$\sigma = \frac{\Delta f \sqrt{2\log(1.25/\delta)}}{\epsilon}, \tag{7}$$

and Δf is the sensitivity of the function, which is determined by the specific task being performed, while the privacy budget ϵ governs the amount of noise added.

2) Privacy Requirement: As discussed in Section III, drivers exhibit heterogeneous privacy needs. We capture this heterogeneity through a per-vehicle *privacy sensitivity score* $s_i \in [1,3]$ (continuous; higher values indicate stricter privacy concern). Based on s_i, we define the total privacy budget that v_i may consume across an entire T-round FL session as

$$E_i = \frac{C}{s_i}, \tag{8}$$

where $C > 0$ is a calibration constant. The per-round privacy budget $\epsilon_i^{(t)}$ is selected adaptively by the DDQN policy of Section VI subject to the cumulative cap $\sum_{\tau=0}^{T-1} \epsilon_i^{(\tau)} \leq E_i$, which yields a session-level (ϵ,δ)-LDP guarantee with $\epsilon = E_i$.

The construction of s_i from observable user attributes is implementation-specific. In our experimental setting on the Helsinki public-transport trajectory dataset, we compute s_i detailed in Section VII from observable mobility statistics that act as data-driven proxies for the latent privacy need.

We use *Privacy Loss* to quantify the cumulative DP exposure of vehicle v_i over the FL session. Assuming v_i uses a GM that satisfies (ϵ_i,δ)-LDP per round, the cumulative privacy loss after spending $\epsilon_i = \sum_\tau \epsilon_i^{(\tau)}$ in total is:

$$p_i = \frac{s_i\, \epsilon_i}{C}, \tag{9}$$

where C is the same calibration constant used to define the budget cap $E_i = C/s_i$ in (8). Sensitivity enters the loss directly: for a fixed spend ϵ_i, a higher s_i produces a larger p_i, so a more privacy-sensitive vehicle loses privacy faster per unit of ϵ. Equivalently, $p_i = \epsilon_i/E_i$ is the fraction of the personal contract consumed (the budget utilisation), so s_i shapes *both* the cap E_i and the rate at which the loss accrues against it.

Because p_i is exactly the budget utilisation, it reaches $p_i = 1$ when a vehicle exhausts its contract E_i; staying within budget ($\epsilon_i \leq E_i$) keeps $p_i \leq 1$ for every sensitivity tier, which the per-round cap enforced by the DDQN policy guarantees throughout the session.

Based on the privacy loss, we define the *Privacy Score* as a smooth, bounded measure of how much of a vehicle's privacy contract remains intact, providing a continuous gradient signal that the reinforcement-learning agent can exploit:

$$\mathcal{P}_i = e^{-p_i} = e^{-s_i \epsilon_i / C}, \tag{10}$$

where $\mathcal{P}_i \in (0,1]$ is the privacy score of the participating vehicle. Since $p_i = \epsilon_i/E_i$, this is equivalently $\mathcal{P}_i = e^{-\epsilon_i/E_i}$: at $\epsilon_i = 0$ no budget is consumed and $\mathcal{P}_i = 1$; at the contract cap $\epsilon_i = E_i$, $\mathcal{P}_i = e^{-1} \approx 0.368$ for every vehicle regardless of s_i; beyond the cap the score decays further.

For the same absolute spend ϵ_i, a more sensitive vehicle (larger s_i) receives a lower $\mathcal{P}_i$: an identical disclosure costs more to a user who values privacy more. Personalization in PerFL therefore acts at two coupled levels: sensitivity s_i tightens the contract cap $E_i = C/s_i$ *and* steepens the score decay per unit ϵ, while all vehicles that fully consume their personal contract converge to the same normalized score e^{-1}.

C. Problem Optimization

The primary goal of PerFL is to enable accurate vehicular dynamics prediction while ensuring privacy preservation using LDP. We use $f_g(\cdot)$ to represent the relationship between model performance and privacy. The reward r_i of v_i can be:

$$r_i = f_g(\mathcal{A}_i, \mathcal{P}_i), \tag{11}$$

where $\mathcal{A}_i$ denotes the performance for v_i and $\mathcal{P}_i$ denotes the privacy score for v_i. f_g is a monotonically increasing function.

In PerFL, a negative correlation between performance and privacy is often observed. High performance is often achieved at the expense of user privacy. In other words, it is challenging for both $\mathcal{A}_i$ and $\mathcal{P}_i$ to attain high values simultaneously. To reasonably evaluate the model, a function $f_g(\cdot)$ is designed to address the above-mentioned tradeoff:

$$f_g(\mathcal{A}_i, \mathcal{P}_i) = \alpha \mathcal{A}_i + (1-\alpha)\,\mathcal{P}_i, \tag{12}$$

where α is a coefficient ranging from 0 to 1, representing the importance of performance and privacy in the task. A higher value of α indicates greater emphasis on model performance.

Finally, PerFL is formulated as the problem of choosing the per-round privacy-budget allocation $\{\epsilon_i^{(t)}\}$ that maximizes the average reward across all PVs:

$$\max_{\{\epsilon_i^{(t)}\}} R = \frac{1}{N}\sum_{i=1}^{N} f_g(\mathcal{A}_i, \mathcal{P}_i) \quad \text{s.t.} \quad \sum_{\tau} \epsilon_i^{(\tau)} \le E_i \quad \forall i, \tag{13}$$

where the allocation jointly determines each PV's prediction performance $\mathcal{A}_i$ (through the global model gW it helps train) and privacy score $\mathcal{P}_i$ (through the budget it consumes), and the constraint enforces each PV's session budget cap E_i. PerFL learns this allocation with the DDQN policy of Section VI.

VI. Methodology

In this section, we introduce the detailed design of the PerFL. We maintain an FL model within a service zone, where the PVs act as clients and the BTS serves as the server. Each PV runs a GRU network for vehicle dynamics prediction. We use a GM that satisfies (ϵ, δ)-LDP to protect the privacy of each PV and employ a DDQN to dynamically adjust the privacy budget according to the individualized privacy needs of each PV. Fig. 3 illustrates the framework of our single-service zone model on a single client.

A. GRU-Based FL for Vehicle Dynamics Prediction

In the modern IoVs, vehicles generate vast amounts of sensitive data in real-time, including location information, driving trajectories, and behavior patterns. As data volume grows, traditional centralized learning methods face privacy leakage and calculation overload. To address these issues, we adopt an FL framework, leveraging the computational power and local data of each PV for model training. This decentralized approach ensures data privacy by keeping sensitive data local and distributing the computational load effectively.

Vehicle dynamics prediction relies on continuous time series data, which exhibits strong temporal correlations. Thus, the model must process data at individual time points and capture temporal dependencies and dynamic patterns. To effectively model this timing characteristic, a GRU network is deployed on each PV; GRUs match LSTMs on short-to-medium-horizon sequence tasks while avoiding the extra gating complexity of an LSTM [31]. Specifically, in each FL round t, every PV v_i receives the current global model $gW^{(t)}$ from the BTS and uses it as the initialization of its local training. The PV runs J epochs of mini-batch gradient descent on its local dataset d_i, producing a locally trained model $oW_i^{(t)}$. The single-step update inside this loop is

$$oW_i^{(t)} \leftarrow oW_i^{(t)} - \eta\, \nabla \mathcal{L}\big(oW_i^{(t)}, d_i\big), \tag{14}$$

where η is the learning rate and $\nabla \mathcal{L}\big(oW_i^{(t)}, d_i\big)$ is the gradient of the loss on d_i.

To bound the per-client influence and to add noise calibrated to the personalized privacy budget $\epsilon_i^{(t)}$ supplied by the DDQN policy in Section VI-B, we adopt a DP-FedAvg style update mechanism that operates on the *model update* rather than on the model itself. Concretely, each PV computes the update

$$\Delta_i^{(t)} = oW_i^{(t)} - gW^{(t)}, \tag{15}$$

clips it to an L_2-norm bound S, and adds Gaussian noise:

$$\tilde{\Delta}_i^{(t)} = \Delta_i^{(t)} \cdot \min\Big(1, \tfrac{S}{\|\Delta_i^{(t)}\|_2}\Big) + \mathcal{N}\big(\mathbf{0}, \sigma_i^2 \mathbf{I}\big), \quad \sigma_i = \tfrac{\zeta\, S}{\epsilon_i^{(t)}}, \tag{16}$$

where $\zeta > 0$ is the noise-multiplier constant of the GM. The PV uploads only $\tilde{\Delta}_i^{(t)}$ to the BTS, so the local model never leaves the device.

