HALT BEFORE DEVIATE! ADAPTIVE LOCAL EFFORT IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The number of local steps is an important control knob in federated learning, shaping both communication cost and the optimization trajectory. However, heterogeneous data can cause local updates to drift and hinder global progress. Choosing when to stop local training is therefore essential to balance communication efficiency and optimization progress. We propose an adaptive stopping rule that selects each client's local step count by comparing both the direction and magnitude of its average update with a reference formed early in the round. Our analysis explains how the rule responds to differences in local curvature and relates local alignment to global descent. Under a common budget of local updates, the proposed rule achieves comparable accuracy with fewer communication rounds.

1 INTRODUCTION

Federated learning (FL; McMahan et al., 2017) collaboratively trains a shared model by repeatedly broadcasting global parameters to decentralized clients, updating them locally, and aggregating the results back at a central server. In each communication round, clients usually perform multiple local updates ($H > 1$) to reduce communication for a given amount of computation. Longer local training allows each client to adapt further to its own data distribution. Under data heterogeneity, the resulting updates may point in different directions, so the choice of H affects both communication cost and the global model learned through aggregation.

A common choice is to fix the number of local steps (or epochs) as a hyperparameter (McMahan et al., 2017; Reddi et al., 2021; Wang et al., 2021). Several other methods adjust it to resource constraints, training progress, or changes in local representations (Wang et al., 2019; Wang & Joshi, 2019; Mills et al., 2023; Shenaj et al., 2025). The resulting schedules can increase or decrease local step counts, depending on the setting and design objective (Wang & Joshi, 2019; Mills et al., 2023; Gu et al., 2024). Early analyses establish when local updates can reduce communication and how data heterogeneity limits this benefit (Stich, 2019; Khaled et al., 2020). More recent results identify further benefits under assumptions on how client gradients and curvatures vary (Patel et al., 2024; Luo et al., 2025). Together, these results show that the value of additional local computation depends on the training setting. The same number of steps need not have the same effect on every client. One client's average update may remain close to its early reference, while the other's turns or shrinks after only a few updates. These behaviors can also change from round to round. A common round-wise schedule cannot respond to both clients separately, and the conditions used in convergence analyses do not directly tell either client when to stop. This leads to the question:

How can a client tell from its local trajectory whether to take another step or to stop?

Each round starts from the current global model, giving every client a fresh starting point for its local trajectory. We allow several base steps to amortize communication and use their average update as a reference for the rest of the round. Additional steps may turn this average away from its early direction or reduce the net movement produced per step. We therefore evaluate both its direction and magnitude relative to the reference, and call the resulting trajectory consistency-based stopping (TCS). In Figure 1, TCS recovers curvature-dependent effort from the observed trajectory. We apply the same criterion to each client, allowing local step counts to differ across clients and rounds. To account for variability in the observed reference, we require the continuation comparison to hold throughout a pre-defined neighborhood around it. Each client continues only while its average up-

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.

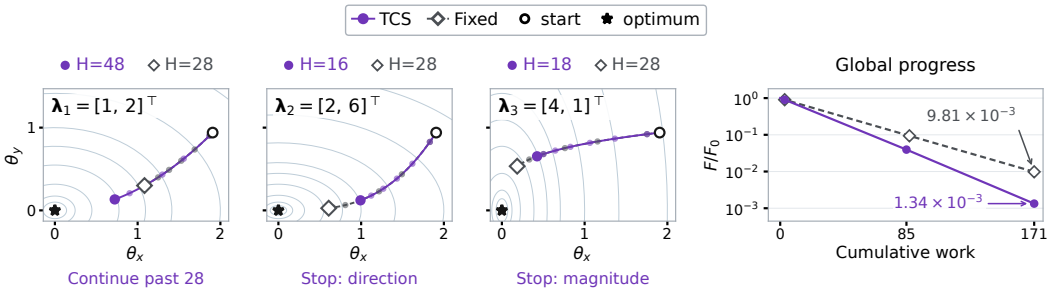

Figure 1: **Local trajectories guide how much each client computes. Left:** Our stopping rule (TCS) continues past fixed steps for client 1 and stops clients 2 and 3 earlier: clients 1 and 2 are stopped due to direction, and client 3 due to magnitude. **Right:** over the first three communications, both methods use exactly 171 total client updates, and TCS reaches $7.34\times$ lower loss than fixed baseline. See Appendix A.1.1 for more details.

date remains closer to every reference in this neighborhood than to zero net movement. This also lets each client decide when to stop from its own observed updates, without additional communication during local training. Our aim is to use local computation where this consistency persists and synchronize when it no longer does.

Our contributions are:

- **Client-wise local step selection.** We propose a rule that evaluates both direction and magnitude of the average local update against a reference formed early in each round, and requires the continuation comparison to hold under all reference perturbations within a prescribed relative radius.
- **Local consistency and global progress.** We relate local consistency to global alignment and separate stopping-induced bias from sampling and local drift, obtaining a nonconvex stationarity bound under partial participation that remains valid although each client's step count is chosen from the same noisy updates it depends on. In deterministic quadratic models, we show how curvature-dependent stopping approximately preserves the relative balance of client reference directions and identify conditions for contraction of the limiting global update.
- **Analysis of computation allocation.** We show how selected step counts vary across clients and training stages, find that most stops would not be reached by magnitude alone, and reduce communication empirically at comparable or higher accuracy than the fixed and adaptive schedules under a common computation budget.

2 Related Work

The selection of local step counts. Table 1 summarizes the adaptive rules compared in our experiments, distinguishing whether counts are shared or client-specific, when they are chosen, and what information guides them. Most listed schedules set a shared count before each round's local training begins. Wang et al. (2019) adapt a shared aggregation interval to a resource budget us-

Table 1: Local step schedules differ in whether counts are shared or client-specific, when they are chosen, and from what signal. Entries describe the variants used in our experiments. Among these methods, only ALT and TCS decide per client during local training; ALT evaluates both local and global encoders, whereas TCS uses quantities already produced by local SGD.

Method	Granularity	Timing	Signal
AdaComm (Wang & Joshi, 2019)	shared	between rounds	training loss, learning rate
Mills (Mills et al., 2023)	shared	between rounds	training loss
QSR (Gu et al., 2024)	shared	between rounds	learning rate
ALT (Shenaj et al., 2025)	per-client	during round	representation difference (activations from local & global models)
TCS (ours)	**per-client**	**during round**	**average local update direction and magnitude**

ing estimates of resource costs and model and data characteristics. ADACOMM adjusts the shared count using training loss and learning rate to balance optimization progress against communication time (Wang & Joshi, 2019). Mills et al. (2023) derive decreasing local-step schedules; we use their training-loss-based variant in our comparisons. QSR increases the count in proportion to the inverse squared learning rate as the learning rate decays (Gu et al., 2024). ALT makes decisions during local training by comparing local and global representations, requiring additional evaluation of the global encoder (Shenaj et al., 2025). TCS uses the average update from base steps as a reference and compares both direction and magnitude during further local training to determine stop timing.

