LEVERAGING LOW-RANK HESSIAN STRUCTURES FOR COMMUNICATION-EFFICIENT FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Communication dominates the cost of federated learning (FL), and second-order curvature can reduce the number of rounds needed to reach a target accuracy. Existing federated second-order methods, however, largely face a dilemma: approaches that operate on curvature at full-Hessian scale incur memory and computation costs quadratic in the parameter count and are therefore restricted to shallow models, whereas lightweight diagonal approximations remain inexpensive but discard cross-coordinate curvature. To address this dilemma, we propose FHaN, a federated optimization method that builds second-order updates from a few locally sampled Hessian columns. Each client selects the columns in a gradient-guided manner and builds from them a compact low-rank correction capturing part of the cross-coordinate curvature. Because the solve is confined to the sampled subspace, neither the full Hessian nor its inverse is ever formed; the only inversion required is that of a small matrix, stabilized by smoothing its singular values. The result is a model-sized update vector per client, so the per-round payload matches that of first-order methods, and the efficiency gain comes from reaching target accuracy in fewer rounds. We show that FHaN converges linearly whenever the aggregated update satisfies a contraction condition, and identify a regime of data heterogeneity and sampling error in which this condition is feasible. On MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100, FHaN's advantage concentrates where curvature matters: federated fine-tuning of frozen pretrained backbones gains up to +13.2 final accuracy points with up to $6\times$ fewer rounds to target, and FC-dominated models trained from scratch gain up to +23.9 points, while saturated tasks, benign convex problems, and end-to-end convolutional training mark its boundaries.

1 INTRODUCTION

In federated learning (FL) (McMahan et al., 2017), the repeated client–server exchange dominates the training cost under realistic bandwidth constraints (Nariman & Hamarashid, 2025; Liu et al., 2024a;c). First-order remedies shrink what is sent—compressed or filtered updates (Zhao et al., 2023; Herzog et al., 2024; Chen et al., 2021; Di et al., 2024)—at the price of limiting the information aggregated each round.

Second-order optimization offers a complementary route: by scaling the gradient with curvature, classical Newton-type methods (Cauchy, 1821; Fletcher & Powell, 1963) converge in far fewer iterations—exactly what a communication-bound setting rewards. Whether the advantage carries over to FL is decided by a trade-off: the client must acquire, store, and solve with the curvature, raising the local cost of every round, and some designs additionally exchange information beyond the model vector. Fewer rounds pay only when the gain in update quality outweighs the cost of the curvature that produces it.

Existing federated Newton-type methods (Elgabli et al., 2022; Dal Fabbro et al., 2024) largely occupy one end of this trade-off: through Hessian compression (Safaryan et al., 2022; Agafonov et al., 2025), Newton sketching (Li et al., 2024; Gupta et al., 2024), or ADMM splitting, they reduce what is communicated but still form full-Hessian-scale quantities locally, which restricts them to relatively few parameters. Diagonal approximations occupy the other end: lightweight, but expressing no cross-coordinate curvature. A few Hessian columns, each encoding the second-order relation between one coordinate and the others, occupy a practical middle ground. *This raises the question studied*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

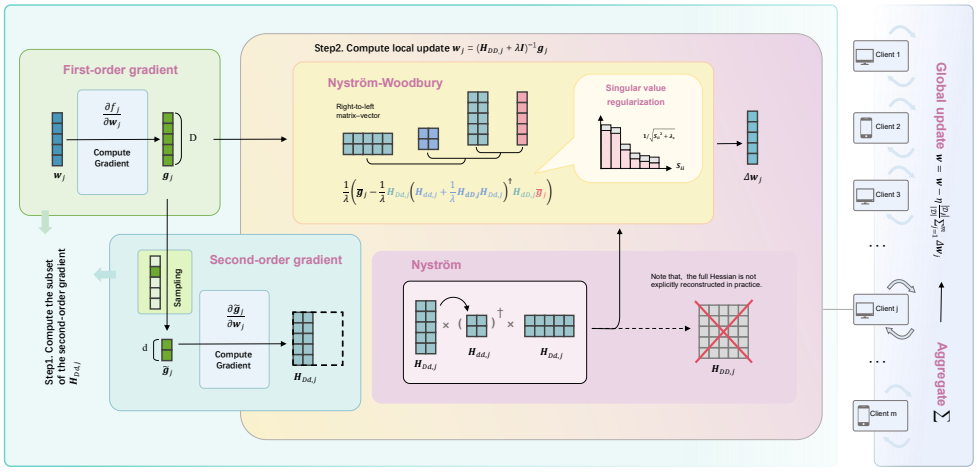


Figure 1: Illustration of FHaN: each client samples a few Hessian columns, builds a low-rank correction with stabilized inversion through matrix-vector products, and the server aggregates model-sized update vectors; the full Hessian is never reconstructed.

in this paper: how can we construct effective second-order updates from only a few local Hessian columns, while keeping communication at the model-vector level, so as to reduce the communication cost of reaching a target accuracy?

In this work, we propose FHaN (Federated Hessian-column approximate Newton's Method), illustrated in Figure 1. FHaN selects a few local Hessian columns in a gradient-guided manner and constructs from them a low-rank correction to the regularized Newton step, at a sampling budget controlled by the user, with each sampled column costing roughly one additional backward pass. The solve is carried out within the small matrix formed by the sampled columns—neither the full Hessian nor its inverse is ever formed—and the inverse singular values of that small matrix are smoothed for numerical stability. Clients upload only update vectors, so the per-round payload matches FedAvg and the efficiency gain comes from reaching target accuracy in fewer rounds; our experiments show that, under the tested conditions, limited local curvature information yields valuable communication savings. We summarize our main contributions as follows:

- We propose FHaN, which obtains second-order updates at first-order per-round payload by solving the regularized Newton system in the span of $d \ll D$ gradient-guided local Hessian columns, with stabilized inversion of a $d \times d$ core matrix.
- We provide a convergence analysis of FHaN in the convex setting: under explicit bounds on heterogeneity and sampling error, the aggregated update satisfies a contraction condition under which the iterates converge linearly from a suitable initialization.
- Experiments on MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100 show that FHaN's advantage concentrates where curvature matters: federated fine-tuning of frozen pretrained backbones (ViT-B/16, ResNet-50, ResNet-18) gains +4.1 to +13.2 final accuracy points with up to $6\times$ fewer rounds to target, and FC-dominated models trained from scratch gain up to +23.9 points; the same suite maps the method's boundaries—saturated tasks, benign convex problems, end-to-end convolutional training, and extreme heterogeneity—and a per-scenario study of λ exposes the stability–performance trade-off.

2 Related Work

Our work applies second-order optimization to FL; for a recent survey of second-order methods in FL, see Sen et al. (2025).

Federated first-order optimizers. FedAvg (McMahan et al., 2017) trains with local SGD; adaptive variants (Reddi et al., 2021; Duchi et al., 2011; Kingma, 2015; Zaheer et al., 2018) and primal-dual designs (Gong et al., 2022) improve per-round progress, while compression (Chen et al., 2021; Chu

et al., 2025), client selection (Di et al., 2024; Ribero & Vikalo, 2024; Pan et al., 2025), and local-step amplification (Herzog et al., 2024) reduce communication at the price of aggregation quality.

Centralized second-order optimizers. Classical accelerations include BFGS, L-BFGS (Broyden, 1965; Liu & Nocedal, 1989), Gauss-Newton (Schraudolph, 2002), and inexact Newton (Dembo et al., 1982); recent scalable variants approximate the Hessian diagonally or block-diagonally, e.g., ADAHESSIAN (Yao et al., 2021), Sophia (Liu et al., 2024b), K-FAC (Martens & Grosse, 2015; Botev et al., 2017), and HesScale (Elsayed et al., 2024). These methods approximate the Hessian with compact, informative substitutes.

Federated second-order optimizers. DANE (Shamir et al., 2014) and GIANT (Wang et al., 2018) use conjugate gradient to approximate Newton updates, while DONE (Dinh et al., 2022) applies Richardson iteration, but they communicate twice per iteration; LocalNewton (Gupta et al., 2021) likewise solves local subproblems with Hessian-free iterations. FedNL and its follow-ups (Safaryan et al., 2022; Burlachenko & Richtárik, 2024), SHED (Dal Fabbro et al., 2024) and FedNew (Elgabli et al., 2022) use compressed Hessians, eigenvector-eigenvalue pairs and ADMM, yet they still form the full Hessian during training. FedNS (Li et al., 2024), FLeNS (Gupta et al., 2024) adopt sketched Hessians, but limited compression ratios constrain their applicability. Recently, Fed-Sophia (Elbakary et al., 2024) is derived from centralized Sophia (Liu et al., 2024b), FAGH (Sen et al., 2024) samples the Hessian's first row, and FedPM (Ishii et al., 2026) mixes locally preconditioned updates on the server, while server-side quasi-Newton designs such as FedSSO (Ma et al., 2022) maintain curvature approximations at the server. Closest to our method, FONN (Nagaraju et al., 2023) also builds local Newton-type updates from sampled Hessian columns, but through a Nyström solve on an aggregated global gradient communicated back to the clients; Nys-FL (Sen et al., 2023) and Nys-Fed (Ghosh et al., 2026) likewise construct federated Newton directions from randomly sketched curvature. FHaN instead couples gradient-guided sampling with smoothed inversion and direct aggregation of update vectors, keeps communication strictly at the model-vector level, and establishes linear convergence under explicit conditions on sampling error and heterogeneity.

