FedA2L: Adaptive Layer-wise Learning Rate Adjustment in Decentralized Federated Learning

Van Truong Vo, Khoa Nguyen and Taehong Kim*

School of Information and Communication Engineering, Chungbuk National University, Cheongju, South Korea

ARTICLE INFO

Keywords:
Decentralized federated learning
Layer-wise learning rate adaptation
Data heterogeneity
Network consensus
Convergence acceleration
Communication efficiency
Internet of Things (IoT)
Edge Computing

ABSTRACT

Decentralized intelligence systems with heterogeneous devices and limited coordination increasingly rely on decentralized federated learning (DFL). However, DFL suffers from convergence inefficiency under data heterogeneity due to the use of a uniform learning rate (LR) that ignores layer-specific optimization needs. Foundational layers are responsible for maintaining network consensus, while specialized layers adapt to local data characteristics, leading to conflicting gradients and degraded performance under non-IID conditions. To address this fundamental tension, this work introduces FedA2L, a method that dynamically adjusts layer-wise LRs based on model divergence signals. By leveraging local update intensity and network consensus constraints, FedA2L seamlessly integrates into existing DFL protocols without additional communication or coordination. Extensive evaluations across DFL algorithms, various model architectures, and datasets demonstrate that FedA2L achieves up to 4.94× faster convergence than vanilla DFL and reduces communication rounds by up to 59% compared to scheduler-based baselines. Furthermore, FedA2L exhibits resilience to severe data heterogeneity, larger network sizes, and sparse topologies, reducing communication overhead and establishing it as a versatile optimization tool for resource-constrained or large-scale distributed learning in edge and IoT deployments. The code is released at https://github.com/nclabteam/FedA2L.

1. Introduction

Recent advances in Industrial IoT (IIoT) and Cyber-Physical Systems (CPS) require robust decentralized intelligence mechanisms to maintain real-time synchronization between physical assets and their Digital Twins (DT) [1]. In such settings, centralized cloud-based learning introduces unacceptable latency, privacy risks, and single points of failure [2]. In this context, federated learning (FL) has emerged as a distributed computing paradigm that enables collaborative model training on decentralized data, preventing direct access to sensitive information at individual nodes [3]. In standard FL, a central server coordinates training by collecting local model updates from participating clients, aggregating them into a global model, and broadcasting updated parameters back to clients for the next round. However, this centralization introduces vulnerabilities, such as communication bottlenecks [4], and scalability issues [5, 6].

DFL mitigates these issues by removing the central server, enabling direct peer-to-peer (P2P) communication among nodes according to a network topology [7]. In DFL, each node independently performs local model training on its private data and exchanges model parameters only with direct neighbors, then aggregates received models to achieve network consensus without central coordination [8]. This architecture enhances robustness, alleviates communication bottlenecks by distributing aggregation load across the network, and improves scalability. These properties make DFL well suited for large-scale networked systems, including edge intelligence systems [9, 10], Internet of Things (IoT) networks [11, 12], and traffic-aware communication environments [13], where centralized coordination is often impractical.

Despite these advantages, DFL systems confront a critical challenge of statistical heterogeneity in distributed data. In practice, data on each node is non-identically and independently distributed (non-IID). This heterogeneity causes each node's local learning objectives to drift away from the consensus objectives enforced by neighbor-wise aggregation, a phenomenon known as client drift [14, 15]. This drift manifests as inconsistency between local gradients and the global optimization direction, leading to conflicting parameter updates. These conflicts slow convergence and reduce model quality [16, 17]. This problem intensifies as data heterogeneity becomes more severe, creating a critical obstacle to achieving efficient and accurate distributed learning.

This problem is further compounded in deep neural networks (DNNs), where client drift emerges unevenly across model layers owing to their hierarchical structure in feature extraction. Foundational layers capture general and shared features to support network consensus, while specialized layers adapt to task-specific features in local data [18, 19]. Within DFL, this architectural hierarchy creates fundamentally different optimization challenges for each layer. Specialized layers are highly adapted to local data characteristics to achieve good performance on local objectives, causing significant dispersion between nodes. Conversely, foundational layers that over-homogenize shared representations limit their capacity to maintain task-general features to support local performance [20]. Recent work on layer-wise personalized FL [21] confirms this asymmetry, demonstrating that optimal layer-wise treatment requires

*Corresponding author
ORCID(s):

balancing local update intensity in specialized layers with network consensus constraints in foundational layers.

This tension is exacerbated by the widespread use of a uniform LR across all model layers in standard DFL algorithms. Treating the LR as a single scalar hyperparameter forces all layers to update at the same rate, ignoring the distinct optimization landscapes, gradient magnitudes, and update sensitivities exhibited by different layers. As a result, uniform LRs in DFL systems suffer from training instability, prolonged convergence time, and suboptimal model quality, becoming more significant in the case of severe non-IID conditions.

To address this critical gap in DFL systems, this study proposes the "**A**daptive **L**ayer-wise **L**earning Rate Adjustment in Decentralized **Fed**erated Learning" (FedA2L), a serverless method for dynamically tuning layer-wise LRs based on model divergence signals. The core innovation is formalizing model states at distinct transitions of each DFL round: the base state (before local training), the trained state (after local gradient updates), and the aggregated state (after network consensus). Through analyzing transitions between these three states, FedA2L derives two layer-wise metrics capturing distinct aspects of the optimization dynamics: weight divergence (σ) measuring local update intensity across nodes and aggregation stability (ζ) measuring network consensus constraints. By leveraging these signals, FedA2L dynamically assigns a unique LR to each model layer, enabling simultaneous optimization of local learning and network consensus without requiring central coordination or additional communication overhead. The primary contributions can be summarized as follows:

- FedA2L introduces a state-based formalization that characterizes model dynamics across three states of each DFL round, enabling precise analysis of layer-wise divergence and aggregation behavior.
- Adaptation of two layer-wise metrics (weight divergence σ and aggregation stability ζ) enables layer-wise dynamic LR adjustment to quantify and balance local update intensity and network consensus constraints.
- Layer-wise adaptive LRs based on local model divergence metrics can be seamlessly integrated into existing DFL algorithms without modifying the core aggregation logic or introducing additional communication overhead.
- Extensive empirical validation demonstrates that FedA2L significantly accelerates convergence across six DFL algorithms and DFedHPO, spanning multiple neural network architectures and diverse datasets, achieving up to 4.94× faster convergence than vanilla DFL and up to 59% fewer communication rounds than the best-performing scheduler-based baselines.

These contributions are particularly valuable for decentralized intelligence systems operating under dynamic conditions, where layer-wise adaptive mechanisms balance localized learning and stable network consensus across distributed entities.

The remainder of this paper is organized as follows. Section 2 reviews related work on DFL and adaptive hyperparameter optimization methods. Section 3 presents the problem formulation, the three-state model of DFL node procedures, and the proposed dual-metric layer-wise LR adaptation mechanism with its algorithmic integration. To formally validate these properties, Section 4 establishes convergence guarantees for both non-convex and strongly convex settings. Section 5 evaluates convergence speed, model accuracy, robustness under severe data heterogeneity, node scalability, and sparse network topologies, together with ablation and hyperparameter sensitivity analyses of FedA2L. Section 6 discusses the advantages and limitations of FedA2L and outlines directions for future work. Section 7 concludes the paper.

2. Related work

2.1. Decentralized federated learning

DFL eliminates the reliance on a central server, enabling robust and scalable P2P collaboration. Foundational work such as D-PSGD [28] provided convergence guarantees using P2P model averaging. Recent surveys [29, 6] systematically categorize DFL advances, with research focusing on enhancing communication efficiency, convergence speed under non-IID conditions, and robustness to security and privacy threats [30, 31]. This includes analyzing the impact of network topology [32, 33] and developing advanced protocols to enhance communication efficiency, such as using sparsification and quantization to compress model updates [34, 35], reducing transmission overhead in bandwidth-constrained edge environments. Advanced protocols, such as FedAWA [36] and FedAW [37], enhance aggregation by reweighing client contributions based on local data characteristics or model similarity, improving alignment with the overall learning dynamics. However, these methods assume uniform LRs during local training or apply only basic decay strategies across all layers, limiting adaptability to heterogeneous data and hierarchical model dynamics. While promising for general scenarios, such limitations hinder the deployment of robust decentralized intelligence in complex cyber-physical infrastructures that require coordinated learning across diverse and heterogeneous devices.

2.2. Hyperparameter optimization in FL and DFL

The LR is a critical hyperparameter in distributed optimization, affecting both convergence speed and model quality. In centralized FL, server-side adaptive optimization methods have been developed to enhance performance. FedYogi [22] employs a server-side adaptive optimizer that stabilizes convergence by maintaining coordinate-wise second moment estimates. FLARE [23] dynamically adjusts LRs for selected devices based on their individual computational capabilities. Recent work on federated hyperparameter optimization [38, 39] highlights the challenge of balancing

Table 1
Comparison of adaptive learning-rate methods for federated learning and decentralized federated learning.

Method	Layer-wise LR	Dynamic per Round	Server-free	No Extra Comm.	Signal Source
FedYogi [22]	×	✓	×	✓	Server-side second moment
FLARE [23]	×	✓	×	✓	Device capability
DFedHPO [24]	×	×	✓	✓	Pre-training search
AutoLR [25]	✓	✓	×	×	Gradient norms
Fed-LAMB [26]	✓	✓	×	×	Gradient statistics
FLAYER [21]	✓	✓	×	×	Global gradient/loss signals
FedLWS [27]	✓	✓	×	×	Weight shrinking
FedA2L	✓	✓	✓	✓	Local state transitions

efficiency with adaptive tuning. However, all these methods are fundamentally server-dependent and incompatible with DFL. In DFL settings, DFedHPO [24] performs a one-time, decentralized hyperparameter search prior to model training and outputs a consensus hyperparameter configuration that remains fixed during subsequent communication rounds. While these methods eliminate the need for a server, both server-dependent adaptive methods and one-shot static search are unable to accommodate evolving layer-specific heterogeneity or continual drift during training.

Layer-wise adaptive methods, motivated by hierarchical feature learning [18, 19], have been explored in centralized FL. AutoLR [25] adjusts per-layer rates based on gradient norms, while Fed-LAMB [26] incorporates layer-wise and dimension-wise adaptivity. More recently, FLAYER [21] adjusts layer-specific LRs using gradient or loss signals, and FedLWS [27] proposes adaptive layer-wise weight shrinking. However, these methods face fundamental limitations in DFL: (1) they require centralized coordination for global layer-wise statistics, (2) incur significant computational overhead, and (3) treat aggregation as a post-processing step rather than an optimization signal source. Table 1 summarizes these limitations and positions FedA2L against existing hyperparameter optimization methods across the four properties that define an effective DFL solution.

Current DFL methods often overlook the internal dynamics of local training, relying on static LRs that fail to account for node-specific data heterogeneity and layer-specific learning dynamics. Existing adaptive LR methods are either server-dependent, operate at a coarse model level, or perform static hyperparameter searches.

To the best of our knowledge, no prior work has developed a dynamic, layer-wise adaptive LR method tailored to the serverless nature of DFL. This highlights the need for decentralized mechanisms that can respond to layer-specific optimization behavior without a centralized coordinator. To address this gap, an effective solution should rely solely on locally available information, incur no additional communication overhead, and support layer-wise adaptation to capture heterogeneous optimization roles across model layers while preserving training stability under severe non-IID conditions.

Based on these requirements, we design a serverless, communication-efficient, and layer-wise adaptive LR mechanism for DFL. The key insight is that the standard DFL training loop already contains locally observable signals: the parameter transitions during local training and network consensus directly encode both local update intensity and network consensus constraints at the layer level. By extracting and combining these signals without any additional coordination, the proposed method achieves lightweight yet effective per-layer rate control that integrates seamlessly into existing DFL protocols, as detailed in the following section.

3. Methodology

3.1. Decentralized federated learning setup

We consider a DFL system comprising a set of nodes $\mathcal{N}$ interconnected in a P2P topology. Each node $i \in \mathcal{N}$ holds a private local dataset $\mathcal{D}_i$, which remains strictly local and unshared. Instead of relying on a centralized server, each node i collaborates by exchanging model parameters only with its direct neighbors $\mathcal{V}_i$. The objective is to learn model parameters that minimize the average of local loss functions across the network:

$$\min_{\theta_1,\dots,\theta_{|\mathcal{N}|}} \mathcal{F}(\theta_1,\dots,\theta_{|\mathcal{N}|}) \triangleq \frac{1}{|\mathcal{N}|}\sum_{i=1}^{|\mathcal{N}|}\mathcal{L}_i(\theta_i). \tag{1}$$

where $\mathcal{L}_i(\theta_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\ell(\theta_i; x, y)]$, and $\ell(\cdot)$ denotes a standard loss function. Solving this optimization problem requires iterative training through communication rounds, alternating between local training on heterogeneous data and network consensus aggregation.

