Expert Systems With Applications 318 (2026) 131992

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Robust federated learning for dynamic and heterogeneous environments: an adaptive client selection approach

Rui Chen, Dongyang Bao, Ning Lu, Jing Zhao *

School of Software Technology, Dalian University of Technology, 116024, Dalian, China

ARTICLE INFO

Keywords:
Heterogeneous federated learning
Adaptive client selection
Truth discovery
Hidden markov model
Dynamic optimization

ABSTRACT

While federated learning (FL) offers a privacy-preserving framework for collaborative training, the process of selecting which clients participate is a pivotal factor that dictates the system's ability to handle real-world dynamics and heterogeneity. However, existing methods suffer from two critical limitations that undermine selection reliability: the inability to accurately estimate true client value amidst complex, multi-faceted errors, and the overreliance on suboptimal, heuristic-based strategies that lack theoretical guarantees and risk critical information loss. To address these challenges, we propose FedTD-HMM, an adaptive client selection framework integrating Truth Discovery (TD) with Hidden Markov Models (HMM) to systematically resolve the dual challenges of client quality assessment and dynamic strategy adjustment in heterogeneous FL. First, we employ TD to counteract biased estimations by iteratively estimating client quality and an ideal aggregated update, using cosine similarity to estimate the directional consistency among multi-dimensional client updates. Subsequently, we model the training process using an HMM with four operational states, applying the Viterbi algorithm for real-time inference of current training states and adaptive adjustment of selection weights. This enables an adaptive selection strategy that optimizes decisions for the current state of training, transforming the process from a static configuration into a dynamic, state-aware optimization that significantly enhances system robustness and convergence. Finally, experiments on real-world datasets demonstrate that FedTD-HMM outperforms state-of-the-art baselines, achieving up to 3.11% improvement in test accuracy while reducing communication rounds by up to 31.4%, under challenging conditions with 50% low-quality clients. Code is available at https://github.com/NihaoRay/FedTD-HMM.

1. Introduction

Federated Learning (FL) (Seo et al., 2024; Tang et al., 2024) enables multiple clients to collaboratively train models on distributed data while ensuring that data remains stored locally and is privately accessible (You et al., 2025). Compared to traditional distributed optimization, FL offers distinct advantages in privacy preservation and communication efficiency, making it ideal for extracting knowledge from privacy-sensitive data at the edge (Gao et al., 2023; Li et al., 2024; Zhang et al., 2023). Consequently, FL has been widely deployed in data-sensitive domains such as smart homes, financial services, and healthcare (Kashyap et al., 2025; Moorthy et al., 2023; Wu et al., 2024; Xue et al., 2024). However, as FL scales to massive numbers of clients, the central server coordinating the process faces significant challenges, including communication bottlenecks and high latency, necessitating an effective client selection mechanism (Chen et al., 2024; Trindade et al., 2024).

Effective client selection is crucial for mitigating these issues, preventing stragglers from degrading performance, and ensuring model accuracy (Cai et al., 2025; Hu et al., 2025; You et al., 2023). The complexity of this task is magnified by the systemic heterogeneity inherent in FL. data that is not independently and identically distributed (Non-IID) (Wang et al., 2020). Furthermore, in dynamic settings like vehicular networks, varying communication capabilities and intermittent connectivity introduce spatiotemporal dynamics that impede stable updates (Luo et al., 2022; Wang et al., 2023). The asynchronous nature of client participation further compounds these issues, leading to disparities in data richness and update staleness, which can cause inefficient aggregation and model overfitting (Sun et al., 2025; Zhang et al., 2023).

In such heterogeneous and dynamic environments, selecting an optimal client subset presents two fundamental challenges that existing methods fail to adequately address. First, client value assessment models suffer from systematic biases. Current approaches, ranging from

* Corresponding author.
E-mail addresses: 32317044@mail.dlut.edu.cn (R. Chen), bdy631@mail.dlut.edu.cn (D. Bao), luning0408@mail.dlut.edu.cn (N. Lu), 72117004@mail.dlut.edu.cn (J. Zhao).

https://doi.org/10.1016/j.eswa.2026.131992
Received 25 September 2025; Received in revised form 17 February 2026; Accepted 5 March 2026
Available online 12 March 2026

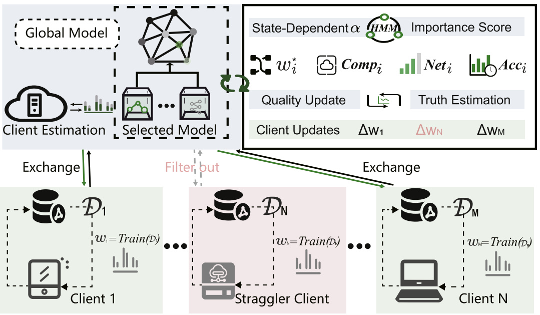


Fig. 1. High-level overview of the FedTD-HMM framework. The server-side pipeline consists of two synergistic stages: a Truth Discovery module for multi-dimensional client quality assessment, followed by an HMM-based module that infers the current training phase and adapts the selection strategy accordingly.

random sampling (Liang et al., 2022, 2025) to basic Truth Discovery (TD) strategies (Li et al., 2025; Xu et al., 2022), struggle to accurately estimate the quality of client contributions. They often treat errors in client uploads as a monolithic entity, failing to capture their complex origins. In reality, these errors arise from multiple overlapping factors: intrinsic random errors stemming from data quality or computational limitations, and extrinsic uncertainties caused by environmental factors like network jitter or location changes (Vono et al., 2022; Wang et al., 2024; Xu et al., 2022; You et al., 2023). This interplay fundamentally undermines the reliability of existing estimation models.

Second, selection strategies rely excessively on rigid, heuristic rules without theoretical optimality guarantees. Greedy algorithms that rank clients by a single quality metric and eliminate a fixed proportion are common (Lin et al., 2024; Ma et al., 2021). These methods lack the flexibility to adapt the selection scale to different training stages. More critically, their rigid elimination criteria risk creating "incomplete task coverage." Specifically, clients who possess rare but critical data (i.e., "long-tail" clients) may be collectively excluded due to lower overall quality rankings, causing irreversible information loss and leading to a suboptimal global model (Hu et al., 2025; Manzoor et al., 2022). These heuristic methods are not optimized for an explicit objective function related to the global model's performance, rendering their results inherently suboptimal.

To address these two core challenges, we propose FedTD-HMM (illustrated in Fig. 1), an adaptive client selection framework that integrates Truth Discovery (TD) (Xu et al., 2022) with Hidden Markov Models (HMM) (Hu et al., 2025). Our framework employs a two-phase optimization approach. **In the first phase**, to counteract biased assessments, we construct a quality assessment mechanism based on Truth Discovery theory. Instead of treating client errors monolithically, it iteratively estimates client reliability weights and an ideal "ground-truth" aggregated update. By using an improved cosine similarity to measure the directional consistency of each client's update against this evolving consensus, the mechanism robustly identifies genuine contributions amidst noise. This process is further enhanced by integrating multi-dimensional metrics (e.g., accuracy, network quality) (Hu et al., 2025; You et al., 2023) to produce a holistic and reliable quality score for each client.

In the second phase, to replace suboptimal heuristics, we introduce a dynamic, model-driven approach using an HMM. This directly addresses the lack of adaptability and the risk of incomplete task coverage. We model the FL training process as a stochastic system transitioning between hidden states (Hu et al., 2025) (exploration, stable convergence, fine-tuning, and oscillation recovery). Using real-time training dynamics as observations, the Viterbi algorithm (Forney, 2005) infers the current training phase. This inference allows the system to adapt its selection strategy-for instance, prioritizing infrastructure reliability during early exploration to establish a stable training baseline. This transforms client selection from a static procedure into an adaptive optimization process aligned with phase-dependent objectives, providing theoretical interpretability and ensuring robustness.

In summary, the major contributions of this work are as follows

- We propose FedTD-HMM, an adaptive client selection framework that integrates Truth Discovery with Hidden Markov Models to address both client quality assessment and dynamic strategy adjustment in heterogeneous FL environments.
- We design a Truth Discovery-based quality assessment mechanism that simultaneously estimates client reliability weights and ground-truth aggregated updates through iterative optimization. This mechanism employs cosine similarity to measure update consistency and incorporates multi-dimensional metrics (accuracy, network quality, computational efficiency) to generate comprehensive client importance scores.
- We formulate the FL training process as a stochastic state transition system and employ the Viterbi algorithm for phase inference. This enables our framework to dynamically adjust selection strategies according to current training dynamics, preventing premature exclusion of clients with rare but critical data.
- Extensive experiments on six datasets spanning image, audio, and text modalities (MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, Speech Commands, and Shakespeare) demonstrate that FedTD-HMM achi-eves accuracy improvements of 0.49%-3.11% while reducing communication rounds by 13.5%-31.4% compared to state-of-the-art methods, validating its effectiveness in challenging federated scenarios.

2. Related work

Client selection is a critical component of Federated Learning. Existing strategies can be broadly categorized into two families: heuristic-based methods, which use simple, predefined rules, and learning-based methods, which formulate selection as a sequential decision problem. In this section, we review the state-of-the-art in both categories to position our work and highlight the unaddressed challenges that motivate FedTD-HMM.

(1) Heuristic-Based Selection Methods. These methods make selection decisions based on predefined rules, often considering factors like data distribution, system resources, or model update characteristics.

Several works focus on data heterogeneity. For instance, HiCS-FL (Chen et al., 2024) guides client sampling based on data heterogeneity, and FedPHE (Li et al., 2025) uses clustering to group clients with similar data distributions. Others, like FedBSS (Xu et al., 2025), mitigate client drift by focusing on high-loss samples. However, these approaches primarily address only one dimension of client quality-data distribution-while failing to solve the challenge of biased value assessment. They do not provide a mechanism to distinguish whether a client's poor performance stems from intrinsically low-quality data or from extrinsic factors like transient network issues or computational limitations. Consequently, a client with valuable but complex data might be unfairly penalized, leading to suboptimal selection.

Other heuristic methods attempt a more direct quality assessment. TDFL (Xu et al., 2022) applies Truth Discovery (TD) to assign weights based on client contributions and to filter out malicious updates. AiFed (You et al., 2023) adaptively integrates clients based on model staleness and data richness. While these methods move towards a more nuanced evaluation, they still fall short of providing a truly robust assessment. For example, TDFL's TD mechanism, while effective against overt maliciousness, tends to treat client errors as a monolithic block. It does not explicitly model or disentangle the intrinsic random errors from extrinsic uncertainties that we identified in the introduction. This makes its reliability estimation susceptible to systemic bias. Furthermore, the final selection in these frameworks often resorts to a rigid filtering or ranking mechanism (e.g., eliminating the bottom-ranked clients), which exemplifies the second challenge of suboptimal, heuristic-based strategies. Such greedy approaches lack adaptability to the global training phase and risk "incomplete task coverage" by consistently excluding clients that may be temporarily underperforming but hold critical, rare data.

(2) Learning-Based Selection Methods. To overcome the rigidity of heuristics, learning-based methods formulate client selection as a sequential decision-making problem, often using Reinforcement Learning (RL) or state-based models.

RL-based approaches, such as FedPref (Hartmann et al., 2025) (Q-learning), FedMarl (Zhang et al., 2022) (multi-agent RL), and FedDRL (Lin et al., 2024), aim to learn an optimal selection policy over time. While these methods offer greater adaptability, they introduce new problems and, more fundamentally, fail to address our first core challenge: biased value assessment. The primary issue lies in the nature of the reward signal (e.g., global model accuracy improvement). This signal is an aggregated outcome that still cannot disentangle the underlying causes of a client's contribution. For instance, a client might receive a low reward due to temporary network lag, and the RL agent may incorrectly learn that the client is intrinsically low-quality, leading to its persistent exclusion. Consequently, RL policies, while adaptive, are learning from a flawed and biased feedback loop, perpetuating the very assessment problem we aim to solve. Furthermore, their "black-box" nature makes it difficult to interpret why certain clients are selected, hindering trust and debuggability in critical systems.

A more interpretable approach involves using Hidden Markov Models (HMMs). For example, FedClamp (Manzoor et al., 2022) employs them to identify anomalous nodes, and TRAIL (Hu et al., 2025) uses a hidden semi-Markov (McDonald et al., 2024) model to estimate client states. These methods are closer in spirit to our work, as they acknowledge the dynamic nature of the FL process. However, their core limitation lies in a conceptual separation between state modeling and decision-making, which leaves our second challenge-suboptimal selection strategies-unresolved. For instance, after inferring a client's or system's state, methods like TRAIL (Hu et al., 2025) and FedClamp (Manzoor et al., 2022) still revert to a separate, often greedy or heuristic, selection logic (e.g., select clients in the "good" state, or apply a fixed threshold). This means they model the dynamics but then fail to translate this insight into a globally aware, theoretically grounded selection strategy. This two-step process misses the opportunity to create a truly integrated policy that adapts its objectives based on the global training phase, thus risking the exclusion of clients crucial for diversity, especially in the early training phases.

In summary, existing works leave a significant research gap by failing to holistically address the two intertwined challenges of client selection. Heuristic-based methods suffer from both (1) biased assessment due to simplistic metrics and (2) suboptimal, rigid selection rules. Advanced learning-based approaches only offer partial solutions. RL methods, while adaptive, perpetuate biased assessment because their reward signals are ambiguous and cannot disentangle error sources. HMM-based methods like TRAIL and FedClamp, while modeling system dynamics, still rely on suboptimal, decoupled heuristic rules for the final selection.

This creates a clear opportunity for a unified framework. Our proposed FedTD-HMM is designed to bridge this gap by being the first to simultaneously and synergistically address both challenges. It tackles the assessment challenge with a Truth Discovery mechanism for robust, multi-dimensional quality estimation that disentangles client value from transient noise. Concurrently, it resolves the selection challenge by using an HMM not merely to identify a state, but to directly parameterize the selection strategy itself, allowing the framework to dynamically shift its objective-from promoting diversity to prioritizing quality-based on the inferred global training phase. This integrated, "modeling-drives-decision" design moves beyond fixed heuristics to achieve a more robust, efficient, and theoretically grounded client selection.

3. Preliminaries

To facilitate understanding of the mathematical formulation, we summarize the key notations used throughout this paper in Table 1.

3.1. Truth discovery for client selection in FL

FL involves a server coordinating M clients $\mathcal{C} = \{C_1, \dots, C_M\}$ to train a global model via decentralized local updates (McMahan et al., 2017). In each round t, the server selects a subset $\mathcal{S}_t \subseteq \mathcal{C}$ to participate, aiming to optimize convergence and robustness amid challenges like data heterogeneity and adversarial attacks. A truth discovery algorithm enhances this process by estimating client "truth values" $w_i(t) \in \mathbb{R}^+$, quantifying client utility (e.g., gradient quality) or reliability (e.g., update consistency) (Wang et al., 2024).