3 PRELIMINARIES

Federated learning. We consider a standard FL setting with a global server and m local devices, where device $j \in [m]$ holds local data $\mathcal{D}_j$ and the datasets form the disjoint union $\mathcal{D} = \cup_{j=1}^{m} \mathcal{D}_j$. The FL objective is

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) = \sum_{j=1}^{m} \frac{|\mathcal{D}_j|}{|\mathcal{D}|} f_j(\boldsymbol{w}), \tag{1}$$

where $f_j(\boldsymbol{w})$ is the j-th device's loss function and $\boldsymbol{w} \in \mathbb{R}^D$ denotes the model parameters. Gradient descent updates $\boldsymbol{w}^t = \boldsymbol{w}^{t-1} - \eta \boldsymbol{g}^{t-1}$ with the gradient $\boldsymbol{g}^t := \nabla f(\boldsymbol{w}^t)$; Newton's method additionally scales the gradient by the inverse Hessian,

$$\boldsymbol{w}^t = \boldsymbol{w}^{t-1} - \eta (\boldsymbol{H}^{t-1})^{-1} \boldsymbol{g}^{t-1}, \tag{2}$$

where $\boldsymbol{H}^t := \nabla^2 f(\boldsymbol{w}^t) \in \mathbb{R}^{D \times D}$, and $\eta = 1$ recovers standard Newton's method. In a federated Newton step both quantities aggregate from the clients as sample-weighted sums, $\boldsymbol{H}^t = \sum_j \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \nabla^2 f_j(\boldsymbol{w}^t)$ and $\boldsymbol{g}^t = \sum_j \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \nabla f_j(\boldsymbol{w}^t)$.

4 METHODOLOGY

In Algorithm 1, we instantiate the update design outlined above. The construction revolves around a single regularized Newton-type operator: we first build its low-rank representation from sampled Hessian columns, then solve the regularized update within the sampled subspace without forming any $D \times D$ matrix, and finally stabilize the inversion at the point where it is performed. Communication remains at the level of first-order methods, while computational and memory costs scale with the sampling budget rather than with D^2.

Algorithm 1 FHaN

Input: Local training data subset $\mathcal{D}_j,\ \forall j \in [m]$, local loss function $F_j(\boldsymbol{w})$, number of communication rounds T, the number of local examples $|\mathcal{D}_j|$ and total examples $|\mathcal{D}|$.
Parameter: Learning rate η, regularization parameter λ, smoothing parameter λ_s.
Output: The global weight $\boldsymbol{w}$

for each round $t = 1, \dots, T$ **do**
  Communicate $\boldsymbol{w} = (x_1, \dots, x_D)^\top$ to all clients.
  for each client $j \in [m]$ **do**
    $\boldsymbol{w}_j \leftarrow \boldsymbol{w}$.
    for each step (batches in each epoch) **do**
      Compute $\boldsymbol{g}_j \leftarrow \frac{\partial f_j}{\partial \boldsymbol{w}_j}$, $\bar{\boldsymbol{g}}_j \leftarrow \boldsymbol{g}_j + \lambda \boldsymbol{w}_j$.
      Sample gradient subset $\widetilde{\boldsymbol{g}}_j \subseteq \boldsymbol{g}_j$ via gradient-guided sampling in (5)
      Compute $\boldsymbol{H}_{Dd,j} \leftarrow \frac{\partial \bar{\boldsymbol{g}}_j}{\partial \boldsymbol{w}_j}, \boldsymbol{H}_{dd,j} \subseteq \boldsymbol{H}_{Dd,j}$ and $\widetilde{\boldsymbol{H}}_{dd,j} = \boldsymbol{H}_{dd,j} + \frac{1}{\lambda}\boldsymbol{H}_{dD,j}\boldsymbol{H}_{Dd,j}$
      Compute the regularized inverse $(\widetilde{\boldsymbol{H}}_{dd,j})^\dagger = \boldsymbol{V}\widetilde{\boldsymbol{S}}^+\boldsymbol{U}^\top$ from the SVD $\widetilde{\boldsymbol{H}}_{dd,j} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ via (8), with $\widetilde{S}^+_{ii} = 1/\sqrt{S^2_{ii} + \lambda_s}$.
      Compute the final local update step $\Delta \boldsymbol{w}_j = \frac{1}{\lambda}\left(\bar{\boldsymbol{g}}_j - \frac{1}{\lambda}\left(\boldsymbol{H}_{Dd,j}\left(\left(\widetilde{\boldsymbol{H}}_{dd,j}\right)^\dagger\left(\boldsymbol{H}_{dD,j}\bar{\boldsymbol{g}}_j\right)\right)\right)\right)$ according to (7) with matrix-vector multiplications.
    end for
    Communicate the local update $\Delta \boldsymbol{w}_j$ to server.
  end for
  On the global server:
  Update global model $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \sum_{j=1}^m \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\Delta \boldsymbol{w}_j$.
end for

4.1 Problem Formulation

We add ℓ_2 regularization with parameter λ, so each client minimizes $F_j(\boldsymbol{w}) = f_j(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$ and the global objective is $F = \sum_j \frac{|\mathcal{D}_j|}{|\mathcal{D}|}F_j$. The Newton solution is expressed as:

$$\begin{aligned} \boldsymbol{w}^t &= \boldsymbol{w}^{t-1} - \eta \sum_{j=1}^m \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\Delta \boldsymbol{w}_j^{t-1}, \\ \Delta \boldsymbol{w}_j^{t-1} &= (\boldsymbol{H}_j^{t-1} + \lambda \boldsymbol{I})^{-1}\bar{\boldsymbol{g}}_j^{t-1}, \end{aligned} \tag{3}$$

where $\bar{\boldsymbol{g}}_j^t := \boldsymbol{g}_j^t + \lambda \boldsymbol{w}^t$, and other notation follows (2).

4.2 Hessian Matrix with Nyström Approximation

The Nyström method (Williams & Seeger, 2000) approximates a matrix from a subset of its columns, reducing Hessian acquisition to the sampled columns alone. On a device with parameters $\boldsymbol{w} = (x_1, \dots x_D)^\top \in \mathbb{R}^D$ and gradient $\boldsymbol{g} = (g_1, \dots, g_D)^\top$ ($g_i = \frac{\partial f}{\partial x_i}$), we sample $\widetilde{\boldsymbol{g}} = (\widetilde{g}_1, \dots, \widetilde{g}_d)^\top \subseteq \boldsymbol{g}$ with $d \ll D$ and form the sampled Hessian columns $\boldsymbol{H}_{Dd} := (\frac{\partial \widetilde{\boldsymbol{g}}}{\partial x_1}, \dots, \frac{\partial \widetilde{\boldsymbol{g}}}{\partial x_D})^\top \in \mathbb{R}^{D\times d}$. The Nyström approximation of the update (3) is

$$\begin{aligned} \boldsymbol{H}_{DD} &\approx \boldsymbol{H}_{Dd}(\boldsymbol{H}_{dd})^\dagger \boldsymbol{H}_{Dd}^\top, \\ (\boldsymbol{H}_{DD} + \lambda \boldsymbol{I})^{-1}\bar{\boldsymbol{g}} &\approx (\boldsymbol{H}_{Dd}(\boldsymbol{H}_{dd})^\dagger \boldsymbol{H}_{Dd}^\top + \lambda \boldsymbol{I})^{-1}\bar{\boldsymbol{g}}, \end{aligned} \tag{4}$$

where $\boldsymbol{H}_{DD} := (\frac{\partial \boldsymbol{g}}{\partial x_1}, \dots, \frac{\partial \boldsymbol{g}}{\partial x_D})^\top \in \mathbb{R}^{D\times D}$, $\boldsymbol{H}_{dd} \in \mathbb{R}^{d\times d} \subseteq \boldsymbol{H}_{Dd}$, $\boldsymbol{H}^\dagger$ is the Moore-Penrose inverse of the matrix $\boldsymbol{H}$.

The accuracy of a rank-d Nyström approximation is governed by which columns are seen: uniform draws oversample coordinates that contribute little to the update and can miss the few that dominate

Table 1: Communication complexity and memory overhead of related methods. DONE and DANE communicate twice per round; κ denotes the condition number and $\delta_\kappa < 1$ the DONE contraction factor.

METHOD	MEMORY COST	ITERATIONS T	PER-ROUND PAYLOAD	COMMUNICATION COMPLEXITY
FEDAVG	$\mathcal{O}(D)$	$\mathcal{O}(\frac{1}{\varepsilon})$	$\mathcal{O}(D)$	$\mathcal{O}(\frac{D}{\varepsilon})$
FEDPROX	$\mathcal{O}(D)$	$\mathcal{O}(\frac{1}{\varepsilon})$	$\mathcal{O}(D)$	$\mathcal{O}(\frac{D}{\varepsilon})$
DANE	$\mathcal{O}(D)$	$\mathcal{O}(\frac{\kappa^2}{\lvert\mathcal{D}_j\rvert}\log(Dm)\log\frac{1}{\varepsilon})$	$\mathcal{O}(D)$	$\mathcal{O}(D\frac{\kappa^2}{\lvert\mathcal{D}_j\rvert}\log(Dm)\log\frac{1}{\varepsilon})$
DONE	$\mathcal{O}(D)$	$\mathcal{O}(\delta_\kappa\log\frac{1}{\varepsilon})$ $\mathcal{O}(\log\log\frac{1}{\varepsilon})$	$\mathcal{O}(D)$ $\mathcal{O}(D)$	$\mathcal{O}(D\delta_\kappa\log\frac{1}{\varepsilon})$ $\mathcal{O}(D\log\log\frac{1}{\varepsilon})$
FEDNL	$\mathcal{O}(D^2)$	$\mathcal{O}(\log\frac{1}{\varepsilon})$	$\mathcal{O}(D)$	$\mathcal{O}(D\log\frac{1}{\varepsilon})$
SHED	$\mathcal{O}(D^2)$	$\mathcal{O}(\log\frac{1}{\varepsilon})$	–	$\mathcal{O}(D^2)$
FEDNEWTON	$\mathcal{O}(D^2)$	$\mathcal{O}(\log\log\frac{1}{\varepsilon})$	$\mathcal{O}(D^2)$	$\mathcal{O}(D^2\log\log\frac{1}{\varepsilon})$
FHAN	$\mathcal{O}(dD)$	$\mathcal{O}(\log\frac{1}{\varepsilon})$	$\mathcal{O}(D)$	$\mathcal{O}(D\log\frac{1}{\varepsilon})$

NOTE: ALL ANALYSES ARE CARRIED OUT UNDER THE ASSUMPTION OF A ε-ACCURATE SOLUTION. DONE AND DANE ACTUALLY COMMUNICATE TWICE PER ROUND. κ IN DANE REPRESENTS THE CONDITION NUMBER, $\delta_\kappa < 1$ IS ASSUMED IN DONE, AND FEDNEWTON DENOTES THE IDEALIZED VARIANT ANALYZED BY SAFARYAN ET AL. (2022). FHAN'S $\mathcal{O}(\log\frac{1}{\varepsilon})$ IS THE LOCAL RATE OF THEOREM 5.6.

it. We score each coordinate by the squared magnitude of the gradient it carries—the quantity the current step must resolve—and sample proportionally:

$$p_i = \widehat{l}(i)/\sum_{i=1}^{D}\widehat{l}(i), \quad \widehat{l}(i) = g_i^2/\sum_{i=1}^{D} g_i^2, \tag{5}$$

where $\widehat{l}(i)$ is the normalized squared magnitude of $g_i \in \boldsymbol{g}$, serving as an inexpensive surrogate for the leverage scores of Lemma 5.4, and $i \in [D]$. Sampling operates within each parameter block (e.g., per layer), so the captured cross-coordinate curvature remains confined within blocks.