3.2. Problem definition

Statistical heterogeneity in distributed federated systems creates uneven, layer-specific optimization challenges with significant implications for system convergence efficiency and practical deployment. During local training, specialized layers adapt rapidly to node-specific data, whereas foundational layers preserve task-general representations. Conversely, during network consensus, parameter averaging applies uniform pressure across all layers, stabilizing

foundational representations but potentially suppressing necessary adaptation in specialized layers. This layer-wise disparity creates competing optimization objectives: local learning drives specialized layers toward node-specific optima, while network consensus pushes foundational layers toward network-wide agreement.

Despite the disparity observed across layers, existing DFL methods typically apply a single scalar LR to all model layers, treating η as a global hyperparameter that modulates updates uniformly regardless of layer position or role. This uniform approach creates a fundamental optimization conflict. Low LRs help maintain consensus in foundational layers and stabilize the network, but they limit the ability of specialized layers to adapt to local data. In contrast, high LRs accelerate adaptation in specialized layers but destabilize foundational layers, causing parameter oscillations and degraded convergence. This dilemma becomes more pronounced under severe non-IID conditions, where conflicting local optima across heterogeneous nodes further widen the gap between local update intensity and network consensus constraints.

As a result, uniform LRs produce unstable parameter trajectories, prolonged convergence measured in communication rounds, and suboptimal model quality. In bandwidth-constrained and latency-sensitive edge deployments, resolving this conflict efficiently is critical: every communication round saved through improved convergence translates directly to reduced bandwidth consumption, lower energy expenditure, and faster system response time. Such efficiency gains are especially critical for large-scale IoT and mobile edge network scenarios where communication and energy costs directly impact system viability. Addressing this limitation requires dynamic, layer-wise LR adjustment derived from layer-specific signals that are extracted locally, without global coordination.

3.3. Node-level procedure and layer-wise architecture

Standard DFL methods apply a single scalar LR η uniformly to all model layers, ignoring their distinct optimization needs. In contrast, the proposed approach assigns a layer-wise LR vector $\eta_i^r = [\eta_{i,1}^r, \dots, \eta_{i,L}^r]$ at each node i and round r, where each component $\eta_{i,l}^r$ represents the LR for layer l. This vectorized design allows different layers to adopt LRs that better match their roles: foundational layers can use conservative rates to preserve network consensus, while specialized layers can use larger rates to accelerate adaptation to local training. The core challenge is to compute these per-layer LRs adaptively using only information available locally within each DFL round.

To enable such computation, the standard DFL node procedure is formalized in terms of three intra-round model states, as illustrated in Fig. 1(a). Within each communication round r, node i performs three sequential operations: (1) local training on its private dataset $\mathcal{D}_i$ for E epochs, (2) exchange of updated parameters with neighbors $j \in \mathcal{V}_i$, and (3) aggregation of neighbor models to maintain network consensus. This iterative process repeats over R communication rounds.

FedA2L characterizes the node procedure using three model states within each round. The base state $\theta_{i,l}^{B,r}$ represents the parameters of layer l at node i before local training in round r. After local training on $\mathcal{D}_i$ for E epochs, node i reaches the trained state $\theta_{i,l}^{T,r}$, reflecting local update intensity from node-specific data distributions. This local training transition, $\theta_{i,l}^{B,r} \rightarrow \theta_{i,l}^{T,r}$, captures how local optimization pushes the layer toward node-specific optima. After model exchange and aggregation with neighbors $j \in \mathcal{V}_i$, node i obtains the aggregated state $\theta_{i,l}^{A,r}$, incorporating neighbor information. This consensus transition, $\theta_{i,l}^{T,r} \rightarrow \theta_{i,l}^{A,r}$, captures network consensus constraints at the layer level. The aggregated state then becomes the base state for the next round ($\theta_{i,l}^{B,r+1} := \theta_{i,l}^{A,r}$), ensuring continuity across communication rounds.

3.4. FedA2L methodology

FedA2L dynamically computes and assigns layer-wise LRs from real-time model divergence signals extracted from the three-state model of DFL node procedures. The approach operates entirely locally at each node, requiring no centralized coordination or additional communication, thereby preserving the decentralized architecture of DFL. Fig. 1(b) illustrates the integrated FedA2L method procedure within the standard DFL workflow shown in Fig. 1(a).

To provide a formal abstraction of the overall process, FedA2L can be viewed as a per-node mapping from intra-round model states to layer-wise LRs. Specifically, at each node i and round r, the method defines the following mapping:

$$M_i^r\left(\theta_i^{B,r}, \theta_i^{T,r}, \theta_i^{A,r}\right) \rightarrow \eta_i^{r+1} = \left[\eta_{i,1}, \eta_{i,2}, \dots, \eta_{i,L}\right]^{r+1} \quad (2)$$

This mapping is realized through three sequential steps. First, dual layer-wise metrics are extracted from the intra-round state transitions, capturing both local update intensity and network consensus constraints. Next, these metrics are normalized using recent temporal statistics, filtering transient fluctuations to ensure that LR adjustments respond to persistent optimization trends. Finally, the normalized signals are fused into per-layer adaptive LRs through a bounded mechanism. The following subsections detail each step in turn.

3.4.1. Step 1: Dual metrics for characterizing layer dynamics

Two complementary metrics are extracted from the three-state model to capture distinct aspects of layer-wise optimization dynamics. Weight divergence quantifies local update intensity during the local training, while aggregation stability quantifies network consensus constraints during the network consensus. These metrics provide layer-specific diagnostic signals encoding the competing forces of local learning and network consensus. FedA2L leverages these signals to compute per-layer LRs balancing both requirements.

Local update intensity via weight divergence. To quantify how strongly each layer responds to node-specific

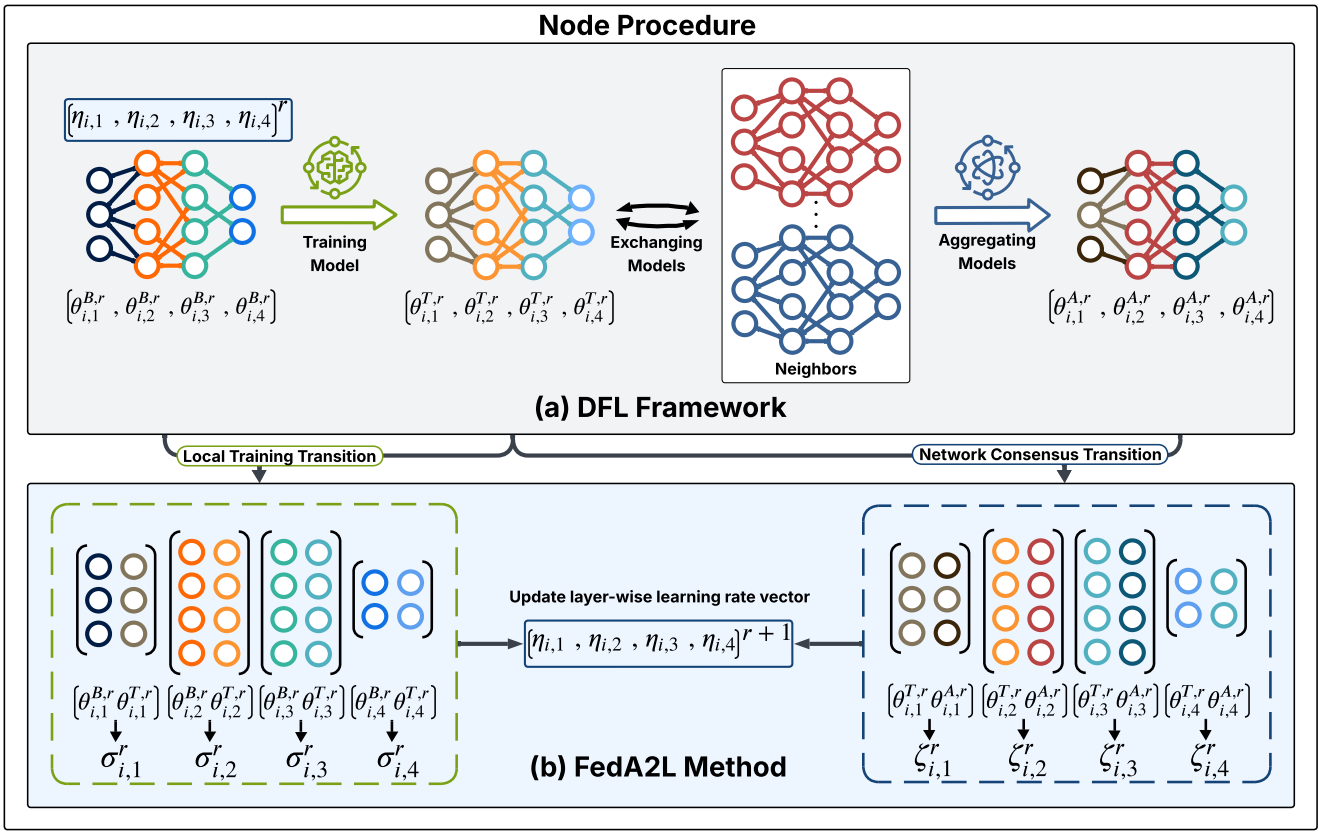


Figure 1: (a) Standard DFL framework with three intra-round model states per layer ($\theta^{B,r}_{i,l}$, $\theta^{T,r}_{i,l}$, $\theta^{A,r}_{i,l}$). (b) FedA2L integrated into the DFL framework. During each round, the three states of layer l are used to compute two layer-wise metrics, $\sigma^r_{i,l}$ (Eq. 3) and $\zeta^r_{i,l}$ (Eq. 4), which in turn determine the adaptive layer-wise learning rate vector $\eta^{r+1}_i = [\eta_{i,1}, \ldots, \eta_{i,L}]^{r+1}$ for the next round.

data, we examine the parameter change during the local training transition. This transition reveals the degree to which local optimization on node-specific data modifies each layer's parameters. After node i completes E epochs of local training, transitioning from the base state $\theta^{B,r}_{i,l}$ to the trained state $\theta^{T,r}_{i,l}$, the per-layer local update intensity is defined as:

$$\sigma^r_{i,l} = \frac{\|\theta^{T,r}_{i,l} - \theta^{B,r}_{i,l}\|_2}{\|\theta^{B,r}_{i,l}\|_2 + \epsilon} \tag{3}$$

where $\|\cdot\|_2$ is the L2-norm and ϵ is a small constant for numerical stability. Large values of $\sigma^r_{i,l}$ reflects substantial parameter modifications during local training, indicating strong adaptation to node-specific data characteristics. Conversely, small values indicate stable layer parameters with minimal changes, suggesting the layer is already well-adapted to local data. This normalized ratio design is scale-invariant, preventing bias from layers with larger absolute parameter magnitudes, and provides a principled measure of relative update magnitude across heterogeneous layer architectures. This per-layer perspective on weight divergence adapts concepts from prior FL work [40] into a diagnostic signal that operates at layer granularity rather than at the model level.

Network consensus constraints via aggregation stability. To quantify how well each layer aligns with neighbors during the network consensus transition, we examine the magnitude of parameter changes that occur when node i aggregates trained parameters with its neighbors, transitioning from the trained state $\theta^{T,r}_{i,l}$ to the aggregated state $\theta^{A,r}_{i,l}$. The aggregation stability metric measures the proportion of layer parameters that remain stable after aggregation, indicating consensus alignment. Motivated by prior work on consensus mechanisms [41], the aggregation stability for layer l is defined as:

$$\zeta^r_{i,l} = \frac{1}{|\theta^r_{i,l}|} \sum_{k=1}^{|\theta^r_{i,l}|} \mathbb{I}\left[\left|\theta^{A,r}_{i,l,k} - \theta^{T,r}_{i,l,k}\right| < \tau\right] \tag{4}$$

where $|\theta^r_{i,l}|$ denotes the total number of parameters in layer l, $\theta^{T,r}_{i,l,k}$ and $\theta^{A,r}_{i,l,k}$ represent the k-th parameter of layer l at node i after local training and after aggregation in round r, respectively, τ is a stability threshold determining parameter stability tolerance, and $\mathbb{I}[\cdot]$ is the indicator function. The metric counts parameters with change magnitude less than τ, normalized by total parameter count.