Formally, let $\mathcal{D}_i$ denote client C_i's local data, and $\mathcal{G}(\Delta w, \mathcal{S}_t)$ the aggregation rule (e.g., FedAvg). The algorithm optimizes a joint objective balancing model performance and truth accuracy:

$$\min_{\{\mathcal{S}_t\}_{t=1}^T} \mathbb{E}_{\substack{\mathcal{S}_t \sim \pi_t \\ \Delta w_t \sim \mathcal{G}}} \left[\mathcal{L}(\Delta w_t; \cup_{i \in \mathcal{S}_t} \mathcal{D}_i) \right] + \lambda \cdot \mathbb{E}_{t=1}^T \left[\| w(\mathcal{S}_t) - \hat{w}(\mathcal{S}_t) \|_2^2 \right] \tag{1}$$

where π_t is the selection policy, $\mathcal{L}$ the global loss, $\lambda > 0$ a regularization parameter (Zou et al., 2005), and $w(\cdot), \hat{w}(\cdot)$ the true and estimated truth values.

Truth values evolve dynamically via $w_i(t+1) = f(w_i(t), \Delta w_i(t), \epsilon_i(t))$, where $\Delta w_i(t)$ is client C_i's update and $\epsilon_i(t)$ encodes perturbations (e.g., data drift, adversarial noise). Selection policies (e.g., bandit-based (Pan et al., 2023)) leverage $\hat{w}_i(t)$ to adapt $\mathcal{S}_t$, ensuring the algorithm prioritizes informative, reliable clients.

3.2. Hidden Markov models

A Hidden Markov Model (HMM) is a statistical model that represents a system with unobservable (hidden) states evolving over time according to a Markov process (Rabiner, 2002). Formally, an HMM is characterized by the tuple λ = (S, O, π, A, B). Here, S = $\{s_1, s_2, \ldots, s_n\}$ denotes the set of hidden states, O = $\{o_1, o_2, \ldots, o_m\}$ represents the set of possible observations. The vector $\pi = \pi_i$ is the initial state probability distribution, where π_i = P($q_1 = s_i$). The matrix A = a_{ij} is the state transition probability matrix, where a_{ij} = P($q_{t+1} = s_j \mid q_t = s_i$). Finally, B = b_j(k) is the observation probability distribution, where b_j(k) = P($o_i \mid q_t = s_j$).

The fundamental assumption of HMMs is the Markov property (Rabiner, 2002), which states that the future state depends only on the current state and not on the sequence of events that preceded it. Additionally, the observation at time t depends solely on the current hidden state, making the model particularly suitable for modeling sequential data with underlying latent dynamics.

3.3. Viterbi algorithm

The Viterbi algorithm (Forney, 2005; Rabiner, 2002) is a dynamic programming approach for finding the most likely sequence of hidden states $Q^* = \{q_1^*, q_2^*, \ldots, q_t^*\}$ given a sequence of observations O = $\{o_1, o_2, \ldots, o_t\}$ and model parameters λ. This problem, known as the decoding problem in HMM literature, seeks to maximize P(Q|O, λ).

The algorithm operates by computing the maximum probability $\delta_t(i)$ of being in state s_i at time t, having observed the partial sequence o_1, o_2, …, o_t:

$$\delta_t(i) = max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_t = s_i, o_1, o_2, \ldots, o_t | \lambda) \tag{2}$$

The recursive computation proceeds as follows. For initialization at $t = 1$, we have $\delta_1(i) = \pi_i b_i(o_1)$ and $\psi_1(i) = 0$. For the recursion step where $2 \le t \le T$, we compute: $\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t)$ and $\psi_t(j) =$

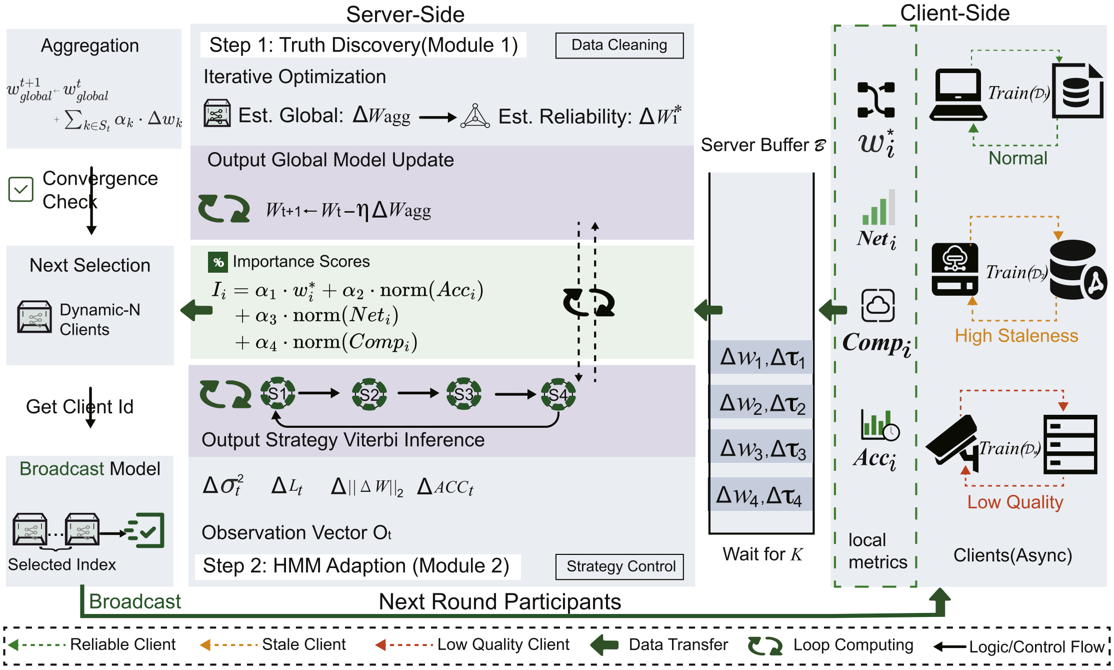

Server-Side
Client-Side
Aggregation
$w^{t+1}_{global} \leftarrow w^{t}_{global} + \sum_{k\in S_t} \alpha_k \cdot \Delta w_k$
Step 1: Truth Discovery(Module 1)
Data Cleaning
Iterative Optimization
Est. Global: ΔW_{agg}
Est. Reliability: ΔW_i^*
Convergence Check
Output Global Model Update
$W_{t+1} \leftarrow W_t - \eta \Delta W_{agg}$
Next Selection
Dynamic-N Clients
Importance Scores
$I_i = \alpha_1 \cdot w_i^* + \alpha_2 \cdot \mathrm{norm}(Acc_i) + \alpha_3 \cdot \mathrm{norm}(Net_i) + \alpha_4 \cdot \mathrm{norm}(Comp_i)$
Get Client Id
S1
S2
S3
S4
Output Strategy Viterbi Inference
Broadcast Model
$\Delta\sigma_t^2$ ΔL_t $\Delta\|\Delta W\|_2$ ΔACC_t
Observation Vector O_t
Selected Index
Step 2: HMM Adaption (Module 2)
Strategy Control
Server Buffer $\mathcal{B}$
$\Delta w_1, \Delta\tau_1$
$\Delta w_2, \Delta\tau_2$
$\Delta w_3, \Delta\tau_3$
$\Delta w_4, \Delta\tau_4$
Wait for K
w_i^*
Net_i
$Comp_i$
Acc_i
local metrics
Train($\mathcal{D}_1$)
Normal
Train($\mathcal{D}_2$)
High Staleness
Train($\mathcal{D}_M$)
Low Quality
Clients(Async)
Broadcast
Next Round Participants
Reliable Client
Stale Client
Low Quality Client
Data Transfer
Loop Computing
Logic/Control Flow

Fig. 2. Detailed architecture of the FedTD-HMM framework (corresponding to Algorithm 1). The framework integrates two core modules: (1) a Truth Discovery (TD) Module that robustly assesses client contribution quality from local updates, and (2) an HMM Adaptive Module that infers the global training phase. The inferred phase then dynamically adapts the client selection strategy for the next round, creating a closed-loop, adaptive system.

Table 1
Notation summary for the FedTD-HMM framework.

Symbol	Description
M	Total number of clients.
Δw_i	Model update submitted by client C_i.
$I_i^{(t)}$	The final importance score of client i at round t.
w_i^*	Converged **contribution quality score** for client i.
Δw_{agg}	The **ideal aggregated update**, co-estimated with quality scores.
$\hat{d}_i$	Combined distance metric of client i's update deviation.
γ	Hyperparameter to balance distance metrics in $\hat{d}_i$.
a_0	Prior parameter for the Dirichlet distribution of client weights.
S	Set of hidden states representing FL training phases.
O_t	Observation vector at round t capturing system dynamics.
A	State transition probability matrix of the HMM.
B	Emission probability distributions of the HMM.
q_t^*	Most likely hidden state (training phase) at round t.
$\vec{\alpha}^{(t)}$	The **adaptive weight vector** for round t, based on state q_t^*.

$\arg\max_{1\le i\le N}\left[\delta_{t-1}(i)a_{ij}\right]$, where $\psi_t(j)$ stores the argument that maximizes the probability, enabling backtracking to reconstruct the optimal path.

The termination step yields: $P^* = \max_{1\le i\le N} \delta_T(i)$ and $q_T^* = \arg\max_{1\le i\le N} \delta_T(i)$. Finally, the path backtracking reconstructs the optimal state sequence by iteratively computing: $q_t^* = \psi_{t+1}(q_{t+1}^*)$ for $t = T-1, T-2, \ldots, 1$.

4. Design of FedTD-HMM

To address the dual challenges of biased client assessment due to conflated error sources and suboptimal selection strategies leading to incomplete task coverage, our framework aligns two synergistic modules directly with these objectives (Fig. 2, Algorithm 1). The first module employs Truth Discovery (TD) (Xu et al., 2022) to decouple intrinsic quality from extrinsic uncertainties (e.g., staleness), ensuring fair assessment. The second module uses a Hidden Markov Model (HMM) (Hu et al., 2025) to infer training phases, dynamically shifting from "diversity-first" during exploration to "efficiency-first" during oscillation. Key notations are in Table 1.

4.1. *Truth discovery-based client quality estimation*

Unlike standard approaches that treat client errors monolithically, this section presents a quality assessment method designed to disentangle intrinsic data quality issues from extrinsic environmental uncertainties in asynchronous FL systems.

4.1.1. *Objective formalization*

The server maintains a buffer $\mathcal{B}$ of the K most recent updates. Let $\Delta w_i \in \mathbb{R}^d$ denote the update from client C_i. We introduce the **reliability weight** w_i and the **ideal aggregated update** Δw_{agg} (estimated ground truth direction), formulating their joint estimation as:

$$\begin{aligned} \min_{w_i, \Delta w_{\text{agg}}} \quad & \sum_{i\in\mathcal{B}} w_i \cdot \mathcal{D}(\Delta w_i, \Delta w_{\text{agg}}) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{B}} w_i = 1, \quad w_i \ge 0, \quad \forall i \end{aligned} \tag{3}$$

where $\mathcal{D}(\cdot,\cdot)$ denotes the distance metric. Stale updates deviating from the consensus direction incur larger penalties, adaptively suppressing the impact of time lag.

4.1.2. *Distance metric and quality modeling*

We adopt a composite metric combining directional alignment and distributional consistency. For directional consistency, we use improved cosine similarity $\text{sim}^*_{\cos}(\Delta w_i,\ \Delta w_{\text{agg}}) = (\Delta w_i \cdot \Delta w_{\text{agg}})^2/(\|\Delta w_i\|^2\|\Delta w_{\text{agg}}\|^2 + \epsilon)$ with distance $d_i = 1 - \text{sim}^*_{\cos}$. To distinguish systematic Non-IID shifts from random noise, we augment this with Rényi divergence $D_2(P_i\|P_{\text{agg}})$ (Van Erven et al., 2014). The **combined distance metric** is:

$$\hat{d}_i = \gamma \cdot d_i + (1-\gamma) \cdot D_2(P_i\|P_{\text{agg}}) \tag{4}$$

where γ balances the two components, allowing the system to penalize poor data quality even when the direction is coincidentally aligned.

4.1.3. *Iterative optimization framework*

Inspired by EM (Dempster et al., 1977), we design an alternating optimization framework.

Truth Estimation: Given weights $\{w_i^{(k-1)}\}$, we solve:

$$\Delta w_{\text{agg}}^{(k)} = \arg\min_{\Delta w} \sum_{i\in\mathcal{B}} w_i^{(k-1)} \cdot \|\Delta w_i - \Delta w\|^2 + \lambda \cdot \Omega(\Delta w) \tag{5}$$

with closed-form solution $\Delta w_{\text{agg}}^{(k)} = \mathcal{S}_\lambda\left(\frac{\sum_{i\in\mathcal{B}} w_i^{(k-1)} \cdot \Delta w_i}{\sum_{i\in\mathcal{B}} w_i^{(k-1)} + \mu}\right)$, where $\mathcal{S}_\lambda(\cdot)$ is the soft-thresholding operator.

Quality Update: Through Bayesian inference with Dirichlet prior $w_i \sim \text{Dirichlet}(a_0)$ (Yang et al., 2021), we derive:

$$w_i^{(k)} = \frac{a_0 - 1 + \exp(-\hat{d}_i^{(k)}/\tau_{temp})}{\sum_{j\in\mathcal{B}}(a_0 - 1 + \exp(-\hat{d}_j^{(k)}/\tau_{temp}))} \tag{6}$$

where τ_{temp} controls sensitivity. Updates with high staleness (large $\hat{d}_i$) receive lower weights, naturally suppressing asynchronous delay effects.

4.1.4. *Multi-dimensional comprehensive assessment*

Finally, to generate the comprehensive score I_i, we integrate the TD-derived reliability (w_i^*) with system metrics:

$$\begin{aligned} I_i &= \alpha_1 \cdot w_i^* \\ &+ \alpha_2 \cdot \text{norm}(Acc_i) \\ &+ \alpha_3 \cdot \text{norm}(Net_i) \\ &+ \alpha_4 \cdot \text{norm}(Comp_i) \end{aligned} \tag{7}$$

This score I_i serves as the basis for the adaptive selection in the next module.