4.3 INVERSE HESSIAN WITH WOODBURY IDENTITY

The update requires the inverse of the Hessian approximation, and no full-sized matrix may be formed at any point. The Woodbury identity (Sherman & Morrison, 1950) sidesteps this: for $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{U} \in \mathbb{R}^{N\times n}, \boldsymbol{C} \in \mathbb{R}^{n\times n}, \boldsymbol{V} \in \mathbb{R}^{n\times N}$ with $n \ll N$,

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}. \tag{6}$$

Applying (6) to the Nyström-approximated inverse (4), the local second-order update becomes

$$(\boldsymbol{H}_{DD}+\lambda\boldsymbol{I})^{-1}\bar{\boldsymbol{g}} \approx \frac{1}{\lambda}\left(\bar{\boldsymbol{g}} - \frac{1}{\lambda}\boldsymbol{H}_{Dd}\left((\widetilde{\boldsymbol{H}}_{dd})^{\dagger}\left(\boldsymbol{H}_{dD}\bar{\boldsymbol{g}}\right)\right)\right). \tag{7}$$

Here $\boldsymbol{H}_{dD} := \boldsymbol{H}_{Dd}^{\top}$, $\widetilde{\boldsymbol{H}}_{dd} := \boldsymbol{H}_{dd} + \frac{1}{\lambda}\boldsymbol{H}_{dD}\boldsymbol{H}_{Dd}$, and the Moore–Penrose pseudoinverse accommodates the case where the core matrix $\widetilde{\boldsymbol{H}}_{dd}$ is singular. The reformulation exposes the structure of the step: (7) is the ridge update $\bar{\boldsymbol{g}}/\lambda$ plus a curvature correction confined to the span of the d sampled columns; on its orthogonal complement the step coincides with the ridge update. All curvature information thus lives in a $D \times d$ sketch and a $d \times d$ solve—nothing larger than a model-sized vector ever leaves the device, in contrast to federated second-order methods that communicate curvature or gradients alongside updates.

Cost of the local solve. The inversion in (4) would cost $\mathcal{O}(D^3)$; the Woodbury reformulation (7) reduces it to $\mathcal{O}(Dd + d^3)$ with $d \ll D$, and the correction factorizes into matrix–vector products, so no dense product is formed.

4.4 SINGULAR-VALUE SMOOTHING

The only numerically sensitive object in (7) is the pseudoinverse of the core matrix: a small singular value σ is amplified by $1/\sigma$ in the inverse, and truncating it discards the associated direction entirely. Prior second-order methods instead clip the curvature or bound it below (Liu et al., 2024b; Elbakary et al., 2024). We stabilize inside the inversion itself: writing the SVD $\widetilde{\boldsymbol{H}}_{dd} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, we damp only the singular-value map, $1/\sigma \mapsto 1/\sqrt{\sigma^2 + \lambda_s}$, retaining every singular direction and capping the amplification exactly where the inverse is sensitive:

$$(\widetilde{\boldsymbol{H}}_{dd})^\dagger = \boldsymbol{V}\widetilde{\boldsymbol{S}}^{+}\boldsymbol{U}^\top, \tag{8}$$

where $\widetilde{\boldsymbol{S}}^{+}$ is the diagonal matrix with $\widetilde{S}^{+}_{ii} = 1/\sqrt{S_{ii}^2 + \lambda_s}$ and $\lambda_s > 0$ is the smoothing parameter.

The smoothing adds no factorization cost and leaves every singular direction intact; we adopt it in place of rank truncation.

5 CONVERGENCE ANALYSIS

We bound the Nyström sampling error (Lemma 5.4), prove a contraction inequality (Theorem 5.5) and its linear rate (Theorem 5.6), and compare complexities with related methods (Table 1). Notation and standard assumptions come first.

Assumption 5.1 (Twice differentiable and strongly convex)**.** The objective F is closed, convex, and twice differentiable, with $\nabla^2 F(\boldsymbol{w}) \succeq \upsilon \boldsymbol{I}$.

Assumption 5.2 (Lipschitz smoothness)**.** The second-order derivatives of F are M-Lipschitz smooth, i.e., $\|\nabla^2 F(\boldsymbol{w}) - \nabla^2 F(\boldsymbol{w}')\| \le M\|\boldsymbol{w} - \boldsymbol{w}'\|$ and gradient satisfies the L-lipschitz condition, i.e., $\|\nabla F(\boldsymbol{w}) - \nabla F(\boldsymbol{w}')\| \le L\|\boldsymbol{w} - \boldsymbol{w}'\|$.

Assumption 5.3 (Bounded heterogeneity)**.** For every client j and all $\boldsymbol{w}$ in the region of analysis: (i) Hessian similarity $\|\nabla^2 F_j(\boldsymbol{w}) - \nabla^2 F(\boldsymbol{w})\| \le \Gamma$; (ii) B-local dissimilarity in the sense of FedProx (Li et al., 2020), $\mathbb{E}[\|\nabla F_j(\boldsymbol{w})\|^2] \le B^2\|\nabla F(\boldsymbol{w})\|^2$; (iii) client curvature lower bound $\nabla^2 F_j(\boldsymbol{w}) \succeq (1-\epsilon)\upsilon \boldsymbol{I}$ for some $\epsilon \in (0, \frac{1}{2})$.

Lemma 5.4 (Nyström approximation error under leverage-score sampling)**.** *With Assumptions 5.1, 5.2, let $c > 0$ and $\delta \in (0,1)$ be fixed constants. For a target accuracy $\epsilon_l \in (0,1)$ and target rank k of the low-rank approximation, if the sampling dimension satisfies $d \ge \frac{c}{\epsilon_l^2} k \ln \frac{k}{\delta}$ with columns drawn from a distribution dominating the leverage scores, the exact Hessian $\boldsymbol{H}$ and its Nyström approximation $\hat{\boldsymbol{H}}$ satisfy*

$$\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \le \rho_{Ny}, \tag{9}$$

where $\rho_{Ny} = (1+\epsilon_l)\lambda_{k+1}(\boldsymbol{H})$, $\lambda_{k+1}(\boldsymbol{H})$ denotes the $(k+1)$-th eigenvalue of $\boldsymbol{H}$, with probability at least $1-\delta$. A detailed proof, following the leverage-score analysis of Gittens & Mahoney (2013), is given in Appendix A.2.

5.1 CONVERGENCE ANALYSIS FOR FHAN

Theorem 5.5. *(Convergence of FHaN). Under Assumptions 5.1–5.3, let $\delta \in (0,1)$ and $\epsilon \in (0, \frac{1}{2})$, and let K denote the number of communication rounds under analysis. Suppose that each $\nabla^2 F_j(\cdot)$ is uniformly upper bounded, i.e., $\nabla^2 F_j(\boldsymbol{w}) \preceq C\boldsymbol{I}$ for all $j \in [m]$. When $|\mathcal{D}_j| \ge \frac{4C}{\upsilon\epsilon^2} \log \frac{2DK}{\delta}$ and $d \ge \frac{c}{\epsilon_l^2} k \ln \frac{kK}{\delta}$, we obtain*

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| \le P(\eta,\lambda)\|\boldsymbol{w}^t - \boldsymbol{w}^*\| + \frac{3M}{2\upsilon}\|\boldsymbol{w}^t - \boldsymbol{w}^*\|^2, \tag{10}$$

with probability at least $1 - 3K\delta$. Here, $P(\eta,\lambda)$ is the contraction factor $P(\eta,\lambda) = \frac{(1-\eta)L}{\upsilon} + \frac{\eta\Gamma L}{(1-\epsilon)\upsilon^2} + \frac{\eta\rho_{Ny}L}{(1-\epsilon)\upsilon\lambda} + \frac{\eta\epsilon_B L}{\lambda}$, where Γ is the Hessian-similarity constant of Assumption 5.3 and $\epsilon_B = 1 + B/\sqrt{\delta}$ is the resulting gradient-dissimilarity level of Assumption 5.3(ii) (Appendix B).

In (10) the algorithm converges when $P(\eta,\lambda) < 1$; the discussion and proof are in Appendix B. The first term is the deviation from standard Newton and vanishes at $\eta = 1$ when Γ, ρ_{Ny} and ϵ_B are zero;

the second and fourth terms are governed by Assumption 5.3 and grow with heterogeneity (Li et al., 2020; Karimireddy et al., 2020); the third is the Nyström error, which depends on sampling quality and the number of sampled columns.

Theorem 5.6. *(Convergence rate of FHaN). Under Assumptions 5.1–5.3 and the iteration of Theorem 5.5, if the initial point satisfies* $\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| \leq \frac{(1-P(\eta,\lambda))v}{3M}$*, then achieving* $\|\boldsymbol{w}^t - \boldsymbol{w}^*\| \leq \varepsilon$ *requires* $T = \mathcal{O}(\log \frac{1}{\varepsilon})$ *iterations.*

The proof is in Appendix C. This result establishes linear convergence, as typical for Newton-type methods. Although the second-order term is not globally dominant, it improves convergence near the optimum, often after a few warm-up steps with a first-order method.

5.2 Theoretical Comparisons

Table 1 compares FHaN with related methods: second-order methods need fewer iterations T than FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) and FHaN retains this without extra communication; FedNL (Safaryan et al., 2022) and SHED (Dal Fabbro et al., 2024) communicate little but incur $\mathcal{O}(D^2)$ memory, while Hessian-free DANE (Shamir et al., 2014) and DONE (Dinh et al., 2022) are memory-efficient but communicate twice per round.