The BTS aggregates the perturbed updates from the active subset $\mathcal{S}^{(t)} \subseteq \{1, \ldots, N\}$ of PVs that participated in this round (a PV may abstain when its remaining budget is exhausted; see Section VI-B) using *SNR-aware* FedAvg. Clients with smaller $\epsilon_i^{(t)}$ inject more noise and therefore carry a lower signal-to-noise ratio; we down-weight their contribution to the aggregate:

$$w_i^{(t)} = \frac{n_i \cdot \big(\epsilon_i^{(t)}\big)^{\alpha_w}}{\sum_{j \in \mathcal{S}^{(t)}} n_j \cdot \big(\epsilon_j^{(t)}\big)^{\alpha_w}}, \qquad i \in \mathcal{S}^{(t)}, \tag{17}$$

where n_i is the number of training samples on v_i and $\alpha_w \ge 0$ is an SNR exponent ($\alpha_w = 0$ recovers standard sample-size weighting; the value used in our experiments is reported in Section VII). The global model is then updated by

$$gW^{(t+1)} = gW^{(t)} + \sum_{i \in \mathcal{S}^{(t)}} w_i^{(t)} \cdot \tilde{\Delta}_i^{(t)}. \tag{18}$$

The refreshed $gW^{(t+1)}$ is broadcast back to all PVs (including those that abstained from this round, so they may resume in subsequent rounds) and reused for the next round of FL.

B. DDQN-Based Privacy Budget Adjustment

1) Markov Decision Process Formulation: For each PV v_i, the problem of choosing a per-round privacy budget across T FL rounds is naturally cast as a finite-horizon Markov Decision Process (MDP) [32]. Because the post-aggregation accuracy of the global model under DP-noisy updates has no closed form, we treat the MDP as model-free and learn a policy directly

via reinforcement learning. For convenience we let $E_i = C/s_i$ denote v_i's total privacy budget (Section V) and $\bar{E}_i = E_i/T$ its per-round *fair share*. At each round $t \in \{0, 1, \ldots, T-1\}$, the agent observes a state, selects an action that determines $\epsilon_i^{(t)}$, and receives a reward computed once the round is complete.

a) State: A purely demographic state cannot capture how the FL trajectory unfolds round by round. We therefore define a seven-dimensional continuous state vector that summarizes both v_i's privacy posture and the current FL progress:

$$\mathbf{s}_i^{(t)} = \big[\, \bar{s}_i,\ \mathcal{A}_i^{\text{loc}},\ \mathcal{A}^{\text{glb}},\ \Delta\mathcal{A}_i,\ \rho_i,\ \tilde{\epsilon}_i,\ \tilde{t}\,\big], \tag{19}$$

where

- $\bar{s}_i \in [0,1]$ is v_i's normalized continuous sensitivity, derived from the data-driven features in Section V;
- $\mathcal{A}_i^{\text{loc}} = \exp(-\text{MAE}_i/M)$ is the focal-client accuracy, with MAE_i the prediction MAE of the current global model on v_i's test split and M a normalisation cap;
- $\mathcal{A}^{\text{glb}} = \exp(-\text{MAE}^{\text{glb}}/M)$ is the global accuracy;
- $\Delta\mathcal{A}_i = (\mathcal{A}_i^{\text{loc}}(t) - \mathcal{A}_i^{\text{loc}}(t-1) + 1)/2 \in [0,1]$ is the marginal accuracy improvement across rounds;
- $\rho_i = \max\!\big(0,\ (E_i - \sum_{\tau<t} \epsilon_i^{(\tau)})/E_i\big)$ is the remaining-budget ratio;
- $\tilde{\epsilon}_i = \min(\epsilon_i^{(t-1)}/\epsilon_i^{\max},\ 1)$ is the previous round's spend, normalized by the maximum per-round spend $\epsilon_i^{\max}$ allowed by the action space;
- $\tilde{t} = t/T$ is the round-progress indicator.

This state is informative both about the user's privacy needs $(\bar{s}_i,\ \rho_i)$ and about the FL process $(\mathcal{A}_i^{\text{loc}}, \mathcal{A}^{\text{glb}}, \Delta\mathcal{A}_i, \tilde{\epsilon}_i, \tilde{t})$, enabling the agent to condition on the joint accuracy–privacy trajectory rather than on static demographic attributes alone.

b) Action: Treating ϵ_i as a continuous variable couples the action magnitude to the very different absolute budgets of clients with different sensitivity levels and complicates exploration. We discretise the action space into K multipliers

$$\mathcal{K} = \{k_1,\ k_2,\ \ldots,\ k_K\}, \tag{20}$$

of the per-round fair share. Choosing action $a^{(t)}$ assigns

$$\epsilon_i^{(t)} = \min\!\Big(\max\big(k_{a^{(t)}}\,\bar{E}_i,\ \epsilon_{\text{floor}}\big),\ E_i - \sum_{\tau<t} \epsilon_i^{(\tau)}\Big), \tag{21}$$

where ϵ_{floor} is a small lower bound that ensures every active client makes a non-trivial contribution to FL aggregation. The fair-share parameterisation makes the policy invariant to the absolute size of E_i, which differs across clients, while the cap on remaining budget enforces (ϵ, δ)-LDP across the entire T-round episode.

c) Reward: Immediately after each FL round we compute the joint accuracy–privacy reward defined in (12):

$$r_i^{(t)} = \alpha\,\mathcal{A}_i^{(t)} + (1-\alpha)\,\mathcal{P}_i^{(t)}, \tag{22}$$

with the accuracy term taken as $\mathcal{A}_i^{(t)} = \exp(-\text{MAE}_i^{(t)}/M)$, post-aggregation, and the privacy term $\mathcal{P}_i^{(t)}$ evaluated on the cumulative budget consumed up to round t. Both terms are continuous in their inputs, providing a smooth gradient signal for value-function learning.

2) DDQN for Privacy Budget: Because the state contains continuous variables and the joint state space is effectively infinite, table-based Q-learning is infeasible. We approximate the action-value function with a multi-layer perceptron $Q(\mathbf{s}, a; \theta)$ and adopt the DDQN variant [33], which mitigates the systematic over-estimation bias of vanilla DQN by decoupling action selection from value estimation. The Bellman update [34] that drives DDQN is therefore:

$$y = r + \gamma\,(1-d)\,Q\Big(\mathbf{s}',\ \arg\max_{a'} Q(\mathbf{s}', a'; \theta);\ \theta^-\Big), \tag{23}$$

where $d \in \{0,1\}$ flags the terminal transition at the end of the T-round horizon ($d = 1$ disables bootstrapping past the horizon), the online network θ selects the next action, and the target network θ^- evaluates its value. Compared with the vanilla DQN target $y = r + \gamma \max_{a'} Q(\mathbf{s}', a'; \theta^-)$, this decoupling reduces the over-estimation that arises when the same network is used for both selection and evaluation.

Algorithm 1 PerFL: Personalized LDP-FL with DDQN

Input: N PVs with local datasets $\{d_i\}_{i=1}^N$, sensitivity scores and total budgets $\{\bar{s}_i, E_i\}_{i=1}^N$; trained DDQN policy θ from Algorithm 2; FL rounds T; DP clip norm S, noise multiplier ζ, SNR exponent α_w, per-round floor ϵ_{floor}.
Output: final global model gW^*.

Step 1: Initialization. BTS initializes $gW^{(0)}$ randomly and broadcasts $gW^{(0)}$, θ, $\{S, \zeta, C, \alpha_w, \epsilon_{\text{floor}}\}$ to all PVs. Each PV sets $U_i \leftarrow 0$, $\mathcal{A}_i^{\text{loc}} \leftarrow 1$, $\mathcal{A}_i^{\text{loc,prev}} \leftarrow 0$, $\tilde{\epsilon}_i \leftarrow 0$; BTS publishes $\mathcal{A}^{\text{glb}} \leftarrow 1$ as the cold-start value.
for $t = 0, 1, \ldots, T-1$ **do**
  Step 2: Local Decision and Training for $i = 1, 2, \ldots, N$ **in parallel**:
    PV v_i assembles state $\mathbf{s}_i^{(t)}$ via (19);
    PV v_i selects $a_i^{(t)} \leftarrow \arg\max_a Q(\mathbf{s}_i^{(t)}, a; \theta)$;
  if $U_i \geq E_i$ **then**
    PV v_i marks itself *inactive* this round; set $\epsilon_i^{(t)} \leftarrow 0$;
  else
    PV v_i maps $a_i^{(t)}$ to $\epsilon_i^{(t)}$ via (21) and runs J local SGD epochs via (14) to obtain $oW_i^{(t)}$;
  end if
  Step 3: Noise Addition and Upload. Each active PV computes the perturbed update $\tilde{\Delta}_i^{(t)}$ via (16) and uploads $\big(\tilde{\Delta}_i^{(t)},\ \mathcal{A}_i^{\text{loc}}\big)$ to BTS; let $\mathcal{S}^{(t)} = \{i : v_i \text{ active}\}$;
  Step 4: Aggregation. BTS computes $\{w_i^{(t)}\}_{i \in \mathcal{S}^{(t)}}$ via (17), updates $gW^{(t+1)}$ via (18), and recomputes $\mathcal{A}^{\text{glb}} \leftarrow \frac{1}{|\mathcal{S}^{(t)}|} \sum_{i \in \mathcal{S}^{(t)}} \mathcal{A}_i^{\text{loc}}$;
  Step 5: Broadcast and Local Refresh. BTS broadcasts $gW^{(t+1)}$ and $\mathcal{A}^{\text{glb}}$ to all PVs (active and inactive). Each PV **in parallel**:
    updates caches: $\mathcal{A}_i^{\text{loc,prev}} \leftarrow \mathcal{A}_i^{\text{loc}}$, $\tilde{\epsilon}_i \leftarrow \epsilon_i^{(t)}/\epsilon_i^{\max}$, $U_i \leftarrow U_i + \epsilon_i^{(t)}$;
    evaluates $gW^{(t+1)}$ on its local test split to refresh $\mathcal{A}_i^{\text{loc}} \leftarrow \exp(-\text{MAE}_i^{\text{post}}/M)$ (post-aggregation, matches DDQN training distribution).
end for
Return $gW^* \leftarrow gW^{(T)}$.