4.2. *Adaptive adjustment of multi-dimensional weights*

The second challenge identified in Section 1 is the rigidity of heuristic selection, which often leads to "incomplete task coverage" by excluding valuable long-tail clients. To address this, we introduce an HMM-based module that adapts the selection weights $\{\alpha_j\}_{j=1}^4$ based on training phase dynamics, balancing exploration (coverage) and exploitation (convergence).

4.2.1. *HMM formal definition*

We model the asynchronous FL process using an HMM tuple $\lambda = (\mathcal{S}, O, A, B, \pi)$:

1) Hidden State Space ($\mathcal{S}$):

We define four states (Hartmann et al., 2025; Lin et al., 2024) corresponding to strategic needs:

- s_1 (*Exploration*): Early stage with high variance, requiring high task coverage to identify long-tail features.
- s_2 (*Stable Convergence*): Model learns consistent patterns; focus shifts to quality exploitation.
- s_3 (*Fine-tuning*): Approaching the optimum, requiring high-precision updates.
- s_4 (*Oscillation/Staleness*): Regression state caused by high latency or non-IID divergence.

2) Observation Space (O): At time t, the observation vector $O_t \in \mathbb{R}^3$ captures system status:

$$O_t = \left[\Delta\mathcal{L}_t, \bar{\tau}_t, \sigma_t^2\right]^T \tag{8}$$

where $\Delta\mathcal{L}_t = \mathcal{L}_{t-1} - \mathcal{L}_t$ indicates learning progress, $\bar{\tau}_t$ is the average gradient staleness (high values indicate outdated updates harming convergence), and σ_t^2 measures client heterogeneity and training instability.

3) Transition Probability Matrix (A): $A = \{a_{ij}\}$ denotes the probability of transitioning from state s_i to s_j, capturing natural progression (e.g., $s_1 \rightarrow s_2$) or regression (e.g., $s_2 \rightarrow s_4$).

4) Emission Probability (B): We model the relationship between hidden states and observations using multivariate Gaussian distributions. For state s_j:

$$\begin{aligned} B_j(O_t) &= P(O_t \mid q_t = s_j) \\ &= \frac{1}{\sqrt{(2\pi)^k|\Sigma_j|}} \exp\left(-\frac{1}{2}(O_t - \mu_j)^T\Sigma_j^{-1}(O_t - \mu_j)\right) \end{aligned} \tag{9}$$

where μ_j and Σ_j are estimated via the Baum-Welch algorithm.

5) Initial State Probability (π): $\pi = \{\pi_i\}$ represents the prior distribution, typically initialized to favor the exploration state (s_1).

4.2.2. *Strategy parameterization and optimization*

The HMM module maps inferred states to optimal client selection strategies via adaptive weight vectors $\vec{\alpha}^{(j)} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$ (Hartmann et al., 2025; Lin et al., 2024). We define the immediate reward function:

$$R_t(\vec{\alpha}) = \underbrace{\Delta Acc_t(\vec{\alpha})}_{\text{Performance Gain}} - \lambda_1 \cdot \underbrace{\bar{\tau}_t(\vec{\alpha})}_{\text{Staleness Penalty}} - \lambda_2 \cdot \underbrace{\sigma_t^2(\vec{\alpha})}_{\text{Instability Penalty}} \tag{10}$$

where λ_1, λ_2 balance performance against staleness and instability. The optimal strategy for state s_j maximizes expected cumulative reward:

$$\vec{\alpha}^{(j)*} = \arg\max_{\vec{\alpha}} \mathbb{E}\left[\sum_{k=0}^{H} \gamma^k R_{t+k} \mid q_t = s_j, \vec{\alpha}\right] \tag{11}$$

Based on this optimization, we define four states:

- **Exploration** (s_1): Prioritizes reliable infrastructure for efficient initialization. **Config:** $\vec{\alpha}^{(1)} \approx [0.1, 0.1, 0.4, 0.4]$ (high weights on Network Stability and Computation Speed to avoid cold-start delays).
- **Stable Convergence** (s_2): Prioritizes high-quality contributions. **Config**: $\vec{\alpha}^{(2)} \approx [0.6, 0.2, 0.1, 0.1]$ (high weight on TD Quality).
- **Fine-tuning** (s_3): Requires high-precision local models. **Config**: $\vec{\alpha}^{(3)} \approx [0.1, 0.7, 0.1, 0.1]$ (high weight on Local Accuracy).
- **Oscillation/Staleness Recovery** (s_4): Restores stability when performance drops. **Config:** $\vec{\alpha}^{(4)} \approx [0.4, 0.1, 0.4, 0.1]$ (high weights on TD Quality and Network Stability to suppress noise).

4.2.3. *Online inference and adaptive updates*

During training, the server infers the current state q_t from observation sequence $\mathbf{O}_{1:t}$ using the Viterbi Algorithm (Forney, 2005). We define $\delta_t(i)$ as the maximum probability of reaching state s_i at time t:

$$\delta_t(i) = \max_{q_1,\ldots,q_{t-1}} P(q_1, \ldots, q_{t-1}, q_t = s_i, O_1, \ldots, O_t|\lambda) \tag{12}$$

The recursive computation is:

$$\delta_t(j) = \left[\max_{1\le i\le N}(\delta_{t-1}(i)a_{ij})\right] \cdot B_j(O_t) \tag{13}$$

where $B_j(O_t)$ is the Gaussian emission probability (Eq. (9)). The current state is $q_t^* = \arg\max_j \delta_t(j)$.

The server then updates client selection scores using the corresponding strategy $\vec{\alpha}^{(t)}$:

$$I_k^{(t)} = \vec{\alpha}^{(t)} \cdot \mathbf{f}_k = \sum_{m=1}^{4} \alpha_m^{(t)} \cdot f_{k,m} \tag{14}$$

This adaptive mechanism ensures that when high staleness is detected (State s_4), the system automatically shifts preference to faster clients (via α_3, α_4), mitigating asynchronous delays. The complete process is described in Algorithm 1.

4.3. *Convergence analysis*

We present a unified convergence analysis that treats FedTD-HMM as a single dynamical process. The analysis focuses on how the client quality assessment (TD) and adaptive strategy (HMM) jointly optimize the convergence bound constants.

Algorithm 1 Adaptive Client Selection with TD-HMM Algorithm.

```
Require: Clients $C$, Max rounds $T_{\max}$, Buffer size $K$, HMM params $\lambda = (A, B, \pi)$
Ensure: Global model $w_{global}$
Initialize: $w_{global}$, Buffer $\mathcal{B} \leftarrow \emptyset$, Iteration $t \leftarrow 0$
while $t < T_{\max}$ do
    Async Receive: Wait for update $(\Delta w_i, \tau_i)$ from any client $C_i$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\Delta w_i, \tau_i)\}$;   Update client status to IDLE
    if $|\mathcal{B}| \geq K$ then                    ▷ Buffer Full: Start Aggregation
        Phase 1: Truth Discovery (Reliability Estimation)
        Initialize reliability weights $\{q_i\}_{i \in \mathcal{B}}$ uniformly
        repeat
            Update global gradient truth: $\Delta w_{agg} \leftarrow \frac{\sum_{i \in \mathcal{B}} q_i \cdot \Delta w_i}{\sum_{i \in \mathcal{B}} q_i}$
            Update client weights: $q_i \leftarrow f_{weight}(||\Delta w_i - \Delta w_{agg}||^2)$
        until Convergence of $\{q_i\}$
        Phase 2: Global Update
        $w_{global} \leftarrow w_{global} - \eta \cdot \Delta w_{agg}$
        $t \leftarrow t + 1$
        Phase 3: HMM-based Strategy Adaptation
        Calculate metrics from $\mathcal{B}$:
            $\Delta \mathcal{L}_t$ (Loss Change), $\bar{\tau}_t$ (Avg Staleness), $\sigma_t^2$ (Gradient Variance)
        Form observation sequence $O_{1:t}$ and estimate current state:
            $s_t^* \leftarrow \text{Viterbi}(O_{1:t}, \lambda)$
        Map state to selection weights: $\vec{\alpha}^{(t)} \leftarrow \text{LookupTable}(s_t^*)$
        Phase 4: Adaptive Client Selection
        for each idle client $C_n \in C_{idle}$ do
            Calculate score: $S_n \leftarrow \vec{\alpha}_1^{(t)} \cdot \text{Perf}_n + \vec{\alpha}_2^{(t)} \cdot \text{Staleness}_n + \dots$
        end for
        Select set $C_{next}$ with highest scores $S_n$ to participate
        $\mathcal{B} \leftarrow \emptyset$                              ▷ Clear Buffer
    end if
end while
return $w_{global}$
```

4.3.1. *Preliminaries and assumptions*

Let $F(w)$ denote the true global objective function. The global model is updated at round t as $w_{t+1} = w_t - \eta_t \Delta w_{\text{agg}}^{(t)}$, where η_t is the learning rate. We rely on the following standard assumptions:

Assumption 1 (L-smoothness). The function F is L-smooth, i.e., $\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\|$.

Assumption 2 (Bounded Effective Variance). The variance of the aggregated update is bounded. Specifically, $\mathbb{E}[\|\Delta w_{\text{agg}}^{(t)} - \mathbb{E}[\Delta w_{\text{agg}}^{(t)}]\|^2] \leq \sigma_{eff}^2$. Here, σ_{eff}^2 represents the *effective* variance after TD-based aggregation.

Assumption 3 (Directional Alignment & Bounded Bias). Due to asynchronous staleness and non-IID data, the expected update direction may deviate from the true gradient. We assume this bias is bounded by $\kappa > 0$ and a bounded bias $\delta_{bias} \geq 0$, such that:

$$\mathbb{E}[\langle \nabla F(w_t), \Delta w_{\text{agg}}^{(t)} \rangle] \geq \kappa \|\nabla F(w_t)\|^2 - \delta_{bias} \tag{15}$$

4.3.2. *Convergence of the unified FedTD-HMM system*

We demonstrate that FedTD-HMM converges to a stationary point, with constants directly linked to system behavior.

Theorem 1. Under Assumptions 1–3, with a constant learning rate η satisfying $\eta \leq \frac{\mu}{L}$, the sequence of global models $\{w_t\}$ satisfies:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t)\|^2] \leq \frac{F(w_0) - F^*}{\eta T \mu} + \underbrace{\frac{L\eta}{2\mu} \sigma_{eff}^2}_{C_1} + \underbrace{\frac{1}{\mu} \delta_{bias}}_{C_2} \tag{16}$$

where F^* is the optimal value. Crucially, C_1 and C_2 are not arbitrary constants but reflect the practical system behavior optimized by our design.

Proof Sketch: Starting from the L-smoothness inequality:

$$\begin{aligned} &\mathbb{E}[F(w_{t+1})] \\ &\leq \mathbb{E}[F(w_t)] - \eta \mathbb{E}[\langle \nabla F(w_t), \Delta w_{\text{agg}}^{(t)} \rangle] \\ &+ \frac{L\eta^2}{2} \mathbb{E}[\|\Delta w_{\text{agg}}^{(t)}\|^2] \end{aligned} \tag{17}$$

Substituting Assumptions 2 and 3 into the inequality, rearranging the terms to isolate $\|\nabla F(w_t)\|^2$, and summing over $t = 0$ to $T - 1$ yields the bound in Eq. (16). The terms associated with variance group into C_1, and terms associated with bias group into C_2.

4.3.3. *Physical interpretation of constants*

To clarify the practical implications of the convergence bound, we explicitly link constants C_1 and C_2 to our proposed modules.

1) Constant C_1 (Gradient Variance) ↔ Truth Discovery Module The term $C_1 \propto \sigma_{eff}^2$ represents the noise level of the aggregated gradient. In standard FedAvg, this variance is high due to low-quality or malicious updates.

- *System Behavior*: Our TD module (Phase 1 in Algorithm 1) computes reliability weights q_i. By performing a weighted aggregation $\Delta w_{\text{agg}} = \sum q_i \Delta w_i$, the system effectively **filters out outliers** that deviate from the consensus.
- *Impact*: This mechanism mathematically minimizes the effective variance ($\sigma_{eff(TD)}^2 < \sigma_{uniform}^2$), directly reducing C_1 and tightening the convergence bound.

2) Constant C_2 (Systematic Bias) ↔ HMM Module The term $C_2 \propto \delta_{bias}$ represents the systematic error introduced by gradient staleness (time lag τ) and data heterogeneity.

- *System Behavior*: In asynchronous FL, high staleness ($\tau \gg 0$) causes the update direction to deviate from the current true gradient, increasing δ_{bias}. Our HMM module (Phase 3) actively monitors the staleness state ($\bar{\tau}_t$). When the system enters the "Staleness" state (s_4), the HMM adjusts selection weights $\vec{\alpha}$ to **prioritize fresh clients** (low τ).
- *Impact*: By dynamically penalizing high-staleness participants, the HMM ensures the aggregated update remains aligned with the current model version. This actively suppresses the bias term δ_{bias}, reducing C_2 and preventing divergence in high-latency environments.

In summary, FedTD-HMM improves convergence not merely through algorithmic techniques, but by structurally reducing the noise (C_1) and lag (C_2) inherent in asynchronous systems.

5. Performance evaluation

To comprehensively evaluate the effectiveness and efficiency of our proposed FedTD-HMM framework, we align our experiments with the theoretical premise established in the introduction: *can a system effectively decouple intrinsic data fidelity from extrinsic environmental uncertainty?* The complete implementation is available at https://github.com/NihaoRay/FedTD-HMM. We design a series of experiments to address the following five core research questions (RQs):

RQ1 (Convergence Efficiency via Uncertainty Decoupling): How does FedTD-HMM perform in terms of convergence speed? Does the mechanism of distinguishing *stale but valid updates* (extrinsic delay) from *low-quality data* (intrinsic error) accelerate the global model's stabilization?

RQ2 (Communication Efficiency): How does FedTD-HMM perform in reducing communication overhead? Specifically, can the system identify "high-information-density" clients to reduce redundant transmissions while achieving **comparable or superior accuracy**?