A high $\zeta^r_{i,l}$ value (approaching 1) indicates most parameters remained stable during aggregation, reflecting strong consensus alignment and minimal disagreement among neighbors. Conversely, a low value indicates substantial parameter

corrections during aggregation, suggesting weak consensus alignment that may require stabilization in subsequent rounds.

3.4.2. Step 2: Z-score-based normalization for anomaly detection

These raw metrics ($\sigma^r_{i,l}$ and $\zeta^r_{i,l}$) can fluctuate significantly due to stochastic gradient variations and transient network dynamics, particularly during early training when limited historical information is available. To prevent LR adjustments from reacting to noise rather than genuine optimization dynamics, FedA2L employs two mechanisms that stabilize metric estimation: a warm-up initialization period and temporal normalization.

For a warm-up initialization, the base LR is applied uniformly across all layers during the initial R_{warm} rounds, allowing sufficient metric history to accumulate and enabling early consensus to stabilize. This warm-up period provides a stable foundation for reliable metric estimation, after which FedA2L transitions to temporal normalization to refine LR modulation.

After the warm-up period, FedA2L transitions to temporal normalization to ensure that LR adjustments respond to persistent trends rather than transient fluctuations. For rounds $r > R_{\text{warm}}$, raw metrics are standardized using Z-score normalization computed over a sliding temporal window of the last ρ rounds. This process produces the normalized intensity $\omega^r_{i,l}$ (derived from $\sigma_{i,l}$) and $\delta^r_{i,l}$ (derived from $\zeta_{i,l}$) signals, defined as:

$$\omega^r_{i,l} = \frac{\sigma^r_{i,l} - \frac{1}{\rho}\sum_{k=r-\rho}^{r} \sigma^k_{i,l}}{\text{STD}(\{\sigma^k_{i,l}\}^r_{k=r-\rho}) + \epsilon} \tag{5}$$

$$\delta^r_{i,l} = \frac{\zeta^r_{i,l} - \frac{1}{\rho}\sum_{k=r-\rho}^{r} \zeta^k_{i,l}}{\text{STD}(\{\zeta^k_{i,l}\}^r_{k=r-\rho}) + \epsilon} \tag{6}$$

where STD$(\cdot)$ computes the sample standard deviation over the last ρ rounds. These Z-scores quantify deviation from recent historical patterns in standardized units: large positive values indicate abnormal behavior compared to recent history, while values near zero reflect typical and stable behavior. This normalization ensures that LR adjustments respond to meaningful changes in optimization dynamics rather than absolute metric magnitudes, improving robustness across varying network heterogeneity and data distributions. By expressing each metric as a deviation from its own recent history, this step also places $\sigma^r_{i,l}$ and $\zeta^r_{i,l}$ on a comparable scale across layers, which is examined further in the ablation study of Section 5.6.

3.4.3. Step 3: Adaptive layer-wise learning rate

To reflect local update intensity and network consensus constraints simultaneously, FedA2L synthesizes the normalized signals into a unified fusion score. This fusion process leverages an exponential transformation to assign higher weights to larger deviations while ensuring that all weights remain positive. The fusion score is computed as:

$$\lambda^r_{i,l} = \beta \cdot e^{\omega^r_{i,l}} + (1-\beta) \cdot e^{\delta^r_{i,l}} \tag{7}$$

where $\beta \in [0,1]$ governs the relative emphasis between local update intensity (first term) and network consensus constraints (second term). This weighted combination produces a positive-valued fusion score that integrates both optimization signals into a single, interpretable measure.

Then, this fusion score is applied to modulate the base LR η^0 by combining with a global decay factor $\gamma^r = (1+\xi \cdot r)^{-0.5}$ to produce layer-wise adaptive LRs. The LR for layer l is defined as:

$$\eta^r_{i,l} = \eta^0 \cdot \Big(1 + \tanh(\log(\lambda^r_{i,l}))\Big) \cdot \gamma^r \tag{8}$$

where ξ controls the convergence speed across rounds, and the logarithmic transformation stabilizes the range of $\lambda^r_{i,l}$ by compressing large variations that could lead to unstable updates. The tanh function provides smooth, bounded modulation, and the decay factor γ^r applies polynomial decay, gradually reducing LRs across rounds while still allowing rapid adaptation in the early stages of training.

Since $\lambda^r_{i,l} > 0$, $\log(\lambda^r_{i,l}) \in \mathbb{R}$ and $\tanh(\log(\lambda^r_{i,l})) \in (-1,1)$, which implies $1 + \tanh(\log(\lambda^r_{i,l})) \in (0,2)$. Together with $\gamma^r \leq 1$, the effective layer-wise LRs satisfy $0 < \eta^r_{i,l} < 2\eta^0\gamma^r$, so FedA2L cannot arbitrarily amplify the step size. Moreover, the Z-score normalization in Eqs. 5 and 6 is computed over a sliding window of ρ rounds; persistent divergence shifts the running mean and variance, driving the normalized scores back toward zero and pulling $\lambda^r_{i,l}$ (and thus the LR multiplier) toward 1 over time.

When the fusion score $\lambda^r_{i,l} = 1$ (typical behavior), the modulation factor equals 1, giving $\eta^r_{i,l} = \eta^0 \cdot \gamma^r$, which aligns with the global decay schedule and represents a steady-state condition. For $\lambda^r_{i,l} > 1$ (high update intensity or weak consensus), the modulation factor exceeds 1, yielding $\eta^r_{i,l} > \eta^0 \cdot \gamma^r$ and allowing faster adaptation to node-specific patterns. In contrast, when $\lambda^r_{i,l} < 1$ (low update intensity or strong consensus), the LR decreases below the base rate, $\eta^r_{i,l} < \eta^0 \cdot \gamma^r$, preserving stable shared representations and network consensus.

FedA2L interprets large deviations in the normalized metrics $\omega^r_{i,l}$ (derived from $\sigma^r_{i,l}$) and $\delta^r_{i,l}$ (derived from $\zeta^r_{i,l}$) as evidence that a layer is under-adapting to its local objective in non-IID settings. Intuitively, reducing the LR under large deviations may suppress the magnitude of these normalized metrics; however, it also suppresses necessary adaptation in non-IID settings and limits layer-wise specialization. Therefore, when these normalized metrics indicate unusually strong local updates or weak consensus, FedA2L assigns a temporarily higher LR to that layer to accelerate adaptation under non-IID data.

Importantly, the bounded multiplier $1 + \tanh(\log \lambda^r_{i,l}) \in (0,2)$, the global decay factor γ^r, and the sliding-window

Z-score normalization ensure that persistent divergence is re-centered and the effective LR is gradually pulled back toward the global schedule. This fine-grained modulation directly addresses the asymmetric layer-wise dynamics by enabling specialized layers to adapt when local signals are strong, while allowing foundational layers to stabilize when consensus signals are strong.

3.5. FedA2L algorithm

Algorithm 1: FedA2L method integrated into a DFL framework.

Initialize: For each node $i \in \mathcal{N}$, initialize base model $\theta_i^{B,r}$ at round r, local dataset $\mathcal{D}_i$, neighbor list $\mathcal{V}_i$, and history lists $\sigma_i, \zeta_i \leftarrow \emptyset$.
for $r = 1$ **to** R **do**
  for *each node* $i \in \mathcal{N}$ **do**
    $\theta_i^{T,r} \leftarrow \text{LocalTraining}(\theta_i^{B,r}, \mathcal{D}_i, \eta_i^r)$;
    Exchange $\theta_i^{T,r}$ with neighbors $\mathcal{V}_i$;
    $\theta_i^{A,r} \leftarrow \text{Aggregation}(\{\theta_j^{T,r}\}_{j \in \mathcal{V}_i \cup \{i\}})$;
    $\theta_i^{B,r+1} \leftarrow \theta_i^{A,r}$
    // FedA2L: Adaptive LR Computation
    for *each layer* $l = 1$ **to** L **do**
      $\sigma_{i,l}^r \leftarrow \frac{\|\theta_{i,l}^{T,r} - \theta_{i,l}^{B,r}\|_2}{\|\theta_{i,l}^{B,r}\|_2 + \epsilon}$
      $\zeta_{i,l}^r \leftarrow \frac{1}{|\theta_{i,l}^r|} \sum_{k=1}^{|\theta_{i,l}^r|} \mathbb{I}\left[\left|\theta_{i,l,k}^{A,r} - \theta_{i,l,k}^{T,r}\right| < \tau\right]$
      Update metric histories σ_i, ζ_i;
      if *round* $r > R_{warm}$ **then**
        Compute Z-scores $\omega_{i,l}^r$ using Eq. 5;
        Compute Z-scores $\delta_{i,l}^r$ using Eq. 6;
        $\lambda_{i,l}^r \leftarrow \beta e^{\omega_{i,l}^r} + (1-\beta) e^{\delta_{i,l}^r}$
        $\eta_{i,l}^{r+1} \leftarrow \eta^0 (1 + \tanh(\log(\lambda_{i,l}^r)))\gamma^r$
      end if
    end for
  end for
end for

Algorithm 1 presents the complete FedA2L method integrated modularly into a standard DFL training workflow over R communication rounds. At each round r, after local training and network consensus, FedA2L executes locally at each node. It computes layer-wise metrics $\sigma_{i,l}^r$ and $\zeta_{i,l}^r$ from locally available model states ($\theta_i^{B,r}$, $\theta_i^{T,r}$, $\theta_i^{A,r}$) and then produces the adaptive LR vector $\eta_{i,l}^{r+1}$ for the next round. This local-only execution preserves decentralization and does not introduce additional data exposure beyond the standard DFL model-sharing protocol.

FedA2L achieves high computational efficiency, with the per-round and per-node computation having complexity $\mathcal{O}(|\theta| \cdot |\mathcal{D}_i| \cdot E)$, where $|\theta|$ denotes the model size, $|\mathcal{D}_i|$ is the dataset size, and E is the number of local steps. In addition, adaptive computation involves per-layer metrics ($\sigma_{i,l}^r, \zeta_{i,l}^r$) and historical statistics, with complexity $\mathcal{O}(|\theta| + L\rho)$, where L is the number of layers and ρ is the normalization window size. As the adaptive computation relies solely on model states rather than training data, this overhead scales only with model depth and normalization history, independent of the local dataset size and training workload. Since $|\mathcal{D}_i| \cdot E \gg L\rho$ in practical DFL configurations, the resulting overhead remains limited compared with the dominant cost of local training. Critically, FedA2L introduces *zero additional communication overhead*, requiring no extra messages, proxy datasets, or centralized coordination, and preserving the decentralized nature and communication efficiency of DFL.

4. Convergence analysis

We now analyze FedA2L in the DFL setup of Section 3.1. Recall that the network optimizes

$$\min_{\theta_1,\dots,\theta_{|\mathcal{N}|}} \mathcal{F}(\theta_1, \dots, \theta_{|\mathcal{N}|}) \triangleq \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \mathcal{L}_i(\theta_i), \tag{9}$$

where $\mathcal{L}_i(\theta_i) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\ell(\theta_i; x, y)]$ is the local loss at node i. For the analysis, we also consider the centralized surrogate

$$f(\theta) \triangleq \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \mathcal{L}_i(\theta), \tag{10}$$

and track the average model

$$\bar{\theta}^r \triangleq \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \theta_i^{B,r}, \tag{11}$$

where $\theta_i^{B,r}$ denotes the base state of node i at round r (Section 3.3).

Within each round, FedA2L modifies only the local step sizes. If the model is decomposed into L layers, $\theta_i^{B,r} = [\theta_{i,1}^{B,r}; \dots; \theta_{i,L}^{B,r}]$, the effective LR for layer l at node i in round r is given by Eq. 8:

$$\eta_{i,l}^r = \eta_0 \gamma_r \Big(1 + \tanh\big(\log \lambda_{i,l}^r\big)\Big), \qquad \gamma_r = (1+\xi r)^{-1/2}, \tag{12}$$

where $\lambda_{i,l}^r > 0$ is the fusion score computed from the dual metrics ($\sigma_{i,l}^r$ and $\zeta_{i,l}^r$) in Eqs. 3-7. Since $\tanh : \mathbb{R} \to (-1, 1)$ and $\lambda_{i,l}^r > 0$, we always have

$$0 < \eta_{i,l}^r < 2\eta_0\gamma_r \quad \text{for all } i, l, r, \tag{13}$$

i.e., FedA2L keeps all layer-wise LRs strictly positive and uniformly bounded.

Assumptions. We adopt standard assumptions from decentralized optimization and DFL:

A1 (Smoothness). Each $\mathcal{L}_i$ is $\mathbb{L}$-smooth: $\|\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}_i(y)\| \le \mathbb{L}\|x - y\|$ for all x, y. Hence f is also $\mathbb{L}$-smooth.

