Data-Diversity-Aware Quantization for Semi-Synchronous Federated Deep Reinforcement Learning in Vehicular Edge Computing

Abstract

Federated Learning (FL) has emerged as a promising paradigm for enabling privacy-preserving intelligence in Vehicular Edge Computing (VEC). However, the highly dynamic nature of vehicular networks introduces severe communication constraints, data heterogeneity, and frequent client stragglers and dropouts, all of which significantly degrade FL efficiency and convergence. While model quantization can reduce communication overhead, existing approaches typically rely on static or computation-intensive adaptive schemes that are unsuitable for resource-constrained vehicular environments and do not adapt compression or scheduling decisions to the heterogeneous informational value of client updates. This paper introduces DQS-FeDRL, a data-diversity-aware quantization framework for semi-synchronous federated deep reinforcement learning in VEC that integrates three key components to address communication constraints, data heterogeneity, and client dropouts. The proposed framework computes a data diversity index that combines intra-client label heterogeneity with a cohort-relative label novelty term, obtained via normalized Jensen–Shannon divergence between a client's local and the aggregated cohort label distributions. This index quantifies each client's marginal informational contribution and drives data-diversity-aware quantization that assigns adaptive bit-widths to clients, thereby allocating communication resources proportional to information utility, while a separate aggregation diversity index weights received updates based on their informational contribution. In parallel, a Bayesian availability-aware scheduling mechanism dynamically optimizes server waiting time to differentiate stragglers from permanent dropouts while accounting for client reliability uncertainty. These components are coordinated with a TD3-based deep reinforcement learning scheduler for client selection and bandwidth allocation. Extensive experiments on CIFAR-10 and GTSRB under non-IID data partitions and stochastic mobility-induced dropout and latency profiles demonstrate that DQS-FeDRL achieves up to 2.9 times faster wall-clock convergence, up to 75% reduction in communication overhead, and up to 50% lower energy consumption compared to state-of-the-art baselines, demonstrating that combining data-diversity-aware quantization with availability-aware semi-synchronous scheduling enables communication-efficient federated learning in vehicular environments.

Keywords: Adaptive quantization, Semi-synchronous federated learning, Data-diversity, Bayesian availability, Edge computing, Non-IID data

1. Introduction

The rapid emergence of IoT devices has transformed the paradigm from centralized cloud computing towards edge computing. Data is now processed closer to the source, which reduces latency and bandwidth usage [1]. Initially, edge computing in vehicular networks relied heavily on task offloading, where vehicles sent raw data and computationally intensive tasks to roadside units (RSUs) or edge servers to minimize local computation load and latency [2]. While task offloading was effective for simple applications, this centralized paradigm became increasingly inadequate in the deep learning era due to scalability limitations and growing privacy concerns. Transmitting massive volumes of raw sensor and contextual data not only overwhelms limited vehicle-to-cloud (V2C) bandwidth but also raises severe privacy concerns regarding users' locations and surroundings. Consequently, federated learning (FL) has emerged as the next viable paradigm, allowing edge devices to share lightweight model updates and keep sensitive raw data local in order to jointly train a shared global model [3].

Federated learning is not without its challenges. First, the most critical bottleneck is communication overhead: transmitting high-dimensional model parameters over highly dynamic vehicular networks causes significant latency and bandwidth consumption [4]. Furthermore, the highly dynamic nature of vehicular networks makes optimal client scheduling and efficient deployment of FL extremely difficult [5]. While recent frameworks [6] have made progress in data-aware client selection, they primarily focus on convergence speed; they often overlook the communication cost of high-precision models and fail to account for the stragglers and dropouts inherent in mobile environments. Quantization techniques have been adopted to compress model parameters and reduce the communication bottleneck, but many existing methods either apply static, uniform compression [7] or dynamic schemes that rely heavily on the computational power of edge devices [8]. This wastes bandwidth and may hurt accuracy because compression is applied without considering the client data utility. For instance, critical updates warrant higher precision, while redundant observations can be more aggressively compressed, yet static schemes treat them identically.

Moreover, existing diversity metrics rely solely on intra-client statistics such as label entropy, ignoring the inter-client

context of the current training cohort. A vehicle with diverse local data may contribute redundant information if the global model already reflects similar label distributions, while a client with modest local diversity may provide high marginal value if it covers under-represented label regions. Capturing this cohort-relative label novelty is essential for quantization and scheduling decisions that truly reflect information utility.

The second major challenge comes from stragglers and dropouts caused by the high mobility and unstable connectivity of vehicular networks. Vehicles may move in or out of coverage from the central server or RSUs, or experience channel fading due to poor wireless conditions. Standard synchronous FL protocols such as FedProx require the server to wait for all selected clients to upload their updates, which means the entire system can be blocked by just a few stragglers or dropped participants [9]. Asynchronous FL mitigates this problem, but it introduces model staleness and convergence instability because the server aggregates outdated updates from clients with varying delays [10]. Recent efforts have tried to use dynamic timeouts, but they often depend on simple feedback loops that blindly extend wait times to hit a target participation ratio, without properly modeling the uncertainty of whether vehicles are genuine stragglers or permanent dropouts [11].

To overcome these restrictions, we introduce DQS-FeDRL, a data-diversity-aware quantization framework for semi synchronous federated deep reinforcement learning in vehicular edge computing. Unlike prior work that relies on uniform compression, rigid synchronization, or imposes heavy computational burdens on resource-constrained edge devices, the proposed method solves a multi-level optimization problem to dynamically allocate system resources.

Our core premise is that resource expenditure should be proportional to data utility: clients contributing unique and diverse data warrant higher precision and extended latency budgets, while those with redundant data are aggressively compressed and de-prioritized for better resource utilization.

This approach is vital for critical tasks like autonomous driving and traffic sign recognition, where accurately capturing rare "edge case" scenarios is essential for safety. It also supports systems like traffic flow prediction and network security, allowing them to process massive amounts of varied data without clogging the connection.

Unlike traditional methods that look at a vehicle's data in a vacuum, this framework evaluates a vehicle's contribution compared to the rest of the group. By focusing on the unique marginal gain each vehicle brings to the current round, the system ensures that every byte and every second of waiting time is spent on the most meaningful updates.

This is especially relevant for emerging AI-enabled vehicle routing and location planning, which increasingly treat trajectories as a first-class control to gather spatio-temporal evidence for edge learning. Recent formulations in this domain couple mobility decisions directly to downstream model adaptation objectives [12], so upstream learning traffic may reflect deliberate exploration of rare environments as well as local label skew. Our cohort-relative notion of marginal value is inherently aligned with such settings; while the present work does not optimize routing, DQS-FeDRL provides the necessary communication efficiency to support vehicles navigating novel or challenging routes. A natural extension is to compose our scheduling and quantization with trajectory-level policies, defining cohort-relative novelty over route or context descriptors in addition to label histograms.

While individual components (Beta-distributed reliability tracking and Weibull latency modeling) build on established techniques, our contribution lies in integrating them through the cohort-relative diversity signal to enable data-diversity-aware quantization and availability-aware semi-synchronous operation. Critically, we employ a TD3-based deep reinforcement learning agent as an intelligent scheduler for resource allocation, which learns to dynamically coordinate client selection, bandwidth allocation, quantization bit-widths, and semi-synchronous deadlines making the semi-synchronous approach highly effective in vehicular environments.

We summarize the primary contributions of this paper as follows:

1. Cohort-Relative Label Novelty Metric and Data-Diversity-Aware Quantization: We extend our previous DA-FedRL framework [13] by introducing a cohort-relative label novelty metric $N_{t,i}^{\text{lbl}}$ that quantifies each client's marginal informational contribution relative to the current training cohort via normalized Jensen–Shannon divergence. Unlike conventional intra-client diversity measures, this metric drives individualized quantization bit-widths that minimize communication overhead for cohort-redundant updates while preserving high-fidelity information from clients with novel label coverage, enabling data-diversity-aware quantization that allocates communication resources proportional to marginal information utility rather than absolute diversity.

2. Bayesian Availability-Aware Semi-Synchronous Scheduling: We develop a probabilistic semi-synchronous scheduler that models client availability as a Beta distribution and completion times using Weibull distributions. The server intelligently calculates dynamic optimal wait times (τ_t^*) to distinguish temporary stragglers from permanent dropouts, maximizing participation without stalling the global training process. This availability-aware approach accounts for client reliability uncertainty and adapts directly to mobility-induced dropout patterns in vehicular edge networks.

3. Diversity-Weighted Aggregation and DRL-Driven Hierarchical Optimization: Unlike standard FedAvg that weights updates by dataset size, we introduce a diversity-weighted aggregation scheme (Eq. 1) that prioritizes clients based on their true informational contribution. To integrate these mechanisms without over-compressing the Markov Decision Process, we deploy a hierarchical TD3 reinforcement learning agent. This learned policy coordinates bandwidth allocation and client selection, effectively balancing data utility, communication efficiency, and straggler resilience within a unified system architecture.

The remainder of this paper is organized as follows. Section II reviews related work on federated learning in vehicular edge computing, communication-efficient FL, and robust scheduling. Section III describes the system model and data diversity quantification. Section IV formulates the optimization problem. Section V presents the proposed DQS-FeDRL framework in detail. Section VI outlines the experimental setup and metrics. Section VII evaluates DQS-FeDRL against baselines, reports component and fair-baseline analyses, and includes a parameter sensitivity analysis that justifies key hyperparameter choices. Finally, Section VIII concludes the paper and discusses future research directions.

2. Related Work

Existing research on federated learning in vehicular and edge computing environments can be broadly organized into four threads. The first thread surveys FL foundations and system architectures in vehicular edge settings. The second targets the communication bottleneck through model compression and quantization. The third emphasizes data- or utility-aware client selection to handle non-IID data and improve convergence. The fourth addresses training-time robustness through asynchronous or semi-synchronous protocols, dynamic scheduling, and resource allocation under stragglers and dropouts.

2.1. Federated Learning in VEC

The shift from centralized cloud computing to edge intelligence resulted in the adoption of Federated Learning (FL) in the Internet of Vehicles (IoV). Initial works primarily focused on replacing traditional task offloading with privacy-preserving local model training and training of a global model through parameter sharing. Maroua [4] provides a comprehensive survey demonstrating how FL provides collaborative training without disclosing raw data. To improve the convergence of FL in resource-constrained vehicular environments, Xia et al. [14] and Du et al. [15] proposed edge-assisted frameworks that optimize resource allocation and client scheduling. Furthermore, Lu et al. [16] and Pokhrel & Choi [17] integrated blockchain technologies to enhance the security and reliability of FL updates in 6G-enabled vehicular networks. While these works lay the foundation of FL, they assume reliable connectivity and static network topologies. The effect of extreme mobility and frequent dropouts on model convergence is overlooked, focusing mainly on system architecture and privacy. This limitation motivates the need for communication-efficient protocols that can handle dynamic vehicular conditions.

2.2. Communication-Efficient FL

To directly address the communication bottleneck identified above, various client selection strategies and model quantization have been introduced. Efficient selection algorithms such as FedCPF [18] and joint client-selection frameworks like FedSSC [19] have been used to reach model convergence quickly, reducing the total latency. Model compression techniques have been widely explored. Alistarh et al. [20] introduced NUQSGD, a fundamental stochastic quantization method, while Wen et al. [21] proposed ternary gradients (TernGrad) to aggressively reduce update sizes. In the context of FL, Reisizadeh et al. [7] developed FedPAQ, combining periodic averaging with quantization to reduce communication rounds. Specifically for vehicular networks, Shen et al. [22] applied ternary quantization to minimize bandwidth usage, and Amiri & Gündüz [23] optimized transmission over wireless fading channels. More recent works like UVeQFed [24] and adaptive quantization [25] have attempted to balance compression and accuracy. However, most existing methods either adopt static or uniform compression across all clients, treating everyone as equally important, or are computationally intensive for edge devices. Due to the heterogeneous data distribution, the former technique is not very fruitful. Clients with highly diverse features are compressed as aggressively as those with redundant data, leading to loss of accuracy, waste of bandwidth, and slower convergence. Crucially, these quantization methods treat each client in isolation, ignoring whether the data has already been well-represented in recent aggregation rounds. This motivates our cohort-relative label novelty metric, which connects quantization precision to the information-theoretic novelty each client contributes. However, identifying which data is worth preserving at high precision requires understanding data quality, leading researchers to explore data-aware client selection strategies.