RQ3 (Robustness): Under varying proportions of low-quality clients (e.g., clients with poor data quality, weak computational capacity, or unstable connections), what advantages does FedTD-HMM demonstrate

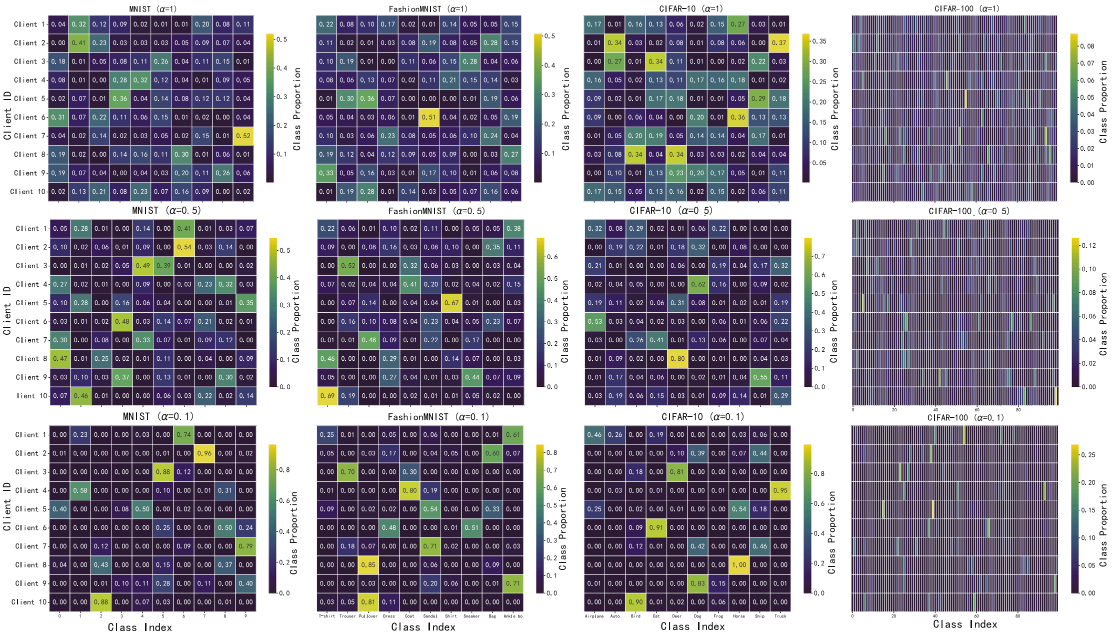


Fig. 3. Client data distribution visualization under varying Non-IID levels ($\alpha = 1.0, 0.5, 0.1$) for MNIST, FMNIST, CIFAR-10, and CIFAR-100. The heatmaps illustrate how decreasing α intensifies data heterogeneity, leading to highly skewed class distributions among clients across all four datasets. Note that these heatmap depict the static data partition initialized prior to the first communication round, which remains constant throughout the training process.

over baseline methods in terms of convergence speed and final accuracy?

RQ4 (Component Contribution): What are the individual contributions of the two core components-the Truth Discovery-based quality assessment mechanism and the HMM-based dynamic adjustment strategy-to the overall performance improvement?

RQ5 (State Inference Accuracy): How accurate is the Hidden Markov Model in inferring training phases? Are the identified training states (exploration state, stable convergence state, fine-tuning state, and oscillation recovery state) consistent with actual training dynamics? Is the timing of state transitions consistent with expected training dynamics?

5.1. Experimental setup

5.1.1. Datasets and model architectures

We evaluate our proposed method on four widely-used benchmark datasets: MNIST (Deng, 2012), Fashion-MNIST (FMNIST) (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), and CIFAR-100 (Krizhevsky et al., 2009). To accommodate the varying complexity of the different datasets, we employ corresponding model architectures: for MNIST and FMNIST datasets, we utilize a lightweight CNN model; for CIFAR-10 and CIFAR-100, we adopt ResNet-18 and ResNet-34 architectures, respectively.

Data Heterogeneity Simulation: To faithfully simulate practical FL scenarios under varying degrees of data heterogeneity, we construct non-IID data partitions using the Dirichlet distribution (Dir(α)), visualized in Fig. 3. Crucially, we interpret these Non-IID settings not just as statistical skew, but as a proxy for **long-tail feature distribution**:

Moderate Non-IID ($\alpha = 1.0$): This setting creates a moderately skewed data distribution where clients hold samples from most classes but in varying proportions.

High Non-IID ($\alpha = 0.5$): This setting generates a more challenging environment where client data distributions are significantly more skewed.

Extreme Non-IID ($\alpha = 0.1$): This setting simulates a severe stress-test scenario where each client's local dataset is dominated by only a few data classes, leading to highly divergent local model updates.

This multi-level evaluation allows us to comprehensively assess the performance of all compared methods as the core challenge of data heterogeneity intensifies. After completing the data partitioning, the resulting partitions are randomly assigned to the 100 clients as their respective local training data.

5.1.2. FL training configuration

Our experimental setup consists of $N = 100$ clients. In each communication round, the system selects $K = 10$ clients (participation rate $C = 10\%$) to participate in training according to different selection strategies. Each selected client performs $E = 14$ epochs of stochastic gradient descent (SGD) optimization on their local dataset, with batch size set to $B = 32$ and learning rate $\eta = 0.015$. The global model is trained for a total of $T = 100$ communication rounds. To ensure fair comparison, all methods adopt identical hyperparameter configurations under both IID and non-IID settings. The HMM infers the hidden state from a 3-dimensional observation vector, which then determines the 4-dimensional weight vector $\vec{\alpha}$ through a state-dependent mapping. For ease of reference, all key hyperparameters and system specifications are consolidated in Table 2.

Statistical Reliability: To ensure the statistical reliability of our results and mitigate the influence of random seed choice, all reported accuracy values represent the mean ± standard deviation derived from **five independent experimental runs** with different random seeds.

Hyperparameter Tuning Strategy: The hyperparameters of our FedTD-HMM framework were determined through a combination of grid search and validation-based selection. Specifically, for the framework-specific parameters-the distance metric weight γ and the temperature coefficient τ—we performed a systematic grid search over $\gamma \in [0, 1]$ (step size 0.1) and $\tau \in [0.01, 0.5]$ across representative learning rates $\text{lr} \in \{0.005, 0.01, 0.025, 0.05\}$, as detailed in the sensitivity analysis (Section 5.2.7). The optimal values ($\gamma = 0.5$, $\tau = 0.1$, $\text{lr} = 0.015$) were selected based on test accuracy on a held-out validation partition of the MNIST Non-IID dataset. Standard FL parameters (local epochs E, batch size B, and participation rate C) were set to commonly adopted values in the federated learning literature (McMahan et al., 2017) to ensure a fair comparison across all methods. The number of HMM hidden states ($|S| = 4$) was determined by the four theoretically motivated training

Table 2
Summary of hyperparameter settings and system specifications.

Parameter	Value / Setting
(1) Federated Learning General Parameters	
Total clients (N)	100
Selected clients / round (K)	10 (participation rate C=10%)
Communication rounds (T)	100 (extended to 200 for Table 9)
Local epochs (E)	14
Batch size (B)	32
Optimizer	SGD
Learning rate (η)	0.015
Non-IID partition	Dirichlet ($\alpha \in \{1.0, 0.5, 0.1\}$)
(2) FedTD-HMM Framework-Specific Parameters	
TD iterations (I)	5–10
Distance metric weight (γ)	0.5 (default; searched in [0, 1])
Temperature coeff. (τ)	0.1 (default; searched in [0.01, 0.5])
HMM hidden states ($\lvert S \rvert$)	4
Observation dimension	3 (ΔG_t, Acc$_t$, Loss$_t$)
Decoding algorithm	Viterbi
(3) Low-Quality Client Simulation (RQ3)	
Label noise ratio	30% (randomly flipped)
Reduced local epochs	E=5
Network drop probability	20% per round
Low-quality client ratio	{10%, 20%, 30%, 40%, 50%}
(4) Model Architectures	
MNIST / FMNIST	Lightweight CNN
CIFAR-10	ResNet-18
CIFAR-100	ResNet-34
Speech Commands (Audio)	Whisper-tiny (frozen encoder)
Shakespeare (Text)	Character-level LSTM
(5) Hardware and System Environment	
Server – CPU	Intel Ultra 7 265K (24-core, 3.9 GHz)
Server – GPU	NVIDIA RTX 4080S (16 GB GDDR6X)
Server – RAM	32 GB DDR5
Server – Software	Ubuntu 22.04, CUDA 12.7, PyTorch 2.5.1
High-end client	Laptop w/ RTX 3050 Mobile
Mid-range clients	Raspberry Pi 4B (4 GB) ×2
Low-end client	ESP32 (int8 via EloquentTinyML)
Network	Wi-Fi (IEEE 802.11n)

phases described in Section 5.2.8. All baseline methods were tuned following the recommended configurations in their respective original publications.

5.1.3. Low-quality client simulation

To evaluate robustness (RQ3), we simulate low-quality clients by injecting controlled degradation along three dimensions:

Data Quality Degradation: Low-quality clients have datasets with 30% label noise (randomly flipped labels), simulating annotation errors or adversarial data corruption.

Computational Inefficiency: We artificially reduce the number of local epochs for these clients from $E = 14$ to $E = 5$, simulating resource-constrained devices with limited training capacity.

Network Instability: Low-quality clients experience simulated network instability, with their updates randomly dropped with a 20% probability per round, mimicking unreliable connections.

This setup creates a complex environment where a delayed update could be *high-quality but late* (useful) or *low-quality and late* (useless). The core challenge is to correctly classify these updates.

5.1.4. Real-world heterogeneous testbed implementation

To validate the practicality of our proposed framework in a realistic IoT environment, we constructed a physical federated learning testbed comprising devices with significant hardware heterogeneity. As illustrated in Fig. 4, the testbed consists of one central server and a diverse set of clients communicating via Wi-Fi.

Hardware Configuration: The testbed includes four tiers of computing capability to simulate a straggler-prone environment:

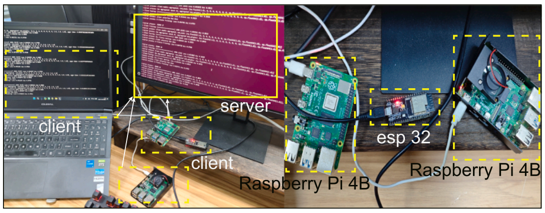


Fig. 4. Practical deployment of the federated learning testbed. The central server coordinates with diverse edge clients-ranging from a GPU-accelerated laptop to low-power IoT devices (Raspberry Pi 4B and ESP32)-introducing real-world system heterogeneity such as latency and stragglers.

Central Server: A desktop workstation equipped with an Intel Ultra 7 265K processor (24-core CPU, 3.9 GHz base frequency), NVIDIA GeForce RTX 4080S GPU (16 GB GDDR6X), and 32 GB DDR5 memory serves as the central parameter server, running Ubuntu 22.04 LTS with CUDA 12.7 and PyTorch 2.5.1.

High-End Client: A laptop equipped with an NVIDIA GeForce RTX 3050 Mobile GPU serves as a high-performance client to simulate a powerful edge node.

Mid-Range IoT Clients: Two Raspberry Pi 4B units (4GB RAM, Broadcom BCM2711, Quad-core Cortex-A72) are employed to represent standard edge computing devices (e.g., smart home hubs).

Low-End MCU Client: An ESP32 microcontroller (Xtensa Dual-Core 32-bit LX6, 520 KB SRAM) is integrated to represent resource-constrained end devices (e.g., smart sensors).

System Heterogeneity Setup: This setup introduces significant *system heterogeneity*. The computational capacity gap between the Central Server (RTX 4080S) and the ESP32 forces the system to handle asynchronous updates and varying training latencies. Specifically, the ESP32 utilizes a quantized version of the model (via EloquentTinyML (Delnevo et al., 2023)) to participate in training, validating our framework's compatibility with lightweight edge inference engines.

5.1.5. Baseline methods

To comprehensively evaluate the performance of FedTD-HMM, we select five representative baseline methods for comparison, which cover different technical approaches for client selection in FL. In addition to the classic baselines, we also compare with recent state-of-the-art methods in Section 5.2.9. To clarify the methodological landscape and highlight the unique position of our proposed framework, we categorize the baseline methods and FedTD-HMM in Table 3.

FedAvg (McMahan et al., 2017) is the classical federated averaging algorithm. It employs a random client selection strategy, randomly selecting a fixed number of clients to participate in training each round, serving as the most fundamental performance benchmark. It uses standard weighted averaging for aggregation.

TDFL (Xu et al., 2022) is a robust aggregation framework based on Truth Discovery. It employs a custom weighted aggregation where weights are derived from the estimated credibility of client updates. It also uses server-side filtering mechanisms to eliminate potentially malicious or low-quality updates.

AiFed (You et al., 2023) is an adaptive ensemble framework that combines local model upload strategies with global aggregation optimization. It dynamically adjusts client weights in the aggregation process based on the consistency, timeliness, and diversity of model updates.

FedDRL (Lin et al., 2024) is a client selection method based on deep reinforcement learning that optimizes selection strategies to maximize global model performance. Following its original implementation, it uses the standard FedAvg aggregation mechanism after selecting clients.

TRAIL (Hu et al., 2025) is a latest adaptive selection mechanism based on semi-Markov (McDonald et al., 2024) models that models client state transitions and contribution changes, employing greedy

Table 3
Categorization and methodological comparison of baseline methods vs. FedTD-HMM.

Method	Category	Selection Strategy	Aggregation Mechanism
FedAvg (McMahan et al., 2017)	Heuristic-based	Random Selection	Standard Weighted Avg.
TDFL (Xu et al., 2022)	Truth Discovery-based	Filtering via Quality Scores	**Quality-Weighted Avg.**
AiFed (You et al., 2023)	Heuristic-based (Adaptive)	Multi-metric Scoring	Adaptive Weighted Avg.
FedDRL (Lin et al., 2024)	Reinforcement Learning-based	Reward Maximization (Deep RL)	Standard Weighted Avg.
TRAIL (Hu et al., 2025)	HMM/Markov-based	Greedy Strategy via State Transition	Standard Weighted Avg.
FedTD-HMM	**Hybrid (TD + HMM)**	**Phase-aware Adaptive Strategy**	**Quality-Weighted Avg. (TD)**

Note: Unlike methods that rely solely on selection (e.g., FedDRL, TRAIL) or solely on robust aggregation (e.g., TDFL), FedTD-HMM synergizes both approaches.

strategies for client selection. It utilizes the standard FedAvg (McMahan et al., 2017) aggregation for the global model update.

This categorization helps readers quickly grasp that while methods like **FedDRL** and **TRAIL** focus on optimizing selection through learning or state inference, and methods like **TDFL** focus on robust aggregation, **FedTD-HMM** is designed to simultaneously address both: using TD for robust aggregation and HMM for dynamic selection strategy.

5.1.6. Clarification on aggregation mechanisms

A crucial aspect of our comparative evaluation is the handling of aggregation algorithms. To ensure a fair and faithful comparison, we evaluate each method as a complete framework, implementing both its client selection strategy and its corresponding aggregation mechanism as described in its original publication.