A2 (Unbiased gradients & bounded variance). Stochastic gradients $g_i(\theta; b)$ (where b denotes a random mini-batch drawn from $\mathcal{D}_i$) satisfy

$$\mathbb{E}[g_i(\theta; b)] = \nabla\mathcal{L}_i(\theta), \qquad \mathbb{E}\big[\|g_i(\theta; b) - \nabla\mathcal{L}_i(\theta)\|^2\big] \le \varrho^2.$$

A3 (Mixing matrix & spectral gap). Each mixing matrix $W^r = [w_{ij}^r]_{i,j}$ used in the aggregation step (Algorithm 1,

line 6) is symmetric and doubly stochastic, $W^r\mathbf{1} = \mathbf{1}$ and $(W^r)^\top\mathbf{1} = \mathbf{1}$, and there exists a spectral gap $\phi \in (0, 1]$ such that

$$\left\| W^r - \tfrac{1}{|\mathcal{N}|}\mathbf{1}\mathbf{1}^\top \right\|_2 \le 1 - \phi \quad \text{for all } r.$$

A4 (Data heterogeneity). There exists $B \ge 0$ such that for all θ,

$$\frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \left\| \nabla\mathcal{L}_i(\theta) - \nabla f(\theta) \right\|^2 \le B^2.$$

A5 (Bounded and slowly varying multipliers). Define layer-wise multipliers $m^r_{i,l}$ via $\eta^r_{i,l} = m^r_{i,l}(\eta_0\gamma_r)$, where $\eta_0 = c/\sqrt{R}$. There exist constants $0 < m_{\min} \le m_{\max} < \infty$ and $\Delta_m < \infty$ such that

$$m_{\min} \le m^r_{i,l} \le m_{\max}, \qquad |m^{r+1}_{i,l} - m^r_{i,l}| \le \Delta_m, \quad \forall\, i, l, r.$$

In FedA2L, $m^r_{i,l} = 1 + \tanh(\log \lambda^r_{i,l})$, so $0 < m^r_{i,l} < 2$ by design, and the Z-score based update over a finite window (Eqs. 5-6) bounds metric volatility, which, when coupled with the bounded tanh transformation, ensures that $|m^{r+1}_{i,l} - m^r_{i,l}| \le \Delta m$ for some finite Δm.

We also measure consensus by the mean-squared disagreement

$$D^r \triangleq \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}}\sum_{l=1}^{L} \left\| \theta^{B,r}_{i,l} - \bar\theta^r_l \right\|^2, \qquad \bar\theta^r_l = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \theta^{B,r}_{i,l}. \tag{14}$$

4.1. Non-convex convergence

We first consider the general non-convex setting. Let $\eta^r_{\min} \le \eta^r_{i,l} \le \eta^r_{\max}$ denote lower/upper bounds on the effective step sizes in round r, induced by (13) and A5.

Theorem 1 (Non-convex FedA2L convergence)**.** *Assume **A1-A5** and one local SGD epoch per round at each node (Algorithm 1 with $E = 1$). Suppose the effective step sizes satisfy $0 < \eta^r_{\min} \le \eta^r_{i,l} \le \eta^r_{\max}$ with $\eta^r_{\max} L$ sufficiently small for all r, which is the standard small-step condition ensuring that higher-order perturbation terms remain controlled under Lipschitz smoothness. Then, for any $R \ge 1$,*

$$\frac{\sum_{r=0}^{R-1}\eta^r_{\min}\,\mathbb{E}\left[\|\nabla f(\bar\theta^r)\|^2\right]}{\sum_{r=0}^{R-1}\eta^r_{\min}} \le \frac{2\big(f(\bar\theta^0) - f_\star\big)}{\sum_{r=0}^{R-1}\eta^r_{\min}} + C_1 L\,\overline{\eta}_{\max}\frac{\varrho^2}{|\mathcal{N}|} + C_2\frac{L^2\,\overline{\eta}^2_{\max}B^2}{\phi^2} + C_3\frac{L\,\overline{\eta}^2_{\max}\Delta_m^2}{\phi^2}, \tag{15}$$

where $f_\star = \inf_\theta f(\theta)$, $\overline{\eta}_{\max} = \max_{0\le r<R}\eta^r_{\max}$, and $C_1, C_2, C_3 > 0$ are constants independent of R.

Sketch of proof. Using $\mathbb{L}$-smoothness, one shows that the average model satisfies

$$\mathbb{E}[f(\bar\theta^{r+1})] \le \mathbb{E}[f(\bar\theta^r)] - \tfrac{\eta^r_{\min}}{2}\,\mathbb{E}\|\nabla f(\bar\theta^r)\|^2 + O\big((\eta^r_{\max})^2\big(\varrho^2 + B^2 + \mathbb{E}D^r\big)\big).$$

A separate recursion for the disagreement D^r follows from the spectral gap ϕ of W^r: consensus contracts by a factor $(1-\phi)$ each round, but is driven by stochastic gradient noise ϱ^2, heterogeneity B^2, and the per-round drift Δ_m of the multipliers, which remains bounded due to the multiplier construction in A5 and the bounded tanh transformation in Eq. 8. Summing the descent inequality over $r = 0, \dots, R-1$, bounding $\mathbb{E}D^r$ by the steady state of this consensus recursion, and dividing by $\sum_{r=0}^{R-1}\eta^r_{\min}$ yields (15).

Diminishing step sizes. For the theoretical analysis, we set $\gamma_r = 1$ and use the budget-dependent base LR $\eta_0 = c/\sqrt{R}$ from A5. Combined with bounded multipliers $m^r_{i,l} \in [m_{\min}, m_{\max}]$ (A5), this gives round-independent effective step sizes $\eta^r_{i,l} = m^r_{i,l}\,\eta_0$, so

$$\overline{\eta}_{\max} = m_{\max}\frac{c}{\sqrt{R}} = O\!\left(\frac{1}{\sqrt{R}}\right),$$
$$\sum_{r=0}^{R-1}\eta^r_{\min} \ge R\cdot m_{\min}\frac{c}{\sqrt{R}} = \Omega(\sqrt{R}).$$

The three error terms in (15) then satisfy $C_1 L\,\overline{\eta}_{\max} = O(1/\sqrt{R})$ and $C_2, C_3 \propto \overline{\eta}^2_{\max} = O(1/R)$, so all terms vanish as $R \to \infty$. Substituting into Theorem 1 yields

$$\frac{\sum_{r=0}^{R-1}\eta^r_{\min}\,\mathbb{E}\|\nabla f(\bar\theta^r)\|^2}{\sum_{r=0}^{R-1}\eta^r_{\min}} = O\!\left(\frac{1}{\sqrt{R}}\right),$$

matching the standard $O(1/\sqrt{R})$ non-convex rate of decentralized SGD. FedA2L does not worsen the convergence order; the bounded multipliers $m^r_{i,l}$ only modify the constants C_1, C_2, C_3. In practice we employ the diminishing schedule $\gamma_r = (1 + \xi r)^{-1/2}$ to improve empirical stability.

4.2. Strongly convex case

If f is additionally μ-strongly convex and the effective step sizes are kept constant over rounds, $\eta^r_{i,l} \in [\eta_{\min}, \eta_{\max}]$ with $\eta_{\max}$ sufficiently small, the same arguments with a Lyapunov function $V^r = \mathbb{E}[f(\bar\theta^r) - f_\star] + \gamma\,\mathbb{E}[D^r]$ show that FedA2L preserves the linear convergence of decentralized SGD: for suitable $\gamma > 0$,

$$\mathbb{E}\big[f(\bar\theta^R) - f_\star\big] \le \rho_0^R\big(f(\bar\theta^0) - f_\star\big) + O\!\left(\frac{L\,\overline{\eta}_{\max}\varrho^2}{|\mathcal{N}|} + \frac{L^2\,\overline{\eta}^2_{\max}B^2}{\phi^2} + \frac{L\,\overline{\eta}^2_{\max}\Delta_m^2}{\phi^2}\right), \tag{16}$$

with contraction factor $\rho_0 \in (0, 1)$ determined jointly by μ and the spectral gap ϕ. Thus, in the constant step-size regime, FedA2L does not degrade the qualitative linear rate of the underlying DFL algorithm; it only changes the constants through bounded, layer-wise multipliers.

This strongly convex result is provided for completeness in the constant step-size regime. Since our experiments adopt the diminishing schedule in Eq. 12, the main theoretical implication for the practical setup is the preservation of the standard non-convex convergence rate established above.

5. Experiments

This section presents the evaluation setup and performance analysis of FedA2L across image classification and time-series forecasting (TSF) tasks, covering multiple benchmark datasets, model architectures, DFL algorithms, and network configurations. The evaluation considers five complementary dimensions. Convergence speed is measured by communication rounds required to reach a target accuracy. Computational efficiency is measured by total time to reach the same target. Model quality is assessed by best test accuracy on classification tasks and MSE on regression tasks, capturing whether convergence gains carry through to model performance. Robustness is examined under varying data heterogeneity, network scales, and sparse topologies. Finally, ablation and sensitivity studies isolate the contribution of each methodological component and examine how key design choices affect convergence behavior. The corresponding analyses are presented in Sections 5.2-5.6.

5.1. Experimental setup

5.1.1. Datasets

Experimental evaluation was conducted on five benchmark datasets spanning image classification and TSF. For image classification, three standard benchmarks are used: CIFAR-10, CIFAR-100, and TinyImageNet. CIFAR-10 and CIFAR-100 [42] contain 60,000 32×32 color images across 10 and 100 classes, respectively, enabling assessment of both coarse and fine-grained classification tasks. TinyImageNet [43] comprises 100,000 64×64 color images across 200 classes, providing a more challenging and realistic benchmark for evaluating scalability on larger-scale vision tasks.

To emulate realistic heterogeneous data distributions (non-IID data distribution), a Dirichlet distribution with parameter α was used to partition datasets across nodes. Each node i receives a subset of samples, where α controls heterogeneity level. Lower α values correspond to severe non-IID conditions where each node receives highly skewed class distributions, while higher values yield more balanced distributions approaching IID. Default experiments used $\alpha = 0.1$, with robustness analysis varying $\alpha \in \{0.01, 0.5\}$.

For TSF, two real-world benchmarks are used: ExchangeRate [44] and BeijingAirQuality [45]. ExchangeRate contains 7,588 daily exchange rate records across 8 currencies, where each currency corresponds to a node. BeijingAirQuality comprises hourly air quality measurements from 12 monitoring stations across Beijing, with 11 variates per node. Both use an input horizon of 96 and a prediction horizon of 96 with naturally defined node partitions (i.e., no Dirichlet-based partitioning). These datasets reflect real-world distributed time-series data with temporal dependencies across nodes.

5.1.2. Models

Model architectures cover varying complexity levels. For image classification, a 4-layer convolutional neural network (CNN) for lightweight evaluation, ResNet-18 for moderate-scale tasks, and ResNet-34 for deeper network evaluation. For TSF, we use Timer [46], an 18-layer transformer-based forecasting model. This diversity of models enables a comprehensive evaluation of FedA2L's effectiveness across classification and regression settings with architectures of different depth and complexity.

5.1.3. DFL configuration

All experiments were conducted with $|\mathcal{N}| = 10$ nodes arranged in a fully connected P2P topology, using synchronous communication unless otherwise specified. Each node performs local training for $E = 1$ epoch per communication round, with initial LR $\eta^0 = 0.01$. Classification experiments train for $R = 500$ communication rounds, and convergence is evaluated by measuring the rounds required to reach predefined target test accuracy. The target accuracies are selected as stable and practically meaningful convergence levels that the majority of baseline methods can reach and sustain, reflecting genuine learning progress in the distributed setting. TSF experiments train for $R = 100$ communication rounds, and convergence is evaluated by the rounds required to reach a target MSE threshold. All experiments are conducted on a server equipped with an NVIDIA GeForce RTX 4090 GPU and 126 GB of RAM, running PyTorch 2.5.0 under Python 3.12. All results represent mean values across 5 independent runs with different random seeds to ensure statistical robustness.