Recent methods such as FAGH (Sen et al., 2024) and Fed-Sophia (Elbakary et al., 2024) incorporate second-order curvature into the update with low memory overhead but lack convergence guarantees; we therefore compare them at the algorithmic level. FAGH, approximating the Hessian by sampling its first row, is the $d{=}1$ special case of the Nyström construction, while ours exposes the budget as a free parameter; Fed-Sophia's diagonal costs $\mathcal{O}(D)$ per iteration, matching ours only at trivial budgets. FHaN thus retains the $\mathcal{O}(D \log \frac{1}{\varepsilon})$ communication of the compressed-Hessian family at $\mathcal{O}(dD)$ memory; Section 6 compares against both baselines empirically.

6 Experiments

The largest gains arise in from-scratch training: TinyViT reaches 59.1% under FHaN against 35.2% under FedAvg ($+23.9$ points). More broadly, FHaN builds a per-block low-rank curvature correction from sampled Hessian columns, so its benefit should concentrate where the trainable parameters are dense matrix blocks and the problem is ill-conditioned enough for curvature to matter; we test this on two task families—federated fine-tuning of frozen backbones and from-scratch training of FC-dominated models—compare against the runnable second-order baselines, and close with the role of λ and explicit boundary cases.

Setup. Two task families probe the design. For federated fine-tuning, ImageNet-pretrained backbones (ViT-B/16, ResNet-50, ResNet-18) are frozen and only a two-layer MLP head (256 hidden units) or a linear head trains, on CIFAR-100 and CIFAR-10; for from-scratch training, dense-matrix models—TinyViT (patch 4, width 64, 2 blocks), a CNN with frozen convolutions and a trainable FC head (CNNFC), and MLPs of varying depth and width—train on Fashion-MNIST and MNIST. The federation protocol is fixed across tasks: 30 clients, full participation, label-wise Dirichlet partitions ($\alpha \in \{1.0, 0.5\}$; the extreme $\alpha{=}0.1$ in the boundary study), 60 rounds, one local epoch, batch size 512. Configurations are selected on seed 0 (four for FHaN over $\eta \in \{0.004, 0.012\} \times \lambda \in \{0.005, 0.02\}$, two per first-order baseline over $\eta \in \{0.05, 0.1\}$) and evaluated over 3 seeds with paired splits and initializations, against FedAvg (McMahan et al., 2017) and SCAFFOLD (Karimireddy et al., 2020) at their tuned rates; DONE, FedNL, and successors form full-Hessian-scale quantities locally and do not run at these scales. FHaN uses sampling rate 3×10^{-4} per block and $\lambda_s = 10^{-4}$ throughout.

Federated fine-tuning. Table 2 and Figure 2 show that with frozen backbones, FHaN improves both final accuracy and rounds-to-target across all tested heads and datasets: on CIFAR-100 the margin is +5.3 points with a ViT-B/16 backbone and +13.2 with ResNet-50, reaching the 60% target in 2–9 rounds where FedAvg needs 17–19 or never. The margin grows as the head optimization becomes harder—the same MLP head yields +0.7 points on CIFAR-10 (near-saturated), +5.3 on CIFAR-100, and +13.2 on the harder ResNet-50 features—matching the expectation that curvature is worth most when the problem is ill-conditioned. The advantage is also not tied to a nonconvex head: on the *linear* head over ViT-B/16 features, a convex problem with a genuinely ill-conditioned spectrum (100 classes, 768 dimensions), FHaN gains +4.1 points and reaches target $3.5\times$ faster, whereas on the

Table 2: Final-round test accuracy (%, mean ± std over 3 seeds) and rounds to target accuracy (per seed). Targets: 60% for CIFAR-100, 80% for Fashion-MNIST and CIFAR-10, 90% for MNIST. Bold marks the better method per task; "–" marks targets unreached within 60 rounds. SCAFFOLD is compared on the four flagship scenarios and tracks FedAvg on each. The MNIST row uses the stable $\lambda = 0.02$ arm (Appendix F.3).

TASK (α)	METHOD	FINAL ACC.	ROUNDS
VIT-B/16 + MLP HEAD CIFAR-100 (1.0)	**FHAN**	**78.42±0.52**	**3/3/3**
	FEDAVG	73.10±0.06	18/19/17
	SCAFFOLD	73.28±0.28	19/19/18
RESNET-50 + MLP HEAD CIFAR-100 (1.0)	**FHAN**	**70.16±0.78**	**9/6/9**
	FEDAVG	56.95±0.81	–/–/–
	SCAFFOLD	57.44±0.69	–/–/–
VIT-B/16 + LINEAR CIFAR-100 (1.0)	**FHAN**	**80.20±0.18**	**2/2/2**
	FEDAVG	76.13±0.24	7/7/7
TINYVIT FROM SCRATCH FASHION-MNIST (1.0)	**FHAN**	**59.08±0.81**	**8/7/8**
	FEDAVG	35.15±0.70	–/–/–
	SCAFFOLD	34.94±0.34	–/–/–
CNNFC (FROZEN CONV) FASHION-MNIST (1.0)	**FHAN**	**83.63±0.38**	**19/22/27**
	FEDAVG	75.18±1.26	–/–/–
	SCAFFOLD	75.24±0.75	–/–/–
MLP FASHION-MNIST (1.0)	**FHAN**	**84.68±0.26**	**7/6/6**
	FEDAVG	83.58±0.10	19/19/18
RESNET-18 + MLP HEAD CIFAR-10 (1.0)	**FHAN**	**86.34±0.25**	**5/7/5**
	FEDAVG	85.65±0.19	11/12/12
MLP3 / MNIST (1.0) (λ=0.02 ARM)	FHAN	92.74±0.07	**10/12/11**
	FEDAVG	**93.77±0.03**	18/20/19

saturated CIFAR-10 ResNet-18 linear head it loses (see *Boundary cases* below). What decides the outcome is conditioning, not convexity.

From-scratch training of FC-dominated models. The pattern holds without pretraining: TinyViT, whose blocks are all dense matrices, reaches 59.1% under FHaN against 35.2% under FedAvg (+23.9 points; FHaN passes the FedAvg final accuracy within 6–11 rounds). CNNFC, in which the convolutional layers are frozen at initialization and only the three FC blocks train, gains +8.5 points over FedAvg with tight variance. The contrast with the *same* architecture trained end-to-end is instructive: full CNN training is the one setting where FHaN loses to FedAvg by 4–6 points (see *Boundary cases* below), because the convolutional blocks force a stabilizing λ that suppresses the curvature benefit on the FC blocks—freezing them flips the outcome from -5 to $+8.5$ points. Standard MLPs show the same direction at smaller margins (+1.1 points and $3\times$ fewer rounds on Fashion-MNIST).

Comparison with second-order baselines. Among methods that use curvature while communicating model-sized vectors, Fed-Sophia and FAGH represent the diagonal and rank-one regimes; DONE and FedNL form full-Hessian-scale quantities and did not complete a run at these scales. Table 3 compares all five optimizers on three task families under three heterogeneity levels: FHaN attains the highest final accuracy in every cell. Fed-Sophia, the strongest alternative in six of nine cells, confirms that curvature pays even in diagonal form; FAGH is competitive on the ResNet heads yet falls below FedAvg elsewhere—a single sampled row recovers too little of the cross-coordinate structure that FHaN's column budget captures.

The role of λ. The parameter λ trades stability against correction strength, with scenario-dependent safe range (Appendix F.3): on MLP3, $\lambda = 0.005$ reaches 90% in 3–6 rounds (4–$5\times$ faster than FedAvg) but collapses late on 4 of 6 runs, while $\lambda = 0.02$ is reliable at $1.8\times$ speed and -1.0 points; on CNNFC the whole +8.5-point advantage lives at low λ without collapse; on TinyViT, low λ is stable and necessary.

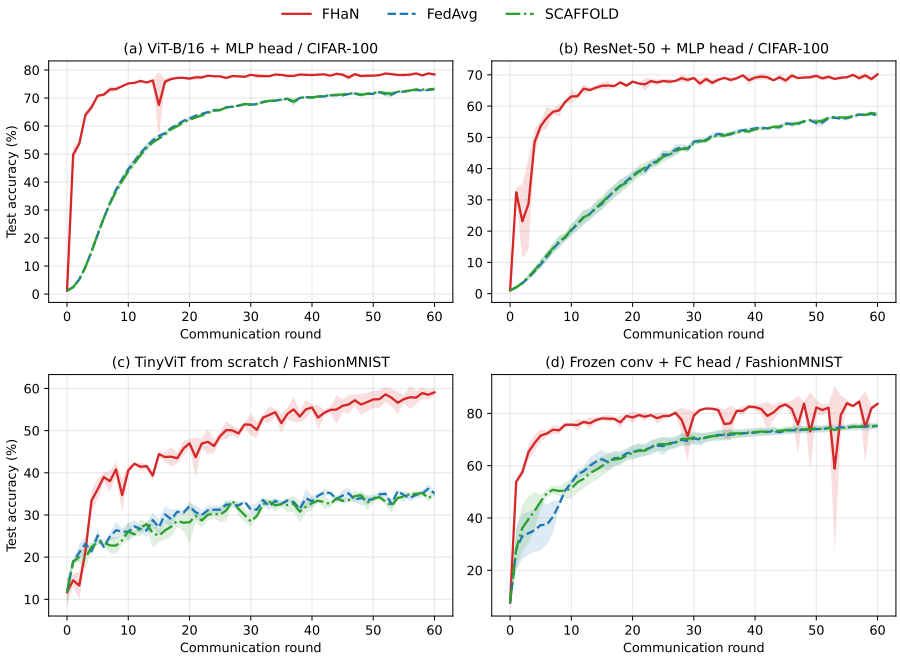


Figure 2: Test accuracy over communication rounds (mean $\pm$ std over 3 seeds, $\alpha = 1.0$). (a)–(b) federated fine-tuning on CIFAR-100 with frozen ImageNet backbones; (c) TinyViT trained from scratch; (d) frozen-convolution CNN with FC head. The advantage of FHaN grows with the difficulty of the head optimization problem.