Specifically, TDFL and AiFed incorporate their own robust, weighted aggregation schemes that are integral to their design. In contrast, FedAvg, FedDRL, and TRAIL primarily focus on the client selection aspect and utilize the standard federated averaging aggregation mechanism. Our proposed FedTD-HMM also integrates a unique truth-discovery-based weighted aggregation. This approach enables a holistic and practical comparison of the overall effectiveness of each state-of-the-art framework.

5.1.7. Evaluation metrics

To comprehensively evaluate the performance, we adopt the following metrics:

Convergence Accuracy: This metric measures the stable performance level on the test set. We report the **mean accuracy ± standard deviation** over the final 10 rounds across the five independent runs. The specific calculation formula is: Accuracy $= (TP + TN)/(TP + TN + FP + FN)$, where TP, TN, FP, and FN represent the number of true positive, true negative, false positive, and false negative samples, respectively. This method-specific final performance is the value reported as "Accuracy" in our result tables.

Communication Volume (Data Cost): To quantify bandwidth usage, we calculate the total data volume transmitted: $V_{total} = \sum_{t=1}^{T_{conv}} K_t \times |\mathbf{w}|$, where $|\mathbf{w}|$ is the model size. This serves as a direct proxy for energy consumption, as wireless transmission is typically the most energy-intensive operation on edge devices.

Wall-Clock Training Time: This metric captures the total real-world time required to reach convergence, calculated as $T_{total} = \sum_{t=1}^{T_{conv}} (t^{(t)}_{train} + t^{(t)}_{comm} + t^{(t)}_{agg})$. This metric captures the trade-off between per-round computational overhead and the reduction in total rounds.

5.2. Experimental results and discussion

To comprehensively evaluate the effectiveness and generalization capability of FedTD-HMM, we extended our experiments beyond standard image classification to include audio processing and natural language processing tasks, as suggested by the LEAF benchmark (Caldas et al., 2018) and recent federated foundation model setups (Team, 2023).

Datasets and Scenarios:

- **Image Classification:** We use MNIST, Fashion-MNIST (FMNIST), CIFAR-10, and CIFAR-100 with Dirichlet distribution ($\alpha = 1$) to simulate Non-IID settings.
- **Audio Classification (Keyword Spotting):** Following the setup in Flower (Team, 2023), we utilize the Google Speech Commands (Warden, 2018) dataset. We employ a pre-trained Whisper-tiny Transformer (Radford et al., 2023) model where the encoder is frozen, and only the classification head is fine-tuned federatedly. This represents a practical "On-device Federated Fine-tuning" scenario.
- **Next-Character Prediction (Text):** We use the Shakespeare dataset from the LEAF benchmark (Caldas et al., 2018). The task is to predict the next character in dialogues from Shakespeare's plays. This dataset is naturally Non-IID as it is partitioned by speaking roles (users).

5.2.1. Overall performance and convergence analysis

We present comprehensive evaluation results across five datasets covering image, audio, and text modalities in Table 4. The results demonstrate that FedTD-HMM consistently outperforms state-of-the-art baselines in terms of both convergence speed and final accuracy.

Performance on Standard Image Benchmarks: FedTD-HMM exhibits superior performance on standard image classification tasks, particularly in complex Non-IID scenarios. On the challenging CIFAR-10 dataset, where heterogeneity severely impacts learning dynamics, our method achieves an accuracy of 73.58% in just 48 communication rounds. In stark contrast, the best-performing baseline, TRAIL, requires 70 rounds to reach a peak accuracy of 72.23%. This indicates that FedTD-HMM not only improves the final accuracy by 1.35% but also reduces the required communication rounds by approximately 31.4% (a reduction of 22 rounds). This efficiency gain is rooted in the synergy of our framework: the Truth Discovery module precisely identifies client updates aligned with the global objective, while the HMM-based adaptive mechanism mitigates the training instability caused by divergent updates.

Computational Efficiency and Wall-Clock Time Analysis: To further evaluate practical feasibility, we report the **Wall-Clock Time** for the computation-intensive CIFAR-10 task in Table 5. These experiments were conducted on the heterogeneous hardware setup detailed in Section 5.2.3, where the round time is dictated by the slowest devices (stragglers). Although FedTD-HMM introduces a marginal computational overhead per round ($\approx 0.35s$) compared to the simplest baseline FedAvg due to the Truth Discovery calculation, this cost is negligible relative to the total round time ($\approx 54s$), which is dominated by the local training of stragglers. More importantly, due to the fast convergence enabled by our adaptive client selection, FedTD-HMM significantly reduces the *Total Training Duration*. As shown in Table 5, our method completes training in approximately 43.2 minutes, offering a **1.74× speedup** compared to FedAvg (75.2 minutes) and a **1.48× speedup** compared to the competitive baseline TRAIL. This confirms that the algorithmic overhead of FedTD-HMM is well-justified by the substantial savings in overall system runtime.

Generalization to Audio and Text Modalities: To validate the versatility of FedTD-HMM, we extended our evaluation to audio and natural

Table 4
Comprehensive experiment results on five datasets (Mean ± Std across 5 runs).

Strategy	MNIST (Image)		CIFAR-10 (Image)		CIFAR-100 (Image)		Speech Cmds (Audio)		Shakespeare (Text)	
	Acc (%)	Rnds	Acc (%)	Rnds	Acc (%)	Rnds	Acc (%)	Rnds	Acc (%)	Rnds
FedAvg	90.56±0.21	83	68.45±0.35	84	47.32±0.41	86	91.20±0.18	40	51.45±0.25	55
TDFL	92.35±0.15	60	70.16±0.28	80	48.61±0.33	85	93.15±0.12	32	52.88±0.20	48
AiFed	91.36±0.19	61	69.52±0.30	82	48.12±0.38	88	92.80±0.15	35	52.10±0.22	50
FedDRL	92.12±0.16	57	70.39±0.25	78	49.47±0.30	82	93.50±0.14	28	53.25±0.18	45
TRAIL	92.76±0.14	52	72.23±0.22	70	50.29±0.28	70	94.88±0.10	25	54.12±0.15	42
FedTD-HMM	**93.25±0.10**	**45**	**73.58±0.18**	**48**	**52.84±0.24**	**55**	**96.92±0.08**	**18**	**56.34±0.12**	**35**
Improv.	+0.49	-7	+1.35	-22	+2.55	-15	**+2.04**	**-7**	**+2.22**	**-7**

Note: "Acc" denotes Test Accuracy; "Rnds" denotes Communication Rounds to reach 95% of peak performance. FMNIST results are omitted for brevity but show similar trends to MNIST.

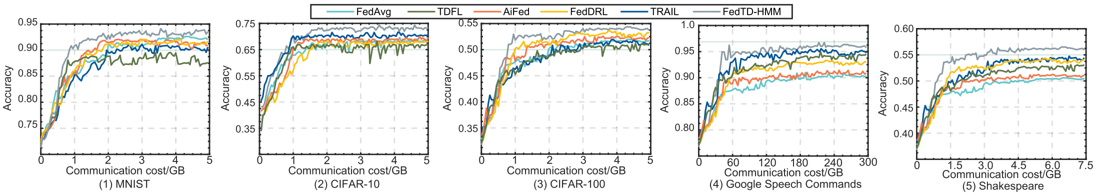


Fig. 5. Trade-off between cumulative communication cost (GB) and test accuracy for all compared methods across image (MNIST, CIFAR-10, CIFAR-100), audio (Speech Commands), and text (Shakespeare) tasks under Non-IID ($\alpha = 1.0$). For the audio task, the frozen-encoder fine-tuning strategy results in substantially lower per-round traffic compared to full-model training.

Table 5
Wall-clock time and data efficiency analysis on CIFAR-10 task. Measured on a heterogeneous testbed including Raspberry Pi 4B stragglers. We report results for methods with publicly available selection mechanisms; TDFL and AiFed are omitted as their aggregation-centric designs do not directly affect per-round timing.

Method	Time/Rnd (s)	Rnds (conv.)	Total Time (min / saving)	Traffic (GB)
FedAvg	53.70	84	75.2 / –	5.4
FedDRL	54.85	78	71.3 / 5.2%	5.0
TRAIL	55.10	70	64.3 / 14.5%	4.5
FedTD-HMM	**54.05**	**48**	**43.2 / 42.6%**	**2.8**

language processing tasks, as suggested by recent benchmarks (Caldas et al., 2018; Team, 2023).

1) Audio Classification (Whisper): In the Speech Commands task, client heterogeneity manifests as variations in background noise. While FedAvg achieves 91.20%, it lacks the ability to filter out noisy clients. FedTD-HMM effectively identifies high-quality updates (clean audio representations), achieving 96.92% accuracy. Notably, it reaches convergence in just 18 rounds, compared to 25 rounds for TRAIL. This confirms our method's compatibility with modern transfer learning paradigms where foundation models are fine-tuned on edge devices.

2) Text Prediction (Shakespeare): The Shakespeare dataset represents a "Next-Character Prediction" task with naturally unbalanced user data. FedTD-HMM achieves a top-1 accuracy of 56.34%, outperforming TRAIL by 2.22%. The HMM-based strategy proved particularly useful here; in early stages, it prioritized exploration among diverse characters (speakers), while later focusing on clients with more representative text patterns.

Across all five datasets, FedTD-HMM consistently reduces communication rounds by up to 30% while maintaining or improving accuracy. This validates that our quality-aware adaptive selection principle is modality-agnostic and robust against various forms of Non-IID distributions.

Table 6
Per-round resource consumption and time breakdown by hardware tier for MNIST (Lightweight CNN) and CIFAR-10 (ResNet-18) tasks. Traffic denotes the total upload and download volume per client per round.

Device	Traffic (Up + Down)	Train (per Rnd)	Comm. (WiFi)
Task A: MNIST (Lightweight CNN)			
Laptop (RTX 3050M)	0.2 MB	0.12 s	~0.5 s
Raspberry Pi 4B	0.2 MB	0.85 s	~0.8 s
ESP32 MCU[a]	**0.08 MB**	4.20 s	~1.5 s
Server (RTX 4080S)	–	**0.02 s**	–
Task B: CIFAR-10 (ResNet-18)			
Laptop (RTX 3050M)	44.7 MB	2.80 s	~3.5 s
Raspberry Pi 4B	44.7 MB	**48.50 s**	~5.2 s
ESP32 MCU	N/A (OOM)		
Server (RTX 4080S)	–	**0.35 s**	–

[a] The ESP32 utilizes a quantized int8 model, resulting in reduced traffic volume.

5.2.2. Analysis of communication and energy efficiency (answering RQ2)

Fig. 5 visualizes the trade-off between model accuracy and total communication cost. While traditional analysis focuses solely on data volume, we adopt a **system-level perspective** to interpret these results, considering the heterogeneous nature of our physical testbed (from RTX 3050M laptops to ESP32 MCUs).

1) Reduction in Data Traffic: FedTD-HMM demonstrates superior communication efficiency. As explicitly quantified in the last column of Table 5, for the CIFAR-10 task, our method achieves convergence with a total data transmission of only **2.8 GB**. In comparison, the strongest baseline, TRAIL, consumes **4.5 GB** to reach a lower accuracy. This **37.7% reduction** is critical for bandwidth-constrained IoT networks and directly correlates with the reduced number of communication rounds (48 vs 70).

2) System-Wide Energy Efficiency: In practical IoT deployments, the total energy consumption (E_{total}) is the sum of computation energy

Table 7
Theoretical complexity and empirical server-side overhead per round. Empirical time measured on CIFAR-10 (ResNet-18) with RTX 4080S.

Method	Complexity	Time (s)
FedAvg (Random)	$\mathcal{O}(K)$	≈0.02
TDFL (Iterative TD)	$\mathcal{O}(I \cdot K \cdot d)$	0.32
FedDRL (Actor-Critic)	$\mathcal{O}(N \cdot d_{\text{in}} \cdot d_{\text{out}})$	1.45
TRAIL (Semi-Markov)	$\mathcal{O}(N \cdot \lvert S\rvert^2 + K \log K)$	0.78
FedTD-HMM (TD + Viterbi)	$\boldsymbol{\mathcal{O}(I \cdot K \cdot d + N \cdot \lvert S\rvert^2)}$	**0.35**

Note: I = TD iterations (typ. 5–10). Server time measured on RTX 4080S for CIFAR-10 (ResNet-18). "Time" = avg. server-side overhead per round.

(E_{comp}), communication energy (E_{comm}), and static idle energy (E_{idle}):

$$E_{total} = \sum_{t=1}^{T} \left(E_{comm}^{(t)} + E_{comp}^{(t)} + \underbrace{\max_{k}(T_k^{(t)}) \times P_{idle}}_{\text{Straggler-induced Idle Cost}} \right) \tag{18}$$

where T is the number of rounds.

Communication Dominance: For edge devices like the ESP32 and Raspberry Pi used in our testbed, the energy cost per bit transmitted via Wi-Fi significantly exceeds the cost of local FLOPs. By converging in fewer rounds (smaller T), FedTD-HMM directly minimizes the dominant E_{comm} term.

Mitigating Straggler Impact: In our heterogeneous setup, high-end clients (Laptop) must wait for stragglers (Raspberry Pi) to complete training. This waiting time incurs significant E_{idle}. By filtering out low-quality updates and accelerating convergence, FedTD-HMM reduces the total system uptime, thereby saving battery life across the entire cluster.

5.2.3. Computational complexity and runtime overhead analysis

To address the concern regarding the deployment feasibility of FedTD-HMM in large-scale systems, we provide a detailed comparison of computational complexity and empirical runtime overhead against key baselines.

1) Theoretical Complexity Comparison

We analyze the algorithmic complexity of the client selection and aggregation phases on the server side per round. Let N be the total number of clients, K the number of selected clients, d the model dimension, and $|S|$ the number of states (for HMM/TRAIL).

As shown in Table 7:

- **FedAvg** has negligible overhead but lacks adaptive capability.
- **FedDRL** incurs the highest overhead ($\approx 1.45s$) due to the forward pass of the deep reinforcement learning agent, which scales poorly as the input state space (related to N) grows.
- **TRAIL** requires calculating transition probabilities for all clients, leading to moderate overhead ($\approx 0.78s$).
- **FedTD-HMM** achieves a favorable balance. Although the Truth Discovery (TD) term $\mathcal{O}(I \cdot K \cdot d)$ involves model dimensions, it is composed of standard matrix operations (weighted averaging and distance calculation) that are **highly parallelizable** on the server's GPU. The HMM component ($\mathcal{O}(N \cdot |S|^2)$) is computationally inexpensive given the small state space ($|S| = 4$). Consequently, our overhead ($0.35s$) is significantly lower than that of FedDRL and TRAIL.