Algorithm 2 DDQN Training for Privacy-Budget Allocation

Input: client pool $\mathcal{V} = \{v_1, \ldots, v_N\}$ with $\{\bar{s}_i, E_i\}$; episodes E_{ep}; FL rounds per episode T; reward weight α; DP clip norm S, noise multiplier ζ, per-round floor ϵ_{floor}; standard DDQN hyperparameters (replay capacity $|R|$, batch B, discount γ, learning rate η, exploration schedule β, target-update period C_{tgt}, gradient clip norm $g_{\max}$).
Output: online-network weights θ.

Initialize θ randomly, $\theta^- \leftarrow \theta$, $R \leftarrow \emptyset$;
for episode $e = 1, 2, \ldots, E_{\text{ep}}$ **do**
  Sample focal $v_f \sim \mathcal{V}$ uniformly; re-initialize gW randomly; reset $U_f \leftarrow 0$ and compute $\mathbf{s}_f^{(0)}$ via (19);
  for round $t = 0, 1, \ldots, T-1$ **do**
    Select $a^{(t)}$ by β-greedy on $Q(\mathbf{s}_f^{(t)}, \cdot; \theta)$; if $U_f \geq E_f$, focal abstains, is excluded from $\mathcal{S}^{(t)}$, and set $\epsilon_f^{(t)} \leftarrow 0$, else map $a^{(t)}$ to $\epsilon_f^{(t)}$ via (21);
    Set every background $\epsilon_j^{(t)} \leftarrow \max\big(k_{a^{(t)}} \bar{E}_f, \epsilon_{\text{floor}}\big)$ (self-play); run one DP-FedAvg round on $\mathcal{S}^{(t)}$ via (14)–(18);
    Evaluate the post-aggregation $gW^{(t+1)}$ on the focal test split to get $\mathcal{A}_f^{(t)}$; compute $\mathcal{P}_f^{(t)}$, reward $r^{(t)}$ via (22), next state $\mathbf{s}_f^{(t+1)}$; update $U_f \leftarrow U_f + \epsilon_f^{(t)}$;
    Push $(\mathbf{s}_f^{(t)}, a^{(t)}, r^{(t)}, \mathbf{s}_f^{(t+1)}, d^{(t)})$ into R;
    if $|R|$ is sufficient **then** sample a mini-batch from R, compute the DDQN target via (23), and update θ by Adam on the Huber loss (24) with gradient clipping (skip the step if the loss is non-finite);
  end for
  Decay β; every C_{tgt} episodes hard-update $\theta^- \leftarrow \theta$;
end for
Return θ.

3) Training Stability and Target Network: Vanilla mean-squared loss is sensitive to the large temporal-difference errors that appear at the start of training, when reward magnitudes can swing across an episode. We therefore train the online network by minimizing the smoothed-L_1 (Huber) loss over a mini-batch $\mathcal{B}$ of transitions sampled from a fixed-capacity replay buffer:

$$L(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{s},a,r,\mathbf{s}') \in \mathcal{B}} \text{Huber}\big(y - Q(\mathbf{s}, a; \theta)\big), \tag{24}$$

where y is computed via (23). Gradients are clipped to a maximum L_2 norm $g_{\max}$ to prevent occasional gradient spikes from destabilizing training. Exploration follows a β-greedy schedule that decays exponentially from β_{start} toward β_{end} across episodes, and the target network is hard-updated by copying θ to θ^- every C_{tgt} episodes.

4) Episode Design and Self-Play Approximation: Two design choices bridge the gap between training the policy in isolation and deploying it in a live FL system.

Independent episodes with fresh global models. Each episode samples a focal client v_i uniformly at random and re-initializes the global GRU model from random weights. The agent therefore experiences the full FL trajectory under its current policy, from cold start to round T, every episode. If we instead carried the warm-started global model across episodes, late-round rewards would be inflated by accumulated FL convergence rather than by the agent's budget choices, obscuring the policy signal.

Self-play approximation for background clients. Within each episode the focal client selects its multiplier via the policy. To prevent the agent from learning a free-riding strategy in which the focal client conserves ϵ while implicitly relying on background clients to provide informative gradients, we constrain every background client to use the same per-round budget $\epsilon^{(t)} = k_{a^{(t)}} \bar{E}_f$ as the focal in each round. The aggregated noise level reflects the realistic scenario in which all clients run independent copies of the trained policy, and the agent is rewarded only for behaviors that remain effective when its own choices are mirrored across the federation.

The training procedure is summarized in Algorithm 2.

C. Integration Algorithm of FL and DDQN

The trained DDQN policy and the DP-FedAvg training loop together form the complete PerFL framework. The per-vehicle data flow is illustrated in Fig. 3, and the end-to-end deployment loop in Fig. 2. Algorithm 1 formalizes one full deployment.

In Step 1 the BTS initializes and broadcasts the global model $gW^{(0)}$, the policy weights θ, and the DP/SNR hyperparameters; each PV initializes its budget tracker and state caches, with the broadcast accuracy $\mathcal{A}^{\text{glb}}$ taking a unit cold-start value. In Step 2, each PV v_i assembles its 7-dimensional state $\mathbf{s}_i^{(t)}$ from local information ($\bar{s}_i$, the cached post-aggregation $\mathcal{A}_i^{\text{loc}}$, the broadcast $\mathcal{A}^{\text{glb}}$, the round-over-round accuracy gain, the remaining-budget ratio, the previous spend, and round progress), runs the policy θ *locally* to obtain the multiplier $a_i^{(t)}$, and either marks itself *inactive* when its accumulated spend U_i has reached E_i or proceeds to map $a_i^{(t)}$ to $\epsilon_i^{(t)}$ via (21) and trains via (14) to obtain the locally updated model $oW_i^{(t)}$. In Step 3, each active PV forms the perturbed update $\tilde{\Delta}_i^{(t)}$ via (16) and uploads it alongside a single scalar $\mathcal{A}_i^{\text{loc}}$ summarizing its accuracy on the previous round's global model. In Step 4, the BTS aggregates the perturbed updates over the active subset $\mathcal{S}^{(t)}$ via (18) and recomputes a scalar $\mathcal{A}^{\text{glb}}$ from the uploaded $\mathcal{A}_i^{\text{loc}}$ values. In Step 5, the BTS broadcasts $gW^{(t+1)}$ and $\mathcal{A}^{\text{glb}}$ to all PVs (active and inactive), and each PV evaluates $gW^{(t+1)}$ on its own test split to refresh $\mathcal{A}_i^{\text{loc}}$ for the next round, thereby aligning the deployment-time state distribution with the post-aggregation signal seen during DDQN training (Section VI-B4).

Because θ is produced offline by Algorithm 2 and shared as a parameter, the BTS never needs to inspect any PV's raw sensor data or unperturbed gradients; the only scalars leaving each device are the noisy update $\tilde{\Delta}_i^{(t)}$ and the local accuracy summary $\mathcal{A}_i^{\text{loc}}$, both of which can be additionally protected by per-message DP if a stricter side-channel guarantee is required.

Algorithm 2 gives the offline training procedure that supplies θ, and therefore determines the per-round privacy budgets used in Algorithm 1.