Table 3: Runnable second-order baselines: final-round test accuracy (%), one run per cell at each method's selected configuration, from the campaign preceding the three-seed protocol of Table 2; bold marks the best per cell.

Task (α)	FedAvg	SCAFFOLD	Fed-Sophia	FAGH	FHaN
MLP (0.1)	87.85	82.80	89.09	84.99	**90.69**
MLP (0.5)	89.95	89.37	91.82	90.00	**92.92**
MLP (1.0)	91.86	91.27	91.96	93.79	**95.00**
CNN (0.1)	75.89	76.83	78.94	75.00	**82.03**
CNN (0.5)	82.43	77.74	85.34	83.00	**86.00**
CNN (1.0)	85.29	74.80	86.28	81.00	**88.00**
ResNet head (0.1)	54.40	46.99	39.83	60.84	**61.03**
ResNet head (0.5)	71.95	72.87	71.46	69.07	**73.96**
ResNet head (1.0)	78.00	76.00	81.00	72.19	**82.12**

Boundary cases. The negative results are as informative as the wins: (i) end-to-end CNN training loses to FedAvg by 4–6 points—convolutional blocks need a large λ that suppresses the correction on the FC blocks; (ii) saturated tasks (CIFAR-10 linear head) lose 7–10 points to sampling variance without curvature benefit; (iii) benign convex problems tie, as the theory predicts for concentrated spectra; (iv) under extreme heterogeneity ($\alpha = 0.1$) FHaN loses across models, matching the heterogeneity term in the contraction condition.

7 Conclusion

We presented FHaN, a Newton-type federated optimizer that retains second-order benefits at first-order communication: its aggregated update satisfies a contraction condition under explicit heterogeneity and sampling-error bounds, and its advantage concentrates where curvature matters—+5.3 to +13.2 points on frozen-backbone fine-tuning with up to $6\times$ fewer rounds to target, +23.9 points from scratch, and every cell against the second-order baselines—while four boundary regimes mark its limits, and adaptive per-block λ schedules and sampling budgets are natural continuations.

IMPACT STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

AI USE STATEMENT

During the preparation of this manuscript, the authors used a generative AI assistant (Claude, Anthropic) as a writing aid: language editing and restructuring of drafts written by the authors, mechanical consistency checks (terminology, cross-references, and bibliography hygiene), literature verification of cited related work, and conversion of the manuscript between conference templates. All technical content, including the method, the theory, the experimental design, and all substantive claims, was developed and decided by the authors, and every AI-assisted edit was reviewed and accepted by the authors. We take responsibility for the final content of this work, including text produced with the aid of generative AI.

REFERENCES

Artem Agafonov, Dmitry Kamzolov, Rachael Tappenden, Alexander Gasnikov, and Martin Takáč. Flecs: A federated learning second-order framework via compression and sketching. *Optimization Methods and Software*, 40(5):1222–1248, 2025.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 557–565. PMLR, 06–11 Aug 2017.

C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593, 1965.

Konstantin Burlachenko and Peter Richtárik. Unlocking fednl: Self-contained compute-optimized implementation, 2024.

Augustin-Louis Cauchy. *Cours d'analyse de l'École Royale Polytechnique*. Cambridge Library Collection - Mathematics. Cambridge University Press, 1821.

Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17): e2024789118, 2021.

Yun-Wei Chu, Dong-Jun Han, and Christopher G. Brinton. Only send what you need: Learning to communicate efficiently in federated multilingual machine translation, 2025.

Nicolò Dal Fabbro, Subhrakanti Dey, Michele Rossi, and Luca Schenato. Shed: A newton-type algorithm for federated learning based on incremental hessian eigenvector sharing. *Automatica*, 160:111460, 2024. ISSN 0005-1098.

Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

Yicheng Di, Hongjian Shi, Ruhui Ma, Honghao Gao, Yuan Liu, and Weiyu Wang. Fedrl: A reinforcement learning federated recommender system for efficient communication using reinforcement selector and hypernet generator. *ACM Trans. Recomm. Syst.*, July 2024. Just Accepted.

Canh T. Dinh, Nguyen H. Tran, Tuan Dung Nguyen, Wei Bao, Amir Rezaei Balef, Bing B. Zhou, and Albert Y. Zomaya. DONE: Distributed approximate newton-type method for federated edge learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2648–2660, 2022.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Ahmed Elbakary, Chaouki Ben Issaid, Mohammad Shehab, Karim Seddik, Tamer ElBatt, and Mehdi Bennis. Fed-sophia: A communication-efficient second-order federated learning algorithm. In *IEEE International Conference on Communications (ICC)*, pp. 950–955, 2024.

Anis Elgabli, Chaouki Ben Issaid, Amrit Singh Bedi, Ketan Rajawat, Mehdi Bennis, and Vaneet Aggarwal. FedNew: A communication-efficient and privacy-preserving Newton-type method for federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5861–5877. PMLR, 17–23 Jul 2022.

Mohamed Elsayed, Homayoon Farrahi, Felix Dangel, and A. Rupam Mahmood. Revisiting scalable hessian diagonal approximations for applications in reinforcement learning, 2024.

R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 08 1963. ISSN 0010-4620.

Sagnik Ghosh, Hardik Tankaria, Arnav Bhavsar, Varun Dutt, and Dinesh Singh. Nys-Fed: Communication and memory efficient nyström approximated curvature for federated learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2026.

Alex Gittens and Michael Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 567–575, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Yonghai Gong, Yichuan Li, and Nikolaos M. Freris. Fedadmm: A robust federated deep learning framework with adaptivity to system heterogeneity, 2022.

Sunny Gupta, Mohit Jindal, Pankhi Kashyap, Pranav Jeevan, and Amit Sethi. Flens: Federated learning with enhanced nesterov-newton sketch. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 7774–7783, 2024.

Vipul Gupta, Avishek Ghosh, Michal Derezinski, Rajiv Khanna, Kannan Ramchandran, and Michael Mahoney. Localnewton: Reducing communication bottleneck for distributed learning, 2021.

Alexander Herzog, Robbie Southam, Othmane Belarbi, Saif Anwar, Marcello Bullo, Pietro Carnelli, and Aftab Khan. Selective updates and adaptive masking for communication-efficient federated learning. *IEEE Transactions on Green Communications and Networking*, 8(2):852–864, 2024.

Hiro Ishii, Kenta Niwa, Hiroshi Sawada, and Akinori Fujino. FedPM: Federated learning using second-order optimization with preconditioned mixing of local parameters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 13–18 Jul 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jian Li, Yong Liu, and Weiping Wang. Fedns: A fast sketching newton-type algorithm for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13509–13517, 2024.

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.

Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent advances on federated learning: A systematic survey. *Neurocomputing*, 597:128019, 2024a. ISSN 0925-2312.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2024b.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024c. doi: 10.1109/TKDE.2024.3352628.

Xin Ma, Renyi Bao, Jinpeng Jiang, Yang Liu, Arthur Jiang, Jun Yan, Xin Liu, and Zhisong Pan. Fedsso: A federated server-side second-order optimization algorithm. *arXiv preprint arXiv:2206.09576*, 2022.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, Lille, France, 07–09 Jul 2015. PMLR.

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*, pp. 1273–1282. PMLR, 2017.

Challa Nagaraju, Mrinmay Sen, and C. Krishna Mohan. FONN: Federated optimization with nys-newton. In *IEEE Region 10 Conference (TENCON)*, pp. 530–534, Chiang Mai, Thailand, 2023. IEEE.

Goran Saman Nariman and Hozan Khalid Hamarashid. Communication overhead reduction in federated learning: a review. *International Journal of Data Science and Analytics*, 19(2):185–216, 2025.

Zhenhui Pan, Yawen Li, Zeli Guan, Meiyu Liang, Ang Li, Jia Wang, and Feifei Kou. Rfcsc: Communication efficient reinforcement federated learning with dynamic client selection and adaptive gradient compression. *Neurocomputing*, 612:128672, 2025.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Mónica Ribero and Haris Vikalo. Reducing communication in federated learning via efficient client sampling. *Pattern Recognition*, 148:110122, 2024. ISSN 0031-3203.

Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. Fednl: Making newton-type methods applicable to federated learning, 2022. URL `https://arxiv.org/abs/2106.02969`.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002. doi: 10.1162/08997660260028683.

Mrinmay Sen, C. Krishna Mohan, and Kai Qin. Nys-FL: A communication efficient federated learning with Nyström approximated global newton direction. In *IEEE International Conference on High Performance Computing and Communications, Data Science and Systems, Smart City, and Dependability (HPCC/DSS/SmartCity/DependSys)*, pp. 217–224, 2023.

Mrinmay Sen, A. K. Qin, and Krishna Mohan C. Fagh: Accelerating federated learning with approximated global hessian, 2024.

Mrinmay Sen, Sidhant R. Nair, and C. Krishna Mohan. Accelerated training of federated learning via second-order methods. *arXiv preprint arXiv:2505.23588*, 2025.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1000–1008, Bejing, China, 22–24 Jun 2014. PMLR.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. ISSN 00034851.

Shusen Wang, Fred Roosta, Peng Xu, and Michael W Mahoney. Giant: Globally improved approximate newton method for distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10665–10673, 2021.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zihao Zhao, Yuzhu Mao, Yang Liu, Linqi Song, Ye Ouyang, Xinlei Chen, and Wenbo Ding. Towards efficient communications in federated learning: A contemporary survey. *Journal of the Franklin Institute*, 360(12):8669–8703, 2023.

A Auxiliary Lemmas

A.1 Local Second-Order Gradient Bound (from Lemma A.1. in Gupta et al. (2021))

We directly adopt the result from Lemma A.1. in Gupta et al. (2021). Under Assumptions 5.1, 5.2, and $\nabla^2 F_j(\boldsymbol{w}) \preceq C\boldsymbol{I}$, let $\epsilon \in (0, \frac{1}{2})$ and $\delta \in (0, 1)$. Then, for a given $|\mathcal{D}_j| \geq \frac{4C}{\upsilon\epsilon^2} \log \frac{2D}{\delta}$, we obtain

$$(1-\epsilon)\upsilon \preceq \nabla^2 F_j(\boldsymbol{w}) \preceq (1+\epsilon)L, \tag{11}$$

for all $\boldsymbol{w} \in \mathbb{R}^D$ and $j \in [m]$ with probability at least $1-\delta$.