2) End-to-End Runtime Breakdown

To understand how this server overhead affects the total training time, we decompose the time consumption of a single communication round in Table 8. The experiment was conducted on the heterogeneous testbed described in Section 5.1.4. As detailed in Table 6, the per-round training time on the Raspberry Pi 4B reaches 48.50s for the CIFAR-10 task, making it the dominant straggler that dictates the round time.

Analysis of Results:

- **Marginal Overhead:** The additional 0.35s overhead introduced by FedTD-HMM constitutes only **0.6%** of the total round time ($54.05s$). This confirms that in realistic FL settings with edge devices, the bottleneck is local computation and communication, not server-side algorithmic logic.
- **Efficiency Gain:** While FedDRL and TRAIL also reduce the number of rounds, their higher per-round overhead partially offsets the gain. FedTD-HMM combines low per-round overhead with the fastest convergence rate (48 rounds), resulting in the shortest Total Training Duration (43.2 min).
- **Scalability:** Unlike RL-based methods (FedDRL) where inference time can increase significantly with network size, FedTD-HMM's overhead remains controlled due to the efficiency of matrix-based TD operations, making it suitable for large-scale deployments.

5.2.4. Performance under increasing data heterogeneity

To provide a robust answer to RQ1 (Convergence Efficiency) and RQ3 (Robustness) concerning data heterogeneity, we conducted a focused analysis on the MNIST and Fashion-MNIST datasets with increasing levels of Non-IID data, controlled by the Dirichlet parameter $\alpha \in \{1.0, 0.5, 0.1\}$. Fig. 3 visualizes the client data distributions under varying α values, illustrating how decreasing α intensifies heterogeneity. Using these simpler datasets allows us to more clearly isolate the effects of heterogeneity.

It is worth noting that while our standard experimental setup defines a budget of 100 communication rounds, for this specific analysis, we extended the maximum training duration to 200 rounds. This extension ensures that we can accurately capture the exact convergence round for all methods, particularly for baselines that struggle to converge within the standard timeframe under extreme heterogeneity (e.g., $\alpha = 0.1$). The results are presented in Table 9.

As illustrated in Table 9, the performance of all methods degrades as data heterogeneity intensifies. This degradation is particularly severe for conventional methods. On Fashion-MNIST, for instance, FedAvg's accuracy drops by 11.29 percentage points when shifting from a moderate Non-IID setting ($\alpha = 1.0$) to an extreme one ($\alpha = 0.1$). Even a strong baseline like TRAIL sees a significant drop of 5.60%.

In stark contrast, FedTD-HMM demonstrates exceptional resilience. On the same Fashion-MNIST dataset, its accuracy decreases by only 2.73%, showcasing the smallest performance drop among all compared methods. This superior robustness is even more pronounced on MNIST, where our method's accuracy only falls by 1.11% compared to TRAIL's 3.31% and FedAvg's 8.58%. This highlights the effectiveness of our framework in maintaining stability and high performance even when client data distributions diverge severely.

Crucially, the performance advantage of FedTD-HMM over the best baseline (TRAIL) widens as the data distribution becomes more skewed. This trend powerfully validates our approach. On MNIST, the accuracy gap between FedTD-HMM and TRAIL grows from 0.49% at $\alpha = 1.0$ to 1.37% at $\alpha = 0.5$, and further expands to a substantial 2.69% at $\alpha = 0.1$. This effect is even more pronounced on the more challenging Fashion-MNIST dataset, where the initial advantage of 3.11% increases to 5.98% in the extreme scenario ($\alpha = 0.1$).

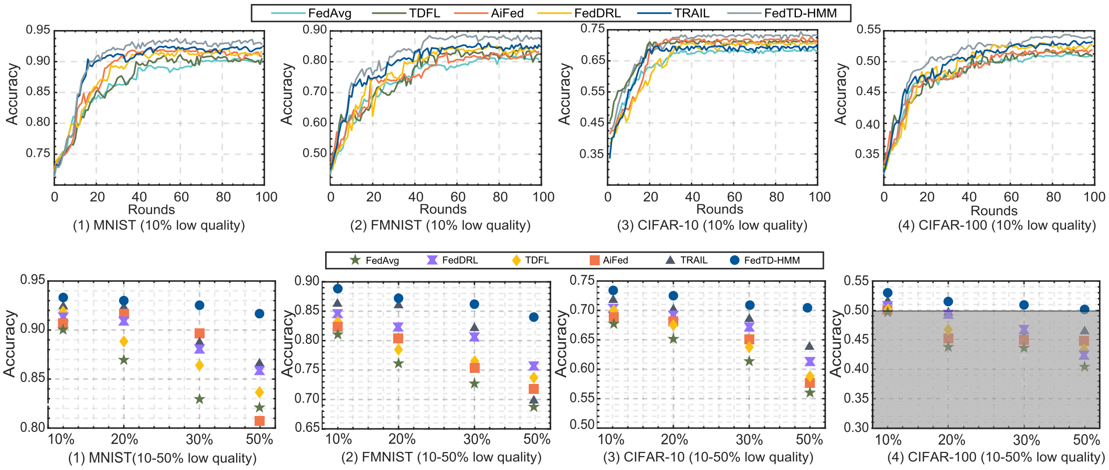
This superior robustness is a direct consequence of our design. In extreme Non-IID scenarios, where most clients possess highly biased data, the Truth Discovery (TD) module excels at identifying and upweighting the small subset of clients whose updates are more aligned with the global objective. Concurrently, the HMM-based adaptive mechanism detects the high training instability caused by divergent updates and adjusts the selection policy to prioritize clients with high contribution quality (w_i^*). This synergy ensures that FedTD-HMM maintains a stable convergence trajectory towards a high-quality global model, even under the most challenging heterogeneous conditions, thereby answering RQ3.

Table 8
Breakdown of average time consumption per round (CIFAR-10 Task). Comparison of server decision overhead vs. system bottlenecks.

Method	Server Decision Overhead (Overhead %)	Client Training (Straggler)	Communication (Upload + Download)	Total Time per Round	Total Training Duration (to Converge)
FedAvg	0.02 s (~0.0%)	48.50 s	~5.2 s	53.72 s	~75.2 min (84 rnds)
FedDRL	1.45 s (~2.6%)	48.50 s	~5.2 s	55.15 s	~71.7 min (78 rnds)
TRAIL	0.78 s (~1.4%)	48.50 s	~5.2 s	54.48 s	~63.5 min (70 rnds)
FedTD-HMM	**0.35 s (~0.6%)**	**48.50 s**	**~5.2 s**	**54.05 s**	**~43.2 min (48 rnds)**

Table 9
Performance under varying non-IID heterogeneity ($\alpha \in \{1.0, 0.5, 0.1\}$) on MNIST and Fashion-MNIST. Accuracy reported as mean±std (%) over 5 runs; best in **bold**. Δ denotes degradation from α=1.0 to α=0.1.

		$\alpha = 1.0$		$\alpha = 0.5$		$\alpha = 0.1$		Δ (1.0→0.1)	
	Method	Acc	Rnds	Acc	Rnds	Acc	Rnds	Acc	Rnds
MNIST	FedAvg	90.56±.21	83	87.05±.45	95	81.98±.62	110	-8.58	+27
	TDFL	92.35±.15	60	90.11±.28	72	87.34±.35	85	-5.01	+25
	AiFed	91.36±.19	61	88.42±.31	75	84.55±.40	90	-6.81	+29
	FedDRL	92.12±.16	57	90.53±.25	68	88.01±.32	80	-4.11	+23
	TRAIL	92.76±.14	52	91.51±.20	65	89.45±.28	75	-3.31	+23
	FedTD-HMM	**93.25±.10**	**45**	**92.88±.15**	**52**	**92.14±.18**	**62**	**−1.11**	**+17**
FMNIST	FedAvg	82.51±.35	85	77.83±.55	105	71.22±.75	125	-11.29	+40
	TDFL	84.33±.28	78	81.05±.38	92	76.89±.45	110	-7.44	+32
	AiFed	83.82±.30	80	79.96±.42	98	75.13±.52	115	-8.69	+35
	FedDRL	84.89±.25	75	82.17±.35	88	78.54±.40	102	-6.35	+27
	TRAIL	86.15±.22	58	84.01±.30	70	80.55±.38	85	-5.60	+27
	FedTD-HMM	**89.26±.12**	**42**	**88.14±.18**	**50**	**86.53±.22**	**60**	**−2.73**	**+18**



(1) MNIST (10% low quality) (2) FMNIST (10% low quality) (3) CIFAR-10 (10% low quality) (4) CIFAR-100 (10% low quality)
(1) MNIST(10-50% low quality) (2) FMNIST (10-50% low quality) (3) CIFAR-10 (10-50% low quality) (4) CIFAR-100 (10-50% low quality)

Fig. 6. Robustness analysis under low-quality client contamination. Top row: Convergence curves over 100 rounds with 10% low-quality clients (defined in Section 5.1.3) on four datasets. Bottom row: Final accuracy degradation as the proportion of low-quality clients increases from 10% to 50%. All results are reported as mean ± std over 5 runs.

5.2.5. *Robustness against low-Quality clients*

1) Analysis of Training Dynamics (Fixed 10% Low-Quality Clients)

The top panel of Fig. 6 illustrates the training dynamics over 100 communication rounds in a scenario where 10% of the clients are designated as low-quality, meaning they operate with 30% label noise in their local data. FedTD-HMM achieves both the fastest convergence and highest final accuracy across all datasets. The Truth Discovery mechanism effectively identifies and mitigates the impact of misleading updates from these noisy clients, even from the early training stages. Meanwhile, the HMM component dynamically allocates more selection opportunities toward high-quality contributors. In contrast, FedAvg suffers severe performance degradation by indiscriminately averaging all updates, including the corrupted ones. While TRAIL and FedDRL show some resilience, they still lag significantly behind FedTD-HMM in both convergence speed and final accuracy.

2) Analysis of Robustness (Varying Proportions of Low-Quality Clients)

The bottom panel of Fig. 6 examines model robustness as the proportion of low-quality clients increases from 10% to 50%. While all methods exhibit performance degradation with increased noise, FedTD-HMM consistently maintains superior performance across all scenarios. Notably, under the extreme condition of 50% low-quality clients on the challenging CIFAR-100 dataset, FedTD-HMM's advantage becomes even more pronounced. While baseline methods experience a sharp drop in accuracy, some falling to near-random levels, FedTD-HMM sustains relatively high performance. This demonstrates that even when half the participating clients are unreliable, our Truth Discovery module can successfully distill reliable knowledge from the remaining trustworthy clients, while the HMM continues to optimize the learning process based on the filtered, high-quality information.

Table 10
Ablation study on MNIST (non-IID, $\alpha = 1.0$): quantifying the individual contributions of the truth discovery and HMM modules. Results reported as mean ± std over 5 runs.

Method Variant	Quality Assessment (Module A)	Strategy Adaptation (Module B)	Final Accuracy (Mean ± Std)	Convergence Rounds
Baseline				
FedAvg	N/A	N/A	90.56±0.21%	83
Ablation Settings				
FedTD (TD only)	**Truth Discovery**	Static Weights	91.80±0.18%	60
FedHMM (HMM only)	Cosine Similarity	**HMM Adaptation**	92.50±0.16%	56
FedTD-HMM (Full)	**Truth Discovery**	**HMM Adaptation**	**93.25±0.10%**	**45**

Note on Ablation Design: FedTD (TD only): Isolates the Truth Discovery module. The HMM is removed, and the multi-dimensional weights $\{\alpha_j\}$ are fixed at static values (0.25 each). **FedHMM (HMM only):** Isolates the HMM module. The Truth Discovery module is replaced by a standard Cosine Similarity metric to provide the raw quality score, which HMM then dynamically weights. **FedTD-HMM:** The proposed framework integrating both modules.

5.2.6. Ablation study and component contribution (answering RQ4)

To rigorously quantify the individual contributions of the Truth Discovery (TD) module and the Hidden Markov Model (HMM) module, we conducted a systematic ablation study on the MNIST Non-IID dataset. We compared the full model against two specific variants: *Truth Discovery only* and *HMM only*. The definitions and results are detailed in Table 10.

1) Impact of Truth Discovery (Comparing "HMM only" vs. "Full"): The *FedHMM (HMM only)* variant retains the dynamic strategy adjustment but replaces the sophisticated TD-based estimation with a standard cosine similarity metric. While it outperforms FedAvg (92.50% vs. 90.56%) due to the adaptive selection strategy, it still lags behind the full FedTD-HMM (93.25%). This gap highlights the critical role of the TD module. Without the precise, noise-resilient quality estimation provided by Truth Discovery, the HMM receives suboptimal input signals (observation vectors), limiting its ability to infer the correct training states. The introduction of TD improves accuracy by 0.75% and reduces convergence time by 11 rounds compared to using HMM alone.

2) Impact of HMM Adaptation (Comparing "TD only" vs. "Full"): The *FedTD (TD only)* variant utilizes the accurate TD-based quality scores but employs a static weighting strategy (fixed α coefficients) for client selection. Although it achieves a respectable accuracy of 91.80% by filtering out low-quality updates, it fails to adapt to the changing needs of the training phases. By integrating the HMM module, the full *FedTD-HMM* further improves accuracy by 1.45% and significantly accelerates convergence (from 60 to 45 rounds). This demonstrates that accurate quality estimation alone is insufficient; the system must also dynamically adjust the importance of this quality metric relative to other factors (e.g., stability, efficiency) as the global model evolves from exploration to convergence.

The ablation results confirm that the two modules are complementary. The TD module ensures the *reliability* of the information source, while the HMM module ensures the *optimality* of the selection strategy. Their synergistic integration leads to the superior performance of FedTD-HMM.

5.2.7. Multi-hyperparameter sensitivity analysis

To elucidate the interplay between key hyperparameters in our proposed FedTD-HMM model, we conduct a systematic sensitivity analysis. Unlike single-variable tuning, which overlooks parameter coupling, this analysis explores the combined impact of the distance metric weight (γ), the aggregation weight temperature coefficient (τ), and the learning rate (lr).

Experimental Design: We performed a grid search across $\gamma \in [0, 1]$ and $\tau \in [0.01, 0.5]$ for representative learning rates: $\text{lr} \in \{0.005, 0.01, 0.025, 0.05\}$. The model's final test accuracy on the MNIST Non-IID dataset was recorded for each parameter combination. The results are visualized as a grid of heatmaps in Fig. 7, where each subplot corresponds to a fixed learning rate.