The role of local computation in distributed learning. The benefit of local computation depends on the objective and data assumptions. For a shared smooth, strongly convex objective with bounded gradient second moments, a suitable local step count preserves the asymptotic mini-batch SGD rate (Stich, 2019). Under data heterogeneity, longer local training can move clients toward different local solutions (Khaled et al., 2020) and, in smooth, strongly convex settings with partial participation, slow convergence enough to offset communication savings (Li et al., 2020). Worst-case analyses also show that local SGD can be less effective than mini-batch SGD on heterogeneous objectives (Woodworth et al., 2020; Patel et al., 2024). More recent work identifies conditions under which local computation helps. Qin et al. (2022) favor increasing counts for strongly convex objectives and fixed counts for convex and nonconvex objectives. Luo et al. (2025) identify conditions under which multiple local steps improves over mini-batch SGD, including weak convexity, with further gains from higher-order similarity and smoothness. Patel et al. (2025; 2026) show how similarity in client curvature can improve the convergence under multiple local steps for strongly convex and general convex objectives, respectively. For heterogeneous, separable logistic regression, Crawshaw et al. (2025) show how decreasing curvature allows additional local steps to improve convergence. These results highlight the role of curvature in determining when additional local computation is beneficial. Motivated by this connection, we use the realized local trajectory to guide stopping and analyze how the resulting step selections respond to differences in local curvature.

3 PROPOSED METHOD

We let each client decide whether to take another local step by comparing both the direction and magnitude of its average update with a reference formed early in the round. We then use work-normalized aggregation to combine updates from heterogeneous clients that stop at different steps.

3.1 PRELIMINARIES

Canonical objective. For K clients, each with a dataset $\mathcal{D}_i$ of size $n_i = |\mathcal{D}_i|$, the FL objective is

$$F(\boldsymbol{\theta}) = \sum\nolimits_{i=1}^{K} \frac{n_i}{\sum_{j=1}^{K} n_j} f_i(\boldsymbol{\theta}), \qquad f_i(\boldsymbol{\theta}) = \frac{1}{n_i} \sum\nolimits_{\xi \in \mathcal{D}_i} \ell(\boldsymbol{\theta}; \xi). \tag{1}$$

At communication round $t = 0, \ldots, T-1$, the server samples $\max\{1, \lfloor C \cdot K \rfloor\}$ clients uniformly without replacement to form $\mathcal{S}_t$, where $C \in (0, 1]$. Each selected client $i \in \mathcal{S}_t$ starts from the global model $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{i,t}^{(0)}$ and uses its own minibatches $\mathcal{B}_{i,h}$ of size B for local updates:

$$\boldsymbol{g}_{i,h} = \frac{1}{B} \sum\nolimits_{\xi \in \mathcal{B}_{i,h}} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_{i,t}^{(h-1)}; \xi), \qquad \boldsymbol{\theta}_{i,t}^{(h)} = \boldsymbol{\theta}_{i,t}^{(h-1)} - \eta_t \boldsymbol{g}_{i,h}, \tag{2}$$

where ℓ is a loss function, h indexes local steps, and $\eta_t > 0$ is fixed within the round. We omit t from auxiliary client variables below; H_i denotes the realized local step count in round t.

Base local steps. We start each client with $H_{\min} \geq 1$ base steps. Choosing $H_{\min} > 1$ allows multiple updates per communication; the rule also permits a single base step. We use their average stochastic gradient as a fixed *reference* $\boldsymbol{r}_i$ for the rest of the round:

$$\boldsymbol{d}_{i,h} = \frac{\boldsymbol{\theta}_t - \boldsymbol{\theta}_{i,t}^{(h)}}{\eta_t h} = \frac{1}{h} \sum\nolimits_{s=1}^{h} \boldsymbol{g}_{i,s}, \qquad \boldsymbol{r}_i \triangleq \boldsymbol{d}_{i,H_{\min}}. \tag{3}$$

Algorithm 1 TCS: Trajectory Consistency-based Stopping

Require: communication rounds T, number of clients K, client sampling rate C, batch size B, learning rate $\{\eta_t\}$, robustness parameter $\kappa \in [0, 1)$, base step count $H_{\min} \geq 1$, maximum step count $H_{\max} \geq H_{\min}$.

Randomly initialize a global model $\boldsymbol{\theta}_0$.
for $t = 0, \ldots, T-1$ **do**
  Sample clients $\mathcal{S}_t$ uniformly without replacement, $|\mathcal{S}_t| = \max\{1, \lfloor C \cdot K \rfloor\}$
  Broadcast $\boldsymbol{\theta}_t$, η_t, and κ
  for each $i \in \mathcal{S}_t$ in parallel **do**
    Set $\boldsymbol{\theta}_{i,t}^{(0)} \leftarrow \boldsymbol{\theta}_t$ and run $H_{\min}$ SGD steps via (2)
    $h \leftarrow H_{\min}$; form $\boldsymbol{d}_{i,h}$ via (3) and set $\boldsymbol{r}_i \leftarrow \boldsymbol{d}_{i,h}$
    if $t > 0$ **then** ▷ The first round uses only base steps
      while $h < H_{\max}$ and $\psi_i(\boldsymbol{d}_{i,h}) > 0$ **do**
        $h \leftarrow h + 1$; take a single step and update $\boldsymbol{d}_{i,h}$
    $H_i \leftarrow h$ and return $(\boldsymbol{\Delta}_{i,t} \equiv (\boldsymbol{\theta}_{i,t}^{(H_i)} - \boldsymbol{\theta}_t),\ H_i)$
  Update $\boldsymbol{\theta}_{t+1}$ via (7)
return $\boldsymbol{\theta}_T$

3.2 Local step selection through trajectory consistency

Direction and magnitude along the local trajectory. Within each round, clients start from the same global model but can move differently because they train on different data. The base steps establish each client's own reference $\boldsymbol{r}_i$ for that round. We then assess whether the average update $\boldsymbol{d}_{i,h}$ remains *consistent* with this early movement by comparing its direction and magnitude with those of the reference. Up to a fixed learning rate, $\|\boldsymbol{d}_{i,h}\|_2$ measures this net movement per step. Note that extra steps may turn the average away from the reference or reduce its magnitude, even if its direction stays the same. Thus, we assess both changes together as *local trajectory consistency*.

A robust comparison with the reference. In a word, we propose that local updates should continue while the average update stays close to the initial reference in both *direction* and *magnitude*.

To implement this principle, we first define the *reference* as the movement observed during the base steps and *zero* to represent no net movement from the starting point (i.e., current global model). The reference $\boldsymbol{r}_i$ summarizes the movement observed during the base steps and reflects both the evolving local trajectory and minibatch variability. Our criterion permits further local steps thereafter only if the current average remains closer to the reference than to zero. However, the reference itself depends on the sampled minibatches, which have sampling noises. To account for the variation therein, we use a Euclidean ball as a simple model of representing closeness to the reference in any direction:

$$\mathcal{U}_i = \{\boldsymbol{u} : \|\boldsymbol{u} - \boldsymbol{r}_i\|_2 \leq \kappa \|\boldsymbol{r}_i\|_2\}, \qquad 0 \leq \kappa < 1.$$

The parameter κ specifies the perturbation radius relative to the observed reference magnitude. We permit another local step only when the current average update is closer to every reference in this neighborhood than to zero. A larger κ requires the continuation condition to hold over a wider perturbation set, making the stopping rule more conservative.

Next, as a criterion for extra steps, we allow further updates *only if* the current average update is closer to every other reference in $\mathcal{U}_i$ than to zero:

$$\max_{\boldsymbol{u} \in \mathcal{U}_i} \|\boldsymbol{d}_{i,h} - \boldsymbol{u}\|_2 < \|\boldsymbol{d}_{i,h} - \boldsymbol{0}\|_2, \tag{4}$$

motivated by robust optimization (Ben-Tal & Nemirovski, 1998).