2.3. Data Quality & Client Selection

Building on the insight that not all updates have equal value, recent research has focused on data-aware client selection. Oort [26] and AUCTION [27] introduced utility-aware selection mechanisms that prioritize participants based on training loss and data quality. Convergence analysis on client selection [28] provided theoretical convergence guarantees for such selection strategies, while active learning principles in FL [29] have been explored for faster convergence. In our own previous work, DA-FeDRL [13], we proposed a reinforcement learning-based selector to prioritize high-utility clients. However, DA-FeDRL evaluated clients using only intra-client statistics without measuring distributional novelty relative to the training cohort, and focused solely on client selection without addressing bandwidth bottlenecks or stragglers and dropouts. The present work extends DA-FeDRL by introducing the cohort-relative label novelty metric, coupling it with adaptive quantization, and integrating Bayesian availability modeling for semi-synchronous scheduling.

While these frameworks are effective in identifying which clients to select for training, they often overlook other bottlenecks such as bandwidth usage, and stragglers/dropouts. Once a client is selected, it must take part in the training process, and that is where the bottlenecks occur: transmitting model parameters to the central server and handling of stragglers/dropouts, which are often not of primary concern of these approaches. Furthermore, data-aware selection frameworks quantify data value using only local statistics (e.g., training loss, gradient norms, or intra-client label entropy), without measuring how a client's data distribution diverges from the current training cohort. Clients

with locally diverse but globally redundant data may be over-prioritized, while clients offering genuinely novel label coverage are overlooked. DQS-FeDRL addresses this through the cohort-relative label novelty $N_{t,i}^{\text{lbl}}$, which captures inter-client distributional novelty via normalized Jensen–Shannon divergence.

2.4. Dynamic Scheduling and Resource Allocation

To ensure selected high-quality clients can actually complete their uploads despite network volatility, asynchronous and semi-asynchronous protocols have been explored. FedAsync [10] allows the server to aggregate updates as they take place, minimizing waiting time while creating model staleness. FedProx [9] addresses heterogeneity by adding a proximal term to the local objective. SAFA [11] introduced a semi-asynchronous protocol with a cache mechanism to handle laggards without completely breaking synchronization. Similarly, Lu et al. [30] and Zheng et al. [31] proposed dynamic scheduling for edge intelligence.

The issue with asynchronous protocols is data staleness. Old data that is currently irrelevant may be integrated with newer data, affecting model convergence. Moreover, the semi-synchronous approaches rely on simple feedback loops that blindly rely on hitting a target participation and overlook the inclusion of probabilistic uncertainty of devices in a dynamic vehicular network to distinguish between stragglers and dropouts, leading to suboptimal resource optimization.

Additionally, recent fairness-aware and co-distillation approaches have been explored to mitigate straggler impact. LESSON [32] proposes learned semi-synchronous deadline optimization, while SALF [33] employs layer-wise salvaging for partial updates. However, these methods do not optimize quantization and availability-aware scheduling together.

Furthermore, recent advances in edge intelligence have effectively leveraged multi-agent RL and TD3-based control to orchestrate dynamic environments. For instance, strong RL frameworks have been proposed for adaptive network slicing [34], SLA-aware task offloading and service orchestration [35], and backhaul traffic optimization in UAV-enabled communications [36]. While these schedulers perform well for workload orchestration and QoS targets, federated learning introduces different bottlenecks: the informational utility of model updates under data heterogeneity, and strict synchronization barriers imposed by mobile stragglers. Zheng et al. [12] study a complementary class of AI-defined vehicle problems: online location planning that jointly optimizes order serving and spatio-temporally heterogeneous edge-assisted model fine-tuning using multi-agent reinforcement learning with graph-structured state representations, explicitly trading off service deadlines and data staleness when deciding where vehicles should move to collect informative urban signal. That mobility-side control is orthogonal to server-side federated uplink scheduling and adaptive quantization, but it clarifies how routing policies can reshape the heterogeneity of edge-learning traffic that FL systems must ingest.

Beyond horizontal diversity, the VEC architecture mirrors hierarchical Vertical Federated Learning (VFL) structures where vehicles and infrastructure maintain disjoint feature views. While this work evaluates a single-tier RSU–vehicle setup, realistic deployments will likely adopt a Server–RSU–Vehicle hierarchy.

In such multi-tier arrangements, our framework can be utilized to manage both horizontal diversity across RSU clusters and vertical collaborations between vehicles (holding onboard sensor data) and RSUs (holding infrastructure-side observations). Jiang et al. [37] address critical governance challenges such as sample and label unlearning in these vertical settings. DQS-FeDRL is complementary to these removal objectives; by utilizing our adaptive quantization and selection rules to optimize stringent feature-exchange requirements, DQS-FeDRL supports privacy-preserving vertical aggregation in dynamic environments.

To bridge this gap, DQS-FeDRL introduces an adaptive resource management framework tailored to the strict constraints of federated learning. Our framework systematically couples data-diversity-aware quantization (via $I_{t,i}^{\text{sel}}$) and diversity-weighted aggregation (via $I_{t,i}^{\text{agg}}$) with Bayesian availability-aware scheduling. This semi-synchronous approach dynamically calculates optimal wait times based on probabilistic vehicle availability, effectively mitigating the impact of temporary stragglers without stalling the global aggregation barrier. By leveraging a cohort-relative label novelty metric $N_{t,i}^{\text{lbl}}$ as a shared baseline, DQS-FeDRL ensures that precision assignment, aggregation weighting, and optimal deadline calculations are strictly driven by each client's marginal informational contribution to the current training round, resulting in a cohesive and highly efficient resource allocation scheme.

3. System Model and Assumptions

3.1. Network Architecture and FL Setup

We consider a Vehicular Edge Computing (VEC) system consisting of a set of vehicles $\mathcal{N} = \{1, 2, \ldots, N\}$ and a central server (RSU or edge server) that coordinates learning within its coverage area. Each vehicle $i \in \mathcal{N}$ maintains a private dataset D_i collected from local sensors (e.g., cameras, LiDAR, GPS) and trains the same lightweight deep neural network using local SGD.

Uplink communication uses OFDMA with total bandwidth B. In each round t, a subset $\mathcal{S}_t \subseteq \mathcal{N}$ is selected, with each vehicle $i \in \mathcal{S}_t$ allocated bandwidth $B_{t,i}$ such that $\sum_{i \in \mathcal{S}_t} B_{t,i} \leq B$. The server broadcasts the global model w_t to selected vehicles, which perform E local epochs of SGD, apply data-diversity-aware quantization (Eq. 22) and upload the compressed updates.

Unlike standard FedAvg [3], which weights updates by dataset size, our framework prioritizes information content. The server de-quantizes received updates $\hat{w}_{t,i}$ from the set $\mathcal{R}_t \subseteq \mathcal{S}_t$ of vehicles that complete within the deadline, then aggregates using diversity-weighted averaging:

$$w_t = \sum_{i \in \mathcal{R}_t} \frac{I_{t,i}^{\text{agg}}}{\sum_{j \in \mathcal{R}_t} I_{t,j}^{\text{agg}}} \hat{w}_{t,i}. \tag{1}$$

This aggregation strategy ensures that clients contributing novel label coverage to the cohort receive proportionally higher influence in the global model.

3.2. Latency and Energy Consumption Models

The total latency for each vehicle comprises local training time and model upload time:

$$\tau_{t,i}^{\text{train}} = \frac{Lc_i|D_{t,i}|}{f_i}, \quad \tau_{t,i}^{\text{up}} = \frac{Q_{t,i}}{r_{t,i}^{\text{up}}}, \tag{2}$$

where L is the number of local epochs, c_i is CPU cycles per data sample, $|D_{t,i}|$ is local dataset size, f_i is computation capacity, and $Q_{t,i}$ is the quantized payload size. The uplink rate follows Shannon capacity:

$$r_{t,i}^{\text{up}} = B_{t,i} \log_2 \left(1 + \frac{P_{t,i}G_{t,i}}{N_0 B_{t,i}}\right), \tag{3}$$

where $P_{t,i}$ is transmit power, $G_{t,i}$ is channel gain, and N_0 is noise power spectral density. Total delay is $\tau_{t,i}^{\text{total}} = \tau_{t,i}^{\text{train}} + \tau_{t,i}^{\text{up}}$.

Similarly, energy consumption comprises computation and transmission components:

$$E_{t,i}^{\text{cmp}} = L\zeta_i c_i |D_{t,i}| f_i^2, \quad E_{t,i}^{\text{up}} = \frac{P_{t,i}Q_{t,i}}{r_{t,i}^{\text{up}}}, \tag{4}$$

where ζ_i is the effective capacitance coefficient. Total energy is $E_{t,i}^{\text{total}} = E_{t,i}^{\text{cmp}} + E_{t,i}^{\text{up}}$. Notably, the payload size $Q_{t,i}$ directly depends on the assigned quantization bit-width $\tilde{b}_{t,i}$, creating a direct coupling between the data-diversity-aware quantization decisions and both communication latency and energy consumption.

3.3. Data Diversity Quantification

We quantify each vehicle's dataset value through two diversity indices: the selection index $I_{t,i}^{\text{sel}}$ drives client selection, quantization bit-width assignment, and deadline optimization, while the aggregation index $I_{t,i}^{\text{agg}}$ determines update weighting during aggregation.

Both indices combine normalized components. First, label heterogeneity $H_{t,i}$ measures the distribution across classes using the Gini-Simpson index:

$$H_{t,i} = 1 - \sum_{c=1}^{C} p_{i,c}^2 \tag{5}$$

where $p_{i,c}$ is the proportion of class c in vehicle i's dataset.

Although $H_{t,i}$ captures intra-client label heterogeneity, it does not quantify whether vehicle i provides new label information relative to the current cohort. We define a cohort-level label distribution $\bar{p}_t$ at round t from aggregated label statistics and measure cohort-relative novelty using the normalized Jensen–Shannon divergence:

$$N_{t,i}^{\text{lbl}} = \frac{\text{JSD}(p_i \parallel \bar{p}_t)}{\log 2} \in [0, 1], \tag{6}$$

where $\text{JSD}(p_i \parallel \bar{p}_t) = \frac{1}{2}\text{KL}(p_i \parallel m) + \frac{1}{2}\text{KL}(\bar{p}_t \parallel m)$ with $m = \frac{1}{2}(p_i + \bar{p}_t)$. This cohort-relative label novelty metric is a key methodological contribution of this work. While the Gini-Simpson index $H_{t,i}$ captures only intra-client label balance, $N_{t,i}^{\text{lbl}}$ measures inter-client distributional novelty by comparing each vehicle's label profile against the aggregated label statistics of the current training cohort. This distinction is critical: a vehicle with high local diversity may still contribute redundant information if similar label distributions have already been aggregated, whereas a vehicle with modest local diversity may provide high marginal value by covering under-represented regions.

Existing data-aware FL approaches evaluate clients using only local statistics (training loss, gradient norms, or intra-client label entropy), which measure a client's internal data characteristics but not its value relative to what has already been aggregated. As a result, clients with locally diverse but globally redundant data may be over-prioritized. By measuring distributional divergence via Jensen–Shannon divergence, $N_{t,i}^{\text{lbl}}$ quantifies each client's marginal information gain.

We then define an enhanced label diversity term that jointly captures intra-client heterogeneity and cohort-relative novelty:

$$H'_{t,i} = \xi H_{t,i} + (1 - \xi)N_{t,i}^{\text{lbl}}, \quad \xi \in [0, 1]. \tag{7}$$

Second, the age of update $U_{t,i}$ measures how long a vehicle has been waiting since its last selection:

$$U_{t,i} = \frac{\Delta t_i}{\max_{j \in \mathcal{N}}(\Delta t_j) + \epsilon_u} \tag{8}$$

where Δt_i is the number of rounds since vehicle i was last selected, and $\epsilon_u > 0$ is a tiny constant that keeps division by zero from happening. A higher $U_{t,i}$ indicates the vehicle has not been selected for a longer period, increasing its priority for selection.

Third, the relative dataset size $S_{t,i}$ is incorporated to capture gradient stability. This ensures that updates from data-rich clients, which provide more reliable gradient directions, are prioritized to improve convergence.

$$S_{t,i} = \frac{|D_i|}{\max_{j \in \mathcal{N}}(|D_j|)}. \tag{9}$$

The selection diversity index combines all three components:

$$I_{t,i}^{\text{sel}} = w_h H'_{t,i} + w_u U_{t,i} + w_s S_{t,i}, \tag{10}$$

where the weights satisfy $w_h + w_u + w_s = 1$. This index is used for client selection, bit-width quantization, and deadline optimization.