5.1.4. Baseline methods

FedAvg [3] (vanilla averaging baseline), FedProx [14] (proximal term for heterogeneity), FedYogi [22] (adaptive second-moment optimization), FedNTD [47] (knowledge distillation approach), FedAWA [36] (adaptive aggregation method), DFedSAM [48] (sharpness-aware minimization for flat minima under non-IID conditions), and DFedHPO [24] (decentralized hyperparameter optimization). DFedHPO conducts a one-time decentralized search to provide a single optimal fixed LR configuration before training begins, while FedA2L continuously adapts layer-wise LRs during training. As a result, the fixed LR from DFedHPO serves as an initialization, and subsequent optimization is governed by FedA2L. Accordingly, DFedHPO is evaluated as a standalone reference baseline, using its searched LR as a fixed configuration without additional tuning procedures or schedulers. This enables a direct comparison between static pre-training hyperparameter search and dynamic layer-wise adaptation during training. These baselines represent classical, regularized, adaptive, network-aware, geometric, and hyperparameter optimization strategies in DFL, providing a solid foundation for comparison with FedA2L, a layer-wise adaptive LR method. For each base algorithm, three distinct LR strategies are evaluated, with the exception of DFedHPO, which is assessed only under its searched LR configuration as described above:

- **Vanilla (Unchanged LR):** Single constant LR maintained throughout training, serving as the reference for comparison.

- **Scheduler-based:** Global LR adjusted per round according to established schedulers: StepLR (exponential decay), CosineAnnealingWarmRestarts (CAWR) with periodic warm restarts [49], OneCycleLR (OCLR) [50] with triangular scheduling, and Hyperbolic scheduler [51] providing efficient decay.
- **FedA2L (Proposed):** Layer-wise adaptive LRs dynamically computed per round.

These three strategy categories encompass the spectrum from static (vanilla) through globally adaptive (schedulers) to locally adaptive layer-wise (FedA2L), providing a comprehensive evaluation framework. This evaluation structure enables fair assessment: unchanged and scheduler baselines provide performance bounds for global (non-adaptive) strategies, while FedA2L demonstrates the advantage of per-layer adaptation.

5.2. How fast does FedA2L converge?

5.2.1. Comparison across models and datasets

FedA2L demonstrates consistent acceleration across diverse models and datasets, with the performance gain increasing as model complexity and dataset diversity increase. Across the 36 scheduler-strategy settings in Table 2, FedA2L achieves the fastest convergence in 32 cases and the second-fastest in 3, confirming robust and generalizable performance. FedA2L also converges faster than DFedHPO across all evaluated configurations, confirming the advantage of dynamic layer-wise adaptation over static hyperparameter search. The *Improv.* rows quantify the percentage reduction in communication rounds relative to the best-performing baseline per setting, providing a direct measure of practical efficiency gain.

Across ResNet-18 and ResNet-34 architectures, FedA2L consistently reduces convergence rounds across all seven algorithms, with the strongest gains on TinyImageNet ranging from 2.77% under DFedSAM to 59.34% under FedYogi. Against the best scheduler, FedA2L achieves further reductions on all ResNet settings. Additionally, FedA2L exhibits notably low variance across independent runs: on TinyImageNet with ResNet-34 under FedAvg, FedA2L converges in 71±1.25 rounds compared to 148±38.81 for CAWR, reflecting stable and reproducible behavior that is important for deployment in resource-constrained edge environments.

On CNN architectures, FedA2L achieves positive gains on CIFAR-10 across most algorithms, reaching up to 28.75% improvement under DFedSAM, while ranking second in the remaining settings where the base algorithm already partially captures layer-level variance. In all such cases, FedA2L still outperforms the vanilla baseline, confirming that the layer-wise mechanism does not destabilize training. The reasons for these interactions are discussed in the following subsection.

5.2.2. Comparison across DFL algorithms

A key property of FedA2L is its algorithmic orthogonality, enabling seamless integration with existing DFL protocols without modifying the core aggregation logic. This is particularly important for practitioners deploying mature, established DFL systems. Table 2 shows consistent acceleration across six DFL algorithms, confirming that this property holds across fundamentally different DFL design philosophies under diverse update mechanisms. Under FedProx, which applies proximal regularization to reduce client drift, FedA2L still achieves 2.24× speedup on CNN CIFAR-10 (127 vs. 285 rounds), confirming that layer-wise rate adaptation targets an orthogonal optimization dimension to loss-based regularization. Under FedAWA, which reweighs aggregation contributions based on model similarity, FedA2L reaches 72 rounds versus 356 rounds on ResNet-34 TinyImageNet (4.94× speedup), demonstrating that adaptive aggregation and adaptive layer-wise LRs are complementary rather than competing mechanisms.

The only exceptions arise in two CNN configurations under FedNTD and two CIFAR-100 settings under FedYogi. In these cases, the base algorithm already partially captures layer-level variance through knowledge distillation in FedNTD and per-parameter moment estimation in FedYogi, reducing the available margin for further layer-wise modulation. This limits, but does not eliminate, the benefit of additional layer-wise adaptation. Nonetheless, FedA2L still outperforms the vanilla baseline in all four cases, confirming that the layer-wise mechanism remains compatible with the base protocol even when divergence signals are partially attenuated.

5.2.3. Comparison across learning rate strategies

Global LR schedulers address temporal heterogeneity by adjusting rates over training rounds. However, they cannot address spatial heterogeneity, which refers to the differing optimization needs across model layers. FedA2L consistently outperforms these scheduler-based baselines (Table 2) by leveraging local layer signals to detect each layer's specific optimization needs during each communication round. This distinction becomes critical in deep heterogeneous models. With ResNet-18 on CIFAR-100 (45% target accuracy), FedA2L converges in 177 rounds compared to 491 rounds for the vanilla baseline under FedAvg (2.77× speedup) and 197 rounds for the best scheduler, CAWR (10.15% further improvement). On TinyImageNet with ResNet-18, the advantage becomes more pronounced. Under FedNTD, all schedulers fail to converge, whereas FedA2L reaches the target in 175 rounds, demonstrating that global scheduling is insufficient under the combined effects of deep layer heterogeneity and severe non-IID conditions.

5.2.4. Comparison across model architectures

The effectiveness of FedA2L scales consistently with architectural depth, following a pattern visible across all algorithm rows in Table 2. On ResNet-34, the deepest architecture evaluated, FedA2L delivers the strongest gains across all algorithms. On TinyImageNet, improvements range from 2.77% under DFedSAM to 59.34% under FedYogi, while the vanilla baseline fails to converge in several settings where FedA2L successfully reaches the target accuracy. This pattern confirms that deep feature hierarchies combined with

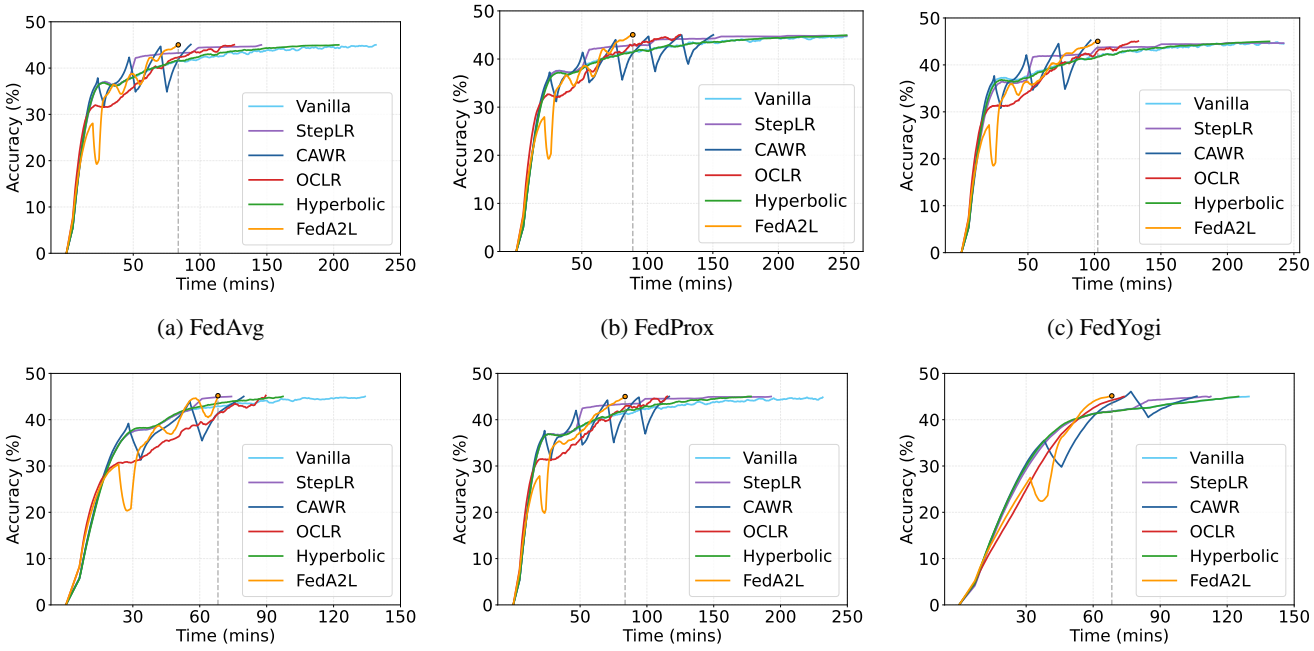
Table 2
Communication rounds to target accuracy in DFL (FedA2L vs. baselines). Results use a sliding average (window = 10). Bold: fastest; underline: second-fastest; "_": no convergence. '1^{st} count' = number of fastest cases. DFedHPO is reported only under its searched LR, as it does not incorporate additional LR scheduling.

Model		CNN		ResNet-18		ResNet-34		1^{st} Count
Dataset		CIFAR-10	CIFAR-100	CIFAR-100	TINY	CIFAR-100	TINY	
Algorithm	Method	64%	29%	45%	35%	41%	30%	
FedAvg	Vanilla	374±38.93	29±22.07	491±53.13	_	406±50.26	_	0
	StepLR	190±23.25	27±0.4	309±33.86	314±44.85	308±8.52	214±29.17	0
	CAWR	191±40.63	**23±0.64**	197±46.2	_	199±23.84	148±38.81	1
	OCLR	165±17.81	58±1.09	266±11.23	450±9.29	142±9.94	225±12.53	0
	Hyperbolic	221±17.82	27±36.8	432±37.5	_	_	256±32.93	0
	FedA2L	**136±8.21**	**23±0.49**	**177±6.97**	**187±15.83**	**120±12.1**	**71±1.25**	**6**
	Improv.	*17.57%*	*0.0%*	*10.15%*	*40.44%*	*15.49%*	*52.03%*	
FedProx	Vanilla	285±51.04	25±0.8	_	_	389±75.5	_	0
	StepLR	144±14.6	25±0.8	_	_	406±66.09	_	0
	CAWR	148±64.78	**23±0.63**	297±23.63	_	244±41.62	194±22.35	1
	OCLR	149±29.97	48±0.8	247±11.38	447±4.72	132±10	182±11.92	0
	Hyperbolic	213±27.6	25±0.75	_	_	266±38.96	251±52.89	0
	FedA2L	**127±24.89**	**23±0.49**	**175±15.14**	**185±7.88**	**115±3.24**	**82±0.82**	**6**
	Improv.	*11.81%*	*0.0%*	*29.15%*	*58.61%*	*12.88%*	*54.95%*	
FedYogi	Vanilla	311±34.98	27±1.79	_	_	367±51.4	308±5.63	0
	StepLR	163±29.92	27±1.79	_	_	306±51.39	207±27.81	0
	CAWR	189±35.88	**23±0.63**	**198±48.88**	248±35.39	198±67.97	195±21.82	2
	OCLR	161±22.01	49±0.98	273±8.45	454±11.23	152±17.69	182±12.03	0
	Hyperbolic	214±31.26	27±1.09	361±27.6	_	324±42.15	251±72.24	0
	FedA2L	**141±25.82**	25±0.49	210±8.7	**240±6.98**	**134±7.678**	**74±1.67**	**4**
	Improv.	*12.42%*	*-8.69%*	*-6.06%*	*3.22%*	*11.84%*	*59.34%*	
FedNTD	Vanilla	225±68.65	24±0.89	242±29.17	_	239±45.26	_	0
	StepLR	**141±27.91**	24±0.89	133±40.57	_	209±51	_	1
	CAWR	190±72.52	**21±0.4**	143±3.06	_	136±20.44	149±1.5	1
	OCLR	150±17.74	47±1.02	161±4.21	_	99±1.94	124±3.29	0
	Hyperbolic	218±24.68	24±1.2	175±16.02	_	_	_	0
	FedA2L	167±17.29	23±0.49	**122±7.81**	**175±17.15**	**88±1.02**	**72±1.69**	**4**
	Improv.	*-18.44%*	*-9.5%*	*8.3%*	*100%*	*11.11%*	*41.94%*	
FedAWA	Vanilla	387±51.94	30±29.67	_	_	264±75	356±74.96	0
	StepLR	195±14.03	27±53.83	409±49.50	_	202±41.41	201±15.92	0
	CAWR	243±77.98	**23±0.8**	247±68.22	248±41.42	197±24.31	146±1.25	1
	OCLR	174±18.11	48±1.02	244±17.68	440±29.56	139±3.76	209±0.82	0
	Hyperbolic	218±13.95	26±1.74	372±86.86	_	286±70.81	495±113.5	0
	FedA2L	**152±14.66**	**23±0.8**	**177±7.49**	**195±8.24**	**112±3.26**	**72±0.94**	**6**
	Improv.	*12.64%*	*0.0%*	*27.45%*	*21.37%*	*19.42%*	*50.68%*	
DFedSAM	Vanilla	346±55.87	36±5.01	167±17.59	254±17.59	117±10.21	72±2.16	0
	StepLR	160±24.07	36±60.81	142±22.01	207±48.56	101±7.76	73±4.24	0
	CAWR	197±24.09	**27±0.47**	97±22.23	84±1.25	92±2.62	86±1.25	1
	OCLR	170±29.58	53±0.82	95±1.7	94±2.45	85±1.89	73±0.47	0
	Hyperbolic	196±19.13	31±0.47	161±0.47	175±55.5	95±27.82	175±55.75	0
	FedA2L	**114±2.36**	**27±0.49**	**88±1.25**	**81±2.85**	**80±2.62**	**70±0.82**	**6**
	Improv.	*28.75%*	*0.0%*	*7.37%*	*3.57%*	*5.88%*	*2.77%*	
DFedHPO	Vanilla	170±10.27	60±0.82	_	_	243±10.34	141±45.9	

large class diversity produce strong layer-wise divergence signals, precisely the condition under which FedA2L's per-layer rate control is most effective under non-IID settings. On ResNet-18, the reduced depth and lower class diversity yield consistent but more modest improvements across all base algorithms on both CIFAR-100 and TinyImageNet. On CNN, FedA2L acts primarily as a training stabilizer, achieving a 2.75× speedup over vanilla under FedAvg on CIFAR-10