From the comparison to a stopping rule. At a given local step, the distance to zero ($\|\boldsymbol{d}_{i,h}\|_2$) does not depend on $\boldsymbol{u}$. Thus, the left hand side of Eq. (4) for any direction $\boldsymbol{d}$ directly gives

$$\max_{\boldsymbol{u} \in \mathcal{U}_i} \|\boldsymbol{d} - \boldsymbol{u}\|_2 = \|\boldsymbol{d} - \boldsymbol{r}_i\|_2 + \kappa \|\boldsymbol{r}_i\|_2.$$

With the robust comparison using criterion in Eq. (4), our stopping rule simply reduces to:

$$\underbrace{\|\boldsymbol{d}_{i,h} - \boldsymbol{r}_i\|_2 + \kappa \|\boldsymbol{r}_i\|_2}_{\text{distance to the farthest reference}} < \underbrace{\|\boldsymbol{d}_{i,h} - \boldsymbol{0}\|_2}_{\text{distance to zero}}.$$

Rearranging, we express the criterion as $\psi_i(\boldsymbol{d})$, defined for any average update $\boldsymbol{d}$ as follows:

$$\psi_i(\boldsymbol{d}) \triangleq \|\boldsymbol{d}\|_2 - \|\boldsymbol{d} - \boldsymbol{r}_i\|_2 - \kappa\|\boldsymbol{r}_i\|_2. \tag{5}$$

Finally, client i should simply stop local updates under $h \in [H_{\min}, H_{\max}]$ in rounds $t \geq 1$, when if $\psi_i(\boldsymbol{d}_{i,h}) \leq 0$. Note that a positive value permits another step, and a nonpositive value means the current average no longer meets the comparison. Since increasing κ lowers this value, the same trajectory with the same cap is expected to stop at the same step or earlier.

How direction and magnitude interact. The stopping rule also has a useful cosine-similarity interpretation. For nonzero reference $\boldsymbol{r}_i$ and direction $\boldsymbol{d}$, the continuation condition $\psi_i(\boldsymbol{d}) > 0$ can be rewritten as (see Proposition A.3 for the derivation):

$$\frac{\langle \boldsymbol{r}_i, \boldsymbol{d} \rangle}{\|\boldsymbol{r}_i\|_2 \|\boldsymbol{d}\|_2} > \kappa + \frac{1-\kappa^2}{2} \frac{\|\boldsymbol{r}_i\|_2}{\|\boldsymbol{d}\|_2}. \tag{6}$$

As the average shrinks relative to the reference, it must align more closely to continue. Even perfect alignment passes only when $\boldsymbol{d} = c\boldsymbol{r}_i$ with $c > (1+\kappa)/2$. Thus, even if there is no drift in the direction, the rule treats excessive decay in per step progress as a loss of local consistency.

3.3 Work-normalized aggregation

Since a model change accumulates over all local steps, a direct model averaging gives more weight to average directions from clients with longer updates. It has been studied that this can effectively shift the global objective in Eq. (1), we follow the work-normalization principle and divide each model change by the realized step count before averaging directions (Wang et al., 2020). After client i returns $\boldsymbol{\Delta}_i = \boldsymbol{\theta}_{i,t}^{(H_i)} - \boldsymbol{\theta}_t$ and H_i, the server computes and aggregates:

$$\begin{aligned} p_i = \frac{n_i}{\sum_{j \in \mathcal{S}_t} n_j}, \boldsymbol{d}_{i,H_i} = -\frac{\boldsymbol{\Delta}_i}{\eta_t H_i}, \bar{H}_t = \sum_{i \in \mathcal{S}_t} p_i H_i, \\ \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \bar{H}_t \sum_{i \in \mathcal{S}_t} p_i \boldsymbol{d}_{i,H_i}. \end{aligned} \tag{7}$$

Note that equal local step counts recover sample-weighted model averaging over the selected clients. Also, this normalization does not by itself guarantee an unbiased update for the population objective under partial participation and adaptive stopping; the remaining sampling, local-drift, and stopping effects are retained in the bias decomposition is deferred in Eq. (23).

We present our proposed method, TCS, in Algorithm 1.

4 Theoretical Analysis

We analyze how curvature affects local updates, when local consistency leads to a decrease in global loss, and how the stopping rule affects convergence. All proofs are deferred to Appendix A.

4.1 How κ controls curvature imbalance

Curvature affects local stopping through the observed average update. For anisotropic quadratics with exact gradients and $0 \prec \eta_t A_i \prec I$, higher-curvature eigendirections decay faster during local training, so the average update can rotate while its magnitude shrinks. TCS responds to these changes through the direction and magnitude of $\boldsymbol{d}_{i,h}$, without estimating a Hessian. Refer to Figure 1 and Appendix A.1.1 for a concrete example.

To isolate the role of κ, consider the isotropic client quadratic

$$f_i(\boldsymbol{\theta}) = \frac{\lambda_i}{2}\|\boldsymbol{\theta} - \boldsymbol{a}_i\|_2^2, \qquad \lambda_i > 0, \qquad 0 < \eta_t \lambda_i < 1. \tag{8}$$

Here, the average update keeps its direction, so stopping depends on its magnitude relative to the reference. After the base-only first round, a client with exact, initially nonzero gradients that stops

at the rule before any cap truncates training retains approximately a fraction $\rho = (1+\kappa)/2$ of its reference magnitude. Thus, even without directional drift, a sufficient decrease in average-update magnitude triggers stopping, i.e., larger κ requires more magnitude to be retained. If all clients stop this way, it suffices $\sum_i p_i \boldsymbol{d}_{i,H_i} \approx \rho \sum_i p_i \boldsymbol{r}_i$, where each update is attenuated by approximately the same factor relative to its own reference. Appendix A.2 bounds the error from the final step.

Proposition 4.1 (Tolerable curvature imbalance)**.** *For the isotropic model* (8)*, assume full participation, exact gradients, fixed $H_{\min}$, and $0 \le \kappa < 1$. Consider a fixed starting point with nonzero local gradients. Let $H_{\max}$ grow as $\eta_t \to 0$ so that the stopping rule, rather than a step or work cap, determines the local counts, as specified in Assumption A.1. Then $H_i/H_j \to \lambda_j/\lambda_i$, and the limiting work-normalized map* (7) *contracts toward the global minimizer without oscillation if $J_\lambda \le 1/(1-\kappa)$, where $J_\lambda = (\sum_i p_i \lambda_i)(\sum_i p_i/\lambda_i)$.*

In this limit, it implies that clients with larger curvature take fewer steps because their gradients change more quickly. A larger κ stops a fixed local trajectory no later and enlarges the range of curvature imbalance covered by this sufficient condition. For example, $\kappa = 0.8$ permits $J_\lambda \le 5$. That said, a smaller κ permits more steps but narrows the range covered by the condition.

4.2 From local consistency to global progress

Assumption 4.1 (Smoothness)**.** *Each f_i has an L-Lipschitz gradient, $F \ge F_{\inf}$, and $\mathbb{E}\|\boldsymbol{\theta}_0\|_2^2 < \infty$.*

We now ask when updates that follow their own references can reduce the global loss. What matters is whether their aggregate aligns with the global gradient. We bound this alignment from below using local alignment, differences across clients, and the accuracy of their average reference.