The aggregation diversity index excludes the age component:

$$I_{t,i}^{\text{agg}} = \tilde{w}_h H'_{t,i} + \tilde{w}_s S_{t,i}, \tag{11}$$

where $\tilde{w}_h + \tilde{w}_s = 1$. This index is used exclusively for weighting updates during aggregation (Eq. (1)), where the age of update is irrelevant and would introduce undesired bias toward recently unselected clients regardless of their data quality. Both indices are normalized to $[0, 1]$ and computed locally. The cohort distribution $\bar{p}_t$ is maintained from aggregated statistics, and vehicles may report only the resulting scalar scores without disclosing raw data.

The selection weights $(w_h, w_s, w_u) = (0.4, 0.4, 0.2)$ prioritize informational novelty and cohort-scale balance alongside participation; increasing w_u improves fairness to long-idle clients

Table 1: Notations

Symbol	Definition
$\mathcal{N}$	Set of vehicles
$\mathcal{S}_t, \mathcal{R}_t$	Scheduled and successful vehicles at round t
$B, B_{t,i}$	Total and allocated bandwidth
$G_{t,i}, r_{t,i}^{\text{up}}$	Channel gain and uplink rate
$w_t, w_{t,i}, \hat{w}_{t,i}$	Global, local, and dequantized model parameters
$I_{t,i}^{\text{sel}}, I_{t,i}^{\text{agg}}$	Selection and aggregation diversity indices
$H'_{t,i}, U_{t,i}, S_{t,i}$	Components of diversity indices
$N_{t,i}^{\text{lbl}}$	Cohort-relative label novelty (normalized JSD)
$\bar{p}_t$	Cohort-level label distribution at round t
ξ	Fusion weight for enhanced label diversity $H'_{t,i}$
$b_{t,i}, b_{\min}, b_{\max}$	Assigned bit-width and bounds
$q_{t,i}, Q_{t,i}$	Quantized index and payload size
$\tau_{t,i}^{\text{total}}, \tau_t^*$	Vehicle latency and server deadline
$E_{t,i}^{\text{total}}$	Vehicle energy consumption
$\bar{\theta}_{t,i}$	Vehicle reliability probability
α_i, β_i	Beta prior parameters for reliability
$\lambda_{t,i}, k$	Weibull latency distribution parameters
μ, η	Bit-width scaling and deadline trade-off factors

but can slow convergence by admitting more redundant updates, whereas larger w_h or w_s shifts emphasis toward label structure and dataset scale. For aggregation, we use equal weights $(\tilde{w}_h, \tilde{w}_s) = (0.5, 0.5)$ so that local heterogeneity and cohort-relative novelty contribute symmetrically to update influence. These settings were chosen empirically for stable behavior across the evaluated CIFAR-10 and GTSRB partitions.

From a privacy standpoint, $I_{t,i}^{\text{sel}}$ and $I_{t,i}^{\text{agg}}$ are transmitted as scalar summaries rather than raw label histograms or high-dimensional gradients. This provides an inherent layer of abstraction that prevents the direct leakage of raw local data distributions during the selection and aggregation phases. While these scores are calculated locally to reflect label-level structure, their reporting is restricted to a single numerical value, ensuring that the RSU only observes the final utility ranking rather than the underlying record-level payloads.

The dual-index design allows DQS-FeDRL to decouple selection decisions from aggregation weighting, ensuring that clients are chosen based on their overall utility (including staleness) while aggregated based solely on their informational contribution, a distinction absent in existing data-aware FL frameworks. Key notations used throughout this paper are summarized in Table 1.

4. Problem Formulation

In vehicular networks, there is significant uncertainty regarding client participation due to high mobility and poor channel conditions. The server must determine which vehicles to schedule and how much bandwidth and quantization precision to allocate. We formulate a dynamic waiting problem that balances system delay against information loss.

Our framework abstracts vehicular mobility through stochastic models, where client availability is governed by Beta distributions and completion times follow Weibull distributions. This approach effectively captures the effects of movement, such as dropouts and latency variance, without the complexity of explicit trajectory tracking. While this abstraction is an appropriate choice for studying data-diversity-aware quantization and availability-aware scheduling, it focuses on the observable patterns that drive resource decisions.

The DRL agent observes these mobility effects indirectly through channel gains $G_{t,i}$ and availability scores $\bar{\theta}_{t,i}$ within the state space. However, this model does not account for explicit vehicular trajectories, handoff events, or coverage transitions. Validating the framework under coordinate-based movement patterns using trajectory-aware simulators remains a subject for future work.

We model the availability of vehicle i in round t using a binary random variable $x_{t,i} \in \{0, 1\}$. Since the true availability probability is unknown, we employ a Bayesian approach treating θ_i as following a Beta distribution, $\theta_i \sim \text{Beta}(\alpha_i, \beta_i)$, where α_i and β_i represent successful participation and dropout counts. The expected availability is:

$$\bar{\theta}_{t,i} = \mathbb{E}[\theta_i] = \frac{\alpha_i}{\alpha_i + \beta_i}. \tag{12}$$

A vehicle that submits an update increments α_i, while a failed submission increments β_i.

The total completion time $T_{t,i}$ of an available vehicle is modeled using a Weibull distribution. The probability that the vehicle fails to finish by time τ is:

$$P(T_{t,i} > \tau \mid x_{t,i} = 1) = e^{-(\tau/\hat{\lambda}_{t,i})^k}. \tag{13}$$

The Weibull shape parameter is set to $k = 1.5$ because it establishes an increasing hazard rate, which logically models the 'aging' process in high-mobility environments where the probability of a vehicle experiencing a link failure or handoff grows as the server waiting time τ increases

The theoretical estimate $\lambda_{t,i}$ combines computation and communication delays:

$$\lambda_{t,i} = \frac{Lc_i|D_{t,i}|}{f_i} + \frac{Q_{t,i}}{r_{t,i}^{\text{up}}}, \tag{14}$$

where $Q_{t,i} = M\tilde{b}_{t,i}$ is the quantized payload size. The empirical scale $\hat{\lambda}_{t,i}$ is initialized from $\lambda_{t,i}$ and updated using an exponential moving average:

$$\hat{\lambda}_{t,i} \leftarrow (1 - \nu)\hat{\lambda}_{t-1,i} + \nu\, \tau_{t,i}^{\text{total}}, \quad \forall i \in R_t, \tag{15}$$

where $\nu \in (0, 1)$ is the learning rate. High availability $\bar{\theta}_{t,i}$ with large $\hat{\lambda}_{t,i}$ identifies a slow but reliable straggler, while low availability identifies a permanent dropout. This probabilistic two-channel characterization enables the system to make nuanced scheduling decisions that existing fixed-timeout or feedback-based approaches cannot achieve.

For asynchronous late arrivals, we use a continuous-valued success weight based on normalized lateness:

$$w_j = \exp\left(-\frac{\tau_j^{\text{late}} - \tau_t^*}{\tau_t^*}\right). \tag{16}$$

This scale-normalized weighting treats reliability as a spectrum: slightly late arrivals receive partial credit, while heavily delayed

arrivals receive a much smaller correction. In Algorithm 1, w_j is used only to update reliability priors for late clients, while the corresponding late model update is still excluded from aggregation due to staleness.

The server determines an optimal waiting time τ_t^* by minimizing a cost function balancing elapsed time against expected information loss:

$$J(\tau) = \tau + \eta \sum_{i \in \mathcal{S}_t} I_{t,i}^{\text{sel}} \cdot \bar{\theta}_{t,i} \cdot P(T_{t,i} > \tau \mid x_{t,i} = 1), \tag{17}$$

where η balances latency against information loss. We fix $\eta = 15$: in (17), larger η weights expected information loss from unfinished clients more heavily and typically pushes the optimizer toward longer deadlines (higher wall-clock per round), whereas much smaller η favors shorter waits but risks truncating slow yet high-utility participants. Preliminary runs placed $\eta = 15$ near the knee of the accuracy-latency trade-off, where additional waiting yielded diminishing marginal accuracy gains relative to round duration. Substituting the Weibull survival function, the optimization problem becomes:

$$\tau_t^* = \arg\min_{\tau} \quad \tau + \eta \sum_{i \in \mathcal{S}_t} I_{t,i}^{\text{sel}} \bar{\theta}_{t,i} e^{-(\tau/\hat{\lambda}_{t,i})^k} \tag{18}$$

s.t.

$$\tau, \eta, \hat{\lambda}_{t,i}, k > 0 \quad \forall i \tag{19}$$

$$\bar{\theta}_{t,i}, I_{t,i}^{\text{sel}} \in [0, 1] \quad \forall i \tag{20}$$

We solve this using Brent's method with complexity $O(m|S_t|)$, where m is small, making overhead negligible.

5. DQS-FeDRL Framework

In this section, we present the proposed DQS-FeDRL framework, which hierarchically performs data-diversity-aware quantization, Bayesian availability-aware semi-synchronous scheduling, and DRL-based resource allocation in vehicular federated learning. Fig. 1 provides an overview of the interaction between the central DRL agent and the vehicular environment.

5.1. Framework Overview

The proposed DQS-FeDRL framework consists of three tightly integrated modules operating between the vehicular clients and the edge server:

- Client-Side Vehicles: Vehicles compute both diversity indices to quantify dataset utility. The selection index $I_{t,i}^{\text{sel}}$ incorporates cohort-relative label novelty $N_{t,i}^{\text{lbl}}$ to assign bit-widths, preserving novel informational value while compressing redundant updates. Simultaneously, the aggregation index $I_{t,i}^{\text{agg}}$ determines the weighting of successful updates to ensure the global model prioritizes clients with the highest marginal contribution.

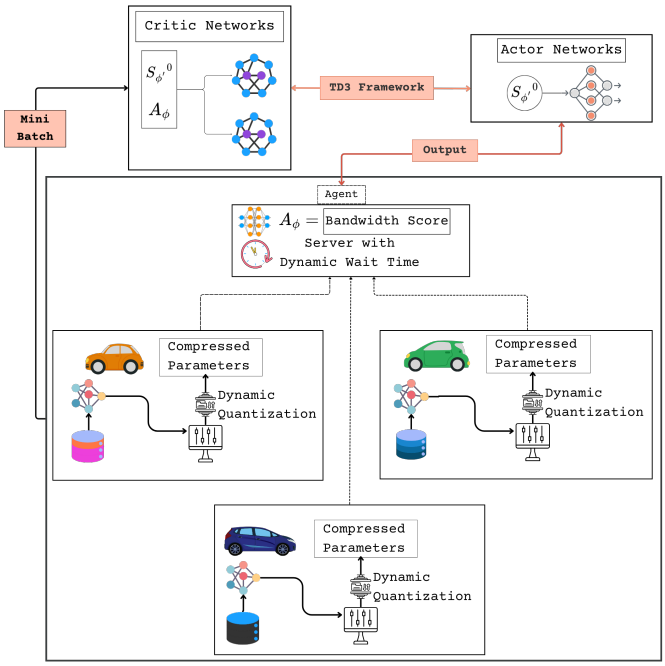


Figure 1: DQS-FeDRL framework

- Bayesian Availability Tracker: To mitigate the impact of mobility-induced dropouts, the server maintains a probabilistic belief of each client's reliability using Beta distributions. This allows the system to distinguish between genuine stragglers and permanent dropouts.

- DRL-Agent: A TD3-based agent running on the server acts as the central scheduler. It observes a composite state aggregated from clients, including channel conditions, data diversity, and reliability scores, to optimize bandwidth allocation and client selection.

These components work in a semi-synchronous loop where the server dynamically optimizes the waiting time τ_t^* to balance global model convergence speed with information loss. Unlike prior work that treats quantization, scheduling, and aggregation as independent design choices, DQS-FeDRL tightly couples these decisions through the cohort-relative diversity signal, enabling the system to adapt resource allocation in real-time based on the evolving informational needs of the training process.

In our proposed framework, we decide the quantization precision based on each client's data diversity. Vehicles with more diverse and informative datasets are assigned higher bit-widths for better accuracy, while those with redundant data are compressed more aggressively to save bandwidth and energy. This behavior is controlled by the selection diversity index $I_{t,i}^{\text{sel}} \in [0, 1]$ introduced in Section 3.3. The key novelty is that $I_{t,i}^{\text{sel}}$ incorporates cohort-relative label novelty, ensuring that bit-width allocation reflects not just local data characteristics but the client's marginal contribution to the current training cohort.