VII. Simulation

In this section, we evaluate the proposed scheme on a publicly available real-world transit-vehicle dataset from the Helsinki metropolitan area, made openly available by Helsinki Region Transport (HSL) through the Digitransit high-frequency positioning API[1].

A. Simulation Setup

1) Environment Configuration: We implement PerFL using PyTorch 2.5 on Python 3.12. All experiments were conducted on a Windows workstation equipped with an Intel Core i9-13980HX CPU (2.20 GHz), 32 GB of RAM, and an NVIDIA GeForce RTX 4080 Laptop GPU.

2) Vehicle Dynamics Dataset: We subscribe to over one month of real-time vehicle position messages streamed from HSL's MQTT broker (Digitransit endpoint), covering the buses, trams, and trains operated under the HSL franchise across the Helsinki region. Each message contains the vehicle identifier, GPS coordinates, instantaneous speed, heading, vehicle type, operator company, and timestamp. The collection therefore captures genuine public-transport mobility patterns across many routes, operators, and time-of-day conditions, and provides a realistic testbed for evaluating our scheme.

We partition the dataset D by vehicle ID into per-vehicle datasets d_i. The raw daily stream covers approximately 3,000 distinct vehicles. From this pool we randomly sample 60 vehicles to form the experimental federation, balanced across the three HSL transit types (20 buses, 20 trams, and 20 trains) so that no modality dominates the analysis. The 60 vehicles also span the three sensitivity tiers (low/medium/high) defined in Section VII-A3, ensuring every privacy profile is represented. For each FL session, a stratified subset of 7 vehicles per type ($N = 21$ in total) is then drawn at random, which (i) keeps each transit modality and sensitivity tier fairly represented, and (ii) keeps the per-round FL training computationally feasible for the hundreds of DDQN training episodes that the policy requires.

3) Sensitivity Computation: Section V introduces the per-vehicle privacy sensitivity score $s_i \in [1, 3]$ abstractly; we here instantiate it for the Helsinki public-transport setting. Since per-driver questionnaires (Section III) are not available for the public-transport fleet, we compute s_i from five passively observable mobility statistics that serve as data-driven proxies for the latent privacy need (distinct from the training-sample count n_i used by the SNR-aware aggregation of Section VI):

- n_i^{rec}: number of raw trajectory records (data volume / temporal coverage);
- span_i: geographic span (sum of latitude and longitude);
- L_i: number of distinct lines v_i services;
- O_i: number of distinct operators associated with v_i;
- σ_i^2: speed variance, reflecting driving-pattern variability.

Each feature is min–max normalized across the vehicle pool. The continuous raw sensitivity is obtained as a weighted sum:

$$\bar{s}_i \;=\; w_{\text{rec}}\,\tilde{n}_i^{\text{rec}} + w_{\text{span}}\widetilde{\text{span}}_i + w_L\tilde{L}_i + w_O\tilde{O}_i + w_\sigma\tilde{\sigma}_i^2, \quad (25)$$

with non-negative weights summing to one; we set $\{w_{\text{rec}}, w_{\text{span}}, w_L, w_O, w_\sigma\} = \{0.15, 0.35, 0.20, 0.15, 0.15\}$, placing the strongest weight on geographic span as the most identifying signal. The score $\bar{s}_i \in [0, 1]$ is then linearly mapped to the sensitivity range $[1, 3]$:

$$s_i = 1 + 2\,\bar{s}_i. \quad (26)$$

The total privacy budget $E_i = C/s_i$ then follows from (8), where s_i is the *continuous* value defined by (26). For grouped reporting in Sections VII-B and VII-C, we additionally bucket vehicles into low/medium/high sensitivity tiers using the $1/3$ and $2/3$ quantiles of $\bar{s}_i$; these three discrete tiers serve *only* as group labels for per-group metrics and do not enter the budget formula or the DDQN reward.

4) Algorithm Description: In our experimental setup, all PVs collaboratively train a shared GRU network for vehicle-dynamics prediction, leveraging the GRU's ability to model temporal dependencies in trajectory time-series data. Concurrently, a DDQN policy learns to allocate the per-round privacy budget ϵ across PVs. At each FL session, one PV is designated as the focal client, and the DDQN observes a state vector that summarizes (i) the focal vehicle's sensitivity s_i, (ii) the current local and global model accuracy together with the marginal improvement over the previous round, (iii) the remaining fraction of the contract budget E_i, (iv) the ϵ spent in the previous round, and (v) the round progress t/T. Based on this state, the DDQN selects an action that scales the per-round fair share E_i/T by one of K multipliers (see Section VII-A5); the chosen multiplier is applied to every PV in the current round, balancing prediction accuracy and privacy protection through the joint reward in (12). The two components are integrated through the FL framework described in Section VI, yielding PerFL. We verify its effectiveness on the prediction-accuracy and privacy-protection metrics defined in Section V.

5) Implementation Hyperparameters: For the privacy-budget calibration in (8), we use $C = 40$ (i.e., $E_i = 40/s_i$). The DP-FedAvg machinery of Section VI uses clip norm $S = 1.0$, noise multiplier $\zeta = 0.1$, and SNR exponent $\alpha_w = 2$. The per-round floor is set to $\epsilon_{\text{floor}} = 0.3 \cdot E_i/T$. The DDQN action space contains $K = 10$ multipliers $\{0.0, 0.2, 0.4, \ldots, 1.8\}$ of the per-round fair share E_i/T, and the MAE normalisation cap is $M = 10$. Each FL session runs for $T = 20$ rounds, and each PV performs $J = 3$ local SGD epochs per round with learning rate 10^{-3} and batch size 16. DDQN training uses replay capacity $|R| = 5000$, mini-batch $B = 64$, discount $\gamma = 0.99$, learning rate $\eta = 5 \times 10^{-4}$, exploration $\beta_{\text{start}} = 1.0 \to \beta_{\text{end}} = 0.05$ at decay rate 0.99 per episode, and target-update period $C_{\text{tgt}} = 10$ episodes; the policy is trained for $E_{\text{ep}} = 400$ episodes.

B. Experimental Results on Prediction Performance

In the context of the IoVs, vehicle dynamics data is highly complex and subject to continuous variation. We evaluate PerFL on the vehicle speed prediction task, a representative regression problem on trajectory time series.

For the speed prediction task, each PV trains a local GRU to capture the temporal dependencies inherent in its trajectory

[1]Dataset description and access protocol: https://digitransit.fi/en/developers/apis/5-realtime-api/vehicle-positions/high-frequency-positioning/

TABLE II: Accuracy comparison at $\alpha = 0.5$, $n = 10$ seeds (mean $\pm$ std). Lower is better for all error metrics; higher is better for R^2; lower Fair.σ indicates more uniform accuracy across sensitivity groups.

Algorithm	gMAE	RMSE	R^2	MAE_i (Low)	MAE_i (Med)	MAE_i (High)	Fair.σ
S-FL	0.522 ± 0.064	0.881 ± 0.174	0.9831 ± 0.0077	0.393 ± 0.016	0.459 ± 0.046	0.699 ± 0.126	0.133 ± 0.058
DP-FL	0.656 ± 0.116	1.035 ± 0.210	0.9768 ± 0.0102	0.466 ± 0.059	0.562 ± 0.082	0.925 ± 0.166	0.199 ± 0.053
RN-FL	0.666 ± 0.111	1.036 ± 0.181	0.9770 ± 0.0090	0.462 ± 0.051	0.585 ± 0.090	0.935 ± 0.183	0.202 ± 0.070
PerFL	$\mathbf{0.622 \pm 0.097}$	$\mathbf{0.967 \pm 0.167}$	$\mathbf{0.9799 \pm 0.0078}$	$\mathbf{0.462 \pm 0.037}$	$\mathbf{0.529 \pm 0.066}$	$\mathbf{0.861 \pm 0.163}$	$\mathbf{0.176 \pm 0.062}$

time series. Each timestep is represented by an 8-dimensional feature vector: instantaneous speed (`spd`), heading (`hdg`), incremental latitude and longitude (Δ`lat`, Δ`lon`, used in place of absolute coordinates so that the learned representations are translation-invariant across different routes), acceleration (`acc`), schedule delay (`dl`, clipped to ± 300 s to suppress outliers), a door-status indicator from HSL telemetry (`drst`), and the time interval between consecutive observations (Δ`t`, which makes irregular sampling explicit to the GRU). These features are jointly informative of vehicle dynamics and provide the temporal context required for accurate prediction. The full FL pipeline (local GRU training, DP-aware aggregation, DDQN-guided per-round ϵ) is then driven by the PerFL framework described in Section VI.