Proposition 4.2 (Local alignment and global descent)**.** *For clients in $\mathcal{S}_t$, let $\boldsymbol{d}_i = \boldsymbol{d}_{i,H_i}$, $\boldsymbol{D}_t = \sum_i p_i \boldsymbol{d}_i$, and $\bar{\boldsymbol{r}} = \sum_i p_i \boldsymbol{r}_i$. All sums in this proposition are over $\mathcal{S}_t$, with $p_i \ge 0$ and $\sum_i p_i = 1$. Define $V_r = \sum_i p_i \|\boldsymbol{r}_i - \bar{\boldsymbol{r}}\|_2^2$ and $V_d = \sum_i p_i \|\boldsymbol{d}_i - \boldsymbol{D}_t\|_2^2$. Assumption 4.1 gives*

$$\underbrace{F(\boldsymbol{\theta}_{t+1}) - F(\boldsymbol{\theta}_t)}_{\textit{global loss change}} \le -\eta_t \bar{H}_t \underbrace{\langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{D}_t \rangle}_{\textit{global alignment}} + \frac{L \eta_t^2 \bar{H}_t^2}{2} \|\boldsymbol{D}_t\|_2^2. \tag{9}$$

The global alignment satisfies

$$\langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{D}_t \rangle \ge \underbrace{\sum_i p_i \langle \boldsymbol{r}_i, \boldsymbol{d}_i \rangle}_{\textit{local alignment}} - \underbrace{\sqrt{V_r V_d}}_{\textit{client disagreement}} - \underbrace{\|\nabla F(\boldsymbol{\theta}_t) - \bar{\boldsymbol{r}}\|_2 \|\boldsymbol{D}_t\|_2}_{\textit{reference error}}. \tag{10}$$

The local-alignment term in (10) measures agreement with each client's own reference; the other terms account for client disagreement and error in the average reference. Combined together, these bounds show that local alignment yields global descent when it outweighs client disagreement, reference error, and the smoothness penalty from the server step. See Lemma A.4 that bounds the possible alignment loss from retaining the first step that fails the test.

4.3 Nonconvex convergence with partial participation

We now consider noisy gradients and partial participation. The step count depends on the updates observed so far, which can bias the returned average (Shin et al., 2021). Even when the noise sums to zero in expectation, dividing it by a count chosen from the same observations need not preserve that mean (Lemma A.5). We account for this dependence in both the returned directions and the server scale $\bar{H}_t$. We use three additional assumptions to describe gradient noise and differences across clients.

Assumption 4.2 (Unbiased gradients)**.** *Conditional on the round history, selected clients, and observations preceding local step h, $\boldsymbol{g}_{i,h}$ has mean $\nabla f_i(\boldsymbol{\theta}_{i,t}^{(h-1)})$.*

Assumption 4.3 (Bounded variance)**.** *Under the same conditioning, the mean squared gradient error is at most σ^2.*

Assumption 4.4 (Bounded dissimilarity)**.** *For some $\gamma_{\mathrm{dis}} \ge 1, \zeta \ge 0$ and every $\boldsymbol{\theta}$, summing over the full client population gives $\sum_i \frac{n_i}{\sum_j n_j} \|\nabla f_i(\boldsymbol{\theta})\|_2^2 \le \gamma_{\mathrm{dis}}^2 \|\nabla F(\boldsymbol{\theta})\|_2^2 + \zeta^2$.*

Theorem 4.3 (Nonconvex convergence bound)**.** *Under Assumptions 4.1–4.4, let $\eta_t > 0$ be deterministic with $L\eta_t H_{\max} \le 1$ and $H_{\min} \le H_i \le H_{\max}$. Let $\mathbb{E}_t$ denote expectation conditional on the history before round t, including randomness in client selection and local training. Consider the scaled direction $\bar{H}_t \boldsymbol{D}_t/(\mathbb{E}_t \bar{H}_t)$. Let $\boldsymbol{\beta}_t$ be its conditional mean minus $\nabla F(\boldsymbol{\theta}_t)$, and V_t its conditional mean squared distance from that mean. Then, for $\Delta = \mathbb{E}[F(\boldsymbol{\theta}_0)] - F_{\inf}$,*

$$\frac{\sum_{t<T} \eta_t \mathbb{E}\|\nabla F(\boldsymbol{\theta}_t)\|_2^2}{\sum_{t<T} \eta_t} \le c_0 \frac{\Delta/H_{\min} + \sum_{t<T} \eta_t \mathbb{E}[\|\boldsymbol{\beta}_t\|_2^2 + L\eta_t H_{\max} V_t]}{\sum_{t<T} \eta_t}, \tag{11}$$

where $c_0 = \max\{2, H_{\max}/H_{\min}\}$.

The bound separates the initial loss gap, bias, and random variation in the server update. We further decompose the bias into effects of client sampling, local gradient drift, and adaptive stopping in Appendix A.4.2. The stopping term captures the dependence between $\bar{H}_t/H_i$ and each client's accumulated gradient noise, and the bound implies stationarity when $\sum_t \eta_t = \infty$ and the weighted bias and variance contributions on its right-hand side vanish. Still, Assumptions 4.1–4.4 alone do not establish that these contributions vanish.

5 Experimental Results

Our evaluation asks whether trajectory-based stopping preserves test accuracy while requiring fewer communications. Every method receives the same total budget of local minibatch updates and methods differ in how they distribute these updates across clients and communication rounds.

5.1 Setup

We use FEMNIST (Caldas et al., 2018), CIFAR-100 (Krizhevsky, 2009), and StackOverflow (The TensorFlow Federated Authors, 2019) FL benchmark datasets. FEMNIST retains $3{,}597$ writer-defined clients, CIFAR-100 assigns its $50{,}000$ training images to 500 clients using a pathological non-IID split of label distributions McMahan et al. (2017) with a configured minimum of 20 classes per client, and StackOverflow retains $342{,}477$ user-corresponded clients. For each dataset, the server samples 10 clients uniformly without replacement every communication round.

We count one local minibatch optimizer update on one client as a single unit of work. That said, the cumulative total work through T rounds is given by $W_T = \sum_{t=0}^{T-1} \sum_{i \in \mathcal{S}_t} H_{i,t}$. We set every method to receive a budget of $W_{\max} = 100{,}000$ on each dataset. We terminate FL procedure when the remaining budget cannot support the required local computation. Following the convention of FL (McMahan et al., 2017), we choose each dataset's batch size so that 10 steps process approximately one epoch, i.e., client's training-set size on average. In the main comparison, we allow fixed baselines to have among $\{10, 20, 50, 100\}$ and adaptive baselines to assign local steps within $[10, 100]$. Otherwise, sample statistics, model architectures, hyperparameters, and method descriptions are specified in Appendix B.

We compare TCS with Fixed, which uses a fixed local count; Mills, which uses a training loss for shared decreasing local-step schedule (Mills et al., 2023); AdaComm, which adjusts a shared count using training loss and learning rate (Wang & Joshi, 2019); QSR, which increases the shared steps as the learning rate decreases (Gu et al., 2024); and ALT, which stops a client-wise update using local–global representation similarity (Shenaj et al., 2025).

5.2 Accuracy and communication under a common work budget

We compare final accuracy and communication under a common budget of $100{,}000$ client updates in Table 2. Note that we report means and sample standard deviation over three paired seeds, and the colored bars show mean communication rounds on a shared scale. We find that in most of cases, frequent communication with a fixed step achieves the best accuracy. Still, at the same computation budget, TCS's mean accuracy stays close to or above Fixed-10 baseline while requiring fewer communications on all three datasets. We highlight $\kappa = 0.7$ for TCS across all three datasets. On FEMNIST, it reaches 83.35% accuracy, comparable to Fixed-10, while using 24.57% fewer rounds. On CIFAR-100, it reaches 26.49%, improving mean accuracy by 1.29 percent points with

Table 2: **Accuracy and communication under a common budget of** $100{,}000$ **client updates.** Accuracy (%) is mean $\pm$ sample standard deviation over three paired seeds. Rounds are means, with cell bars on a shared linear scale from 0 to 1,000 (shorter is better). Each TCS setting uses the same κ across datasets.