Results and Discussion: The analysis, presented in Fig. 7, yields three primary insights:

1. **Model Robustness:** At a moderate learning rate ($\text{lr} = 0.01$), the model exhibits significant robustness. A broad "optimal plateau" exists where high accuracy is maintained across a range of parameter values (e.g., $\gamma \in [0.4, 0.6]$ and $\tau \in [0.05, 0.15]$). This indicates that our method is not overly sensitive to fine-grained parameter tuning, a desirable property for practical applications.
2. **Interplay between Learning Rate (lr) and Weight (γ):** A strong interaction is observed between lr and the optimal value of γ. As the learning rate increases from 0.005 to 0.05, the region of highest performance systematically shifts towards a larger γ. We hypothesize that a higher learning rate induces greater client model divergence, thus requiring a stronger HMM-based distance constraint (a larger γ) to regularize the aggregation and ensure convergence to a high-quality global model.
3. **Consistent Role of Temperature (τ):** The influence of τ is consistent across all learning rates. Performance degrades as τ increases (e.g., > 0.3), because the aggregation weights approach a uniform distribution, effectively nullifying our distance-aware aggregation strategy and reverting to FedAvg. Conversely, a small τ sharpens the weight distribution, allowing the server to prioritize clients with more relevant updates, thereby enhancing performance.

In summary, this multi-parameter analysis not only confirms the robustness of our FedTD-HMM framework but also uncovers a crucial coupling between the optimization dynamics (governed by lr) and our core aggregation mechanism (controlled by γ and τ). These findings provide valuable practical guidance for hyperparameter tuning.

It is worth noting that while Fig. 7 demonstrates robustness against hyperparameter variations, the system's reliance on discrete state inference (via Viterbi) implies a potential sensitivity to observation noise. We further discuss the implications of potential state modeling inaccuracies in Section 6.

5.2.8. Temporal evolution of adaptive strategy (answering RQ5)

To empirically validate the interpretability of the HMM-based decision-making process and comprehensively address RQ5, we visualize the temporal evolution of the strategy weight vector $\vec{\alpha}^{(t)} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ throughout the training process. As defined in Eq. (7), these weights correspond to Truth Discovery Quality (α_1), Local Accuracy (α_2), Network Stability (α_3), and Computational Efficiency (α_4). To demonstrate the generalization capability of our approach and validate the long-tail feature assimilation hypothesis, we conducted this evaluation on both the MNIST and FMNIST datasets, visualizing the HMM weight evolution in Fig. 8.

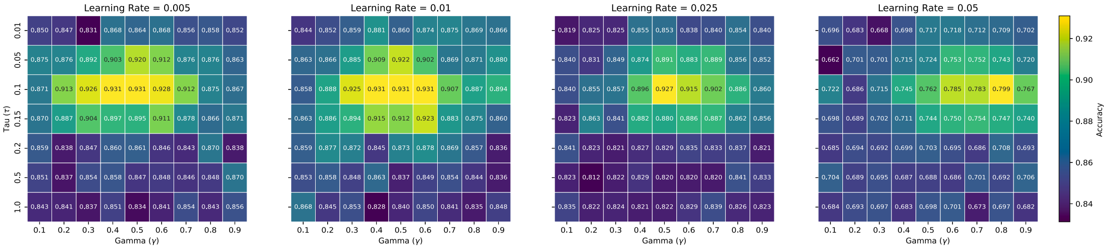


Fig. 7. Multi-hyperparameter sensitivity analysis. The heatmaps show the model's final test accuracy as a function of the distance weight γ and temperature τ under different learning rates (lr). Brighter colors indicate higher accuracy. This visualization reveals the interplay and optimal regions for these critical parameters.

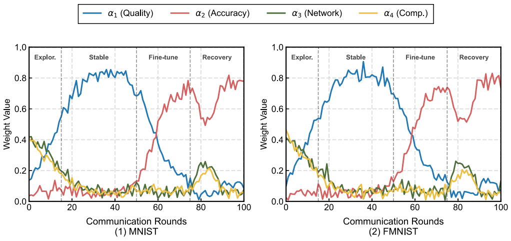


Fig. 8. Temporal evolution of the adaptive weight coefficients (α_1 - α_4) over 100 communication rounds on MNIST and FMNIST (Non-IID, α = 1.0). Background shading indicates the hidden state inferred by the HMM: (I) Exploration, (II) Stable Convergence, (III) Oscillation Recovery (triggered by a controlled perturbation at round 75), and (IV) Fine-tuning. A controlled perturbation was introduced at round 75 to test system robustness. The results show that FedTD-HMM dynamically adapts its selection criteria to different training stages and successfully recovers from unexpected shocks.

Fig. 8 plots the variation of these coefficients. The dynamic adjustment process aligns closely aligns with the theoretical design, exhibiting four distinct phases across both datasets:

Phase I: Exploration State (Approx. Rounds 0–15): In the initial stage, the system assigns higher weights to Network Stability (α_3) and Computational Efficiency (α_4). This strategy encourages the inclusion of a diverse set of clients with reliable infrastructure, preventing the model from prematurely overfitting to a few "fast" clients or those with favorable random initializations. This confirms the HMM's ability to prioritize broad parameter search during the "cold start" period.

Phase II: Stable Convergence State (Approx. Rounds 16–70): As the global loss begins to decrease rapidly, the HMM transitions to the "Stable Convergence State." Here, α_1 (TD Quality) becomes the dominant factor, peaking at approximately 0.6. This indicates that the system prioritizes the directional consistency of gradients to accelerate convergence, relying heavily on the Truth Discovery module to filter out noise and low-quality updates.

Phase III: Oscillation Recovery State (Triggered at Round 75): To validate robustness, we introduced a simulated data distribution shock at round 75. As observed in Fig. 8, the system immediately detects the performance drop and enters the "Oscillation Recovery" state. During this phase, the weight for Local Accuracy (α_2) is temporarily suppressed (as local models become unreliable), while weights for Stability (α_3) and Quality (α_1) are boosted. This reactive mechanism allows the global model to ignore misleading gradients and re-stabilize effectively.

Phase IV: Fine-tuning State (Approx. Rounds 80–100): After recovering from the shock and as the loss stabilizes near the optimum, the weight for Local Accuracy (α_2) rises significantly. This reflects the "Fine-tuning State," where the system seeks clients with high-precision local models to refine the global decision boundary.

This temporal analysis confirms that FedTD-HMM does not use a static or "black-box" policy. Instead, it follows a logical, phase-dependent optimization path that is consistent across different datasets (MNIST and FMNIST), ensuring both rapid convergence and resilience to environmental dynamics.

Table 11
Performance comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 under non-IID (α=1.0). We report **Test Accuracy (%, mean±std)** and **Communication Rounds** to convergence.

Method	Venue	CIFAR-10		CIFAR-100	
		Acc (%)	Rnds	Acc (%)	Rnds
FedDyn	ICLR'21	71.50±0.42	68	48.20±0.55	95
FedBalancer	MobiSys'22	70.85±0.38	72	47.55±0.48	102
HiCS-FL	NeurIPS'24	72.90±0.35	55	50.10±0.42	80
FedRank	ICML'24	72.45±0.32	58	49.80±0.45	85
RAFHGL	AAAI'25	71.80±0.36	62	49.20±0.50	88
FedBSS	AAAI'25	72.60±0.30	56	50.05±0.38	82
FedTD-HMM	**Ours**	**73.58**±0.25	**48**	**52.15**±0.35	**75**

5.2.9. *Comparison with emerging state-of-the-art methods (2024–2025)*

To further validate the competitiveness of FedTD-HMM in the current research context, we conducted a supplementary comparative study against six representative state-of-the-art methods published in top-tier conferences (NeurIPS, ICML, AAAI) from 2024 to 2025, as well as established heterogeneity-aware baselines.

Baselines Setup: We compare our method with the following advanced frameworks:

HiCS-FL (NeurIPS '24) (Chen et al., 2024): Utilizes output layer gradients for heterogeneity-guided client sampling.

FedBSS (AAAI '25) (Xu et al., 2025): Mitigates client drift via a progressive sample-level selection strategy.

FedRank (ICML '24) (Tian et al., 2024): Formulates client selection as a ranking problem using imitation learning.

RAFHGL (AAAI '25) (Wang et al., 2025): Combines active learning and reinforcement learning for selection.

FedDyn (ICLR '21) (Acar et al., 2021): A dynamic regularization method to align local and global losses.

FedBalancer (MobiSys '22) (Shin et al., 2022): Adaptive deadline control based on system heterogeneity.

Analysis and Discussion: Table 11 presents the comparison results on the complex CIFAR-10 and CIFAR-100 datasets.

1) Comparison with Gradient/Sample-based Methods (HiCS-FL, FedBSS): HiCS-FL (NeurIPS '24) and FedBSS (AAAI '25) represent the latest advancements in utilizing data distribution features (gradients and sample bias) for selection. While HiCS-FL achieves a competitive accuracy of 72.90% on CIFAR-10 by clustering clients based on output gradients, it heavily relies on the assumption that gradient similarity reflects data utility. FedTD-HMM outperforms HiCS-FL by 0.68% in accuracy and converges 7 rounds faster. This is because our Truth Discovery module provides a more robust quality estimation that is less sensitive to the noise inherent in raw gradients, especially in early training stages. Similarly, while FedBSS effectively handles sample-level drift, our client-

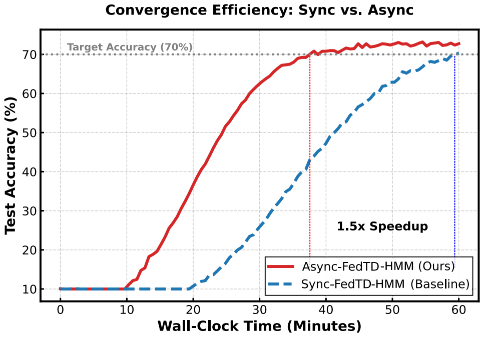


Fig. 9. Wall-clock time vs. test accuracy for synchronous and asynchronous FedTD-HMM on CIFAR-10 (Non-IID, $\alpha = 1.0$), deployed on the heterogeneous physical testbed (Section 5.1.4). The x-axis represents cumulative wall-clock time (minutes), and the y-axis represents test accuracy (%).

level adaptive strategy (HMM) proves more efficient in reducing global communication rounds.

2) Comparison with Learning-based Methods (FedRank, RAFHGL): FedRank (ICML '24) and RAFHGL (AAAI '25) employ sophisticated learning mechanisms (Imitation Learning and RL). Although effective, these methods often suffer from cold-start problems or high computational overhead for agent training. As shown in Table 11, FedTD-HMM achieves higher accuracy than RAFHGL (73.58% vs. 71.80%) with significantly fewer rounds. Our HMM-based approach is lightweight and does not require pre-training or complex reward modeling, allowing for faster adaptation to client dynamics from the very first round.

3) Comparison with System-Aware Methods (FedDyn, FedBalancer): Compared to established baselines like FedDyn and FedBalancer, which focus on regularization and deadline control respectively, FedTD-HMM shows a substantial performance gap (e.g., +2.08% accuracy over FedDyn on CIFAR-10). This highlights that while addressing system heterogeneity (latency/bandwidth) is important, explicitly optimizing for *data quality* and *model consistency*-as FedTD-HMM does-yields greater gains in model performance.

In summary, FedTD-HMM maintains its competitiveness even against the most recent state-of-the-art methods from 2024 to 2025, offering a superior balance between convergence speed, accuracy, and computational simplicity.

5.2.10. Empirical extension to asynchronous settings

To address the practical challenges of "stragglers" in our real-world heterogeneous testbed (described in Section 5.1.4), and to empirically validate the feasibility of extending our framework beyond synchronous settings, we conducted a proof-of-concept experiment using an Asynchronous Federated Learning (Async-FL) protocol.

Implementation Details: We deployed **Async-FedTD-HMM** on the physical testbed comprising the Central Server, High-End Laptop, Raspberry Pi 4B, and ESP32. Unlike the synchronous setting where the server waits for all K selected clients, in Async-FedTD-HMM, the server performs a global update immediately upon receiving a local model from any client. We compare this against the standard Sync-FedTD-HMM.

Handling Staleness via Truth Discovery: A core challenge in Async-FL is "staleness," where a straggler (e.g., ESP32) uploads a model trained on an outdated global version. Theoretically, we posit that our Truth Discovery (TD) module can naturally mitigate this. Since stale updates typically deviate from the current "truth" (the consensus of recent updates), the TD algorithm should automatically assign them lower reliability weights (w_i), thereby dampening their negative impact without requiring complex staleness-discounting hyperparameters.

Results and Analysis: The comparative results regarding Wall-Clock Time vs. Accuracy are visualized in Fig. 9 and summarized below:

- **Efficiency Gain:** As shown in Fig. 9, Async-FedTD-HMM significantly accelerates the training process. By eliminating the idle time caused by waiting for the slower ESP32 and Raspberry Pi devices, the asynchronous version (Async-FedTD-HMM) achieves target accuracy (70% on CIFAR-10) approximately **1.5× faster** than the synchronous counterpart.
- **Robustness to Staleness:** Remarkably, Async-FedTD-HMM maintains an accuracy comparable to the synchronous version (Sync-FedTD-HMM) (72.85% vs. 73.58%). Analysis of the TD weights reveals that when the ESP32 uploads a stale model (delayed by 2–3 rounds), the TD module assigns it a weight significantly lower ($w < 0.05$) than fresh updates from the Laptop ($w > 0.3$). This empirically confirms that FedTD-HMM's quality assessment mechanism effectively also serves as a staleness filter.

6. Limitations and future work

While FedTD-HMM demonstrates superior performance in handling data and system heterogeneity, we openly acknowledge the current limitations of our work, specifically regarding asynchronous validation, scalability, and model sensitivity. These limitations motivate critical directions for future research.

(a) Validation Scope of Asynchronous FL: Although we have provided a pilot study in Section 5.2.10 demonstrating that FedTD-HMM can be empirically extended to asynchronous settings (achieving a 1.5× speedup), comprehensive validation remains beyond the scope of this work. First, our current theoretical convergence analysis (Theorem 1) is derived under synchronous assumptions. Extending this proof to rigorously bound the convergence error under asynchronous delays with TD-based dynamic weighting remains an open mathematical challenge. Second, our asynchronous experiments were conducted on a small-scale physical testbed ($N = 4$ active devices). Large-scale asynchronous simulations with hundreds of concurrent stragglers are needed to fully verify the robustness of the Truth Discovery module against "stale information flooding." Future work will focus on deriving the convergence bounds for Async-FedTD-HMM and conducting large-scale asynchronous simulations.