Table 3: Comparing time to reach 45% target accuracy (total time in minutes) on CIFAR-100 using ResNet-18 (Dirichlet $\alpha = 0.1$).

Algorithm	FedAvg		FedProx		FedYogi		FedNTD		FedAWA		DFedSAM	
Method	Time per round (s)	Total Time (min)	Time per round (s)	Total Time (min)	Time per round (s)	Total Time (min)	Time per round (s)	Total Time (min)	Time per round (s)	Total Time (min)	Time per round (s)	Total Time (min)
Vanilla	28.43	232.63	30.02	_	28.62	_	32.69	131.85	28.18	_	46.61	129.73
StepLR	28.23	145.38	30.02	_	28.60	_	32.46	71.95	28.38	193.46	46.23	67.68
CAWR	28.15	92.41	29.68	146.92	28.82	**95.09**	32.82	78.22	28.48	117.23	46.47	46.59
OCLR	28.15	124.78	29.78	122.60	28.62	130.22	32.35	86.81	28.15	114.48	46.51	45.31
Hyperbolic	28.15	202.71	30.29	_	28.70	172.65	32.94	96.08	28.48	176.59	46.74	77.00
FedA2L	29.28	**86.38**	31.47	**91.79**	30.73	107.55	34.64	**70.44**	30.41	**89.70**	47.05	**45.07**



(a) FedAvg (b) FedProx (c) FedYogi

(d) FedNTD (e) FedAWA (f) DFedSAM

Figure 2: Time to reach 45% target accuracy on CIFAR-100 using ResNet-18 (Dirichlet $\alpha = 0.1$) across six DFL algorithms.

while narrowing scheduler gaps rather than delivering the strongest speedups in every configuration. These results indicate that FedA2L's benefits increase with feature hierarchy depth and representation complexity, rather than with network depth alone.

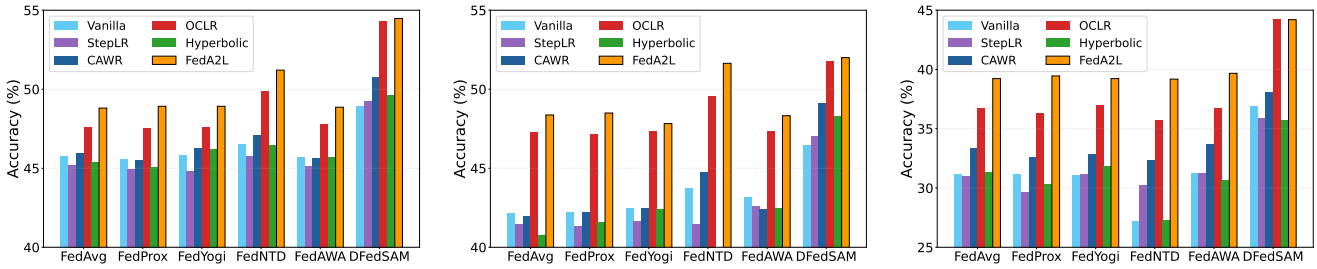
5.2.5. Comparison of convergence time

FedA2L consistently ranks within the top two in total convergence time across all six evaluated algorithms, achieving the lowest total time in five cases. As shown in Table 3, FedA2L incurs a modest increase in time per round relative to vanilla, for example 29.28s versus 28.43s under FedAvg, due to layer-wise metric computation. However, this overhead is consistently outweighed by the reduction in total training time, since fewer rounds are needed to reach the target accuracy. To reach the 45% target accuracy, FedA2L converges in 177 rounds (86.38 minutes), whereas the vanilla baseline requires 491 rounds (232.63 minutes), yielding a 2.69× reduction in total time despite the higher per-round cost.

Fig. 2 further illustrates this trend across all six DFL algorithms, showing that FedA2L consistently achieves the lowest total time, with the most pronounced gains under FedProx and FedAWA, where FedA2L reduces total training time by 25.1% (91.79 vs. 122.60 minutes) and 21.65% (89.70 vs. 114.48 minutes) relative to the best competing baseline, respectively (Table 3). These results demonstrate that FedA2L's layer-wise adaptation provides measurable wall-clock efficiency across diverse DFL protocols, supporting its practicality for time-sensitive industrial IoT, cyber-physical systems, and decentralized intelligence deployments where both communication and computational costs are critical.

5.3. How well does FedA2L perform in accuracy?

Beyond accelerating convergence, FedA2L achieves the highest test accuracy across diverse settings, maintaining both model quality and training stability under non-IID conditions. Fig. 3 compares the highest achieved accuracy as model depth and dataset complexity increase. FedA2L reaches the



(a) ResNet-18 on CIFAR-100 (b) ResNet-34 on CIFAR-100 (c) ResNet-34 on TinyImageNet

Figure 3: Highest test accuracy achieved across algorithms, models, and datasets under non-IID partitioning (Dirichlet $\alpha = 0.1$).

Table 4
Convergence performance of FedA2L vs. baselines in DFL frameworks. Rounds to target accuracies varying data heterogeneity, scalability, and network topologies. Results are smoothed using a sliding average (window size = 15). Formatting: Bold = fastest; Underline = second-fastest; Dash = failure to converge.

Setting		Heterogeneity		Scalability		Topology	
Target Accuracy		40%	45%	42%	40%	33%	45%
Algorithm	Method	*Dir*(0.01)	*Dir*(0.5)	20 nodes	30 nodes	Ring	K-Connected
FedAvg	Vanilla	300	354	–	–	–	–
	StepLR	206	109	–	–	–	–
	CAWR	194	136	398	348	–	–
	OCLR	298	132	341	366	493	296
	Hyperbolic	315	248	–	–	–	–
	FedA2L	**181**	**104**	**278**	**339**	**318**	**216**
FedAWA	Vanilla	380	406	–	–	–	–
	StepLR	209	125	–	–	–	–
	CAWR	193	140	–	451	–	299
	OCLR	316	135	352	379	497	296
	Hyperbolic	314	226	–	–	–	–
	FedA2L	**187**	**106**	**273**	**334**	**319**	**211**

highest accuracy in the majority of configurations across all six algorithms. The accuracy improvement stems from the same underlying mechanism as convergence improvement, by dynamically balancing layer-specific roles based on local divergence metrics. FedA2L avoids the trade-off inherent in uniform LRs where high rates destabilize and low rates under-adapt, maintaining both leading accuracy and stable convergence across model architectures and datasets. Overall, fine-grained per-layer control contributes to consistently higher accuracy without compromising training stability in heterogeneous decentralized environments.

5.4. How robust is FedA2L in practical DFL settings?

To validate practical applicability beyond baseline conditions, this section evaluates FedA2L's robustness under severe non-IID data distributions, increased node scalability, and challenging network topologies (Table 4). Across these demanding conditions, FedA2L consistently reduces communication rounds required to reach target accuracy, demonstrating that adaptive layer-wise LRs remain effective and stable as DFL training conditions become more challenging.

5.4.1. *Comparison under heterogeneous data distributions*

The convergence results under heterogeneous data distributions (Table 4, heterogeneity columns) underscore the robust capability of FedA2L to mitigate the severe impact of non-IID data. Under the highly skewed $\alpha = 0.01$ conditions that create extreme optimization challenges, baseline performance degrades sharply due to client drift where inconsistent local gradients pull the model away from the optimum. In contrast, FedA2L achieves the fastest convergence (181 rounds for FedAvg, 187 for FedAWA) and maintains a substantial 1.66× speedup over the vanilla baseline. This result validates that the layer-wise adaptive mechanism remains effective, enabling a stable and accelerated convergence that is crucial in practical non-IID settings.

5.4.2. *Comparison across node scalability*

The evaluation of node scalability (Table 4, scalability columns), from 20 to 30 clients, confirms that the performance benefits of FedA2L are consistent in increasingly complex DFL environments, with more distributed interactions, even as the consensus challenge grows. Larger networks

introduce more gradients to aggregate, amplifying noise and slowing consensus propagation. FedA2L sustains its performance advantage over all baselines, requiring 278 and 273 rounds under FedAvg and FedAWA, respectively, for 20 nodes, and 339 and 334 rounds for 30 nodes. These results demonstrate that FedA2L effectively mitigates this increased noise and the ensuing consensus delays inherent in larger peer-to-peer networks, allowing layer-wise adaptive LRs to remain effective without proportional degradation as network size increases.

5.4.3. Comparison across network topologies

The evaluation across network topologies (Table 4, topology columns), particularly the challenging sparse Ring topology, provides compelling evidence of FedA2L's resilience to communication bottlenecks. The Ring topology is structurally demanding due to its high diameter and severely restricted neighbor interactions, causing most baseline methods to fail to converge entirely. Despite these severe structural constraints, FedA2L successfully converges, achieving a remarkable 1.55× speedup (318 rounds for FedAvg, 319 for FedAWA) compared to the best-performing converging baseline (OCLR). This demonstrates that layer-wise adaptive LRs effectively compensate for structural communication limitations, enabling faster convergence even under severely constrained neighbor interactions.

5.5. Results on TSF

Table 5 reports communication rounds and total time to reach target MSE on both datasets under FedAvg and FedAWA. FedA2L achieves the fastest convergence across all settings. On ExchangeRate, FedA2L reaches target MSE in 11 rounds versus 16 rounds for the best scheduler (OCLR) under FedAvg, a 1.45× speedup that translates to 31.25% reduction in communication rounds and 26.37% reduction in total time. Under FedAWA, the gap widens: 11 versus 17 rounds (1.55× speedup, 35.29% fewer rounds, 31.67% less time). On BeijingAirQuality, the improvement is more modest: 18 versus 22 rounds under FedAvg (13.64% fewer rounds, 9.26% less time) and 20 versus 22 rounds under FedAWA (9.1% fewer rounds, 5.91% less time). The smaller gains indicate that highly multivariate, high-frequency data (11 variates, hourly) in BeijingAirQuality exhibits less layer-wise optimization conflict than univariate, low-frequency data (1 variate, daily) in ExchangeRate, where per-layer LR tuning has more room to improve. In all cases, FedA2L's modest per-round overhead (28-31 s versus 28-29 s for the baselines) is outweighed by the reduction in total rounds required, confirming the practical benefit of adaptive layer-wise LRs across both temporal forecasting settings.

Table 5
Communication rounds and total time to reach target MSE (0.0575 for ExchangeRate, 4.3582 for BeijingAirQuality) for FedA2L and baseline learning rate strategies.