	FEMNIST		CIFAR-100		StackOverflow	
Method	Acc. (%) ↑	Rounds ↓	Acc. (%) ↑	Rounds ↓	Acc. (%) ↑	Rounds ↓
FIXED-10	$\mathbf{83.67} \pm 0.32$	1000	$\mathbf{25.20} \pm 1.38$	1000	$\mathbf{18.05} \pm 0.23$	1000
FIXED-20	81.10 ± 1.40	500	23.21 ± 2.41	500	17.85 ± 0.07	500
FIXED-50	75.30 ± 0.99	200	19.40 ± 1.15	200	17.62 ± 0.11	200
FIXED-100	69.76 ± 2.91	100	16.40 ± 0.62	100	17.41 ± 0.40	100
MILLS	69.65 ± 2.14	103	17.01 ± 1.05	101	17.37 ± 0.31	102.7
ADACOMM	72.10 ± 2.77	123.3	17.11 ± 0.50	104	17.50 ± 0.05	117
QSR	80.47 ± 0.37	753	22.63 ± 1.17	753	17.78 ± 0.45	753
ALT	69.42 ± 1.93	102	22.94 ± 0.44	522.3	17.28 ± 0.07	101
TCS ($\kappa = .7$)	$\mathbf{83.35} \pm 0.92$	754.3	$\mathbf{26.49} \pm 0.98$	772.3	$\mathbf{18.04} \pm 0.07$	638.7

22.77% fewer rounds. On StackOverflow, it reaches 18.04%, staying almost same with the best case with 36.13% fewer rounds. While the other adaptive schedules assigns different local steps either round-wise or client-wise, they couldn't retain even comparable accuracy. Especially both MILLS and ADACOMM become similar with the fixed baseline with the most local steps. Interestingly, although QSR prescribes a fairly close synchronization interval in two vision datasets, the average accuracy is better in TCS. We attribute this improvement to the granular step assignment at the client level, which QSR cannot implement. ALT assigns client-wise local steps like TCS, but it is not effective in striking balance between accuracy and communication.

5.3 HOW HETEROGENEITY AFFECTS LOCAL COMPUTATION

We examine TCS on MNIST dataset, simulating 10 clients randomly participating out of 100 clients pool, with Dirichlet label partitions at $\alpha \in \{0.1, 10.0\}$, representing stronger and weaker label heterogeneity, respectively (Hsu et al., 2019). We fix $\kappa = 0.5$ and use Figure 2 to examine both the selected local steps and the signals that trigger stopping. Note that, in Appendix B.5, we also report the corresponding accuracy and communication results and comparisons with other schedules. Under the common work budget, TCS uses 36.9% and 27.4% fewer rounds than FIXED-10 at $\alpha = 0.1$ and 10.0, respectively, with mean accuracy differing by at most 0.26 percentage points.

The top panels show within-round medians and interquartile ranges, averaged over work bins and three seeds. TCS selects more local steps at $\alpha = 0.1$ than at $\alpha = 10.0$, with a larger gap later in training. We observe that the local step counts rise around the prescribed learning rate decay timing, and the shaded bands widen, indicating greater variation among clients within a round. Thus, the rule adapts computation across both clients and training phases rather than assigning a shared count.

The bottom panels distinguish stopping decisions for which a change in direction is necessary from those for which magnitude alone is sufficient. At $\kappa = 0.5$, magnitude alone permits continuation when $\|\boldsymbol{d}_{i,h}\|/\|\boldsymbol{r}_i\| > 0.75$. Among stops triggered by the rule, this condition still holds in about 53–65% of cases across training phases at $\alpha = 0.1$, and 95–99% at $\alpha = 10.0$. The direction comparison thus contributes in both partitions, while magnitude alone suffices more often at $\alpha = 0.1$ than at $\alpha = 10.0$. The higher share of direction-dependent stops at weak heterogeneity indicates that update magnitude alone less likely explains stopping, but directional information is more often required. Note that this diagnostic identifies which information is necessary for the observed stopping decisions, and it does not by itself establish the performance advantage of the combined criterion.

6 DISCUSSION

TCS uses local trajectory consistency to guide the allocation of computation between synchronizations. By comparing each client's average update with its own early-round reference, the rule re-

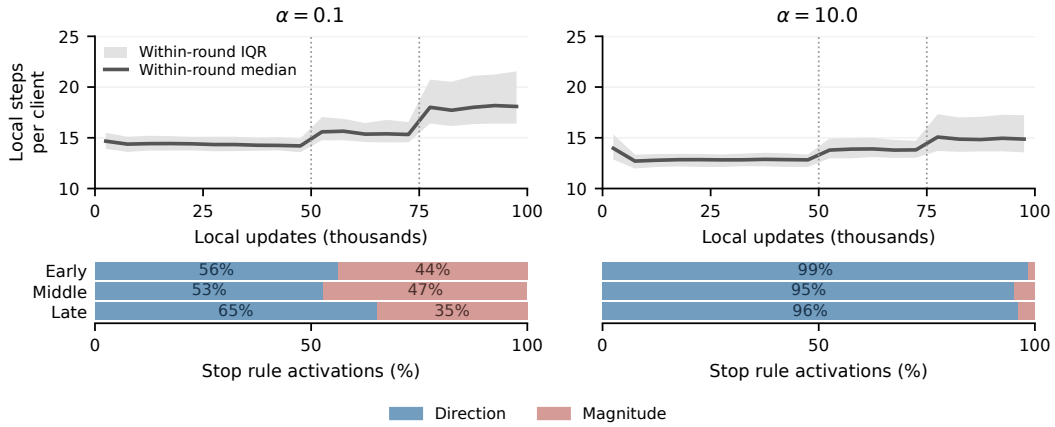

Figure 2: **Local steps and stopping signals change during training.** TCS with $\kappa = 0.5$. **Top**: within-round medians and quartiles, averaged in work bins over three seeds; dotted lines mark learning rate decays. **Bottom**: *Direction* means magnitude alone would permit continuation; *Magnitude* means magnitude alone suffices to stop. See Appendix B.5 for details.

sponds separately as local trajectories evolve across clients and rounds. This decision uses quantities available from local SGD, without additional model evaluations or communication during local training. The same principle suggests several ways to adapt local computation more closely to the training conditions.

One direction is to adapt how the reference is formed and how its uncertainty is represented. Building on the fixed base length used in our experiments, estimates of noise and update variation could guide the choice of $H_{\min}$ across rounds. Such estimates could also inform the radius of the reference neighborhood, providing a data-dependent way to choose κ. In the current method, $H_{\min}$ determines the trajectory segment used to form the reference, while κ controls how conservatively subsequent updates are compared with that reference. Our MNIST sweep examines the effect of κ at a fixed $H_{\min} = 10$. Characterizing their joint sensitivity across noise levels and tasks, and calibrating the reference neighborhood using estimates of update variability, are directions for future work. Together, these extensions would allow both the reference and the stopping threshold to reflect the observed local trajectory.

These algorithmic extensions can also be combined with practical resource constraints. Under a common update budget, our experiments show that TCS can reduce communication relative to short fixed local training while retaining similar mean accuracy. Hence, including estimates of client execution time and communication cost into the local step cap would make it possible to study the same principle under walltime budgets. Integrating with personalization would further clarify how the current client-wise computation allocation benefits clients with different data and resources.

7 Conclusion

We proposed TCS, which selects each client's local step count by comparing the direction and magnitude of its average update with an early update direction as a reference. A robust distance comparison yields a stopping rule that uses the local update trajectory without extra communication overhead during local training, and work-normalized aggregation removes the explicit local step count multiplier from each selected clients average update, preventing longer local trajectories from receiving proportionally larger mixing weights solely because they execute more steps. Our analysis characterizes curvature-dependent stopping in deterministic quadratic models, gives a sufficient condition linking local alignment to global descent, and accounts for stopping-induced bias in a nonconvex convergence bound. Under a shared total computation budget, TCS reduces communication rounds while being comparable to the best test performances across three large-scale FL benchmarks. These results support local trajectory consistency as a practical signal for allocating computation between synchronizations in collaborative machine learning systems.