The cohort-relative label novelty metric $N_{t,i}^{\text{lbl}}$, which is embedded in both diversity indices ($I_{t,i}^{\text{sel}}$ and $I_{t,i}^{\text{agg}}$) is a metric flows through the system: vehicles compute it locally, while the TD3

agent incorporates it as a state feature to determine the priority scores for client selection and bandwidth allocation, the quantization module uses it to assign bit-widths (Eq. 22), the deadline optimizer uses it to balance waiting time (Eq. 18), and the aggregator uses it to weight updates (Eq. 1). By sharing this single signal across all decision points, the framework ensures that resource allocation consistently reflects marginal informational value rather than absolute diversity.

5.2. Data Diversity Aware Quantization

To map $I_{t,i}^{\text{sel}}$ to a bit-width we first apply a logarithmic stretching that gives more emphasis to high-diversity clients while keeping the result in $[0, 1]$:

$$\tilde{I}_{t,i} = \frac{\log(1 + \mu I_{t,i}^{\text{sel}})}{\log(1 + \mu)}, \tag{21}$$

where $\mu > 0$ is a shaping parameter. In this way, clients with highly diverse data obtain a stronger advantage in the bit-width allocation.

The quantization bit-width for vehicle i is then chosen by a simple interpolation between a minimum and maximum value:

$$\tilde{b}_{t,i} = \text{round}\,(b_{\min} + (b_{\max} - b_{\min})\,\tilde{I}_{t,i}), \tag{22}$$

where $b_{\min}$ and $b_{\max}$ are predefined integers. Since $\tilde{I}_{t,i} \in [0, 1]$, the bit-width $\tilde{b}_{t,i}$ automatically lies in the interval $[b_{\min}, b_{\max}]$.

Given $\tilde{b}_{t,i}$, vehicle i quantizes its local model update $w_{t,i}$ before transmission. For each parameter block, the client computes its minimum and maximum values, denoted by $w_{\min}$ and $w_{\max}$, and applies a uniform scalar quantizer with $2^{\tilde{b}_{t,i}}$ levels between these extremes. Each floating-point weight w is mapped to a quantization index via stochastic rounding. First, we normalize the weight to the quantization range:

$$\alpha = (2^{\tilde{b}_{t,i}} - 1)\frac{w - w_{\min}}{w_{\max} - w_{\min}}, \tag{23}$$

The discrete quantization index is then obtained stochastically:

$$q = \begin{cases} \lceil \alpha \rceil & \text{with probability } \{\alpha\} \\ \lfloor \alpha \rfloor & \text{otherwise} \end{cases}, \tag{24}$$

where $\{\alpha\} = \alpha - \lfloor \alpha \rfloor$ is the fractional part of α. This unbiased quantization scheme satisfies $\mathbb{E}[q] = \alpha$, preventing bias accumulation during federated training.

This reduces payload size in proportion to $\tilde{b}_{t,i}$, lowering uplink latency and energy. Our framework employs per-layer quantization blocks with $L = 5$ layers, each receiving a scaling pair $(w_{\min}, w_{\max})$. We use a single client-level bit-width across all layers to minimize metadata overhead (~45 bytes/round) in resource-constrained environments. The stochastic rounding ensures unbiased quantization, preventing error accumulation.

On the server side, the received values are dequantized by reversing the mapping. Using the transmitted $(w_{\min}, w_{\max})$, each index q is converted to an approximate value

$$\hat{w} = w_{\min} + q\,\frac{w_{\max} - w_{\min}}{2^{\tilde{b}_{t,i}} - 1}, \tag{25}$$

yielding a reconstructed update $\hat{w}_{t,i}$ for vehicle i. The de-quantized updates from all vehicles that complete within the current waiting window are then aggregated using the diversity-weighted rule defined in (1), where clients with larger aggregation diversity indices $I_{t,i}^{\text{agg}}$ are assigned higher aggregation weights. In this way, the data-diversity-aware quantization and de-quantization steps tightly link communication cost and learning utility by allocating more bits (via $I_{t,i}^{\text{sel}}$) and more influence (via $I_{t,i}^{\text{agg}}$) to the most informative vehicles. Naturally, this also introduces a controlled loss of accuracy for aggressively compressed low-diversity clients, which is the trade-off we accept in exchange for reduced communication overhead.

The proposed adaptive quantization logic also facilitates a synergistic interplay with privacy-preserving requirements. By assigning higher bit-widths to updates with high $I_{t,i}^{\text{sel}}$, the framework effectively creates a utility-driven priority map for potential noise injection. In environments requiring differential privacy, stronger stochastic noise can be applied to heavily quantized, low-diversity updates with minimal impact on global accuracy. Conversely, high-utility updates identified by our novelty metric are preserved with the precision necessary to maintain model convergence. Thus, our quantization approach serves as a first-level filter, ensuring that any added privacy mechanisms prioritize the most informative marginal updates.

5.3. Federated DRL-Based Resource Allocation

We extend our previous federated deep reinforcement learning (DA-FeDRL) framework [13], in which scheduling and resource allocation in VEC were modeled as a Markov Decision Process (MDP) and solved using a TD3-based actor-critic network. We retain the core learning pipeline of the DA-FeDRL framework but extend the state space, improve the action design, and redefine the reward function to incorporate client availability while avoiding redundant payload penalization. The key distinctions from DA-FeDRL are: (1) the diversity signal now captures cohort-relative novelty $N_{t,i}^{\text{lbl}}$ via two separate indices ($I_{t,i}^{\text{sel}}$ and $I_{t,i}^{\text{agg}}$) for selection and aggregation, (2) we introduce diversity-weighted aggregation (Eq. 1) that prioritizes informational contribution over dataset size, (3) we couple this with data-diversity-aware quantization that allocates bit-widths based on marginal information value, and (4) we integrate Bayesian availability-aware semi-synchronous scheduling to handle mobility-induced stragglers and dropouts.

The state space is modified to include the availability probability $\bar{\theta}_{t,i}$ from (12) and the dataset size $|D_i|$ so that the scheduler can favor clients that are both informative and likely to respond in time, while also accounting for computational load to prevent stragglers. At round t, the state of vehicle i is defined as

$$S_{t,i} = [\, I_{t,i}^{\text{sel}},\, G_{t,i},\, r_{t,i},\, \bar{\theta}_{t,i},\, |D_i|\,], \tag{26}$$

where $I_{t,i}^{\text{sel}}$ is the selection diversity index, $G_{t,i}$ is the channel gain, $r_{t,i}$ is the uplink rate, and $|D_i|$ is the dataset size. The term $\bar{\theta}_{t,i}$ is the reliability score calculated via (12), representing the probability that client i will successfully return its update. By incorporating dataset size, the DRL agent can allocate more bandwidth to clients with both high diversity and large datasets,

ensuring they complete training without becoming stragglers that delay the entire system.

The actor network π_ϕ outputs a continuous proto-action for each device, given by

$$a_{t,i} = \pi_\phi(S_{t,i}) + \mathcal{N}_t, \tag{27}$$

where $\mathcal{N}_t$ is the exploration noise. In our prior work, the DRL agent produced both selection indicators and bandwidth scores. Here, we streamline the design by interpreting $a_{t,i}$ as a single priority score that determines both selection and allocation. Specifically, the server selects the subset $\mathcal{S}_t$ of the top-K clients with the highest scores. The total bandwidth B is then allocated among these selected clients proportional to their scores:

$$B_{t,i} = B \cdot \frac{\exp(a_{t,i})}{\sum_{j\in\mathcal{S}_t} \exp(a_{t,j})}, \quad \forall i \in \mathcal{S}_t. \tag{28}$$

This normalization ensures that bandwidth is distributed dynamically, granting larger shares to higher-priority clients while strictly satisfying the capacity constraint $\sum B_{t,i} \le B$. Unlike equal bandwidth division, the DRL agent optimally allocates resources to minimize overall system latency and energy consumption while maximizing data diversity. For example, a slow client with high-diversity data receives more bandwidth to ensure timely completion, preventing it from becoming a straggler, whereas a faster client with redundant data is assigned less bandwidth, enabling the system to collectively minimize round latency without sacrificing informative updates.

To avoid an over-compressed MDP, we enforce a strict separation of concerns: the TD3 priority score $a_{t,i}$ is used only for top-K client selection and bandwidth allocation. Quantization is computed independently from the diversity index $I^{\text{sel}}_{t,i}$, and semi-synchronous scheduling (round deadline τ^*_t) is handled by a deterministic Bayesian-Weibull scheduler via (18), conditioned on $\mathcal{S}_t$, $B_{t,i}$, and $r^{\text{up}}_{t,i}$.

To balance diversity utility against communication and system costs while maintaining the diversity–efficiency trade-off structure from our prior work, we redefine the reward at round t as

$$R_t = \frac{\Gamma(a_t)}{\sum_{i\in\mathcal{S}_t} (c_E E^{\text{total}}_{t,i} + c_T \tau^{\text{total}}_{t,i}) + \epsilon_r} - \rho \, \|a_t\|_2^2, \tag{29}$$

where $\Gamma(a_t) = \log(1 + \sum_{i\in\mathcal{S}_t} I^{\text{sel}}_{t,i})$ is the data diversity factor defined in our prior work, representing the cumulative utility of the selected subset $\mathcal{S}_t$. $E^{\text{total}}_{t,i}$ and $\tau^{\text{total}}_{t,i}$ are the energy and delay costs. The term $\epsilon_r > 0$ is a tiny constant that keeps division by zero from happening. The coefficients c_E and c_T are scaling factors that normalize the energy and time (latency) components, since these have different units and value ranges. Although both transmission delay ($\tau^{\text{up}}_{t,i}$) and transmission energy ($E^{\text{up}}_{t,i}$) scale with payload size ($Q_{t,i}$), we keep them as separate penalties because they constrain orthogonal physical domains: delay controls system-level synchronization (straggler impact), while energy controls client-level sustainability (battery usage). The regularization term $\rho\|a_t\|_2^2$ penalizes overly sharp scheduling patterns and encourages smoother bandwidth allocation. Together, these terms allow the DRL agent to prioritize clients that offer high selection diversity $I^{\text{sel}}_{t,i}$ and reliability relative to their combined energy and latency costs.

Algorithm 1 DQS-FeDRL Algorithm

Initialize: Global model w_0, Actor π_ϕ, Critics Q_{ψ_1}, Q_{ψ_2}, Replay Buffer $\mathcal{B}$.
Initialize: Priors $\alpha_i = 1, \beta_i = 1$ and empirical scales $\hat{\lambda}_{0,i} \leftarrow \lambda_{0,i}$ for all $i \in \mathcal{N}$.
for round $t = 1, \ldots, T$ **do**
  {*Server Side (Scheduling & Resource Allocation):*}
  for each client $i \in \mathcal{N}$ **do**
    Compute reliability score $\bar{\theta}_{t,i} = \alpha_i/(\alpha_i + \beta_i)$ using (12).
    Observe state $S_{t,i} = [I^{\text{sel}}_{t,i}, G_{t,i}, r_{t,i}, \bar{\theta}_{t,i}, |D_i|]$.
  end for
  Generate action scores $a_{t,i} = \pi_\phi(S_{t,i}) + \mathcal{N}_t$ for each client $i \in \mathcal{N}$.
  Select active set $\mathcal{S}_t$ (top-K scores) and allocate bandwidth $B_{t,i}$ via (28).
  Compute optimal wait time τ^*_t via (18).
  Broadcast global model w_{t-1} to clients in $\mathcal{S}_t$.
  {*Client Side (Local Training & Quantization):*}
  for each client $i \in \mathcal{S}_t$ **in parallel do**
    Determine bit-width $\bar{b}_{t,i}$ based on selection diversity $I^{\text{sel}}_{t,i}$ using (22).
    Perform local training: $w_{t,i} \leftarrow \text{SGD}(w_{t-1}, D_i)$.
    Compute quantized indices $q_{t,i}$ from $w_{t,i}$ using (24).
    Upload quantized indices $q_{t,i}$ to server.
  end for
  {*Server Side (Aggregation & Update):*}
  Wait for duration τ^*_t.
  Receive set of successful updates $\mathcal{R}_t \subseteq \mathcal{S}_t$.
  for each client $i \in \mathcal{R}_t$ **do**
    Reconstruct approximate update $\hat{w}_{t,i}$ from $q_{t,i}$ using (25).
  end for
  Aggregate global model w_t using (1).
  {*Synchronous Reliability Update:*}
  for each client $i \in \mathcal{S}_t$ **do**
    if $i \in \mathcal{R}_t$ **then**
      Update priors: $\alpha_i \leftarrow \alpha_i + 1$.
      Update $\hat{\lambda}_{t,i}$ via (15).
    else
      Penalize straggler/dropout: $\beta_i \leftarrow \beta_i + 1$.
    end if
  end for
  {*Asynchronous Soft-Penalty Feedback:*}
  Receive late updates $\mathcal{L}_t$ from previous rounds.
  for each late client $j \in \mathcal{L}_t$ **do**
    Discard stale gradients $\hat{w}_j$ to preserve model stability.
    Compute w_j via (16) from lateness $\tau^{\text{late}}_j - \tau^*_t$.
    Update priors: $\alpha_j \leftarrow \alpha_j + w_j, \beta_j \leftarrow \max(1, \beta_j - w_j)$.
    Update $\hat{\lambda}_{t,j}$ via (15), treating τ^{late}_j as $\tau^{\text{total}}_{t,j}$.
  end for
  Compute reward R_t via (29).
  Store transition (S_t, a_t, R_t, S_{t+1}) in replay buffer $\mathcal{B}$ for mini-batch updates.
  Sample mini-batch from $\mathcal{B}$ and update TD3 actor and critic networks.
end for

The TD3 scheduler uses an actor network π_ϕ with two fully-connected hidden layers of 256 neurons each with ReLU activations, followed by a Tanh output layer that produces a single normalized priority score in $[-1, 1]$ per client. The twin critic networks Q_{ψ_1} and Q_{ψ_2} share the same architecture: the state and action are concatenated and passed through two fully-connected layers with 256 neurons and ReLU activations, followed by a linear output layer that returns the scalar Q-value. All networks are implemented in PyTorch with the default Kaiming-uniform initialization for linear layers.