A.2 Proof of Lemma 5.4

Proof. The approximation error of the Nyström has been extensively studied in previous work. Here, we introduce a result from Gittens & Mahoney (2013) concerning uniform sampling. Let $\boldsymbol{H} \in \mathbb{R}^{D\times D}$ be a symmetric positive definite matrix with eigendecomposition $\boldsymbol{H} = \sum_{i=1}^{D} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$, where $\{\lambda_i\}_{i=1}^D$ are the eigenvalues and $\{\boldsymbol{u}_i\}_{i=1}^D$ are the corresponding orthonormal eigenvectors of $\boldsymbol{H}$. Let $\boldsymbol{H}_k = \sum_{i=1}^{k} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ denote the best rank-k approximation of $\boldsymbol{H}$. Then, there exists a constant $c > 0$ and a parameter $\beta = \frac{D}{k} \max_i \hat{l}(i)$ that measures the distribution of the column space, such that when $d \geq c\beta k \ln \frac{k}{\delta}$, the approximation error between $\boldsymbol{H}$ and its low-rank approximation $\hat{\boldsymbol{H}}$ is bounded by

$$\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \leq (1 + \frac{2D}{d})\|\boldsymbol{H} - \boldsymbol{H_k}\|, \tag{12}$$

with probability at least $1-\delta$ and where $\|\boldsymbol{H} - \boldsymbol{H_k}\| = \|\sum_{i=k+1}^{D} \lambda_i \boldsymbol{u_i} \boldsymbol{u_i}^\top\| = \lambda_{k+1}(\boldsymbol{H})$.

Building on this, we consider the case of leverage score sampling. Unlike uniform sampling, which requires a larger number of samples to ensure quality, leverage score sampling focuses solely on approximation accuracy and does not depend on data distribution. Setting $\epsilon_l \in (0, 1)$ and applying the Matrix Bernstein inequality, we obtain the following result when $d \geq \frac{c}{\epsilon_l^2} k \ln \frac{k}{\delta}$:

$$\|\boldsymbol{H} - \hat{\boldsymbol{H}}\| \leq (1+\epsilon_l)\lambda_{k+1}(\boldsymbol{H}) = \rho_{Ny}, \tag{13}$$

with probability at least $1-\delta$. Using a larger d allows the approximation error ϵ_l to be reduced, leading to improved approximation quality, which is consistent with both our empirical observations and experimental results. Moreover, leverage score sampling reduces the amplification factor $1 + \frac{D}{d}$ to a controllable level determined by the target precision.

□

B Proof of Theorem 5.5

Proof. We analyze the full-batch case with one training epoch. Intuitively, the entire update can be decomposed into the ideal global exact update and an error term, which can then be analyzed separately. Before proceeding with the proof, we define $\bar{\boldsymbol{g}}^t := \nabla F(\boldsymbol{w}^t)$, $\bar{\boldsymbol{g}}_j^t := \nabla F(\boldsymbol{w}_j^t)$,$\boldsymbol{H}_F^t := \nabla^2 F(\boldsymbol{w}^t)$, $\boldsymbol{H}_{F,j}^t := \nabla^2 F_j(\boldsymbol{w}^t)$, and $\hat{\boldsymbol{H}}_{F,j}^t := \hat{\boldsymbol{H}}_{f,j}^t + \lambda \boldsymbol{I}$, where $\hat{\boldsymbol{H}}_{f,j}^t$ is the approximate matrix of $\nabla^2 f_j(\boldsymbol{w}^t)$ derived via the Nyström. We then obtain

$$
\begin{aligned}
\|e^t\| &= \left\| (\boldsymbol{H}_F^t)^{-1}\bar{\boldsymbol{g}}^t - \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}((\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\bar{\boldsymbol{g}}_j^t) \right\| \\
&\leq \left\| \underbrace{\frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}\left((\boldsymbol{H}_F^t)^{-1} - (\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\right)\bar{\boldsymbol{g}}^t}_{e_1^t} + \underbrace{\frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}\left((\hat{\boldsymbol{H}}_{F,j}^t)^{-1}(\bar{\boldsymbol{g}}^t - \bar{\boldsymbol{g}}_j^t)\right)}_{e_2^t} \right\|.
\end{aligned} \tag{14}
$$

We first consider term e_1^t, where the second-order gradient error arises from both data heterogeneity and the Nyström approximation. By introducing the Hessian similarity bound inspired by (A2) of SCAFFOLD Karimireddy et al. (2020), we obtain $\|\boldsymbol{H}_F^t - \boldsymbol{H}_{F,j}^t\| \leq \Gamma$. For any invertible matrices $\boldsymbol{A}$ and $\boldsymbol{A'}$, we have $(\boldsymbol{A'})^{-1} - \boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{A'})(\boldsymbol{A'})^{-1}$. Then with Eq. (11) and $\delta \in (0,1)$, we obtain

$$
\left\|(\boldsymbol{H}_F^t)^{-1} - (\boldsymbol{H}_{F,j}^t)^{-1}\right\| \leq \left\|(\boldsymbol{H}_F^t)^{-1}\right\| \left\|\boldsymbol{H}_F^t - \boldsymbol{H}_{F,j}^t\right\| \left\|(\boldsymbol{H}_{F,j}^t)^{-1}\right\| \leq \frac{\Gamma}{(1-\epsilon)\upsilon^2}, \tag{15}
$$

with probability at least $1-\delta$. Similarly, by applying (13) , we obtain

$$
\begin{aligned}
\left\|(\boldsymbol{H}_{F,j}^t)^{-1} - (\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\right\| &\leq \left\|(\boldsymbol{H}_{F,j}^t)^{-1}\right\| \left\|\boldsymbol{H}_{F,j}^t - \hat{\boldsymbol{H}}_{F,j}^t\right\| \left\|(\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\right\| \\
&\leq \frac{1}{(1-\epsilon)\upsilon}\rho_{Ny}\frac{1}{\lambda} \leq \frac{\rho_{Ny}}{(1-\epsilon)\upsilon\lambda},
\end{aligned} \tag{16}
$$

with probability at least $1-2\delta$. The last step uses $\|(\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\| \leq 1/\lambda$, which holds because the Nyström approximation of a PSD Hessian is PSD, so $\hat{\boldsymbol{H}}_{F,j}^t = \hat{\boldsymbol{H}}_{f,j}^t + \lambda\boldsymbol{I} \succeq \lambda\boldsymbol{I}$. Using Eq. (15) and Eq. (16), the error in the second-order gradient component can be bounded as:

$$
\begin{aligned}
\|e_1^t\| &= \left\| \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}\left((\boldsymbol{H}_F^t)^{-1} - (\boldsymbol{H}_{F,j}^t)^{-1} + (\boldsymbol{H}_{F,j}^t)^{-1} - (\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\right)\bar{\boldsymbol{g}}^t \right\| \\
&\leq \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}(\left\|(\boldsymbol{H}_F^t)^{-1} - (\boldsymbol{H}_{F,j}^t)^{-1}\right\| + \left\|(\boldsymbol{H}_{F,j}^t)^{-1} - (\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\right\|)\left\|\bar{\boldsymbol{g}}^t\right\| \\
&\leq \left(\frac{\Gamma}{(1-\epsilon)\upsilon^2} + \frac{\rho_{Ny}}{(1-\epsilon)\upsilon\lambda}\right)\left\|\bar{\boldsymbol{g}}^t\right\|,
\end{aligned} \tag{17}
$$

with probability at least $1-2\delta$.

Next, we consider term e_2^t, which involves bounding the heterogeneity of the first-order gradients. To this end, we adopt the B-local dissimilarity of Assumption 5.3(ii) (Definition 3 of FedProx Li et al. (2020)): $\mathbb{E}[\|\bar{\boldsymbol{g}}_j^t\|^2] \leq B^2\|\bar{\boldsymbol{g}}^t\|^2$. (The first- and second-order similarity bounds are not directly related, so referencing different works does not affect the validity of our analysis; the two gradient-similarity definitions are equivalent in form.) Applying Markov's inequality to $\|\bar{\boldsymbol{g}}_j^t\|^2$, for each client j and any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$
\|\bar{\boldsymbol{g}}_j^t\|^2 \leq \delta^{-1}B^2\|\bar{\boldsymbol{g}}^t\|^2 \quad \Longrightarrow \quad \|\bar{\boldsymbol{g}}_j^t\| \leq \tfrac{B}{\sqrt{\delta}}\|\bar{\boldsymbol{g}}^t\|, \tag{18}
$$

and therefore, by the triangle inequality, with $\epsilon_B := 1 + \frac{B}{\sqrt{\delta}}$,

$$
\|\bar{\boldsymbol{g}}^t - \bar{\boldsymbol{g}}_j^t\| \leq \epsilon_B\|\bar{\boldsymbol{g}}^t\|, \tag{19}
$$

with probability at least $1-\delta$. Consequently, we derive:

$$
\begin{aligned}
\|e_2^t\| &= \left\| \frac{|\mathcal{D}_j|}{|\mathcal{D}|}\sum_{j=1}^{m}\left((\hat{\boldsymbol{H}}_{F,j}^t)^{-1}(\bar{\boldsymbol{g}}^t - \bar{\boldsymbol{g}}_j^t)\right) \right\| \\
&\leq \frac{1}{\lambda}\|\bar{\boldsymbol{g}}^t - \bar{\boldsymbol{g}}_j^t\| \\
&\leq \frac{\epsilon_B}{\lambda}\|\bar{\boldsymbol{g}}^t\|,
\end{aligned} \tag{20}
$$

with probability at least $1-3\delta$, using $\|(\hat{\boldsymbol{H}}_{F,j}^t)^{-1}\| \le 1/\lambda$ as established above.