AI USE STATEMENT

We used large language models for drafting and editing the manuscript, figure plotting by code generation, document formatting, proof checks, adjacent works search, and drafting experimental codes into reproducibility details in Appendix. The authors carefully audited the final claims, proofs, code, experimental results, and references with responsibility.

REPRODUCIBILITY STATEMENT

We specify the local stopping rule and aggregation procedure in Section 3 and Algorithm 1. We provide Appendix A for the assumptions and proofs, and Appendix B for the all detailed experimental settings, baseline implementations, random seeds, and reporting procedures. The anonymized code used for generating empirical results is also provided in the supplementary material.

REFERENCES

Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998. doi: 10.1287/moor.23.4.769. URL https://doi.org/10.1287/moor.23.4.769.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018. URL https://arxiv.org/abs/1812.01097.

Michael Crawshaw, Blake Woodworth, and Mingrui Liu. Local steps speed up local GD for heterogeneous distributed logistic regression. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e20533d29fe52d49a4b1b0f66a49ca5b-Paper-Conference.pdf.

Xinran Gu, Kaifeng Lyu, Sanjeev Arora, Jingzhao Zhang, and Longbo Huang. A quadratic synchronization rule for distributed deep learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.14423.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4519–4529, 2020. URL https://proceedings.mlr.press/v108/bayoumi20a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1907.02189.

Ruichen Luo, Sebastian U. Stich, Samuel Horváth, and Martin Takáč. Revisiting LocalSGD and SCAFFOLD: Improved rates and missing analysis. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2539–2547, 2025. URL https://proceedings.mlr.press/v258/luo25c.html.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 1273–1282, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Jed Mills, Jia Hu, and Geyong Min. Faster federated learning with decaying number of local sgd steps. *arXiv preprint arXiv:2305.09628*, 2023. URL https://arxiv.org/abs/2305.09628.

Kumar Kshitij Patel, Margalit Glasgow, Ali Zindari, Lingxiao Wang, Sebastian U. Stich, Ziheng Cheng, Nirmit Joshi, and Nathan Srebro. The limits and potentials of local SGD for distributed heterogeneous learning with intermittent communication. In *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 4115–4157, 2024. URL https://proceedings.mlr.press/v247/patel24a.html.

Kumar Kshitij Patel, Ali Zindari, Sebastian U. Stich, and Lingxiao Wang. Revisiting consensus error: A fine-grained analysis of local SGD under second-order data heterogeneity. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1959. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/54cee855c7eb9429b4b3668d06bf670f-Abstract-Conference.html.

Kumar Kshitij Patel, Rustem Islamov, Sebastian U. Stich, Aurelien Lucchi, Eduard Gorbunov, and Lingxiao Wang. What's in a smoothness constant? tighter rates for local SGD with bounded second-order heterogeneity. *arXiv preprint arXiv:2607.14731*, 2026. doi: 10.48550/arXiv.2607.14731. URL https://arxiv.org/abs/2607.14731.

Tiancheng Qin, S Rasoul Etesami, and César A Uribe. The role of local steps in local sgd. *arXiv preprint arXiv:2203.06798*, 2022. URL https://arxiv.org/abs/2203.06798.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2003.00295.

Donald Shenaj, Eugene Belilovsky, and Pietro Zanuttigh. Adaptive local training in federated learning. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, pp. 1907–1911. IEEE, 2025. doi: 10.23919/EUSIPCO63237.2025.11226038. URL https://eurasip.org/Proceedings/Eusipco/Eusipco2025/pdfs/0001907.pdf.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. On the bias, risk and consistency of sample means in multi-armed bandits. *arXiv preprint arXiv:1902.00746v3*, 2021. URL https://arxiv.org/abs/1902.00746v3.

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1805.09767.

The TensorFlow Federated Authors. TensorFlow Federated Stack Overflow dataset, 2019. URL https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data.

Jianyu Wang and Gauri Joshi. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd. In *Proceedings of Machine Learning and Systems*, volume 1, 2019. URL https://proceedings.mlsys.org/paper_files/paper/2019/hash/4a0151b47bd93c5de2a0b57831981a0d-Abstract.html.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2007.07481.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021. URL https://arxiv.org/abs/2107.06917.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 2019. URL https://arxiv.org/abs/1804.05271.

Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local SGD for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper/2020/hash/45713f6ff2041d3fdfae927b82488db8-Abstract.html.

A PROOFS AND SUPPORTING ANALYSIS

We use the notation and assumptions of Sections 3–4. All client sums use the selected-set weights p_i; under full participation, $p_i = n_i / \sum_j n_j$. The returned counts satisfy $H_{\min} \le H_i \le H_{\max}$, including the base-only first round and any budget truncation decided before local training.

A.1 CURVATURE-DEPENDENT STOPPING

A.1.1 ANISOTROPIC QUADRATIC EXAMPLE (CF. FIGURE 1)

For $f_i(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{a}_i)^\top A_i(\boldsymbol{\theta} - \boldsymbol{a}_i)$ with $0 \prec \eta_t A_i \prec I$, put $R_i = I - \eta_t A_i$. Exact gradient descent gives

$$\boldsymbol{d}_{i,h} = \frac{I - R_i^h}{\eta_t h}(\boldsymbol{\theta}_t - \boldsymbol{a}_i). \tag{12}$$

Different eigenvalues give different attenuation rates, so the average update can change both direction and magnitude. Figure 1 uses three equally weighted clients with $\boldsymbol{a}_i = 0$ and

$$A_1 = \mathrm{diag}(1, 2), \quad A_2 = \mathrm{diag}(2, 6), \quad A_3 = \mathrm{diag}(4, 1).$$

We set $\boldsymbol{\theta}_0 = (2, 1)^\top$, $\eta_t = 0.02$, $\kappa = 0.1$, $H_{\min} = 1$, and $H_{\max} = 100$.

Both methods take one base-only round, then use TCS or Fixed-28 with aggregation (7). In the first adaptive round, TCS returns $(48, 16, 18)$ local steps. The first two clients retain magnitude ratios 0.5574 and 0.5633, above the aligned threshold 0.55, but fail the cosine condition (6). The third has ratio 0.5455 and also fails the magnitude threshold. The loss panel counts all executed updates, including the shared first round.

A.1.2 STOPPING THRESHOLDS FOR ISOTROPIC QUADRATIC EXAMPLE

For the model (8), assume exact gradients, $0 < \eta_t \lambda_i < 1$, and nonzero initial local gradients in a round after the base-only first round. Write $\tau_i = 1 - \eta_t \lambda_i$, $\rho = (1+\kappa)/2$, and $q_h(\tau) = h^{-1} \sum_{s=0}^{h-1} \tau^s$.