After each federated round t, once the global reward R_t is

computed from (29), if there are at least 100 transitions available, we sample a mini-batch of 64 transitions to update the TD3 agent after storing the transition (S_t, a_t, R_t, S_{t+1}) in a replay buffer of size 10^5. The critic networks are updated at every round, while the actor is updated every 2 critic updates (policy delay). Target networks are updated using Polyak averaging with coefficient $\tau = 0.005$. Both actor and critics use Adam with learning rate 3×10^{-4} and discount factor $\gamma = 0.99$. During action selection, we add Gaussian exploration noise $\mathcal{N}(0, 0.1)$ to the actor output, and employ target policy smoothing with Gaussian noise of standard deviation 0.2 clipped to $[-0.5, 0.5]$ when computing target Q-values. The TD3 agent is warmed up for 1,000 rounds prior to the main training runs.

5.4. DQS-FeDRL Algorithm

The complete workflow of the proposed framework is summarized in Algorithm 1. The process integrates data-diversity-aware quantization, Bayesian availability tracking, and DRL-based resource allocation into a unified semi-synchronous protocol.

DQS-FeDRL Algorithm is written in causal round order, but control is hierarchical: TD3 learns only the priority scores used for top-K selection and bandwidth allocation, bit-widths follow $I^{\text{sel}}_{t,i}$ through a fixed mapping, and τ^*_t is set by deterministic deadline optimization. The remaining steps implement semi-synchronous aggregation and reliability tracking, and the scalar reward supervises TD3 from the resulting latency and energy outcomes rather than treating quantization or the cutoff as policy outputs. Quantization and the temporal deadline are therefore induced protocols, which stabilizes training while keeping exploration focused on orchestrating heterogeneous clients under tight uplink budgets.

At the start of each round t, the server collects state information from active clients, including channel quality $G_{t,i}$ and local diversity indices $I^{\text{sel}}_{t,i}$ and $I^{\text{agg}}_{t,i}$. The Bayesian availability module first updates the reliability scores $\bar{\theta}_{t,i}$ using the accumulated history (α_i, β_i). These inputs are then fed into the TD3-based actor network to determine the bandwidth priority scores $a_{t,i}$, which are used to select the active subset $\mathcal{S}_t$ and allocate bandwidth resources $B_{t,i}$.

Once selected, clients perform local training and compress their model updates using the diversity-aware bit-width determined in (22) based on $I^{\text{sel}}_{t,i}$. Simultaneously, the server estimates the latency scale parameters $\lambda_{t,i}$ via (14) and solves the optimization problem (18) using $I^{\text{sel}}_{t,i}$ to derive the optimal cutoff time τ^*_t. The server waits for duration τ^*_t; any updates received before this deadline are considered successful, while late updates are excluded from aggregation to prevent model staleness.

Upon the deadline expiry, the server de-quantizes the received updates and aggregates them using the diversity-weighted rule defined in (1). Following aggregation, the availability priors are updated based on participation: successfully received updates increment α_i, while missed updates increment β_i.

Unlike rigid binary updates, Eq. (16) lets DQS-FeDRL acknowledge near-on-time clients without accepting stale gradients into aggregation. This preserves strict semi-synchronous convergence while improving the Bayesian tracker's ability to distinguish temporary stragglers from permanent dropouts.

Finally, the DRL agent computes the reward R_t, stores the transition (S_t, a_t, R_t, S_{t+1}) in its replay buffer, and updates the actor and critic networks to refine the scheduling policy for future rounds.

Through this iterative process, DQS-FeDRL creates a self-optimizing feedback loop where resource allocation is driven by real-time data utility and client reliability.

6. Experimental Setup

We evaluate DQS-FeDRL against several federated learning baselines that span synchronous, semi-asynchronous, and quantized communication frameworks, all within a shared vehicular-edge simulation. The subsections below specify the environment, datasets, baselines, training protocol, and evaluation metrics.

6.1. Simulation Environment and Datasets

The FL environment is implemented using Flower Framework with PyTorch for local training and the TD3-based reinforcement learning and model evaluation. The simulations were conducted on a workstation equipped with AMD Ryzen 7 processor and NVIDIA RTX 3060 GPU, alongside Apple M2 hardware.

We utilize two standard benchmarks for image classification suitable for a vehicular network setting: CIFAR-10 and the German Traffic Sign Recognition Benchmark (GTSRB). The CIFAR-10 is a standard baseline for FL comparisons, while GTSRB is specifically chosen as a domain-specific dataset that is highly relevant to vehicular applications, as it involves traffic sign recognition, which is directly consistent with our vehicular network setup and represents realistic tasks that autonomous vehicles would perform.

The training data is partitioned among $N = 50$ vehicles using a Dirichlet distribution with concentration parameter $\alpha = 0.1$ to simulate non-IID data heterogeneity representative of vehicular settings. The value $N = 50$ reflects a typical upper bound on the number of active vehicles simultaneously participating in FL within a single RSU coverage area under moderate vehicular densities. This introduces severe data heterogeneity and label skew across clients that captures the diverse data distribution characteristics expected in vehicular networks.

We simulate a VEC network with OFDMA communication, Rayleigh fading channels with dynamic gains $G_{t,i}$, and bandwidth $B = 10$ MHz. Device heterogeneity includes CPU frequencies $f_i \in [1, 4]$ GHz, computational complexity $c_i \in [1, 10] \times 10^6$ cycles/sample, and Weibull-distributed ($k = 1.5$) straggler behavior.

6.2. Baseline Selection and Comparative Setup

To evaluate DQS-FeDRL, we compare it against three established frameworks that address isolated challenges in vehicular networks, such as heterogeneity, communication bottlenecks, and straggler effects.

Table 2: Simulation Parameters and Network Configurations

Category	Parameter	Value
VEC Environment	Total Clients (N)	50
	Communication-Bandwidth (B)	10 MHz
	Weibull Shape Factor (k)	1.5
Federated Learning	Local Training Epochs (E)	1
	Mini-batch Size	32
	Dirichlet Non-IID Factor (α)	0.1
DQS-FeDRL	Quantization Bit-width Range	[8, 16] bits
	Bit-width Shaping Factor (μ)	2.0
	Deadline Trade-off Factor (η)	15
	I^{sel} Weights (w_h, w_s, w_u)	0.4, 0.4, 0.2
	I^{agg} Weights ($\tilde{w}_h, \tilde{w}_s$)	0.5, 0.5
	Diversity fusion factor (ξ)	0.5
TD3 Scheduler	Actor/Critic Learning Rate	3×10^{-4}
	Discount Factor (γ)	0.99
	Client Selection (K)	10
Availability	Bayesian EMA coefficient (ν)	0.1
Wireless Channel	Transmit Power ($P_{t,i}$)	20–30 dBm
	Noise PSD (N_0)	−174 dBm/Hz

- FedProx: A synchronous protocol that introduces a proximal term ($\mu = 0.001$) to the local objective. This baseline serves to evaluate standard methods designed for system heterogeneity without the use of data compression or dynamic scheduling.
- FedPAQ: A communication-efficient framework utilizing fixed 20-bit quantization ($s = 2^{20}$) and periodic averaging. It acts as a benchmark for static quantization schemes that do not adapt to the informational value of client updates.
- SAFA: A semi-asynchronous protocol featuring a cache mechanism to mitigate straggler impacts. While it relaxes synchronization barriers, it lacks adaptive quantization and Bayesian-driven deadline optimization, relying instead on a fixed lag tolerance.

All models are trained using a common architecture: a custom CNN comprising 3 convolutional and 2 fully connected layers with approximately 629k parameters. Training is conducted via Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a learning rate of 0.01 to ensure a uniform experimental baseline. Detailed environmental and network configurations are summarized in Table 2.

6.3. Evaluation Metrics

All results are averaged across ten simulation runs. We evaluate the framework using:

1. Global Model Accuracy vs. Communication Rounds and Wall-Clock Time: Measures convergence speed and real-world training efficiency. For DQS-FeDRL, round time is determined by the optimal deadline τ_t^*; for baselines, by the slowest client or fallback timeout $T_{\max} = 20$ seconds.
2. Cumulative Energy Consumption: Total energy consumed by all vehicles, computed as the sum of computation ($E_{t,i}^{\text{cmp}}$) and transmission ($E_{t,i}^{\text{up}}$) energy.
3. Communication Overhead to Target Accuracy: Total data volume (GB) transmitted until the model reaches and sustains target accuracy (80% for CIFAR-10, 85% for GTSRB) for 10 consecutive rounds.
4. Time to Reach Target Accuracy: Wall-clock time required to reach and sustain the target accuracy threshold for 10 consecutive rounds.
5. Ablation Study: Contribution of individual components analyzed by disabling each module and measuring performance degradation together with intermediate DRL-FedAvg and DQS-SAFA configurations.
6. Parameter sensitivity analysis: We conduct a granular sensitivity analysis on four core parameters to justify their selection through empirical trade-offs.

7. Results & Analysis

We evaluate the proposed DQS-FeDRL framework across two datasets (CIFAR-10 and GTSRB). In addition to comparing against standard baselines, we report intermediate configurations (DRL-FedAvg and DQS-SAFA) and per-module ablations on GTSRB (Table 4) to isolate the contributions of TD3 selection, diversity-aware quantization, and Bayesian deadline control under a uniform protocol.

7.1. Illustrative Example

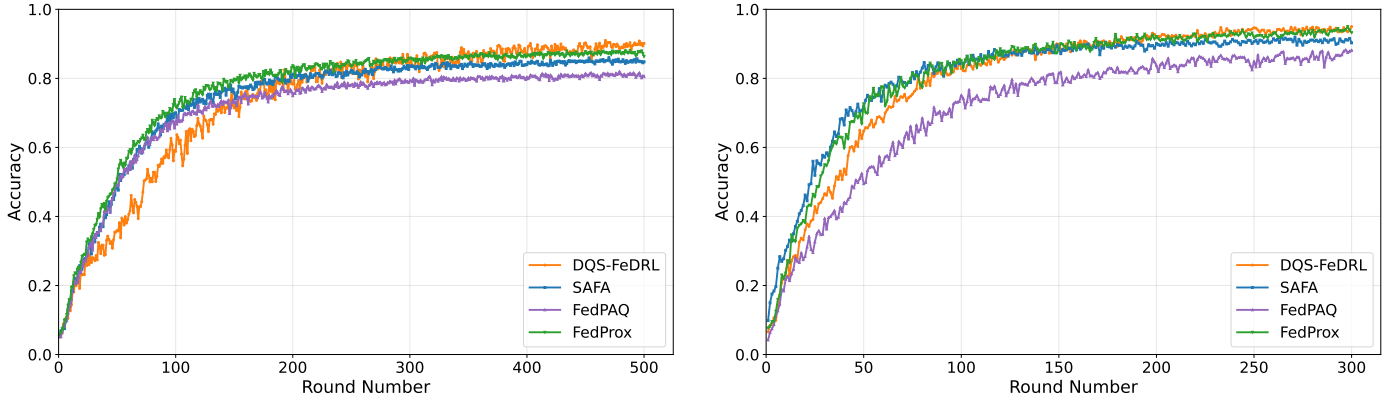
To demonstrate how DQS-FeDRL adapts to heterogeneous compute and communication conditions, we simulate a single training round with five representative vehicles.