Setting		ExchangeRate		BeijingAirQuality	
Algorithm	Method	Round	Total time (s)	Round	Total time (s)
FedAvg	Vanilla	47	1170.46	48	1195.36
	StepLR	55	1365.63	–	–
	CAWR	89	2180.64	38	1376.40
	OCLR	16	393.86	22	806.05
	Hyperbolic	88	2158.42	42	1519.98
	FedA2L	**11**	**289.98**	**18**	**731.38**
	Improv.	*31.25%*	*26.37%*	*13.64%*	*9.26%*
FedAWA	Vanilla	58	1391.18	40	1547.33
	StepLR	–	2121.43	–	–
	CAWR	98	2360.4	34	1307.71
	OCLR	17	423.39	22	852.10
	Hyperbolic	–	2379.8	38	1470.10
	FedA2L	**11**	**307.31**	**20**	**801.69**
	Improv.	*35.29%*	*31.67%*	*9.1%*	*5.91%*

5.6. Hyperparameter sensitivity analysis and ablation study

We conduct a two-part analysis to examine the design choices and configuration parameters of FedA2L. First, we analyze the dynamics and stability of the layer-wise LR feedback mechanism to verify that the adaptive behavior is driven by meaningful optimization signals rather than noise. Second, we conduct a hyperparameter sensitivity and component ablation analysis to evaluate how the design parameters and the dual-metric structure affect both convergence speed and model performance.

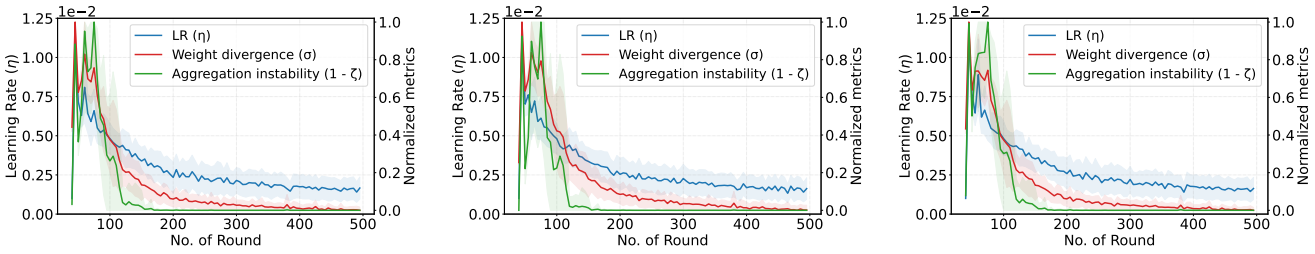
5.6.1. LR feedback dynamics and stability

To inspect the behavior of FedA2L, we analyze the joint evolution of the layer-wise LR and the local divergence metrics. Fig. 4 shows the top three layers ranked by their highest local divergence metrics in a representative configuration (ResNet-18, CIFAR-100, FedAvg at $\alpha = 0.1$). For each selected layer, we plot across rounds the effective LR $\eta^r_{i,l}$ together with the normalized weight divergence $\sigma^r_{i,l}$ and aggregation instability $1 - \zeta^r_{i,l}$ [1].

Across all three layers, we observe large spikes in both $\sigma^r_{i,l}$ and $1-\zeta^r_{i,l}$ during early rounds, indicating strong local updates and non-negligible corrections by neighbors. In response, the corresponding LRs temporarily increase but remain within a narrow range and then decay smoothly over time. As training progresses, the divergence and aggregation-instability signals rapidly diminish toward zero, and $\eta^r_{i,l}$ converges toward a stable plateau determined by the global decay factor γ^r. Importantly, even when a layer exhibits large divergence or strong correction, the LR does not grow without bound; transient boosts are followed by reductions, consistent with the bounded $1 + \tanh(\log(\lambda^r_{i,l}))$ modulation and the self-correcting Z-score normalization described in Section 3.4.3.

These trajectories show that FedA2L briefly increases LRs when local signals are strong, enabling short-term beneficial divergence in specialized layers under non-IID data, while the normalization and decay mechanisms gradually return the rates toward the global schedule and prevent runaway behavior.

[1] We use $1-\zeta^r_{i,l}$ instead of aggregation stability ($\zeta^r_{i,l}$) so that higher values consistently indicate stronger neighbor corrections, aligning the direction of both instability signals. All metric curves are linearly rescaled to $[0, 1]$ for visualization.



(a) Highest divergence layer (b) Second highest divergence layer (c) Third highest divergence layer

Figure 4: LR feedback dynamics for the three layers with the highest local divergence (ResNet-18, CIFAR-100, FedAvg, Dirichlet $\alpha = 0.1$). The plots show the layer-wise learning rate $\eta^r_{i,l}$ (left axis), together with the normalized weight divergence $\sigma^r_{i,l}$ and aggregation instability $1 - \zeta^r_{i,l}$ (right axis), over communication rounds.

Table 6
Hyperparameter sensitivity and ablation analysis of FedA2L on CIFAR-100 ($\alpha = 0.1$). Results are reported in terms of communication rounds to reach the target accuracy and best test accuracy (%).

Model	Metric	τ			R_{warm}			ρ			ξ			β		
		10^{-2}	10^{-3}	10^{-4}	10	20	40	5	10	20	0.05	0.1	0.3	0.0	0.6	1.0
CNN	Rounds@29%	22	**21**	23	**24**	38	29	**23**	28	25	30	29	**24**	23	**21**	23
	Best accuracy	**33.39**	**33.39**	33.31	**33.11**	32.59	31.14	**33.29**	32.97	32.88	32.33	32.87	**33.33**	32.58	**33.63**	32.46
ResNet-18	Rounds@45%	173	**171**	192	462	207	**175**	197	**176**	179	179	**177**	_	176	**175**	180
	Best accuracy	48.46	**48.81**	48.53	45.46	47.78	**48.63**	46.94	**48.78**	48.69	48.73	**48.86**	44.90	49.23	**49.50**	49.10
ResNet-34	Rounds@45%	148	**146**	153	249	167	**156**	160	**156**	**156**	168	**157**	_	158	**150**	159
	Best accuracy	**48.67**	**48.67**	48.61	46.52	48.63	**48.87**	48.27	48.78	**48.86**	48.73	**48.86**	44.29	47.22	**48.63**	47.22

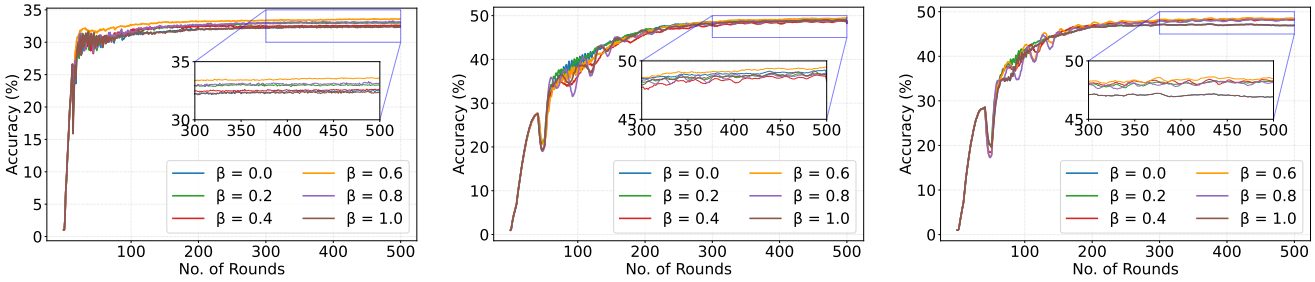
5.6.2. Hyperparameter sensitivity and configuration

To assess robustness and reproducibility, we conduct a sensitivity study on CIFAR-100 across CNN (29% target), ResNet-18 and ResNet-34 (45% target), varying each of the five FedA2L design parameters (R_{warm}, ρ, ξ, τ, and β) independently while holding the others at their defaults. Table 6 reports both convergence efficiency and best test accuracy for each parameter, providing a joint view of how each parameter affects both dimensions of performance. Fig. 5 presents the full accuracy trajectories under varying β, where the extreme settings $\beta = 0$ and $\beta = 1$ serve as direct ablations of the individual ζ and σ components, respectively, enabling assessment of their contribution to both convergence behavior and final model accuracy.

Influence of stability threshold (τ). The stability threshold τ in Eq. 4 sets the tolerance boundary for counting a parameter as stable during aggregation, directly controlling the resolution of the consensus signal ζ. Too small a value makes ζ uninformative by classifying nearly all parameter changes as unstable; too large a value loses discrimination between genuine consensus and client drift. As shown in Table 6, CNN is negligibly affected across all tested values, while $\tau = 10^{-4}$ degrades convergence on ResNet-18 and ResNet-34 from 171 to 192 rounds and from 146 to 153 rounds respectively, confirming that ζ loses its discriminative power at excessively fine thresholds. A stable operating region exists between 10^{-2} and 10^{-3}, and $\tau = 10^{-3}$ achieves competitive accuracy across all architectures, confirming that convergence efficiency and model quality are jointly optimized. We adopt $\tau = 10^{-3}$ across all model architectures.

Influence of warm-up rounds (R_{warm}). The warm-up period R_{warm} defines the number of initial rounds during which FedA2L applies the base LR uniformly across all layers before adaptive modulation begins. Its sensitivity scales with model depth, where shallow architectures accumulate reliable divergence statistics within a few rounds, whereas deeper models require a longer history for per-layer signals to stabilize. As shown in Table 6, reducing R_{warm} to 10 rounds degrades convergence from 175 to 462 rounds on ResNet-18 and from 156 to 249 rounds on ResNet-34, while all CNN settings yield comparable results. Critically, very short warm-up periods also reduce best accuracy, as insufficient metric history produces unreliable early LR modulation that degrades the full training trajectory. We adopt $R_{\text{warm}} = 10$ for CNN and $R_{\text{warm}} = 40$ for ResNet architectures.

Influence of temporal window size (ρ). The window size ρ controls the number of recent rounds used in Z-score normalization, directly governing the temporal horizon of the adaptive signal. FedA2L is largely insensitive to this parameter because Z-score normalization inherently reduces short-term fluctuations regardless of window length, making the adaptive signal robust to the exact history size. Table 6 confirms this robustness: convergence varies within a narrow range across $\rho \in \{5, 10, 20\}$ on all architectures, ranging from 23 to 25 rounds on CNN and from 176 to 197 rounds on ResNet-18, without a consistent directional trend, and best accuracy follows the same pattern across all architectures. ResNet architectures show marginally better convergence with $\rho = 10$, consistent with their greater need for stable statistical history. We adopt $\rho = 5$ for CNN and $\rho = 10$ for



(a) CNN (b) ResNet-18 (c) ResNet-34

Figure 5: Effect of fusion balance β on convergence speed and model performance under FedAvg on CIFAR-100. The extreme values $\beta = 0$ (ζ-only) and $\beta = 1$ (σ-only) constitute direct component ablations of ζ and σ, respectively, in Eq. 7, while intermediate values reflect the full dual-metric fusion.

ResNet architectures as minimal window sizes that ensure sufficient statistical stability.

Influence of global decay constant (ξ). The global decay factor $(1 + \xi \cdot r)^{-0.5}$ in Eq. 8 controls how aggressively the base LR decreases across rounds, setting the pace at which global optimization slows relative to layer-wise adaptation. The sensitivity of ξ is architecture-dependent: deeper models require a gentler decay to allow layer-wise signals to fully differentiate per-layer rates during the critical mid-training phase, while shallower models converge faster and tolerate earlier decay. As shown in Table 6, $\xi = 0.3$ leads to convergence failure on ResNet-18 and ResNet-34, with best accuracy collapsing to 44.90% and 44.29% respectively, confirming that aggressive decay suppresses layer-wise adaptation before the per-layer signals become effective. In contrast, reducing ξ to 0.1 restores stable convergence at 177 and 157 rounds on ResNet-18 and ResNet-34. Meanwhile, CNN achieves its lowest convergence in 24 rounds under the same aggressive setting, as its simpler feature hierarchy reaches the target earlier under faster decay. We adopt $\xi = 0.1$ for ResNet architectures and $\xi = 0.3$ for CNN.

Influence of fusion balance (β). The fusion balance β in Eq. 7 is a key operational parameter, as it controls how FedA2L balances local update intensity σ and network consensus constraints ζ when constructing the per-layer LR signal. The extreme cases $\beta = 0$, which relies solely on network consensus constraints ζ, and $\beta = 1$, which relies solely on local update intensity σ, serve as direct ablations of each individual component. Both single-signal configurations exhibit slower convergence and lower best accuracy than any balanced setting, as evidenced by Table 6 and the accuracy trajectories in Fig. 5. The performance degradation observed in both settings indicates that neither signal alone is sufficient for effective layer-wise adaptation. For ResNet-18 on CIFAR-100, the best test accuracy drops from 49.50% at $\beta = 0.6$ to 49.23% at $\beta = 0.0$ and 49.10% at $\beta = 1.0$, demonstrating that σ and ζ provide complementary and non-redundant information whose combination is necessary for full performance. For deeper architectures such as ResNet-34, performance degradation at $\beta = 1.0$ is more pronounced than at $\beta = 0.0$, with the required rounds increasing from 150 to 159 at $\beta = 1.0$, compared to 158 at $\beta = 0.0$.