1) Algorithms Compared: We compare PerFL against three baselines that span the privacy–accuracy spectrum:

- **S-FL** (Standard FL) [2]: vanilla FedAvg without any DP mechanism ($\epsilon = 0$ for every client, no noise injection). Included only as a reference for the no-privacy accuracy upper bound, not as a fair privacy-preserving competitor.
- **DP-FL** (Uniform DP-FedAvg) [35]: every client uses an identical per-round budget $\epsilon = \bar{E}/T$, where $\bar{E} = \frac{1}{N}\sum_i E_i$ is the mean contract budget across the federation. Each client therefore consumes the same total budget $\bar{E}$ regardless of its individual sensitivity, modeling the classical "one-size-fits-all" DP-FedAvg policy widely adopted in subsequent DP-FL work [18], [36].
- **RN-FL** (Random-Noise FL): a fully specified ablation baseline that we constructed for this study. At every round each client picks an action uniformly at random from the same K-multiplier set used by PerFL and applies it to its own fair share E_i/T. The scheduler is privacy-agnostic (it does not consult s_i or the current state), but the resulting per-round ϵ is still bounded by the same per-round floor ϵ_{floor} and contract cap E_i as PerFL.
- **PerFL** (ours): each client's per-round ϵ is decided by the trained DDQN policy conditioned on its sensitivity s_i and the current state (Section VI-B), subject to the same ϵ_{floor} and $E_i = C/s_i$ cap.

All four algorithms share the same FL training pipeline, including an identical GRU architecture, $N = 21$ stratified clients per round, $T = 20$ rounds, and identical local SGD hyperparameters. The three privacy-preserving algorithms additionally share the same DP-FedAvg machinery with clip norm $S = 1.0$, noise multiplier $\zeta = 0.1$, and SNR-aware aggregation. They differ only in the per-round ϵ allocation rule. Multi-seed deployment uses 10 independent random seeds, which jointly control client sampling, FL initialization, and (for RN-FL) the random action draws.

Fig. 4 traces the per-round global MAE of the four algorithms over 20 FL rounds at $\alpha = 0.5$, averaged across 10 deployment seeds. All algorithms exhibit a sharp drop in the first 3–4 rounds followed by gradual stabilisation. S-FL, which trains without any DP noise, attains the lowest MAE and converges fastest, serving as the no-privacy reference. DP-FL and RN-FL converge nearly as fast as S-FL but plateau approximately 0.13–0.14 MAE units above it because each round of training is corrupted by Gaussian noise. PerFL is the slowest of the three privacy-preserving methods during the first 10 rounds, reflecting the noisier per-round gradient profile produced by its personalized allocation policy. By round 20, however, its MAE is the closest of the three privacy-preserving methods to the S-FL curve, and its std band across seeds is visibly tighter than those of DP-FL and RN-FL, indicating a more consistent final model.

Table II quantifies the converged behavior. PerFL attains the lowest global MAE among the three privacy-preserving methods (0.622 ± 0.097 vs. 0.656 for DP-FL and 0.666 for RN-FL), a 5–7% relative improvement, while also achieving the highest R^2 among the privacy-preserving algorithms (0.9799 vs. 0.9768 and 0.9770). The gap to S-FL (0.522) is intrinsic to any DP mechanism: every round of clipped, noised gradient aggregation costs accuracy. The per-client MAE (MAE_i, computed on each client's own test split and averaged within each sensitivity tier) shows that PerFL is on par with the best baseline in the low-sensitivity tier (PerFL 0.462 vs. RN-FL 0.462 vs. DP-FL 0.466) and clearly outperforms both baselines in the medium and high tiers (e.g., 0.861 vs. 0.925/0.935 on high-sensitivity clients). The unfairness measure Fair.σ is also the lowest among DP methods (0.176 vs. 0.199/0.202), indicating that PerFL's personalized budget allocation does not penalise any group disproportionately.

Fig. 5 visualizes the per-group breakdown. In the low-sensitivity tier the four algorithms are essentially indistinguishable on MAE_i: noise has little impact when each client's contract budget E_i is large. From the medium tier onwards PerFL pulls clearly below DP-FL and RN-FL, and the gap is the largest in the high-sensitivity tier, where the absolute increase of MAE_i relative to S-FL is also largest. The latter effect is by design: high-s clients receive a tighter contract cap $E_i = C/s_i$, so even an oracle policy is forced to inject more noise per unit of training signal.

C. Experimental Results on Privacy Performance

This subsection evaluates the privacy behavior of the three privacy-preserving algorithms (S-FL omitted, as it applies no DP mechanism). The aggregate privacy score $\mathcal{P}$ is computed per-vehicle using Equation 10 and averaged over all N clients:

TABLE III: Privacy comparison at $\alpha = 0.5$, $n = 10$ seeds (mean $\pm$ std). Higher $\mathcal{P}$ is better; lower ϵ, BU, and RDP indicate stronger formal privacy. S-FL is omitted as it applies no DP mechanism. RDP composition uses $\delta = 10^{-5}$.

Algorithm	$\mathcal{P}$ (All)	$\mathcal{P}$ (Low)	$\mathcal{P}$ (Med)	$\mathcal{P}$ (High)	ϵ_{mean}	BU (All)	RDP (All)
DP-FL	0.5048 ± 0.0091	0.5048 ± 0.0091	0.5048 ± 0.0091	0.5048 ± 0.0091	27.350 ± 0.720	0.943 ± 0.006	2829.9 ± 148.2
RN-FL	0.5421 ± 0.0152	0.4930 ± 0.0215	0.5307 ± 0.0119	0.6006 ± 0.0157	24.744 ± 1.138	0.906 ± 0.026	3163.6 ± 287.4
PerFL	$\mathbf{0.7185 \pm 0.0048}$	$\mathbf{0.7466 \pm 0.0028}$	$\mathbf{0.7243 \pm 0.0053}$	$\mathbf{0.6874 \pm 0.0060}$	$\mathbf{13.253 \pm 0.271}$	$\mathbf{0.506 \pm 0.025}$	$\mathbf{864.0 \pm 45.9}$

$$\mathcal{P} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{P}_i, \tag{27}$$

where $\mathcal{P}_i \in (0, 1]$ and higher values indicate stronger per-vehicle protection. We additionally report the budget utilisation $\text{BU} = \epsilon_i / E_i$ and the RDP composition cost (at $\delta = 10^{-5}$) to quantify formal (ϵ, δ)-LDP strength.

Fig. 6 and Table III report the per-group privacy scores. PerFL achieves the highest $\mathcal{P}_i$ in *every* sensitivity tier: 0.747 (low), 0.724 (medium), and 0.687 (high), substantially above DP-FL's flat 0.505 (uniform ϵ for all clients) and RN-FL's 0.493/0.531/0.601. Aggregated over all clients, PerFL's $\mathcal{P} = 0.7185$ is 0.21 higher than DP-FL (0.5048) and 0.18 higher than RN-FL (0.5421). The std bands of PerFL across seeds are extremely tight (≤ 0.008 for every group), reflecting that the DDQN policy converges to a stable per-vehicle budget pattern.

Fig. 7 clarifies the mechanism. DP-FL spends a uniform $\epsilon \approx 27.35$ for every client regardless of need; because the contract cap $E_i = C/s_i$ shrinks with sensitivity, this uniform allocation pushes low-sensitivity clients to BU $= 0.86$ and *saturates* the high-sensitivity contract at BU $= 1.000$, leaving no headroom for the most privacy-sensitive group. RN-FL adopts a random per-round multiplier and ends with $\epsilon \approx 24.7$ on average; its per-tier spending (28.4/25.5/20.5 from low to high) follows the contract sizes mechanically but the allocation is not steered by any accuracy or sensitivity signal. PerFL spends *less than half* the budget overall ($\bar{\epsilon} = 13.25$) yet still attains higher $\mathcal{P}_i$ in every group. The DDQN agent learns to allocate ϵ asymmetrically along the contract: low-sensitivity clients (which have the largest cap) receive 11.7 on average for BU $= 0.37$, leaving comfortable headroom; high-sensitivity clients (whose cap is one-third as large) consume 15.0 for BU $= 0.68$, using a larger *fraction* of their tighter cap without ever saturating it. This adaptive, contract-aware policy is the source of PerFL's simultaneous improvement on accuracy (Sec. VII-B) and privacy.

Formal (ϵ, δ)-LDP composition (RDP) confirms the same picture. PerFL's RDP cost is 864.0 ± 45.9, roughly $3.3\times$ tighter than DP-FL (2829.9) and $3.7\times$ tighter than RN-FL (3163.6). In other words, PerFL provides a substantially stronger formal guarantee in addition to its higher $\mathcal{P}$.

D. Comprehensive Analysis of Experimental Results

The accuracy and privacy experiments in Sections VII-B and VII-C examine PerFL along orthogonal axes. This subsection brings the two views together and reads off the practical advantages PerFL offers over both the uniform-DP and random-noise baselines.