Table 3: Numerical illustration of client-level allocation and latency parameters.

Client	$I_{t,i}^{\text{sel}}$	$b_{t,i}$ (bits)	$\theta_{t,i}$	$B_{t,i}$ (MHz)	$\lambda_{t,i}$ (s)
1	0.10	9	0.05	1.13	11.040
2	0.30	11	0.30	1.41	3.120
3	0.50	13	0.55	1.72	2.998
4	0.70	14	0.70	2.00	2.847
5	0.90	16	0.85	3.74	3.001
Optimal round deadline τ_t^*					**5.974 s**

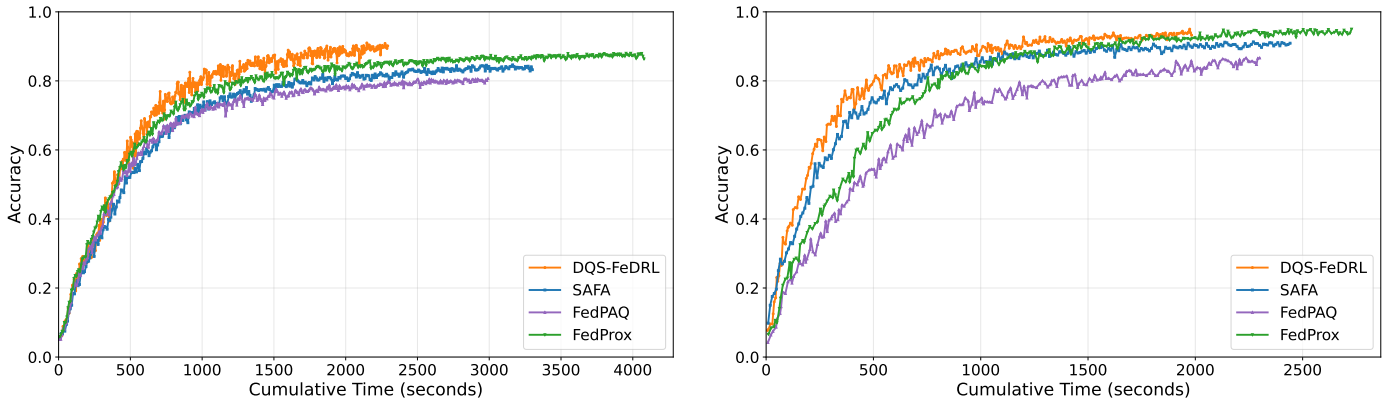
Table 3 presents the selection diversity index, quantization bit-width, bandwidth allocation, reliability, and estimated latency for each client with parameters from Table 2. The global deadline τ_t^* is derived by solving the optimization problem in (18).

The quantization mechanism assigns bit-widths proportional to selection diversity: Client 1 receives 9 bits ($I_{t,1}^{\text{sel}} = 0.10$) while Client 5 gets 16 bits ($I_{t,5}^{\text{sel}} = 0.90$). The DRL scheduler allocates bandwidth accordingly (3.74 MHz vs. 1.13 MHz). Combined with Bayesian reliability scores ($\theta_{t,1} = 0.05$ vs. $\theta_{t,5} = 0.85$) and



(a) CIFAR-10 (b) GTSRB

Figure 2: Model learning accuracy vs. communication rounds.



(a) CIFAR-10 (500 rounds) (b) GTSRB (300 rounds)

Figure 3: Model learning accuracy vs. wall-clock time.

latency estimates ($\lambda_{t,1} = 11.04$s vs. $\lambda_{t,5} = 3.00$s), the optimal deadline $\tau_t^* = 5.974$s excludes Client 1 while aggregating Clients 2–5 using $I_{t,i}^{\text{agg}}$.

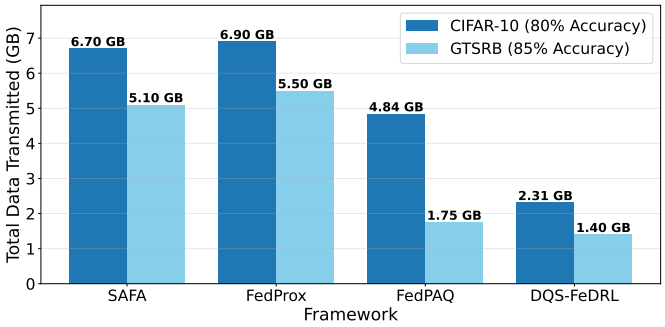
This example also clarifies how to interpret the empirical results that follow. Because DQS-FeDRL enforces a semi-synchronous cutoff τ_t^* and allocates fewer bits to low-utility updates, it may not always appear strictly faster when viewed only as accuracy versus rounds; our primary objective is to reduce wall-clock time, uplink volume, and energy while preserving stable convergence under non-IID data and mobility-induced stragglers.

7.2. *Learning Performance*

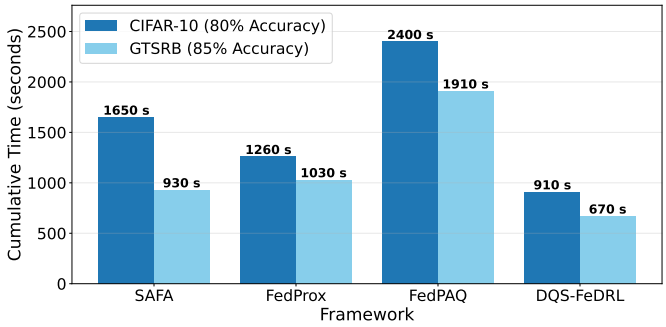
Fig. 2 shows global model learning accuracy over communication rounds for CIFAR-10 (Fig. 2a) and GTSRB (Fig. 2b). On CIFAR-10, DQS-FeDRL achieves 92% final accuracy, compared with SAFA (86%), FedProx (90%), and FedPAQ (82%). DQS-FeDRL reaches 85% accuracy at round 300, later than SAFA and FedProx, because adaptive quantization (Eq. (22)) reduces information from redundant clients and the dynamic deadline τ_t^* (Eq. (18)) excludes stragglers. Despite this slower per-round convergence, DQS-FeDRL achieves higher final accuracy than all baselines. We see similar results on GTSRB. SAFA and FedProx both reach 90% at around round 170, while DQS-FeDRL reaches 90% at round 200 but completes with final accuracy of 95%, matching FedProx and exceeding SAFA (92%). FedPAQ fails to reach 90% on both datasets.

Fig. 3 demonstrates the critical advantage of DQS-FeDRL in wall-clock time, the metric most relevant for practical vehicular deployments. On CIFAR-10 (Fig. 3a), DQS-FeDRL completes 500 communication rounds in around 2250 seconds, achieving final accuracy higher than SAFA and comparable to FedProx. This 1.5–1.8× wall-clock speedup stems from two mechanisms: (1) dynamic deadline optimization (τ_t^*) reduces idle waiting by excluding stragglers that exceed the cutoff, and (2) adaptive quantization minimizes per-round transmission time while maintaining information quality through data-diversity-aware bit-width allocation. In contrast, SAFA requires around 3300 seconds to complete the same 500 rounds, as it relies on full-precision updates without aggressive compression. FedPAQ, despite using fixed 20-bit quantization, requires 3500 seconds due to higher per-round overhead compared to DQS-FeDRL's adaptive 8–16 bit scheme. FedProx, lacking both quantization and dynamic deadline mechanisms, requires over 4000 seconds.

On GTSRB, FedPAQ requires around 2,300 seconds, SAFA

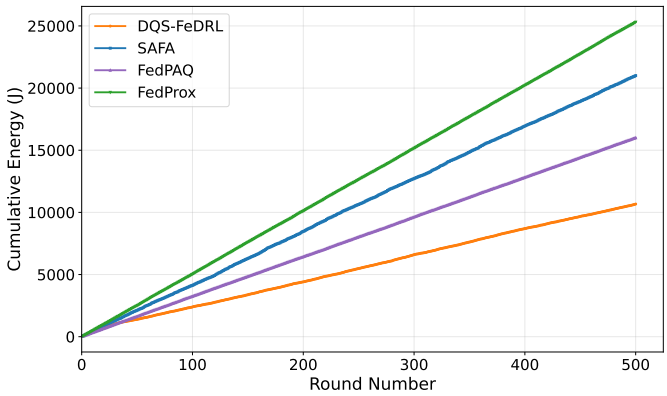


(a) Communication overhead

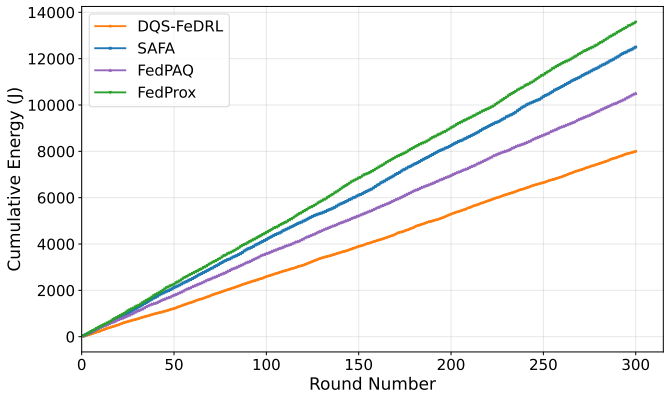


(b) Time to accuracy

Figure 4: Resource efficiency metrics: (a) Communication overhead to reach target accuracy, (b) Time to reach target accuracy.



(a) CIFAR-10



(b) GTSRB

Figure 5: Cumulative energy consumption over rounds.

about 2,450 seconds, and FedProx over 2,700 seconds, so the wall clock gaps versus DQS-FeDRL are smaller than on CIFAR-10 at 500 rounds (about 1.5 to 1.8× on CIFAR-10 as above). At the 80% and 85% accuracy targets, DQS-FeDRL is up to about 2.9× faster than the slowest baseline (Fig. 4(b)). Overall, DQS-FeDRL remains faster in wall clock time, which is the primary concern in vehicular edge computing.

7.3. Resource Efficiency

Fig. 4 presents resource efficiency metrics essential for practical vehicular deployments. Fig. 4(a) shows total data transmitted to reach target accuracy (80% CIFAR-10, 85% GTSRB), while Fig. 4(b) reports wall-clock time required to achieve these thresholds.

On communication overhead,DQS-FeDRL requires 2.31 GB on CIFAR-10 to reach 80% accuracy, substantially less than SAFA (6.70 GB), FedProx (6.90 GB), and FedPAQ (4.84 GB). On GTSRB, DQS-FeDRL transmits 1.40 GB to reach 85% accuracy compared to SAFA (5.10 GB), FedProx (5.50 GB), and FedPAQ (1.75 GB), yielding roughly 3×–4× communication reduction over the synchronous baselines (SAFA, FedProx) and still a noticeable gain over FedPAQ.

On wall-clock time to accuracy, DQS-FeDRL reaches 80% on CIFAR-10 in 910 seconds, compared to 1,650 seconds (SAFA), 1,260 seconds (FedProx), and 2,400 seconds (FedPAQ). On GTSRB, DQS-FeDRL achieves 85% in 670 seconds versus 930 seconds (SAFA), 1,030 seconds (FedProx), and 1,910 seconds (FedPAQ), delivering consistent 1.4–2.9× wall-clock speedup across datasets.

The improvements come from adaptive quantization assigning bit-widths based on data diversity, keeping high-quality clients at full precision while compressing redundant ones. The dynamic deadline (Eq. (18)) also helps by cutting off stragglers instead of waiting for everyone. SAFA shortens rounds by relaxing synchronization, but it still transmits full-precision updates and relies on a fixed lag tolerance with cached stale models, which increases communication and energy and limits final accuracy. In contrast, DQS-FeDRL combines data-diversity-aware quantization with an optimized deadline τ_t^* to avoid stale gradients and compress redundant clients, explaining its consistently lower time, communication, and energy for comparable or better accuracy.