Proposition A.1 (Relative attenuation)**.** *The average update satisfies*

$$\boldsymbol{d}_{i,h} = q_h(\tau_i) \nabla f_i(\boldsymbol{\theta}_t) = \phi_i(h) \boldsymbol{r}_i, \qquad \phi_i(h) = \frac{H_{\min}(1 - \tau_i^h)}{h(1 - \tau_i^{H_{\min}})}. \tag{13}$$

For $h \ge H_{\min}$, $\phi_i(h)$ decreases strictly from one to zero. TCS *continues exactly when $\phi_i(h) > \rho$; its uncapped stopping count is $N_i = \min\{h \ge H_{\min} : \phi_i(h) \le \rho\}$.*

Proof. The local gradient at step $s+1$ equals $\tau_i^s \nabla f_i(\boldsymbol{\theta}_t)$; averaging the geometric sum proves (13). Since $\tau_i^h < q_h(\tau_i)$, $q_{h+1} = (h q_h + \tau_i^h)/(h+1) < q_h$, and $q_h \to 0$. Substituting $\boldsymbol{d}_{i,h} = \phi_i(h) \boldsymbol{r}_i$ into the test gives $\psi_i = 2(\phi_i(h) - \rho)\|\boldsymbol{r}_i\|_2$. □

A cosine-only test with threshold below one continues to the cap in this model, since the update direction remains unchanged.

Corollary A.2 (Common attenuation up to the crossing step)**.** *If each client reaches its first crossing before any cap truncates it, then $H_i = N_i$ and*

$$\rho(1 - 1/H_i) < \phi_i(H_i) \le \rho, \qquad \left\| \sum_i p_i \boldsymbol{d}_{i,H_i} - \rho \sum_i p_i \boldsymbol{r}_i \right\| \le \rho \sum_i \frac{p_i \|\boldsymbol{r}_i\|_2}{H_i}. \tag{14}$$

Proof. The previous prefix has $\phi_i(H_i - 1) > \rho$. The positive last gradient component gives $\phi_i(H_i) > (1 - 1/H_i)\phi_i(H_i - 1) > (1 - 1/H_i)\rho$. The upper bound is the stopping condition. Sum the resulting errors using the triangle inequality. □

A.1.3 Proof of Proposition 4.1

Assumption A.1 (Nonbinding caps in the quadratic limit)**.** *Use the isotropic model above with full participation, fixed $H_{\min}$, $0 \le \kappa < 1$, and a fixed starting point with nonzero local gradients. Let $x_\kappa > 0$ solve $(1 - e^{-x_\kappa})/x_\kappa = \rho$. As $\eta_t \downarrow 0$, let $H_{\max}$ grow so that $\liminf \eta_t H_{\max} \min_i \lambda_i > x_\kappa$, and assume no remaining-work budget truncates the round.*

The function $f(x) = (1 - e^{-x})/x$ decreases strictly from one to zero, so x_κ is unique. For $h = \lfloor x/(\eta_t \lambda_i) \rfloor$, $\phi_i(h) \to f(x)$, since $q_{H_{\min}}(\tau_i) \to 1$. Bracketing the first crossing by $x_\kappa - \varepsilon$ and $x_\kappa + \varepsilon$, then letting $\varepsilon \downarrow 0$, gives

$$\eta_t \lambda_i H_i \to x_\kappa, \qquad \frac{H_i}{H_j} \to \frac{\lambda_j}{\lambda_i}, \qquad \boldsymbol{d}_{i,H_i} \to \rho \nabla f_i(\boldsymbol{\theta}_t). \tag{15}$$

Put $\bar{\lambda} = \sum_i p_i \lambda_i$, $\boldsymbol{\theta}^\star = \sum_i p_i \lambda_i \boldsymbol{a}_i / \bar{\lambda}$, and $J_\lambda = \bar{\lambda} \sum_i p_i / \lambda_i \ge 1$. Then $\boldsymbol{D}_t \to \rho \bar{\lambda}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star)$ and $\eta_t \bar{H}_t \to x_\kappa \sum_i p_i / \lambda_i$.

Substitution into (7) yields the one-round limit

$$\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^\star \longrightarrow [1 - J_\lambda (1 - e^{-x_\kappa})](\boldsymbol{\theta}_t - \boldsymbol{\theta}^\star). \tag{16}$$

For $x > 0$, $\tanh(x/2) \le x/2$ implies $(1 - e^{-x})(1 + 2/x) \le 2$. At $x = x_\kappa$, this gives $1 - e^{-x_\kappa} \le 2(1 - \rho) = 1 - \kappa$. Hence $J_\lambda \le 1/(1 - \kappa)$ makes the coefficient in (16) lie in $[0, 1)$, proving the proposition.

A.2 Geometry of the stopping rule

Proposition A.3 (Robust comparison and alignment)**.** *For nonzero $\boldsymbol{r}_i$, the continuation condition is equivalent to*

$$\mathcal{R}_i(\boldsymbol{d}) := \langle \boldsymbol{r}_i, \boldsymbol{d} \rangle - \kappa \|\boldsymbol{r}_i\|_2 \|\boldsymbol{d}\|_2 > \frac{1 - \kappa^2}{2} \|\boldsymbol{r}_i\|_2^2. \tag{17}$$

An aligned update $\boldsymbol{d} = a\boldsymbol{r}_i$, $a > 0$, passes exactly when $a > (1 + \kappa)/2$. On a fixed local trajectory with a fixed cap, H_i is nonincreasing in κ.

Proof. The farthest point in the reference ball gives $\max_{\boldsymbol{u} \in \mathcal{U}_i} \|\boldsymbol{d} - \boldsymbol{u}\|_2 = \|\boldsymbol{d} - \boldsymbol{r}_i\|_2 + \kappa \|\boldsymbol{r}_i\|_2$. Squaring the continuation inequality yields (17). Conversely, Cauchy–Schwarz and (17) imply $\|\boldsymbol{d}\|_2 > (1 + \kappa)\|\boldsymbol{r}_i\|_2/2 > \kappa \|\boldsymbol{r}_i\|_2$, so taking square roots recovers the test. Dividing (17) by $\|\boldsymbol{r}_i\|_2 \|\boldsymbol{d}\|_2$ proves (6); setting $\boldsymbol{d} = a\boldsymbol{r}_i$ gives the aligned threshold. Increasing κ lowers ψ_i at each fixed prefix, so its first failure cannot occur later. If $\boldsymbol{r}_i = 0$, the rule stops at the base. □

Lemma A.4 (Alignment at the returned step)**.** *Let $\boldsymbol{d}_i = \boldsymbol{d}_{i,H_i}$. Define $o_i = 0$ if $H_i = H_{\min}$ and otherwise $o_i = (1 + \kappa)\|\boldsymbol{r}_i\|_2 \|\boldsymbol{g}_{i,H_i} - \boldsymbol{d}_{i,H_i - 1}\|_2 / H_i$. Then*

$$\langle \boldsymbol{r}_i, \boldsymbol{d}_i \rangle \ge \frac{1 - \kappa^2}{2} \|\boldsymbol{r}_i\|_2^2 + \kappa \|\boldsymbol{r}_i\|_2 \|\boldsymbol{d}_i\|_2 - o_i. \tag{18}$$

Proof. $\mathcal{R}_i$ is $(1 + \kappa)\|\boldsymbol{r}_i\|_2$-Lipschitz. If $H_i > H_{\min}$, the preceding prefix passed the test and $\boldsymbol{d}_i - \boldsymbol{d}_{i,H_i - 1} = (\boldsymbol{g}_{i,H_i} - \boldsymbol{d}_{i,H_i - 1})/H_i$.