Fig. 8 synthesises the joint behavior of the three privacy-preserving algorithms on the accuracy–privacy plane. Each algorithm is plotted as a cluster of 10 per-seed dots together with a mean marker and a min–max envelope. PerFL is the only configuration that occupies the upper-left region of the plane. Its center lies at higher $\mathcal{P}$ *and* lower gMAE, and its envelope does not extend into the high-MAE region that both DP-FL and RN-FL visit on their unlucky seeds.

Combining the per-round convergence behavior (Fig. 4), the per-group accuracy breakdown (Fig. 5, Table II), the per-group privacy results (Figs. 6–7, Table III), and the trade-off plot (Fig. 8), we draw several consistent findings.

First, PerFL's personalized allocation produces simultaneously better accuracy (lower gMAE by 5–7% relative to DP-FL/RN-FL) and stronger privacy (higher $\mathcal{P}$ in every sensitivity tier), validating that personalization is not merely a fairness intervention but a Pareto improvement on both objectives. Second, half the budget for stricter formal privacy. PerFL consumes only about half of the cumulative privacy budget that the two baselines use ($\bar{\epsilon} = 13.25$ vs. 27.35 and 24.74), and the resulting RDP composition guarantee at $\delta = 10^{-5}$ is 3.3–3.7$\times$ tighter (864 vs. 2830 and 3164). The formal (ϵ, δ)-LDP strength of PerFL is therefore substantially stronger than that of either baseline. Third, contract-aware allocation that never saturates the cap. A unique structural property of PerFL is that the DDQN policy respects every client's contract cap $E_i = C/s_i$ with non-trivial headroom: budget utilisation ranges from BU $= 0.37$ on the most flexible (low-sensitivity) clients to BU $= 0.68$ on the tightest (high-sensitivity) clients. DP-FL, by contrast, applies a uniform ϵ that drives the high-sensitivity tier all the way to BU $= 1.000$, fully exhausting their contract; RN-FL hovers around BU ≈ 0.9 in every tier without sensitivity awareness. Only PerFL keeps every tier inside its contract with margin to spare.

In addition, fairer accuracy distribution across sensitivity tiers. The per-client accuracy spread across the three tiers, measured by Fair.σ, is the lowest among privacy-preserving methods (0.176 vs. 0.199 for DP-FL and 0.202 for RN-FL). High-sensitivity clients pay the largest absolute accuracy price under *any* DP method (a consequence of their tighter cap), but PerFL minimizes this cross-tier disparity. Finally, robustness across deployment seeds. The variance of PerFL's results across the 10 random seeds is the lowest among the three privacy-preserving methods on every metric we report: gMAE std 0.097 (vs. 0.116 and 0.111), $\mathcal{P}$ std below 0.01 in every group (vs. up to 0.02 for RN-FL), and no seed produces a gMAE above 0.78 for PerFL whereas DP-FL and RN-FL each have one seed with gMAE near 0.92. The DDQN policy is therefore robust to the random sampling of the deployed client subset, an important property for production deployments.

Taken together, these findings support the claim that a learned, contract-bounded, sensitivity-aware budget allocator strictly dominates uniform and random allocation along the accuracy–privacy trade-off frontier, while offering additional structural guarantees on contract respect, fairness across groups, and seed-to-seed robustness.

VIII. Conclusion

In this paper, we proposed PerFL, an FL framework for vehicle dynamics prediction that personalizes the privacy budget across drivers rather than imposing a uniform LDP noise level. PerFL pairs a GRU predictor with a DDQN-based per-round budget allocator, and aggregates the resulting perturbed updates through an SNR-aware FedAvg that down-weights noisier contributions. Motivated by a questionnaire study of drivers' privacy preferences that revealed heterogeneous concerns systematically correlated with observable mobility attributes, each vehicle's privacy sensitivity is inferred from passively observable mobility statistics, so the personalization requires no demographic data collection at deployment time. Comprehensive experiments on a real-world public-transit trajectory dataset from Helsinki show that PerFL Pareto-dominates the uniform-DP and random-noise FL baselines. It attains lower per-tier prediction error while consuming under half of the cumulative privacy budget, and yields a 3 to 4 times tighter formal (ϵ, δ)-LDP composition bound.

References

[1] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," *Journal of Advanced Transportation*, vol. 2021, pp. 1–18, 1 2021.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication efficiency in federated learning: Achievements and challenges," *arXiv preprint arXiv:2107.10996*, vol. 1, no. 1, p. 13, 7 2021. [Online]. Available: http://arxiv.org/abs/2107.10996

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, no. 1, p. 11, 2019.

[6] Z. Zhao, M. Luo, and W. Ding, "Deep leakage from model in federated learning," *arXiv preprint arXiv:2206.04887*, vol. 1, no. 1, p. 7, 2022.

[7] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*, IEEE. City, state: publisher name, 2019, pp. 691–706.

[8] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. City, country: nd, 2020, pp. 377–390.

[9] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays," *PLoS genetics*, vol. 4, no. 8, p. e1000167, 2008.

[10] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Measuring membership privacy on aggregate location time-series," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 2, pp. 1–28, 2020.

[11] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multiparty computing," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6178–6186, 2020.

[12] J. Park and H. Lim, "Privacy-preserving federated learning using homomorphic encryption," *Applied Sciences*, vol. 12, no. 2, p. 734, 2022.

[13] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[14] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[15] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning," in *2020 USENIX annual technical conference (USENIX ATC 20)*, 2020, pp. 493–506.

[16] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th annual symposium on foundations of computer science*. IEEE, 2013, pp. 429–438.

[17] ——, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.

[18] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, 2020, pp. 61–66.

[19] Y. U. Yim and S.-Y. Oh, "Modeling of vehicle dynamics from real vehicle measurements using a neural network with two-stage hybrid learning for accurate long-term prediction," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1076–1084, 2004.

[20] B. Jiang and Y. Fei, "Vehicle speed prediction by two-level data driven models in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1793–1801, 2016.

[21] C. Zhao, J. Gong, C. Lu, G. Xiong, and W. Mei, "Speed and steering angle prediction for intelligent vehicles based on deep belief network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 301–306.

[22] Z. Cheng, M.-Y. Chow, D. Jung, and J. Jeon, "A big data based deep learning approach for vehicle speed prediction," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2017, pp. 389–394.

[23] J. Shin and M. Sunwoo, "Vehicle speed prediction using a markov chain with speed constraints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3201–3211, 2018.

[24] L. P. Qian, A. Feng, N. Yu, W. Xu, and Y. Wu, "Vehicular networking-enabled vehicle state prediction via two-level quantized adaptive kalman filtering," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7181–7193, 2020.

[25] X. Zhou, M. Chen, L. Peng, and H. Li, "Acpp: An effective privacy preserving scheme for precise location sharing in internet of vehicles," in *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2017, pp. 883–887.

[26] Q. Kong, L. Su, and M. Ma, "Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4889–4898, 2020.

[27] D. Yu, Z. Zou, S. Chen, Y. Tao, B. Tian, W. Lv, and X. Cheng, "Decentralized parallel sgd with privacy preservation in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5211–5220, 2021.

[28] N. Wang, W. Yang, X. Wang, L. Wu, Z. Guan, X. Du, and M. Guizani, "A blockchain based privacy-preserving federated learning scheme for internet of vehicles," *Digital Communications and Networks*, 2022.

[29] H. Batool, A. Anjum, A. Khan, S. Izzo, C. Mazzocca, and G. Jeon, "A secure and privacy preserved infrastructure for vanets based on federated learning with local differential privacy," *Information Sciences*, vol. 652, p. 119717, 2024.

[30] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[32] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[33] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[34] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[35] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2018. [Online]. Available: https://arxiv.org/abs/1712.07557

[36] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu, "Practical and private (deep) learning without sampling or shuffling," in *International Conference on Machine Learning*, PMLR. City, state: publisher name, 2021, pp. 5213–5225.

BIOGRAPHIES

Wenjun Zhang is currently pursuing a Ph.D. degree at the Department of Computer Science, University of Helsinki, Finland. Her research interests focus on Distributed Machine Learning, Artificial Intelligence, and Information Privacy. She is also active in Mobile and Ubiquitous Computing and Environmental Sensing and Monitoring.

Boyu Wang received the B.E. degree in the School of Cyberspace Science and Technology from Beijing Institute of Technology, China, in 2024. He is currently pursuing the Master's degree at the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. His research interests include deep learning and edge computing.