Fig. 5 demonstrates DQS-FeDRL's energy efficiency advantage. On CIFAR-10, DQS-FeDRL consumes roughly 2× less energy than FedProx and SAFA, and about 1.5× less than FedPAQ. On GTSRB, the advantage is similar, with DQS-FeDRL using around 1.6–1.7× less energy than FedProx and approximately 1.5× less than SAFA and FedPAQ.

This energy efficiency stems from adaptive quantization reducing transmission energy through data-diversity-aware bit-

width assignment, combined with the DRL reward function explicitly minimizing energy consumption. By optimizing these objectives, DQS-FeDRL selects low-energy, high-diversity clients while excluding expensive stragglers, achieving substantial efficiency gains.

7.4. Component Analysis and Fairness Evaluation

To rigorously evaluate the isolated contributions of individual components within DQS-FeDRL, and to ensure a strictly fair comparison against standard baselines, we conduct an expanded component analysis on the GTSRB dataset (Table 4). We adopt a two-pronged approach. First, we introduce two intermediate configurations: DRL-FedAvg (applying our TD3 scheduler without quantization) and DQS-SAFA (equipping the SAFA protocol with our adaptive quantization) to isolate how our mechanisms enhance existing protocols. Second, we perform a systematic ablation by independently disabling adaptive quantization, Bayesian availability-aware deadline optimization, and DRL-based client selection from the fully integrated framework. This allows us to quantify both the standalone value of each mechanism and their necessary synergy.

Table 4: Ablation Study

Configuration	**Time (s) to 90%**	**Rounds to 90%**	**Data (GB)**	**Avg. Lat. (s)**
Baseline fairness configurations				
DRL-FedAvg	1,780	220	6.10	8.09
DQS-SAFA	1,495	300	3.05	4.98
Component ablations				
w/o Dynamic Quantization	1,050	165	4.80	6.36
w/o Dynamic Wait Time	1,560	195	1.50	8.00
w/o DRL Selection	1,280	250	2.90	5.12
DQS-FeDRL	**950**	**200**	**1.60**	**4.75**

Evaluating the intermediate DRL-FedAvg configuration isolates the specific impact of the TD3 selection agent when applied to a standard FL protocol. By enabling intelligent client selection without adaptive quantization or dynamic deadlines, the system successfully improves non-IID convergence (requiring 220 rounds to reach 90% accuracy) as the agent learns to prioritize faster, high-utility clients. However, the absence of data compression results in a massive communication overhead (6.1 GB), and the lack of a Bayesian deadline leaves the system vulnerable to straggler-induced delays, pushing the total wall-clock time to 1,780 s. This confirms that while intelligent selection improves data quality, it cannot overcome the physical communication bottlenecks of vehicular networks on its own.

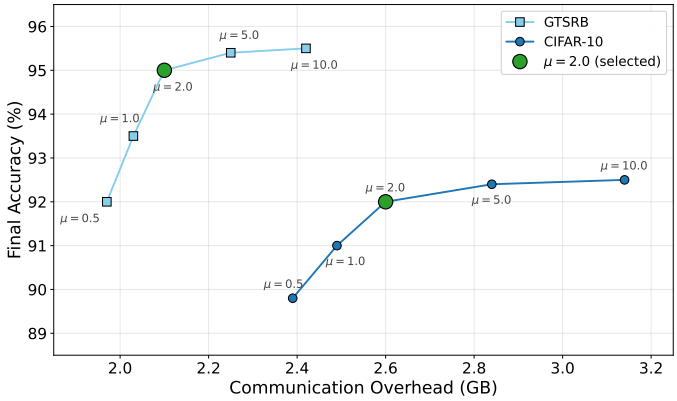
Similarly, the DQS-SAFA configuration isolates the benefits of adding our data-diversity-aware quantization to an existing semi-asynchronous protocol. By equipping SAFA with adaptive bit-widths, the communication footprint drops significantly to 3.05 GB, and the average per-round latency improves to 4.98 s. However, because this configuration lacks the DRL agent for intelligent selection, it struggles with severe non-IID data, requiring 300 rounds to converge. Furthermore, relying on SAFA's cache mechanism rather than our Bayesian deadline optimization introduces stale gradients that hinder overall learning efficiency, resulting in a wall-clock time of 1,495 s. This demonstrates that while adaptive quantization effectively minimizes bandwidth, it must be paired with optimal scheduling and selection to prevent stale or redundant updates from degrading global convergence.

Removing adaptive quantization (*i.e.*, using fixed 32-bit floats) exposes the tension between per-round convergence and wall-clock time. Despite reaching 90% accuracy faster in terms of rounds (165 vs. 200), the wall-clock time is still roughly 10% higher (1,050 s vs. 950 s) as it transmits 4.80 GB versus DQS-FeDRL's 1.6 GB. Although compression slightly slows per-round convergence by introducing quantization noise, the 3.0× data reduction more than compensates. More importantly, this ablation reveals a critical interaction within the framework: without adaptive bit-widths, the payload of every client increases, which directly inflates transmission times. This inadvertently forces the downstream deadline solver to either wait significantly longer or drop valuable updates, demonstrating that minimizing transmission footprints is a strict prerequisite for efficient scheduling in bandwidth-constrained vehicular networks.

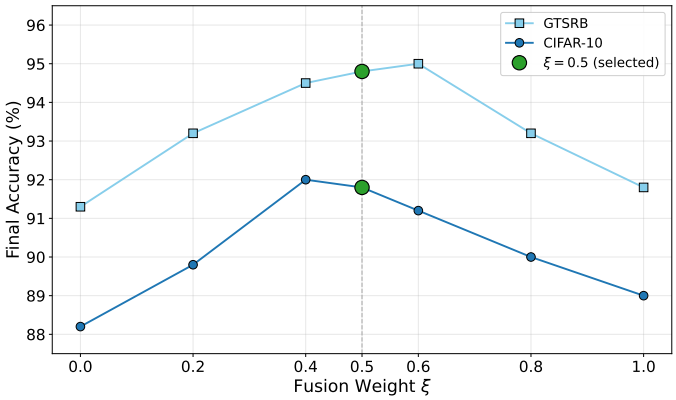
Disabling Bayesian deadline optimization (*i.e.*, reverting to a fixed timeout) has minimal convergence impact (195 rounds) but causes the largest latency penalty: 64% higher wall-clock time (1,560 s vs. 950 s) and an 8 s average latency per round. Without the Weibull survival probabilities to intelligently cut off stragglers, the server is essentially held hostage by the slowest vehicle in the active set. Communication overhead remains low (1.50 GB) since quantization is still active, but excessive idle waiting dominates the training time. This highlights that simply compressing data is insufficient if the system cannot maintain a probabilistic belief to distinguish between transient network latency and a high statistical likelihood of permanent dropout.

Replacing DRL-based diversity selection with random sampling degrades convergence most severely: 25% more rounds (250 vs. 200), 35% higher wall-clock time (1,280 s vs. 950 s), and a 1.8× data overhead (2.90 GB vs. 1.6 GB). This sharp decline occurs because random sampling breaks the framework's dependency chain. Without the DRL agent evaluating $I_{t,i}^{\text{sel}}$, $\bar{\theta}_{t,i}$, and channel conditions, the system frequently selects unreliable vehicles or those with redundant data. Consequently, the adaptive quantization module is forced to waste bandwidth allocating bits to low-utility updates.

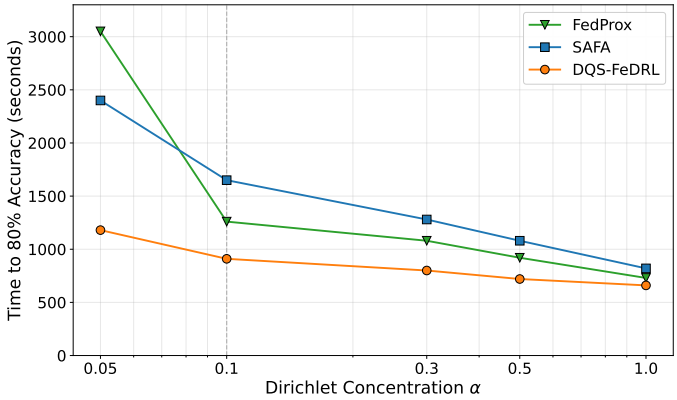
Ultimately, these results validate the necessity of all three components and their tight synergistic loop. Intelligent DRL-based selection ensures only high-utility, reliable vehicles are chosen; adaptive quantization minimizes the communication footprint of those selected clients based on their cohort novelty; and the Bayesian deadline protects the aggregated round from unpredictable mobility failures without sacrificing participation. Removing any single module collapses this careful balance, proving that these interacting mechanisms are required to achieve efficient resource utilization in dynamic VEC environments. The aggregate trends in Table 4 are consistent with the learning dynamics summarized in Fig. 2 and Fig. 3 and with



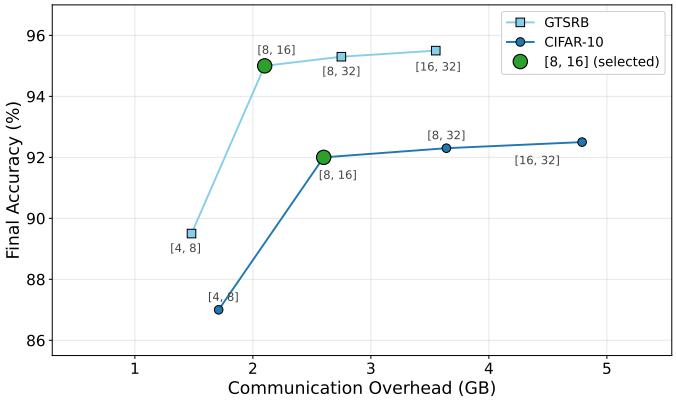
(a) Bit-width shaping factor μ



(b) Fusion weight ξ



(c) Non-IID severity α (CIFAR-10)



(d) Bit-width range $[b_{\min}, b_{\max}]$

Figure 6: Parameter sensitivity analysis on CIFAR-10 and GTSRB. (a) Accuracy–communication trade-off for μ. (b) Accuracy across $\xi \in [0, 1]$. (c) Time to 80% accuracy vs. α for DQS-FeDRL, FedProx, and SAFA on CIFAR-10. (d) Accuracy–communication trade-off for bit-width range.

the resource metrics in Fig. 4.

7.5. Parameter Sensitivity Analysis

To justify our core parameter selections, we conducted sensitivity analysis on four critical parameters that directly influence DQS-FeDRL: the bit-width shaping factor μ, the intra-client heterogeneity and cohort-relative novelty fusion weight ξ, the Dirichlet coefficient that controls the level of data heterogeneity α, and the quantization bit-width range $[b_{\min}, b_{\max}]$. For μ, ξ, and bit-width range, we test on both CIFAR-10 and GTSRB to demonstrate cross-dataset consistency, while for α we compare DQS-FeDRL against FedProx and SAFA on CIFAR-10 across the heterogeneity spectrum. Fig. 6 summarizes all four sweeps in a 2×2 layout.

The parameter μ in Eq. (21) controls the concavity of the diversity-to-bit-width mapping. As $\mu \to 0$ the mapping becomes linear, aggressively differentiating between clients with high diversity and low diversity, while as $\mu \to \infty$ the mapping pushes nearly all clients toward the maximum bit-width regardless of the diversity. Fig. 6a shows the accuracy-communication trade-off for $\mu \in \{0.5, 1.0, 2.0, 5.0, 10.0\}$. At $\mu = 0.5$, moderate-diversity clients are over-compressed, achieving 89.8% accuracy at 2.39 GB on CIFAR-10 and 92.0% at 1.97 GB on GTSRB. At $\mu = 10.0$, the mapping saturates, achieving 92.5% (3.14 GB) and 95.5% (2.42 GB) respectively. Our selected value $\mu = 2.0$ represents the Pareto elbow on both datasets (92.0% at 2.60 GB on CIFAR-10, 95.0% at 2.10 GB on GTSRB): increasing μ beyond 2.0 yields only 0.5% additional accuracy at 15–21% higher communication cost, while decreasing μ sacrifices 2–3% accuracy for only marginal (6–8%) communication savings.