Table 7
FedA2L default configuration.

Parameter	Tested range	CNN	ResNet-18 ResNet-34
R_{warm}	$\{10, 20, 40\}$	10	40
ρ	$\{5, 10, 20\}$	5	10
τ	$\{1e-2, 1e-3, 1e-4\}$	1e-3	1e-3
β	$\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$	0.6	0.6
ξ	$\{0.05, 0.1, 0.3\}$	0.3	$0.1^{\dagger}$

† $\xi = 0.05$ is applied for TinyImageNet due to higher class diversity.

This suggests that network consensus constraints offer more informative layer-wise signals in complex feature hierarchies. FedA2L remains stable across $\beta \in [0.4, 0.8]$, with $\beta = 0.6$ consistently achieving the strongest convergence speed and accuracy across all evaluated architectures. We therefore adopt $\beta = 0.6$ as the default, reflecting a moderate emphasis on local update intensity while retaining sufficient weight on network consensus constraints.

Overall robustness and recommended configuration. The default configuration is summarized in Table 7. Among the five parameters, ρ and τ show limited sensitivity across all tested architectures, while β remains stable within $[0.4, 0.8]$, confirming that the dual-signal fusion is robust to reasonable variation in weighting. The primary configuration parameters are R_{warm} and ξ, both of which interact directly with model depth and should be adjusted when deploying on architectures outside the configurations evaluated here.

6. Discussion

FedA2L dynamically adjusts LRs at the layer level using only locally available model statistics, integrating seamlessly into any DFL protocol without modifying the core aggregation logic or introducing additional communication overhead. This distinguishes it from server-dependent adaptive methods such as FedYogi, layer-wise adaptive approaches that rely on centralized gradient statistics, and divergence-based approaches [52, 53, 54] that utilize deviation signals for representation alignment or aggregation reweighting. Such a mechanism is particularly relevant under non-IID data

and dynamic network conditions where consistent global statistics are difficult to obtain. The layer-wise LR signals in FedA2L may also be extended to personalized DFL, where client-specific adaptation is incorporated into decentralized optimization.

While FedA2L offers flexibility, it involves several practical considerations. First, some hyperparameters (e.g., R_{warm} and ξ) depend on model depth and may require additional tuning across different architectures. Second, the performance gains are less pronounced when the base DFL algorithm already incorporates strong drift correction or adaptive regularization (e.g., FedNTD and FedYogi), where scheduler-based baselines remain competitive. Overall, despite these considerations, FedA2L provides a robust and effective foundation for layer-wise adaptation in decentralized learning.

7. Conclusion

This paper presents FedA2L, a layer-wise adaptive LR method designed specifically for DFL. By exploiting dual state-transition signals, FedA2L enables per-layer LR modulation that captures layer-specific heterogeneity while stabilizing network consensus, without requiring modification to the underlying DFL protocol. Extensive experiments demonstrate that FedA2L consistently accelerates convergence and improves model accuracy across diverse models, datasets, and DFL algorithms, without additional communication overhead. The reduction in communication rounds further improves communication efficiency, which is beneficial in bandwidth-constrained and latency-sensitive environments. These results indicate that layer-wise adaptation provides an effective mechanism for addressing heterogeneity in decentralized learning, establishing FedA2L as a practical and scalable optimization approach for DFL systems.

Funding

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2022R1I1A3072355).

References

[1] Wen Sun, Shiyu Lei, Lu Wang, Zhiqiang Liu, and Yan Zhang. Adaptive federated learning and digital twin for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 17(8):5605–5614, 2021. doi: 10.1109/TII.2020.3034674.

[2] V Padmavathi, R Kanimozhi, and R Saminathan. Digital twin driven smart factories: real time physics based co-simulation using edge ai and federated learning. *Scientific Reports*, 15(1):43373, 2025.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[4] Tian Wang, Yan Liu, Xi Zheng, Hong-Ning Dai, Weijia Jia, and Mande Xie. Edge-based communication optimization for distributed federated learning. *IEEE Transactions on Network Science and Engineering*, 9 (4):2015–2024, 2021.

[5] Shuai Wang, Youliang Tian, Jinbo Xiong, Jianfeng Ma, and Yan Zhang. VerifyDFL: Secure aggregation for decentralized federated learning with input validation in mobile edge intelligence. *IEEE Transactions on Cognitive Communications and Networking*, 12:2526–2541, 2026. doi: 10.1109/TCCN.2025.3587771.

[6] Liangqi Yuan, Ziran Wang, Lichao Sun, Philip S. Yu, and Christopher G. Brinton. Decentralized federated learning: A survey and perspective. *IEEE Internet of Things Journal*, 11(21):34617–34638, 2024. doi: 10.1109/JIOT.2024.3407584.

[7] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. Fully decentralized federated learning. In *Third workshop on bayesian deep learning (NeurIPS)*, volume 12, 2018.

[8] Peng Wang, Wen Sun, Haibin Zhang, Wenqiang Ma, and Yan Zhang. Distributed and secure federated learning for wireless computing power networks. *IEEE Transactions on Vehicular Technology*, 72(7):9381–9393, 2023.

[9] Zheyi Chen, Qingnan Jiang, Lixian Chen, Xing Chen, Jie Li, and Geyong Min. MC-2PF: A multi-edge cooperative universal framework for load prediction with personalized federated deep learning. *IEEE Transactions on Mobile Computing*, 24(6):5138–5154, 2025. doi: 10.1109/TMC.2025.3528404.

[10] Zheyi Chen, Jie Liang, Zhengxin Yu, Hongju Cheng, Geyong Min, and Jie Li. Resilient collaborative caching for multi-edge systems with robust federated deep learning. *IEEE Transactions on Networking*, 33 (2):654–669, 2024.

[11] Jianchun Liu, Jiaming Yan, Hongli Xu, Lun Wang, Zhiyuan Wang, Jinyang Huang, and Chunming Qiao. Accelerating decentralized federated learning with probabilistic communication in heterogeneous edge computing. *IEEE Transactions on Networking*, 34:486–501, 2026. doi: 10.1109/TON.2025.3600015.

[12] Mbasa Joaquim Molo, Lucia Vadicamo, Claudio Gennaro, and Emanuele Carlini. Decentralized edge learning: A comparative study of distillation strategies and dissimilarity measures. *Future Generation Computer Systems*, 176:108171, 2026. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2025.108171.

[13] Zheyi Chen, Junjie Zhang, Geyong Min, Zhaolong Ning, and Jie Li. Traffic-Aware Lightweight Hierarchical Offloading Toward Adaptive Slicing-Enabled SAGIN. *IEEE Journal on Selected Areas in Communications*, 42(12):3536–3550, 2024. doi: 10.1109/JSAC.2024.3459020.

[14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[15] Radwan Selo, Majid Kundroo, and Taehong Kim. FedTVD: Balancing data quality and quantity for robust federated learning. *Future Generation Computer Systems*, page 108177, 2025.

[16] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. FedDC: Federated learning with non-IID data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

[17] Junyoung Park, Sungpil Woo, and Joohyung Lee. Def-Ag: An energy-efficient decentralized federated learning framework via aggregator clients. *Future Generation Computer Systems*, 175:108114, 2026. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2025.108114.

[18] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[19] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[20] Ahmed Elhussein and Gamze Gürsoy. PLayer-FL: A principled approach to personalized layer-wise cross-silo federated learning, 2025. URL https://arxiv.org/abs/2502.08829.

[21] Weihang Chen, Cheng Yang, Jie Ren, Zhiqiang Li, and Zheng Wang. Optimizing personalized federated learning through adaptive layer-wise learning. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 4860–4868. International Joint Conferences on Artificial

Intelligence Organization, August 2025. doi: 10.24963/ijcai.2025/541. Main Track.

[22] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2021.

[23] Bingnan Xiao, Jingjing Zhang, Wei Ni, and Xin Wang. FLARE: A new federated learning framework with adjustable learning rates over resource-constrained wireless networks. *IEEE Transactions on Wireless Communications*, 2025.

[24] Anam Nawaz Khan, Qazi Waqas Khan, Atif Rizwan, Rashid Ahmad, and Do Hyeun Kim. Consensus-driven hyperparameter optimization for accelerated model convergence in decentralized federated learning. *Internet of Things*, 30:101476, 2025. ISSN 2542-6605. doi: https://doi.org/10.1016/j.iot.2024.101476.

[25] Youngmin Ro and Jin Young Choi. AutoLR: Layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2486–2494, 2021.

[26] Belhal Karimi, Ping Li, and Xiaoyun Li. Fed-LAMB: layer-wise and dimension-wise locally adaptive federated learning. In *Uncertainty in Artificial Intelligence*, pages 1037–1046. PMLR, 2023.

[27] Changlong Shi, Jinmeng Li, He Zhao, Dandan Guo, and Yi Chang. FedLWS: Federated learning with adaptive layer-wise weight shrinking. *arXiv preprint arXiv:2503.15111*, 2025.

[28] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017.

[29] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013, 2023.

[30] Lingling Wang, Xueqin Zhao, Zhongkai Lu, Lin Wang, and Shouxun Zhang. Enhancing privacy preservation and trustworthiness for decentralized federated learning. *Information Sciences*, 628:449–468, 2023. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.01.130.

[31] Yuhao Zhou, Minjia Shi, Yuxin Tian, Qing Ye, and Jiancheng Lv. DeFTA: A plug-and-play peer-to-peer decentralized federated learning framework. *Information Sciences*, 670:120582, 2024. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2024.120582.

[32] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[33] Vo van Truong, Pham Khanh Quan, Dong-Hwan Park, and Taehong Kim. Performance evaluation of decentralized federated learning: Impact of fully and k-connected topologies, heterogeneous computing resources, and communication bandwidth. *IEEE Access*, 13:32741–32755, 2025. doi: 10.1109/ACCESS.2025.3542772.

[34] Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487, 2019.

[35] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. GossipFL: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE Transactions on Parallel and Distributed Systems*, 34(3):909–922, 2022.

[36] Changlong Shi, He Zhao, Bingjie Zhang, Mingyuan Zhou, Dandan Guo, and Yi Chang. FedAWA: Adaptive optimization of aggregation weights in federated learning using client vectors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30651–30660, 2025.

[37] Yitong Tang. Adapted weighted aggregation in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23763–23765, 2024.

[38] Majid Kundroo and Taehong Kim. Federated learning with hyperparameter optimization. *Journal of King Saud University-Computer and Information Sciences*, 35(9):101740, 2023.

[39] Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendis, Xue Jiang, and Xuebing Zhou. Federated hyperparameter optimization through reward-based strategies: Challenges and insights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4236–4244, 2024.

[40] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-IID data. *arXiv preprint arXiv:1806.00582*, 2018.

[41] Shu Zheng, Tiandi Ye, Xiang Li, and Ming Gao. Federated learning via consensus mechanism on heterogeneous data: A new perspective on convergence. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7595–7599. IEEE, 2024.

[42] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[43] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[44] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[45] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205), 2017.

[46] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[47] Gihun Lee, Yongjin Shin, Minchan Jeong, and Se-Young Yun. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. *CoRR*, abs/2106.03097, 2021.

[48] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31269–31291. PMLR, 23–29 Jul 2023.

[49] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts, 2017. URL https://arxiv.org/abs/1608.03983.

[50] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018. URL https://arxiv.org/abs/1708.07120.

[51] Tae-Geun Kim. HyperbolicLR: Epoch insensitive learning rate scheduler, 2025. URL https://arxiv.org/abs/2407.15200.

[52] Kai Hu, Yaogen Li, Shuai Zhang, Jiasheng Wu, Sheng Gong, Shanshan Jiang, and Liguo Weng. FedMMD: A Federated weighting algorithm considering Non-IID and Local Model Deviation. *Expert Systems with Applications*, 237:121463, 2024.

[53] Jaewon Jang and Bong Jun Choi. Personalized federated learning via deviation tracking representation learning. In *2024 International Conference on Information Networking (ICOIN)*, pages 762–766. IEEE, 2024.

[54] Wenjie Yao, Guanglu Sun, Suxia Zhu, Ruidong Wang, Xinzhong Zhu, HuiYing Xu, and Xiguang Wei. FedRDA: Representation Deviation Alignment in Heterogeneous Federated Learning. *IEEE Transactions on Industrial Informatics*, 2025.