Rasolondrazanaka Ryzzyie Vickene is a PhD researcher at Information System and Security & Countermeasures Experimental Center, Beijing Institute of Technology. She received his M.Sc. in Cyberspace Science and Technology from Beijing Institute of Technology China in 2024 and B.Sc. in Computer Science from Ecole Nationale d'Informatique University of Fianarantsoa, Madagascar in 2021. Her research interest includes network security, traffic analysis, and machine learning.

Chao Zhu received the Ph.D. degree from the Department of Computer Science, Aalto University, Espoo, Finland, in 2021. He is currently an Assistant Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China.

Xiaoli Liu received the Ph.D. degree in mathematics and statistics from the University of Helsinki, Helsinki, Finland, in 2017. She is an Assistant Professor with the School of Technology, University of Vaasa and Adjunct Professor with the Department of Computer Science, University of Helsinki. Her research focuses on AI and networked systems, particularly edge intelligence, distributed learning, large language model (LLM) inference, and agentic systems.

Sasu Tarkoma (Senior Member, IEEE) is a Professor of Computer Science with University of Helsinki. He is a visiting professor with the 6G Flagship at the University of Oulu. He completed his Ph.D. in Computer Science at the University of Helsinki in 2006. He has authored four textbooks and has published over 500 scientific articles. He holds ten granted U.S. patents. His research interests include Internet technology, distributed systems, 6G, and mobile and ubiquitous computing.

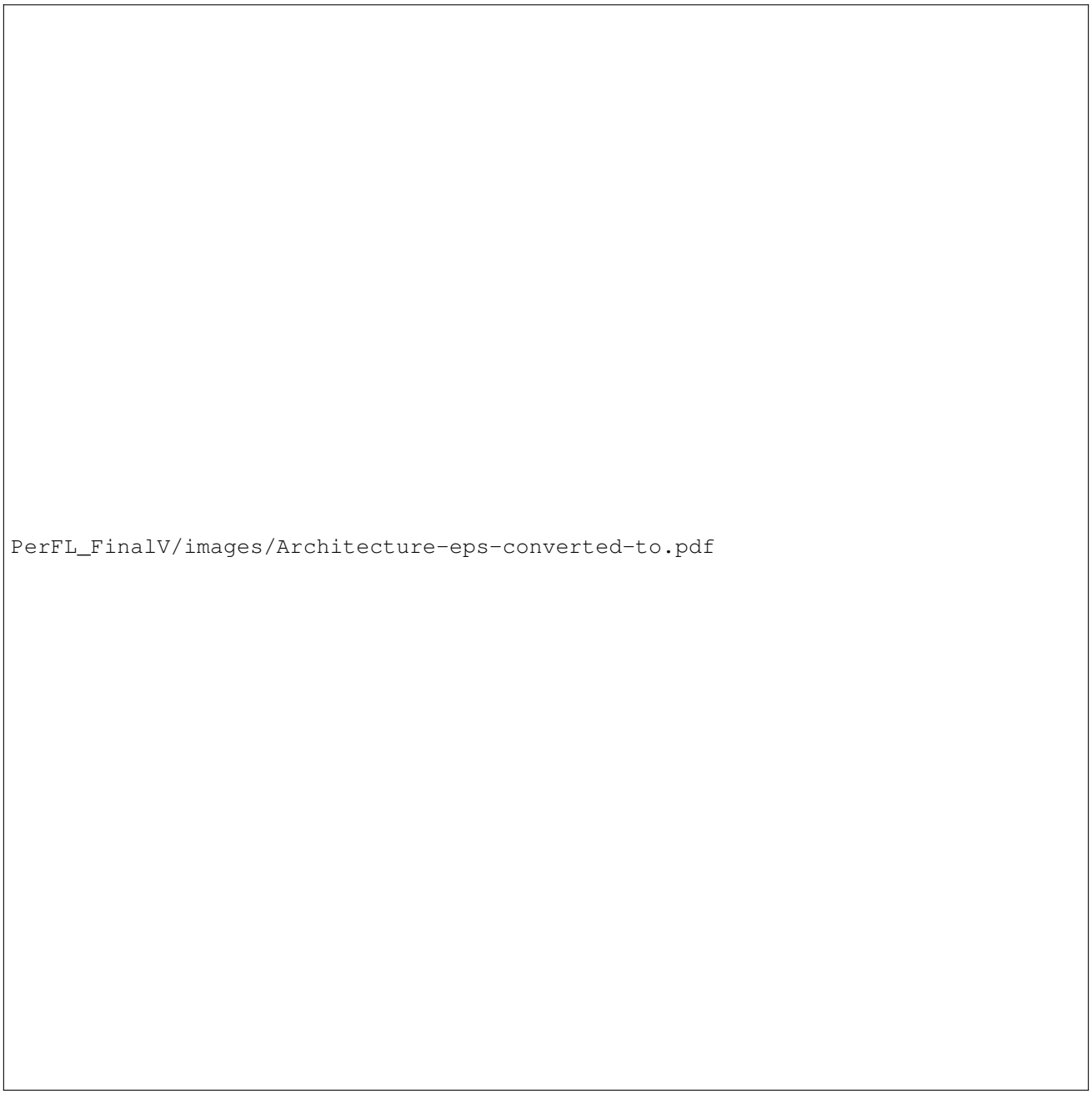

Fig. 2: Workflow of PerFL within a single service zone. The BTS coordinates PVs with heterogeneous privacy sensitivities (qualitative tiers shown by the color bars below each vehicle). Each FL round proceeds through five numbered steps: ① *Model Distribution* from BTS to PVs; ② *Local Model Training* on each PV's private trajectory data; ③ *Noise Addition* to the local model update according to the per-round privacy budget chosen on-device; ④ *Model Aggregation* at the BTS into a refreshed global model; and ⑤ *Local Model and Privacy Budget Update* at each PV for the next round.