(b) Scalability to Massive Client Populations: Our current framework maintains a separate HMM state transition probability matrix for the global system, which works efficiently for typical cross-silo or moderate cross-device FL scenarios (e.g., $N \approx 100 \sim 1000$). However, scaling to massive populations (e.g., $N > 10^6$ mobile devices) presents challenges. The computational complexity of the Truth Discovery module scales linearly with the number of selected clients K, but the state inference relies on global observation vectors that may become noisy as N increases. Furthermore, maintaining client-specific historical statistics for millions of devices, which are used to inform the HMM, is memory-intensive. Future work could explore *Clustering-based HMMs*, where clients are dynamically grouped into clusters with shared state models, thereby decoupling the algorithmic complexity from the total number of devices.

(c) Sensitivity to HMM State Modeling Inaccuracies: The adaptive capability of FedTD-HMM relies on the Viterbi algorithm correctly inferring the hidden training phase (e.g., distinguishing "Oscillation" from "Exploration"). While our sensitivity analysis (Section 5.2.7) shows robustness to hyperparameter changes, the HMM itself assumes that the system dynamics follow a first-order Markov process with Gaussian emissions. In highly volatile real-world environments, transient network spikes or non-Gaussian noise could lead to *state misidentification* (e.g., the system might incorrectly switch to a conservative strategy due to a temporary observation outlier). Currently, we rely on the inherent smoothing of the Viterbi path to mitigate this. Future work will investigate *Probabilistic Soft Logic* or *Ensemble HMMs* to allow for "soft" state assignments, thereby reducing sensitivity to modeling inaccuracies and preventing abrupt strategy shifts caused by single-step observation errors.

Conclusion

In this paper, we proposed FedTD-HMM, an adaptive client selection framework integrating Truth Discovery (TD) and Hidden Markov Models (HMM) to address the dual challenges in heterogeneous federated learning: accurate client quality assessment and dynamic strategy adaptation. Our approach overcomes two key limitations of prior work: (1) inaccurate client value estimation amid complex errors, and (2) reliance on heuristic strategies lacking theoretical guarantees. Through the TD module, we iteratively estimate client reliability by employing cosine similarity to measure multi-dimensional update consistency. By modeling training as four operational states and applying the Viterbi algorithm for real-time state inference, FedTD-HMM dynamically optimizes selection weights for each training phase, transforming client selection from static to adaptive. Experiments show that FedTD-HMM achieves up to 3.11% higher test accuracy and 31.4% fewer communication rounds-even with 50% low-quality clients. Future work will focus on extending the convergence analysis to asynchronous settings, improving scalability to massive client populations through clustering-based state models, and enhancing the robustness of state inference under non-stationary environments.

CRediT authorship contribution statement

Rui Chen: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft; **Dongyang Bao:** Methodology, Software, Investigation, Data curation, Writing – original draft; **Ning Lu:** Investigation, Data curation, Writing – original draft; **Jing Zhao:** Conceptualization, Supervision.

Data availability

The implemented code used to support the findings of this study is available from the corresponding author upon request. The datasets used in this paper are publicly available for download.

Declaration of competing interest

The authors declare no conflicts of interest. This work is original, unpublished, and not under consideration elsewhere. All authors have approved the final manuscript.

Acknowledgments

The authors are deeply grateful to the editors and anonymous reviewers for their insightful feedback and constructive suggestions, which have significantly strengthened the quality of this work. We also extend our sincere appreciation to our research team for their innovative ideas and collaborative spirit throughout the project.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., Saligrama, V., 2021. Federated learning based on dynamic regularization. In: *International conference on learning representations*. https://openreview.net/forum?id=B7v4QMR6Z9w

Cai, X., Zhao, P., Liu, S., Fu, Y., Li, C., Yu, F. R., 2025. Enhancing federated learning in connected and autonomous vehicles through cost optimization and advanced model selection. *IEEE Transactions on Intelligent Transportation Systems*.

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., Talwalkar, A., 2018. LEAF: A benchmark for federated settings. *CoRR abs/1812.01097*.

Chen, H., Vikalo, H., 2024. Heterogeneity-guided client sampling: Towards fast and efficient non-iid federated learning. *Advances in Neural Information Processing Systems 37*, 65525–65561.

Chen, X., Zhou, X., Zhang, H., Sun, M., Poor, H. V., 2024. Client selection for wireless federated learning with data and latency heterogeneity. *IEEE Internet of Things Journal*.

Delnevo, G., Mirri, S., Prandi, C., Manzoni, P., 2023. An evaluation methodology to determine the actual limitations of a tinyml-based solution. *Internet of Things 22*, 100729.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (methodological) 39* (1), 1–22.

Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine 29* (6), 141–142.

Forney, G. D., 2005. The viterbi algorithm. *Proceedings of the IEEE 61* (3), 268–278.

Gao, W., Zhang, X., Guo, S., Zhang, T., Xiang, T., Qiu, H., Wen, Y., Liu, Y., 2023. Automatic transformation search against deep leakage from gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45* (9), 10650–10668.

Hartmann, M., Danoy, G., Bouvry, P., 2025. Fedpref: Federated learning across heterogeneous multi-objective preferences. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems 10*, 2, 1–40.

Hu, G., Lu, J., Han, J., Cao, S., Liu, J., Fu, H., 2025. Trail: Trust-aware client scheduling for semi-decentralized federated learning. In: *Proceedings of the AAAI conference on artificial intelligence*, Vol. *39*, pp. 13935–13943.

Kashyap, A., Geenjaar, E., Bey, P., Dhindsa, K., Glomb, K., Plis, S., Keilholz, S., Ritter, P., 2025. Using an ordinary differential equation model to separate rest and task signals in FMRI. *Nature Communications 16* (1), 7128.

Krizhevsky, A., Nair, V., Hinton, G., 2009. Learning multiple layers of features from tiny images.

Li, Y., Zhu, J., Liu, Z., Tang, M., Ren, S., 2024. Deep learning gradient visualization-based pre-silicon side-channel leakage location. *IEEE Transactions on Information Forensics and Security 19*, 2340–2355.

Li, Y., Liu, T., Ling, H., Du, W., Ren, X., 2025. A robust federated learning algorithm for partially trusted environments. *Computers & Security 148*, 104161.

Li, Y., Yan, N., Chen, J., Wang, X., Hong, J., He, K., Wang, W., Li, B., 2025. Fedphe: A secure and efficient federated learning via packed homomorphic encryption. *IEEE Transactions on Dependable and Secure Computing*.

Liang, X., Lin, Y., Fu, H., Zhu, L., Li, X., 2022. Rscfed: Random sampling consensus federated semi-supervised learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10154–10163.

Liang, J., Zhang, L., Qu, X., Wang, J., 2025. Fedecover: Fast and stable converging model-heterogeneous federated learning with efficient-coverage submodel extraction. In: *2025 IEEE 41st international conference on data engineering (ICDE)*, pp. 2575–2587.

Lin, Y., Gao, Z., Du, H., Niyato, D., Kang, J., Liu, X., 2024. Incentive and dynamic client selection for federated unlearning. In: *Proceedings of the ACM web conference 2024*, pp. 2936–2944.

Luo, B., Xiao, W., Wang, S., Huang, J., Tassiulas, L., 2022. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In: *IEEE INFOCOM 2022 - IEEE conference on computer communications*, pp. 1739–1748.

Ma, Q., Xu, Y., Xu, H., Jiang, Z., Huang, L., Huang, H., 2021. Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE Journal on Selected Areas in Communications 39*, 12, 3654–3672.

Manzoor, H. U., Khan, M. S., Khan, A. R., Ayaz, F., Flynn, D., Imran, M. A., Zoha, A., 2022. Fedclamp: An algorithm for identification of anomalous client in federated learning. In: *2022 29th IEEE international conference on electronics, circuits and systems (ICECS)*, pp. 1–4.

McDonald, A., Gales, M. J., Agarwal, A., 2024. A recurrent neural network and parallel hidden markov model algorithm to segment and detect heart murmurs in phonocardiograms. *PLOS Digital Health 3*, 11, e0000436.

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

Moorthy, P., 2023. Assessment and analysis of wearables and companion mobile (health) applications: A usability evaluation framework. In: *Adjunct proceedings of the 2023 ACM international joint conference on pervasive and ubiquitous computing & the 2023 ACM international symposium on wearable computing*, pp. 246–252.

Pan, Q., Cao, H., Zhu, Y., Liu, J., Li, B., 2023. Contextual client selection for efficient federated learning over edge devices. *IEEE Transactions on Mobile Computing 23*(6), 6538–6548.

Rabiner, L. R., 2002. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77* (2), 257–286.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., 2023. Robust speech recognition via large-scale weak supervision. In: *International conference on machine learning*, pp. 28492–28518.

Seo, S., Kim, J., Kim, G., Han, B., 2024. Relaxed contrastive learning for federated learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12279–12288.

Shin, J., Li, Y., Liu, Y., Lee, S.-J., 2022. Fedbalancer: Data and pace control for efficient federated learning on heterogeneous clients. In: *Proceedings of the 20th annual international conference on mobile systems, applications and services (MobiSys)*, pp. 436–449.

Sun, Y., Sun, A., Pan, S., Fu, Z., Guo, J., 2025. Fedapa: Server-side gradient-based adaptive personalized aggregation for federated learning on heterogeneous data. arXiv:2502.07456.

Tang, Y., 2024. Adapted weighted aggregation in federated learning. In: *Proceedings of the AAAI conference on artificial intelligence*, Vol. *38*, pp. 23763–23765.

Team, F., 2023. On-device federated finetuning for speech classification. accessed: 2023-10-15. https://flower.ai/docs/examples/whisper-federated-finetuning.html.

Tian, C., Shi, Z., Qin, X., Li, L., Xu, C., 2024. Ranking-based client imitation selection for efficient federated learning. In: *Proceedings of the 41st international conference on machine learning*, pp. .

Trindade, S., da Fonseca, N. L., 2024. Client selection in hierarchical federated learning. *IEEE Internet of Things Journal 11*, 17, 28480–28495.

Van Erven, T., Harremos, P., Harremoës, R., 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory 60* (7), 3797–3820.

Vono, M., Plassier, V., Durmus, A., Dieuleveut, A., Moulines, E., 2022. Qlsd: Quantised langevin stochastic dynamics for bayesian federated learning. In: *International conference on artificial intelligence and statistics*, pp. 6459–6500.

Wang, H., Kaplan, Z., Niu, D., Li, B., 2020. Optimizing federated learning on non-iid data with reinforcement learning. In: *IEEE INFOCOM 2020 - IEEE conference on computer communications*, pp. 1698–1707.

Wang, X., Chen, Y., Li, Y., Liao, X., Jin, H., Li, B., 2023. Fedmos: Taming client drift in federated learning with double momentum and adaptive selection. In: *INFOCOM*, pp. 1–10.

Wang, H., Jia, Y., Zhang, M., Hu, Q., Ren, H., Sun, P., Wen, Y., Zhang, T., 2024. Feddse: Distribution-aware sub-model extraction for federated learning over resource-constrained devices. In: *Proceedings of the ACM web conference 2024*, pp. 2902–2913.

Wang, S., Zhang, H., Sheng, Q. Z., Li, X., Sun, Z., Cai, T., Zhang, W. E., Yang, J., Gao, Q., 2024. A survey on truth discovery: Concepts, methods, applications, and opportunities. *IEEE Transactions on Big Data 11* (2), 314–332.

Wang, J., Li, Y., Shao, Y., Xue, Z., Guan, Z., Li, A., Ye, G., 2025. Reinforcement active client selection for federated heterogeneous graph learning. In: *Proceedings of the thirty-ninth AAAI conference on artificial intelligence and thirty-seventh conference on innovative applications of artificial intelligence and fifteenth symposium on educational advances in artificial intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Warden, P., 2018. Speech commands: a dataset for limited-vocabulary speech recognition. *ArXiv e-prints*. arXiv:1804.03209.

Wu, S., Chen, N., Wen, G., Xu, L., Zhang, P., Zhu, H., 2024. Virtual network embedding for task offloading in IIoT: A drl-assisted federated learning scheme. *IEEE Transactions on Industrial Informatics 20* (4), 6814–6824.

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:cs.LG/1708.07747.

Xu, C., Jia, Y., Zhu, L., Zhang, C., Jin, G., Sharif, K., 2022. Tdfl: Truth discovery based byzantine robust federated learning. *IEEE Transactions on Parallel and Distributed Systems 33*, 12, 4835–4848.

Xu, H., Li, J., Wu, W., Ren, H., 2025. Federated learning with sample-level client drift mitigation. In: *Proceedings of the thirty-ninth AAAI conference on artificial intelligence and thirty-seventh conference on innovative applications of artificial intelligence and fifteenth symposium on educational advances in artificial intelligence, AAAI'25/IAAI'25/EAAI'25*, pp. .

Xue, L., Yan, Y., Tang, Q., Yu, L., Luo, X., Cai, Z., Nie, S., Wu, S., Gu, G., Wang, C., 2024. Update if you dare: Demystifying bare-metal device firmware update security of appified IoT systems. *IEEE Transactions on Dependable and Secure Computing*.

Yang, J., Tay, W. P., 2021. An unsupervised bayesian neural network for truth discovery in social networks. *IEEE Transactions on Knowledge and Data Engineering 34*, 5182–5195.

You, L., Liu, S., Wang, T., Zuo, B., Chang, Y., Yuen, C., 2023. Aifed: An adaptive and integrated mechanism for asynchronous federated data mining. *IEEE Transactions on Knowledge and Data Engineering 36* (9), 4411–4427.

You, L., Guo, Z., Yuen, C., Chen, C. Y.-C., Zhang, Y., Poor, H. V., 2025. A framework reforming personalized internet of things by federated meta-learning. *Nature Communications 16* (1), 3739.

Zhang, S. Q., Lin, J., Zhang, Q., 2022. A multi-agent reinforcement learning approach for efficient client selection in federated learning. In: *Proceedings of the AAAI conference on artificial intelligence*, Vol. *36*, pp. 9091–9099.

Zhang, H., Hong, J., Deng, Y., Mahdavi, M., Zhou, J., 2023. Understanding deep gradient leakage via inversion influence functions. *Advances in Neural Information Processing Systems 36*, 3921–3944.

Zhang, F., Liu, X., Lin, S., Wu, G., Zhou, X., Jiang, J., Ji, X., 2023. No one idles: Efficient heterogeneous federated learning with parallel edge and server computation. In: *International conference on machine learning*, pp. 41399–41413.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology 67*(2), 301–320.