Finally, we derive the following recurrence relation for $\|\boldsymbol{w}^t - \boldsymbol{w}^*\|$:

$$\begin{aligned}
\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| &= \left\| \boldsymbol{w}^t - \boldsymbol{w}^* - \eta \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \sum_{j=1}^{m} \left((\hat{\boldsymbol{H}}_{F,j}^t)^{-1} \bar{\boldsymbol{g}}_j^t \right) \right\| \\
&= \left\| \boldsymbol{w}^t - \boldsymbol{w}^* - \eta (\boldsymbol{H}_F^t)^{-1} \bar{\boldsymbol{g}}^t + \eta e^t \right\| \\
&\le \left\| \boldsymbol{w}^t - \boldsymbol{w}^* - \eta (\boldsymbol{H}_F^t)^{-1} \bar{\boldsymbol{g}}^t \right\| + \left\| \eta e^t \right\| \\
&\le \|(\boldsymbol{H}_F^t)^{-1}\| \|\boldsymbol{H}_F^t (\boldsymbol{w}^t - \boldsymbol{w}^*) - \eta \bar{\boldsymbol{g}}^t\| + \left\| \eta e^t \right\| \\
&\le \frac{1}{\upsilon} \left\| (\boldsymbol{H}_F^t - \boldsymbol{H}_F^*)(\boldsymbol{w}^t - \boldsymbol{w}^*) \right\| + \left\| \boldsymbol{H}_F^* (\boldsymbol{w}^t - \boldsymbol{w}^*) - \eta \bar{\boldsymbol{g}}^t \right\| + \left\| \eta e^t \right\| \\
&\overset{(i)}{\le} \frac{1}{\upsilon} (M \left\| \boldsymbol{w}^t - \boldsymbol{w}^* \right\|^2 + (1-\eta) \left\| \bar{\boldsymbol{g}}^t \right\| + \frac{M}{2} \left\| \boldsymbol{w}^t - \boldsymbol{w}^* \right\|^2) + \left\| \eta e^t \right\| \\
&\le \left(\frac{(1-\eta)L}{\upsilon} + \frac{\eta \Gamma L}{(1-\epsilon)\upsilon^2} + \frac{\eta \rho_{Ny} L}{(1-\epsilon)\upsilon\lambda} + \frac{\eta \epsilon_B L}{\lambda} \right) \left\| \boldsymbol{w}^t - \boldsymbol{w}^* \right\| \\
&\quad + \frac{3M}{2\upsilon} \left\| \boldsymbol{w}^t - \boldsymbol{w}^* \right\|^2,
\end{aligned} \tag{21}$$

with probability at least $1-3\delta$, where (i) is derived from Taylor expansion with integral remainder and $\boldsymbol{H}_F^*$ denotes the Hessian of the global objective at the optimal solution $\boldsymbol{w}^*$.

We have obtained the recursion with coefficient

$$P(\eta, \lambda) = \frac{(1-\eta)L}{\upsilon} + \frac{\eta \Gamma L}{(1-\epsilon)\upsilon^2} + \frac{\eta \rho_{Ny} L}{(1-\epsilon)\upsilon\lambda} + \frac{\eta \epsilon_B L}{\lambda}. \tag{22}$$

We now show that one can choose $\eta \in (0,1)$ and $\lambda > 0$ so that $P(\eta, \lambda) < 1$, and the resulting parameter set is non-empty. (*Sufficient parameter inequalities implying* $P(\eta, \lambda) < 1$). Fix any margin $\alpha \in (0,1)$. Choose η, λ such that

$$\eta > 1 - \frac{(1-\alpha)\upsilon}{L}, \quad \eta < \frac{\alpha(1-\epsilon)\upsilon^2}{4\Gamma L}, \quad \lambda > \max\{ \frac{4\eta\rho_{Ny} L}{\alpha(1-\epsilon)\upsilon}, \frac{4\eta\epsilon_B L}{\alpha} \}. \tag{23}$$

Then (22) can be bounded as:

$$P(\eta, \lambda) \le \frac{L}{\upsilon} \left(\frac{(1-\alpha)\upsilon}{L} \right) + \frac{\alpha(1-\epsilon)\upsilon^2 \Gamma L}{4L\Gamma(1-\epsilon)\upsilon^2} + \frac{\alpha\eta\rho_{Ny} L(1-\epsilon)\upsilon}{4(1-\epsilon)\upsilon\eta\rho_{Ny} L} + \frac{\eta\epsilon_B L\alpha}{4\eta\epsilon_B L} = 1 - \frac{\alpha}{4} < 1. \tag{24}$$

Therefore, any (η, λ) satisfying (23) guarantees $P(\eta, \lambda) < 1$.

(*Feasibility: the parameter set in* (23) *is non-empty*). Define

$$\eta_{low} := 1 - \frac{(1-\alpha)\upsilon}{L}, \quad \eta_{up} := \frac{\alpha(1-\epsilon)\upsilon^2}{4\Gamma L}. \tag{25}$$

To make (23) feasible, it suffices that there exists $\eta \in (0,1)$ satisfying the first two inequalities in (23), i.e., $\eta_{low} < \eta < \eta_{up}$. This holds whenever the interval $(\eta_{low}, \eta_{up}) \cap (0,1)$ is non-empty, which is guaranteed by the following sufficient condition:

$$\Gamma < \frac{\alpha(1-\epsilon)\upsilon^2}{4(L - (1-\alpha)\upsilon)}. \tag{26}$$

Under (23), such an $\eta \in (0,1)$ exists, and hence the parameter set in (23) is non-empty. When Γ is so large that (26) does not hold (i.e., Γ is too large), then the above sufficient-condition argument no longer guarantees a contraction factor $P(\eta, \lambda) < 1$; we discuss this regime and its implications in Corollary D.1. Finally, taking a union bound over the K analyzed rounds $t = 0, \ldots, K-1$ yields the stated sample-size thresholds and the failure probability $3K\delta$. □

C Proof of Theorem 5.6

Proof. This analysis assumes the initialization lies within a sufficiently small neighborhood of $\boldsymbol{w}^*$, which is standard in local convergence theory. Recall from (10) that we can rewrite the expression in a simplified form:

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| \leq P(\eta, \lambda)\|\boldsymbol{w}^t - \boldsymbol{w}^*\| + Q\|\boldsymbol{w}^t - \boldsymbol{w}^*\|^2, \quad Q := \frac{3M}{2\upsilon} > 0. \tag{27}$$

Choose $\gamma > 0$ such that

$$\gamma \leq \frac{1 - P(\eta, \lambda)}{2Q}, \tag{28}$$

where $P < 1$ by Theorem 5.5. Define the convergence rate constant $\rho_r = P(\eta, \lambda) + Q\gamma$, which is bounded as:

$$\rho_r \leq P(\eta, \lambda) + Q\frac{1 - P(\eta, \lambda)}{2Q} = \frac{P(\eta, \lambda) + 1}{2} < 1. \tag{29}$$

The convergence result holds under the assumption that the initial iterate satisfies $\|\boldsymbol{w}^0 - w^*\| \leq \gamma$, where $\gamma \leq \frac{1-P(\eta,\lambda)}{2Q}$ defines a local convergence region. We prove by induction that

$$\|\boldsymbol{w}^t - w^*\| \leq \gamma\rho_r^t, \forall t \geq 0. \tag{30}$$

When $t = 0$, the inequality holds trivially as $\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| \leq \gamma = \gamma\rho_r^0$. Assume now that $\|\boldsymbol{w}^t - \boldsymbol{w}^*\| \leq \gamma\rho_r^t$ holds for some $t > 0$. Then we have

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| \leq P(\eta, \lambda)\|\boldsymbol{w}^t - \boldsymbol{w}^*\| + Q\|\boldsymbol{w}^t - \boldsymbol{w}^*\|^2 \leq P(\eta, \lambda)\gamma\rho_r^t + Q(\gamma\rho_r^t)^2. \tag{31}$$

Since $\rho_r^t \leq 1$, it follows that $\gamma\rho_r^t \leq \gamma$, and hence

$$Q(\gamma\rho_r^t)^2 = Q\gamma^2\rho_r^{2t} \leq Q\gamma^2\rho_r^t. \tag{32}$$

Substituting yields

$$\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| \leq P(\eta, \lambda)\gamma\rho_r^t + Q\gamma^2\rho_r^t = \gamma\rho_r^t(P(\eta, \lambda) + Q\gamma) = \gamma\rho_r^{t+1}, \tag{33}$$

where the last equality uses the definition $\rho_r = P(\eta, \lambda) + Q\gamma$. This proves (30).

To achieve $\|\boldsymbol{w}^t - w^*\| \leq \varepsilon$, it suffices to ensure

$$\begin{aligned} &\gamma\rho_r^T \leq \varepsilon, \\ &T \log \frac{1}{\rho_r} \geq \log \frac{\gamma}{\varepsilon}, \\ &T \geq \frac{\log(\gamma/\varepsilon)}{\log(1/\rho_r)}, \end{aligned} \tag{34}$$

where the second inequality follows from the fact that $\rho_r < 1$. Since γ and ρ_r are constants independent of ε, the iteration complexity is

$$T = \mathcal{O}(\log \tfrac{1}{\varepsilon}). \tag{35}$$

□

D Local Convergence Under Large Heterogeneity

Theorem 5.5 provides a sufficient parameter feasibility condition ensuring $P(\eta, \lambda) < 1$. When the second-order heterogeneity Γ is large, this feasibility condition may fail to be easily satisfied. Nevertheless, the recursion (27) still implies a standard local basin-of-attraction result: even if $P(\eta, \lambda)$ is close to 1, the iterates contract as long as the initialization lies in a sufficiently small neighborhood of $\boldsymbol{w}^*$, whose radius explicitly depends on $P(\eta, \lambda)$.

Corollary D.1 (Local basin of attraction and contraction radius)**.** . *With Assumptions 5.1, 5.2, consider the update of our proposed method analyzed in Theorem 5.5. On the same event where* (27) *holds, the iterates satisfy the recursion* (27)*. Let $r > 0$ be any radius such that*

$$P(\eta, \lambda) + rQ < 1. \tag{36}$$

Then, for any initialization satisfying $\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| < r$, the following hold:

1. ***(Invariance of the radius)*** $\|\boldsymbol{w}^t - \boldsymbol{w}^*\| < r$ *for all* $t \geq 0$.