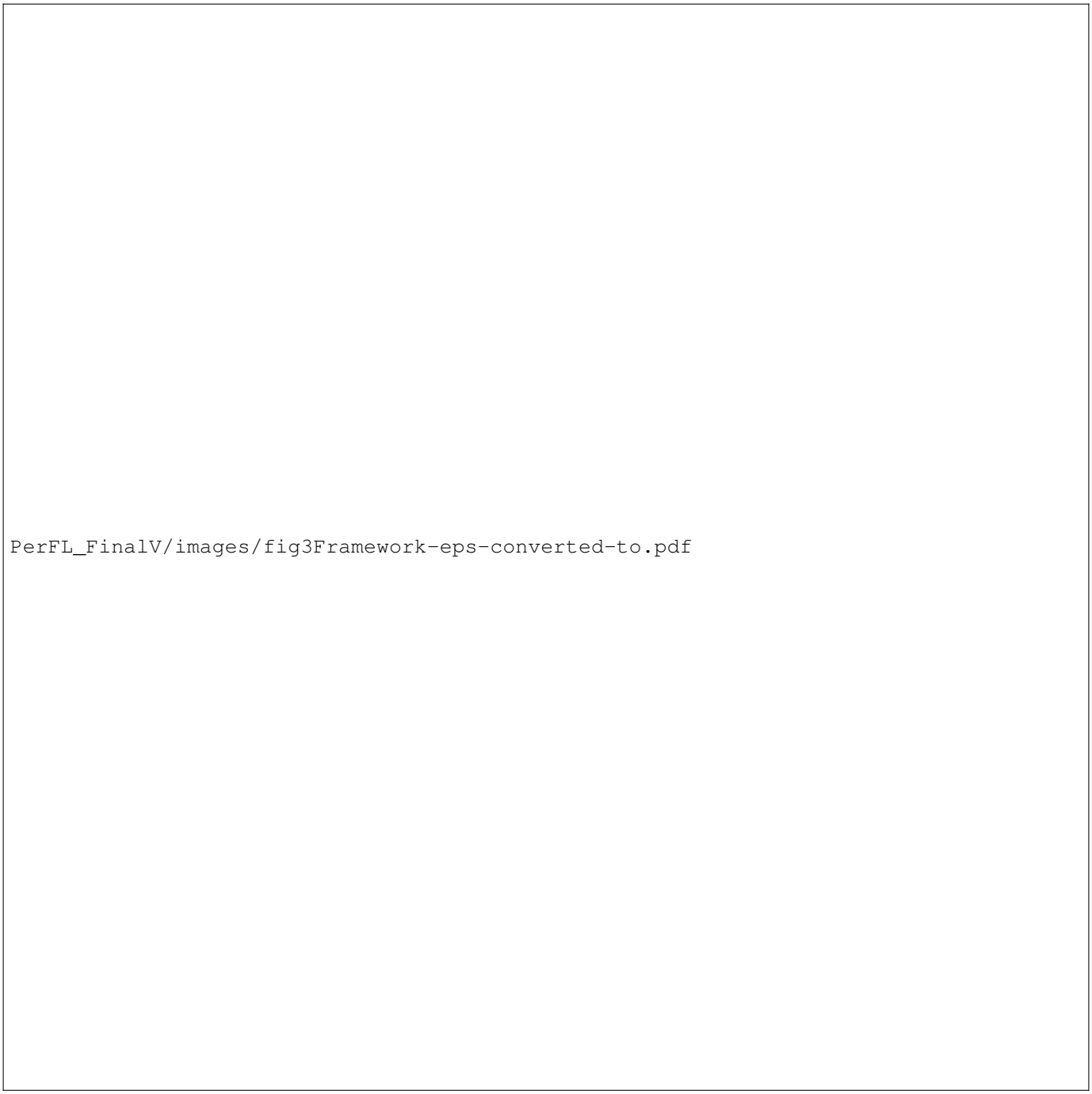

Fig. 3: Framework of PerFL on a single Participant Vehicle (PV). The PV runs a local GRU on its private trajectory data (top dashed box) to produce the round-t local update $\Delta_i^{(t)}$, while a set of passively observable mobility statistics yields a per-vehicle sensitivity score $\bar{s}_i$ (bottom). Both signals feed the 7-dimensional state $\mathbf{s}_i^{(t)}$ of an on-device DDQN policy (right dashed box), whose action sets the per-round privacy budget $\epsilon_i^{(t)}$. The DP-perturbation block then clips $\Delta_i^{(t)}$ to L_2-norm S and injects Gaussian noise with $\sigma_i = \zeta S/\epsilon_i^{(t)}$. The perturbed update $\tilde{\Delta}_i^{(t)}$ and the local accuracy summary $\mathcal{A}_i^{\text{loc}}$ are uploaded for SNR-aware model aggregation, and the refreshed global model is broadcast back to all PVs.

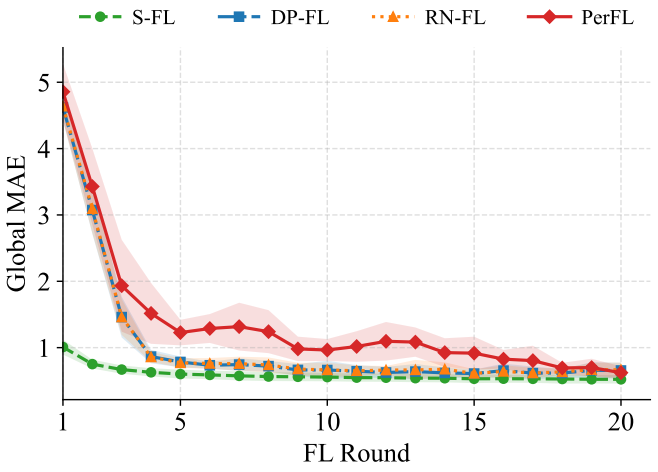

Fig. 4: Global MAE convergence over $T = 20$ FL rounds for the speed prediction task at $\alpha = 0.5$ ($n = 10$ seeds, mean $\pm$ std band). S-FL (no DP) attains the lowest achievable MAE and serves as the no-privacy reference; among the three privacy-preserving algorithms, PerFL converges more slowly than DP-FL and RN-FL during the first half of the session but ends with the smallest gap to the S-FL curve.

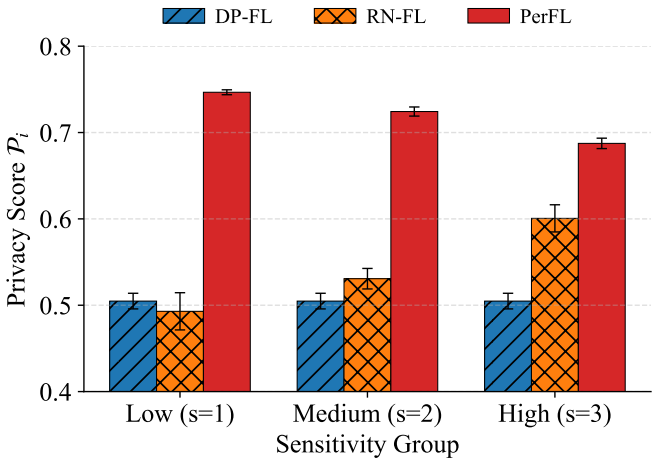

Fig. 6: Per-sensitivity-group privacy score $\mathcal{P}_i$ at $\alpha = 0.5$ ($n = 10$ seeds). PerFL achieves the highest $\mathcal{P}_i$ in every sensitivity tier, substantially above the uniform DP-FL and the random RN-FL baselines.

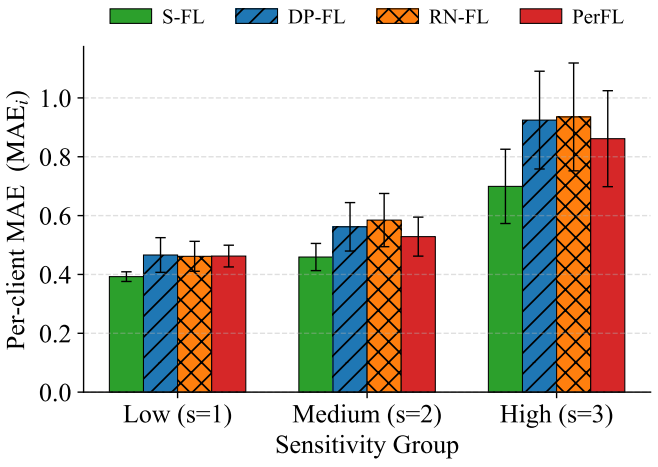

Fig. 5: Per-sensitivity-group per-client MAE (MAE_i) at $\alpha = 0.5$ ($n = 10$ seeds). For each client, MAE is computed on that client's own test split; values shown are tier-wise means. PerFL achieves the smallest gap to the no-DP reference (S-FL) in every tier, with the largest absolute improvement over the baselines in the medium- and high-sensitivity tiers. The gap to S-FL itself widens for high-sensitivity clients due to their tighter contract cap $E_i = C/s_i$.

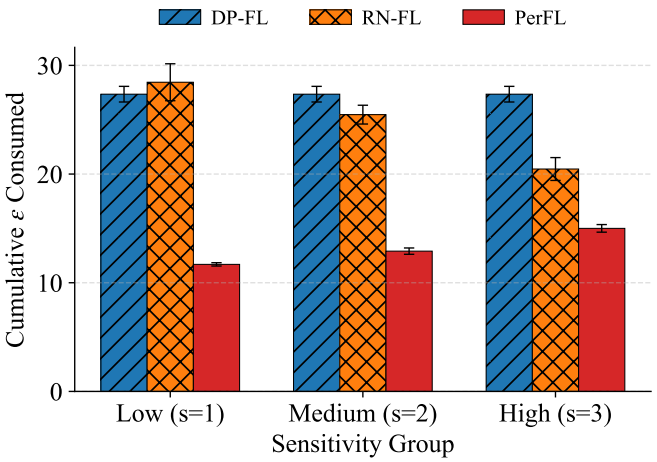

Fig. 7: Per-sensitivity-group cumulative privacy budget ϵ consumed at $\alpha = 0.5$ ($n = 10$ seeds). PerFL spends roughly half of the ϵ used by DP-FL and RN-FL overall, while still attaining higher per-group $\mathcal{P}_i$ in Fig. 6.

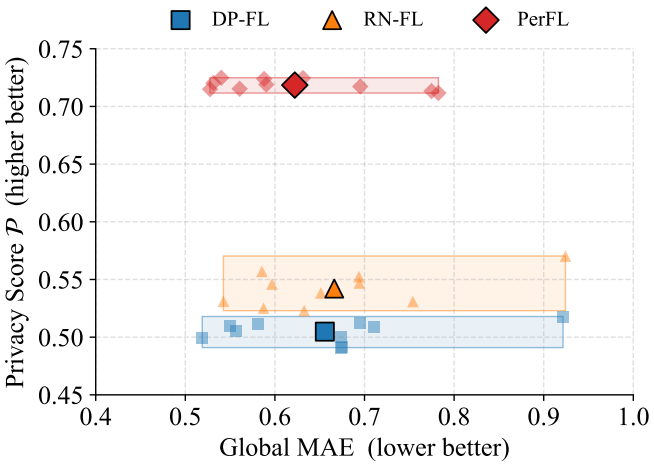

Fig. 8: Accuracy–privacy trade-off of the three privacy-preserving algorithms at $\alpha = 0.5$ ($n = 10$ seeds). Small markers are individual seeds; large markers with black edge are the per-algorithm mean; shaded rectangles span the min–max range. PerFL occupies the upper-left region, dominating DP-FL and RN-FL on both axes simultaneously.