The fusion weight ξ in Eq. (7) balances cohort-relative label novelty $N_{t,i}^{\text{lbl}}$ with local Gini-Simpson heterogeneity $H_{t,i}$. Fig. 6b shows accuracy across $\xi \in [0, 1]$. At $\xi = 0$ (pure cohort-novelty), accuracy drops to 88.5% on CIFAR-10 and 91.5% on GTSRB because the cohort distribution $\bar{p}_t$ is unstable during early training rounds, making JSD-based selection noisy. At $\xi = 1.0$, accuracy reaches only 89.0% and 92.0% respectively because the framework over-prioritizes locally diverse but globally redundant clients. Both datasets exhibit an inverted-U shape but with slightly different optima: CIFAR-10 peaks at $\xi = 0.4$ with 92.3% accuracy, reflecting its globally redundant class structure where cohort-novelty matters more, while GTSRB peaks at $\xi = 0.6$ with 95.2% accuracy, where the highly distinctive traffic-sign classes make local heterogeneity slightly more informative. We adopt $\xi = 0.5$ as a balanced midpoint that performs within 0.3% of the dataset-specific optimum on both datasets (92.0% on CIFAR-10 and 95.0% on GTSRB), avoiding the need for per-deployment tuning. This validates that cohort-relative novelty

and local heterogeneity provide complementary signals, and that an equal-weight fusion serves as a reliable default across diverse vehicular learning tasks.

To evaluate the framework's behavior across data heterogeneity levels, we test DQS-FeDRL alongside FedProx and SAFA with $\alpha \in \{0.05, 0.1, 0.3, 0.5, 1.0\}$ on CIFAR-10. Fig. 6c shows the time required to reach 80% accuracy. Under extreme non-IID conditions ($\alpha = 0.05$), DQS-FeDRL completes in 1,180s, while FedProx requires 3,050s and SAFA 2,400s, yielding 2.58× and 2.03× speedups respectively. Notably, FedProx underperforms SAFA at this extreme heterogeneity level: the proximal term struggles to reconcile radically divergent local gradients, and the synchronous protocol incurs prohibitive waiting times when client completion variance is high. At our selected $\alpha = 0.1$, DQS-FeDRL completes in 910s versus 1,260s (FedProx) and 1,650s (SAFA). Even under near-IID conditions ($\alpha = 1.0$), DQS-FeDRL retains a 1.11× speedup over FedProx, confirming that DQS-FeDRL's gains are strongest where most needed and remain non-detrimental under homogeneous conditions.

For the quantization bit-width range $[b_{\min}, b_{\max}]$, we evaluate four ranges on both datasets: $[4, 8]$, $[8, 16]$, $[8, 32]$, and $[16, 32]$ bits. Fig. 6d shows the accuracy-communication trade-off. The $[4, 8]$ range over-compresses all clients, achieving only 87.0% (CIFAR-10) and 89.5% (GTSRB) accuracy, confirming that 4-bit quantization introduces excessive gradient noise that stochastic rounding cannot fully compensate for. The $[16, 32]$ range maintains 92.5% and 95.5% accuracy but consumes 1.69×–1.84× more communication than $[8, 16]$, defeating the purpose of adaptive compression. The $[8, 32]$ range offers marginal 0.3% accuracy gains at 31–40% higher communication cost. We select $[8, 16]$ as it achieves 92.0% (2.60 GB) on CIFAR-10 and 95.0% (2.10 GB) on GTSRB, representing the Pareto-optimal balance for resource-constrained vehicular networks.

These results confirm that the selected configuration ($\mu = 2.0$, $\xi = 0.5$, $\alpha = 0.1$, $[b_{\min}, b_{\max}] = [8, 16]$) represents principled choices grounded in empirical trade-offs rather than arbitrary tuning.

8. Conclusion

The findings of this work demonstrate that the efficiency of vehicular federated learning is fundamentally constrained by the dynamic tension between communication overhead and data utility. By introducing the cohort-relative label novelty metric $N_{t,i}^{\text{lbl}}$, we have shown that quantization and scheduling should not be treated as independent design choices, but as part of a coordinated resource allocation problem that adapts to a client's marginal informational value and network conditions.

Our evaluation against standard baselines and intermediate configurations, proved that neither availability-aware scheduling nor adaptive quantization is sufficient on its own to handle extreme non-IID vehicular data. The synergy within DQS-FeDRL enables the system to reinvest bandwidth saved from redundant updates into the most informative and reliable participants. Achieving up to a 2.9× speedup in wall-clock convergence, a 52–75% reduction in communication overhead, and a 33–50% lowering of energy consumption, this framework provides an efficient architecture for deploying privacy-preserving intelligence in resource-constrained, dynamic vehicular environments.

While DQS-FeDRL demonstrates strong empirical results at the evaluated scale (50 vehicles), scalability to larger vehicular networks remains a limitation as the DRL agent's state and action spaces grow substantially. Our current simulation does not model explicit vehicular trajectories, handoff events, or coverage transitions; we abstract mobility through stochastic dropout and latency models. Integration with trajectory-based simulators such as SUMO or Veins represents important future work to validate the framework under realistic vehicular movement patterns. The Bayesian availability model may also exhibit limitations under extreme volatility where past reliability history becomes less predictive. Hierarchical DRL architectures and trajectory-aware availability prediction represent promising directions for future investigation.

Acknowledgment

The authors are grateful to King Saud University, Riyadh, Saudi Arabia for funding this work through the Ongoing Research Funding Program- ORF-2026-18.

References

[1] W. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, Future Generation Computer Systems 97 (2019) 219–235.

[2] M. Ahmed, S. Raza, M. Mirza, A. Aziz, M. Khan, W. Khan, J. Li, Z. Han, A survey on vehicular task offloading: Classification, issues, and challenges, Journal of King Saud University - Computer and Information Sciences 34 (2022) 4135–4162.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of AISTATS, 2017, pp. 1273–1282.

[4] D. Maroua, A state-of-the-art on federated learning for vehicular communications, Vehicular Communications 45 (2024) 100709.

[5] C. Smestad, J. Li, A systematic literature review on client selection in federated learning, in: Proceedings of EASE, 2023, pp. 2–11.

[6] T. Rafi, F. Noor, T. Hussain, D.-K. Chae, Fairness and privacy preserving in federated learning: A survey, Information Fusion 105 (2024) 102198.

[7] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, R. Pedarsani, Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization, in: Proceedings of AISTATS 2020, Vol. 108 of PMLR, 2020, pp. 2021–2031.

[8] Y. Li, Y. Guo, M. Alazab, S. Chen, C. Shen, K. Yu, Joint optimal quantization and aggregation of federated learning scheme in vanets, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 19852–19863.

[9] P. Zheng, Y. Zhu, Y. Hu, Z. Zhang, A. Schmeink, Federated learning in heterogeneous networks with unreliable communication, IEEE Transactions on Wireless Communications 23 (2024) 3823–3838.

[10] C. Xu, Y. Qu, Y. Xiang, L. Gao, Asynchronous federated learning on heterogeneous devices: A survey, Computer Science Review 50 (2023) 100595.

[11] W. Wu, L. He, W. Lin, R. Mao, C. Maple, S. Jarvis, Safa: A semi-asynchronous protocol for fast federated learning with low overhead, IEEE Transactions on Computers 70 (2021) 655–668.

[12] B. Zheng, B. Rao, T. Zhu, C. W. Tan, J. Duan, Z. Zhou, X. Chen, X. Zhang, Online location planning for ai-defined vehicles: Optimizing joint tasks of order serving and spatio-temporal heterogeneous model fine-tuning, IEEE Transactions on Mobile Computing 25 (2) (2026) 1777–1794.

[13] R. Ullah, I. Chowdhury, M. Razzaque, P. Roy, M. Hassan, Federated deep reinforcement learning for optimal resource allocation in vehicular edge computing, in: Proceedings of Intelligent Computing (CompCom 2025), Vol. 1425 of LNNS, 2025.

[14] Q. Xia, W. Ye, Z. Tao, J. Wu, Q. Li, A survey of federated learning for edge computing: Research problems and solutions, High-Confidence Computing 1 (2021) 100008.
[15] Z. Du, C. Wu, T. Yoshinaga, K.-L. Yau, Y. Ji, J. Li, Federated learning for vehicular internet of things: Recent advances and open issues, IEEE Open Journal of the Computer Society 1 (2020) 45–61.
[16] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks, IEEE Transactions on Industrial Informatics 17 (2021) 5098–5107.
[17] S. Pokhrel, J. Choi, Federated learning with blockchain for autonomous vehicles: Analysis and design challenges, IEEE Transactions on Communications 68 (2020) 4734–4746.
[18] S. Liu, J. Yu, X. Deng, S. Wan, Fedcpf: An efficient-communication federated learning approach for vehicular edge computing in 6g communication networks, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 1616–1629.
[19] S. Liu, P. Guan, J. Yu, A. Taherkordi, Fedssc: Joint client selection and resource management for communication-efficient federated vehicular networks, Computer Networks 237 (2023) 110100.
[20] A. Ramezani-Kebrya, F. Faghri, I. Markov, V. Aksenov, D. Alistarh, D. Roy, Nuqsgd: Provably communication-efficient data-parallel sgd via nonuniform quantization, Journal of Machine Learning Research 22 (2021) 1–30.
[21] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, H. Li, Terngrad: Ternary gradients to reduce communication in distributed deep learning, in: Advances in Neural Information Processing Systems 30, 2017.
[22] S. Shen, C. Yu, K. Zhang, X. Chen, H. Chen, S. Ci, Communication-efficient federated learning for connected vehicles with constrained resources, in: Proc. IWCMC, 2021, pp. 1636–1641.
[23] M. Amiri, D. Gündüz, Federated learning over wireless fading channels, IEEE Transactions on Wireless Communications 19 (2020) 3546–3557.
[24] N. Shlezinger, M. Chen, Y. Eldar, H. Poor, S. Cui, Uveqfed: Universal vector quantization for federated learning, IEEE Transactions on Signal Processing 69 (2021) 500–514.
[25] H. Tang, K. Zhang, J. Zhang, X. Xie, X. Tong, X. Liu, Aqmfl: An adaptive quantization framework for multi-modal federated learning in heterogeneous edge devices, in: Proceedings of IEEE ISPA, 2024, pp. 98–105.
[26] F. Lai, X. Zhu, H. Madhyastha, M. Chowdhury, Oort: Efficient federated learning via guided participant selection, in: Proceedings of OSDI, 2021, pp. 19–35.
[27] Y. Deng, F. Lyu, J. Ren, H. Wu, Y. Zhou, Y. Zhang, X. Shen, Auction: Automated and quality-aware client selection framework for efficient federated learning, IEEE Transactions on Parallel and Distributed Systems 33 (8) (2022) 1996–2009.
[28] D. Ami, K. Cohen, Q. Zhao, Client selection for generalization in accelerated federated learning: A bandit approach, in: Proceedings of ICASSP, 2023, pp. 1–5.
[29] J.-H. Ahn, Y. Ma, S. Park, C. You, Federated active learning: An efficient annotation strategy for federated learning, IEEE Access 12 (2024) 39261–39269.
[30] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Differentially private asynchronous federated learning for mobile edge computing in urban informatics, IEEE Transactions on Industrial Informatics 16 (2020) 2134–2143.
[31] T. Zang, C. Zheng, S. Ma, C. Sun, W. Chen, A general solution for straggler effect and unreliable communication in federated learning, in: Proc. IEEE ICC, 2023, pp. 1194–1199.
[32] L. Yu, X. Sun, R. Albelaihi, C. Yi, Latency-aware semi-synchronous client selection and model aggregation for wireless federated learning, Future Internet 15 (2023) 352.
[33] N. Lang, A. Cohen, N. Shlezinger, Stragglers-aware low-latency synchronous federated learning via layer-wise model updates, IEEE Transactions on Communications 73 (2025) 3333–3346.
[34] A. Mohajer, J. Hajipour, V. C. M. Leung, Dynamic offloading in mobile edge computing with traffic-aware network slicing and adaptive td3 strategy, IEEE Communications Letters 29 (1) (2025) 95–99.
[35] A. Mohajer, A. Mirzaei, M. Darabi, X. Fernando, Joint sla-aware task offloading and adaptive service orchestration with graph-attentive multi-agent reinforcement learning, IEEE Transactions on Network and Service Management 23 (2026) 3737–3754.
[36] Q. Song, J. Yang, A. Mohajer, Multi-objective resource optimization in uav-enabled heterogeneous cellular networks using serverless federated learning and power-domain noma, Transactions on Emerging Telecommunications Technologies 36 (8) (2025) e70210.
[37] Y. Jiang, X. Tong, Z. Liu, X. Zhang, K.-Y. Lam, C. W. Tan, Certifying the right to be forgotten: Primal–dual optimization for sample and label unlearning in vertical federated learning, IEEE Transactions on Information Forensics and Security 20 (2025) 13143–13158.