2. ***(Linear convergence within the basin)*** *the iterates converge linearly to* $\boldsymbol{w}^*$ *with rate*

$$\|\boldsymbol{w}^t - \boldsymbol{w}^*\| \leq (P(\eta, \lambda) + Qr)^t \, \|\boldsymbol{w}^0 - \boldsymbol{w}^*\| \tag{37}$$

In particular, the largest admissible contraction radius implied by (36) *is*

$$r_{max} := \frac{1 - P(\eta, \lambda)}{Q}, \tag{38}$$

which decreases as $P(\eta, \lambda)$ *increases (e.g., due to larger second-order heterogeneity* Γ*, larger Nyström approximation error* ρ_{Ny}*, or larger first-order dissimilarity constants* ϵ_B *entering P).*

Proof. If $\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| < r$, then $\|\boldsymbol{w}^1 - \boldsymbol{w}^*\| \leq (P(\eta, \lambda) + Qr)\|\boldsymbol{w}^0 - \boldsymbol{w}^*\| < (P(\eta, \lambda) + Qr)r < r$. By induction, $\|\boldsymbol{w}^t - \boldsymbol{w}^*\| \leq r$ for all $t \geq 0$, proving the invariance.

Using (27) we obtain

$$\begin{aligned} \|\boldsymbol{w}^{t+1} - \boldsymbol{w}^*\| &\leq P(\eta, \lambda)\|\boldsymbol{w}^t - \boldsymbol{w}^*\| + Q\|\boldsymbol{w}^t - \boldsymbol{w}^*\|^2 \\ &\leq P(\eta, \lambda)\|\boldsymbol{w}^t - \boldsymbol{w}^*\| + Qr\|\boldsymbol{w}^t - \boldsymbol{w}^*\| = (P(\eta, \lambda) + Qr) \, \|\boldsymbol{w}^t - \boldsymbol{w}^*\|. \end{aligned} \tag{39}$$

Repeatedly applying the above inequality yields (37), proving the convergence within the basin. □

E Experimental Setting

Protocol. All experiments use 30 clients with full participation, 60 communication rounds, one local epoch per round, and batch size 512. Data are partitioned by a Dirichlet distribution over labels with concentration $\alpha \in \{1.0, 0.5\}$ (and 0.1 in the boundary study). For each task, hyperparameters are selected on seed 0 under a fixed tuning protocol—four configurations for FHaN ($\eta \in \{0.004, 0.012\}$, $\lambda \in \{0.005, 0.02\}$) and two for each first-order baseline ($\eta \in \{0.05, 0.1\}$)—and the selected configuration is then run on seeds $\{0, 1, 2\}$ with paired data splits and paired initializations shared across methods. All numbers are mean $\pm$ std over the three seeds. FHaN uses per-block sampling rate 3×10^{-4} with a per-block minimum of 4 sampled columns (1 for the MLP-family models) and smoothing $\lambda_s = 10^{-4}$. Experiments were run on NVIDIA A800 GPUs.

Tasks. Federated fine-tuning tasks freeze an ImageNet-pretrained backbone—ViT-B/16 (768-d features), ResNet-50 (2048-d), or ResNet-18 (512-d)—and train only the head on CIFAR-100 or CIFAR-10: a two-layer MLP (hidden width 256, no bias) or a linear layer. Features are extracted once with the backbone in eval mode (resize-224 bicubic, ImageNet normalization) and shared identically across methods. From-scratch tasks use: TinyViT (patch 4, width 64, depth 2, 4 heads; Fashion-MNIST), CNNFC (the paper's CNN with convolutional layers frozen at initialization, only the three FC blocks train; Fashion-MNIST), and MLPs (784→500→10; 784→500→500→10 as MLP3; Fashion-MNIST and MNIST).

Selected configurations. FHaN: $(\eta, \lambda) = (0.012, 0.005)$ for ViT-B/16 + MLP head at α=1.0 and ResNet-50 at α=1.0; $(0.004, 0.005)$ for all other fine-tuning tasks, TinyViT, CNNFC, and MLP/MLP3 at α=1.0; $(0.004, 0.02)$ for MLP-family models at α=0.5 and the MLP3 stable arm. FedAvg and SCAFFOLD: $\eta = 0.1$ everywhere except TinyViT ($\eta = 0.05$). SCAFFOLD uses the model-drift control-variate update (option I).

F Experimental Supplement

F.1 Results at $\alpha = 0.5$

The $\alpha = 0.5$ picture is consistent with $\alpha = 1.0$: fine-tuning tasks and FC-dominated models keep large margins, while the easiest saturated tasks (MLP on Fashion-MNIST) tilt toward FedAvg on final accuracy with FHaN still faster to target.

Table 4: Final-round test accuracy (%, mean ± std, 3 seeds) at $\alpha = 0.5$; rounds to target per seed in parentheses.

TASK	METHOD	FINAL ACC. (ROUNDS)
VIT-B/16 + MLP HEAD / CIFAR-100	**FHAN**	**78.38±0.13** (4/4/4)
	FEDAVG	72.58±0.80 (17/17/18)
RESNET-50 + MLP HEAD / CIFAR-100	**FHAN**	**68.91±0.07** (16/15/15)
	FEDAVG	57.61±0.35 (–/–/–)
VIT-B/16 + LINEAR / CIFAR-100	**FHAN**	**79.65±0.11** (1/1/1)
	FEDAVG	76.00±0.17 (8/7/8)
TINYVIT / FASHION-MNIST	**FHAN**	**58.78±0.94** (10/11/6)
	FEDAVG	34.46±1.12 (–/–/–)
CNNFC / FASHION-MNIST	**FHAN**	**82.11±1.23** (27/29/21)
	FEDAVG	74.76±1.05 (–/–/–)
MLP / FASHION-MNIST	FHAN	81.02±1.19 (13/14/11)
	FEDAVG	**83.20±0.29** (23/23/20)

F.2 BOUNDARY CASES IN DETAIL

- **End-to-end CNN (Fashion-MNIST).** FHaN trails FedAvg by 4–6 final points across $\alpha \in \{1.0, 0.5\}$. The stability grid shows convolutional blocks require $\lambda \geq 0.02$ for stable training, at which point the curvature correction on the FC blocks is too attenuated to help; a per-block minimum-sampling control (raising conv-block columns from 1 to 8/16) does not close the gap, ruling out conv sampling starvation as the cause.
- **Saturated linear head (CIFAR-10, ResNet-18 features).** FHaN reaches 86.4% early but degrades to 76.5 by round 60 while FedAvg converges stably to 86.3—pure sampling variance with no curvature benefit left at the task ceiling.
- **Benign convex problems.** Multinomial logistic regression on MNIST: FHaN ties FedAvg at moderate targets and trails by 1–2 points at the final round; the spectrum is too concentrated for the correction to matter.
- **Extreme heterogeneity** ($\alpha = 0.1$)**.** FHaN loses across MLP, CNN, and MLR (e.g., 15 vs. 3 rounds to 80% on MLR), consistent with the heterogeneity term in the contraction condition (Theorem 5.5).

F.3 PER-SCENARIO λ INTERACTION

Table 5: Effect of λ by scenario (3 seeds, $\alpha = 1.0$).

SCENARIO	$\lambda = 0.005$	$\lambda = 0.02$	COLLAPSE
CNNFC / FASHION-MNIST	**83.63±0.38**	75.57±1.68	0/6
MLP / FASHION-MNIST	**84.68±0.26**	83.16±0.23	0/6
TINYVIT / FASHION-MNIST	**59.08±0.81**	52.25 (SEED 0)	0/6
MLP3 / MNIST	85.56±11.87	**92.74±0.07**	4/6

Late-round collapse appears only in the MLP3 configuration: all six runs peak at 95.7–96.3% but four of six drop sharply in the final rounds at $\lambda = 0.005$. On CNNFC, raising λ to 0.02 removes the entire advantage (75.6 vs. FedAvg 75.2), showing the gain is carried by the low-λ curvature correction rather than by implementation details.

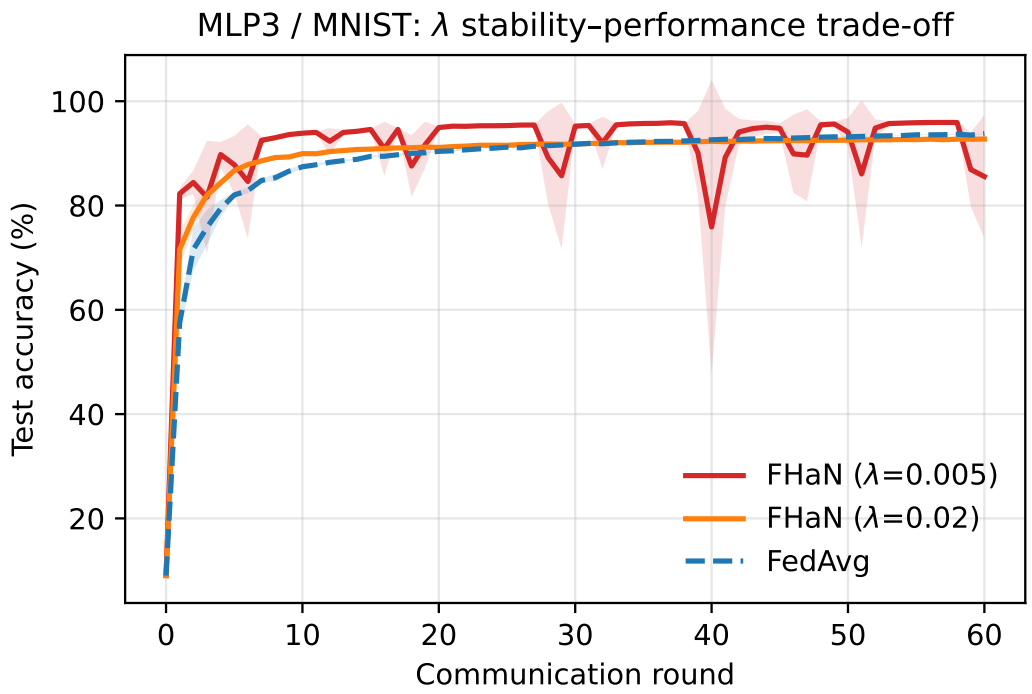


Figure 3: MLP3 / MNIST ($\alpha = 1.0$): the aggressive arm ($\lambda = 0.005$) converges fastest but collapses late on a subset of seeds, while $\lambda = 0.02$ remains stable throughout.