Applying the Lipschitz bound to (17) proves the claim. For $H_i = H_{\min}$, substitute $\boldsymbol{d}_i = \boldsymbol{r}_i$; the remaining margin is $(1 - \kappa)^2 \|\boldsymbol{r}_i\|_2^2 / 2 \ge 0$. □

A.3 Proof of Proposition 4.2

Write $\boldsymbol{G} = \nabla F(\boldsymbol{\theta}_t)$ and use $\boldsymbol{D}_t, \bar{\boldsymbol{r}}, V_r, V_d$ from the proposition. Centering the client vectors gives

$$\langle \boldsymbol{G}, \boldsymbol{D}_t \rangle = \sum_i p_i \langle \boldsymbol{r}_i, \boldsymbol{d}_i \rangle - \sum_i p_i \langle \boldsymbol{r}_i - \bar{\boldsymbol{r}}, \boldsymbol{d}_i - \boldsymbol{D}_t \rangle + \langle \boldsymbol{G} - \bar{\boldsymbol{r}}, \boldsymbol{D}_t \rangle. \tag{19}$$

Weighted Cauchy–Schwarz bounds the second sum in absolute value by $\sqrt{V_r V_d}$ and the last term below by $-\|\boldsymbol{G} - \bar{\boldsymbol{r}}\|_2 \|\boldsymbol{D}_t\|_2$. This proves (10). Smoothness and (7) give

$$F(\boldsymbol{\theta}_{t+1}) - F(\boldsymbol{\theta}_t) \le -\eta_t \bar{H}_t \langle \boldsymbol{G}, \boldsymbol{D}_t \rangle + \frac{L\eta_t^2 \bar{H}_t^2}{2} \|\boldsymbol{D}_t\|_2^2,$$

which is (9). Lemma A.4 supplies a stopping-rule lower bound on each local alignment term, including the retained crossing step.

A.4 Proof of Theorem 4.3

Proof. Let $\mathbb{E}_t$ condition on the history before round t, and $\mathbb{E}_{t,S}$ also condition on its selected clients. Use the assumptions of the theorem and abbreviate

$$\begin{aligned} &\boldsymbol{G}_t = \nabla F(\boldsymbol{\theta}_t), \quad \mu_t = \mathbb{E}_t \bar{H}_t, \quad a_t = \eta_t \mu_t, \\ &\boldsymbol{Y}_t = \frac{\bar{H}_t \boldsymbol{D}_t}{\mu_t}, \quad \boldsymbol{\beta}_t = \mathbb{E}_t \boldsymbol{Y}_t - \boldsymbol{G}_t, \quad V_t = \mathbb{E}_t \|\boldsymbol{Y}_t - \mathbb{E}_t \boldsymbol{Y}_t\|_2^2. \end{aligned} \tag{20}$$

Here $H_{\min} \le \mu_t \le H_{\max}$, and the actual update is $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - a_t \boldsymbol{Y}_t$.

Smoothness and the conditional second-moment identity give

$$\begin{aligned} \mathbb{E}_t F(\boldsymbol{\theta}_{t+1}) &\le F(\boldsymbol{\theta}_t) - a_t \langle \boldsymbol{G}_t, \boldsymbol{G}_t + \boldsymbol{\beta}_t \rangle + \frac{La_t^2}{2} \big(\|\boldsymbol{G}_t + \boldsymbol{\beta}_t\|_2^2 + V_t \big) \\ &\le F(\boldsymbol{\theta}_t) - \frac{a_t}{2} \|\boldsymbol{G}_t\|_2^2 + \frac{a_t}{2} \|\boldsymbol{\beta}_t\|_2^2 + \frac{La_t^2}{2} V_t. \end{aligned}$$

The second inequality discards $-a_t(1 - La_t)\|\boldsymbol{G}_t + \boldsymbol{\beta}_t\|_2^2/2 \le 0$, since $La_t \le L\eta_t H_{\max} \le 1$.

Taking expectations and summing over $t < T$ telescopes the loss, yielding

$$\sum_{t<T} \mathbb{E}[a_t \|\boldsymbol{G}_t\|_2^2] \le 2\Delta + \sum_{t<T} \mathbb{E}[a_t \|\boldsymbol{\beta}_t\|_2^2 + La_t^2 V_t].$$

Using $\eta_t H_{\min} \le a_t \le \eta_t H_{\max}$ and deterministic η_t on both sides gives, for $\chi = H_{\max}/H_{\min}$,

$$\frac{\sum_{t<T} \eta_t \mathbb{E}\|\boldsymbol{G}_t\|_2^2}{\sum_{t<T} \eta_t} \le \frac{2\Delta}{H_{\min} \sum_{t<T} \eta_t} + \chi \frac{\sum_{t<T} \eta_t \mathbb{E}\|\boldsymbol{\beta}_t\|_2^2}{\sum_{t<T} \eta_t} + L\chi H_{\max} \frac{\sum_{t<T} \eta_t^2 \mathbb{E} V_t}{\sum_{t<T} \eta_t}. \tag{21}$$

Taking $c_0 = \max\{2, \chi\}$ proves (11). The bound yields stationarity when $\sum_t \eta_t = \infty$ and its weighted bias and variance terms vanish. □

A.4.1 Why the stopped noise must be retained

Lemma A.5 (Centered stopped sums)**.** *Let $\boldsymbol{\xi}_{i,h} = \boldsymbol{g}_{i,h} - \nabla f_i(\boldsymbol{\theta}_{i,t}^{(h-1)})$ and $\boldsymbol{M}_i = \sum_{h=1}^{H_i} \boldsymbol{\xi}_{i,h}$. Under the main-text noise assumptions,*

$$\mathbb{E}_{t,S} \boldsymbol{M}_i = 0, \qquad \mathbb{E}_{t,S} \|\boldsymbol{M}_i\|_2^2 \le \sigma^2 \mathbb{E}_{t,S} H_i. \tag{22}$$

The mean of $\boldsymbol{M}_i / H_i$ need not be zero.

Proof. Write $\boldsymbol{M}_i = \sum_{h=1}^{H_{\max}} \mathbf{1}_{\{H_i \ge h\}} \boldsymbol{\xi}_{i,h}$ on a potential local trajectory. TCS decides whether to execute step h before observing that gradient, so each indicator is measurable with respect to the preceding local history. Conditional mean-zero noise gives the first identity and eliminates all cross terms in the second moment. Then, summing the variance bounds proves the inequality. □

A.4.2 Bias decomposition

Define the selected initial gradient and each client's average drift by

$$\boldsymbol{G}_{S_t} = \sum_{i \in \mathcal{S}_t} p_i \nabla f_i(\boldsymbol{\theta}_t), \qquad \boldsymbol{b}_i = \frac{1}{H_i} \sum_{h=1}^{H_i} [\nabla f_i(\boldsymbol{\theta}_{i,t}^{(h-1)}) - \nabla f_i(\boldsymbol{\theta}_t)].$$

Since $\boldsymbol{d}_{i,H_i} = \nabla f_i(\boldsymbol{\theta}_t) + \boldsymbol{b}_i + \boldsymbol{M}_i/H_i$, the bias separates exactly as

$$\boldsymbol{\beta}_t = \underbrace{\frac{\mathbb{E}_t[\bar{H}_t(\boldsymbol{G}_{S_t} - \boldsymbol{G}_t)]}{\mu_t}}_{\text{sampling}} + \underbrace{\frac{\mathbb{E}_t[\bar{H}_t \sum_i p_i \boldsymbol{b}_i]}{\mu_t}}_{\text{local drift}} + \underbrace{\frac{\mathbb{E}_t \sum_i p_i (\bar{H}_t/H_i)\boldsymbol{M}_i}{\mu_t}}_{\boldsymbol{\beta}_t^{\text{stop}}}. \tag{23}$$

Lemma A.5 rewrites the last term as

$$\boldsymbol{\beta}_t^{\text{stop}} = \frac{1}{\mu_t}\mathbb{E}_t \sum_i p_i \mathbb{E}_{t,S}\left[\left(\frac{\bar{H}_t}{H_i} - \mathbb{E}_{t,S}\frac{\bar{H}_t}{H_i}\right)\boldsymbol{M}_i\right]. \tag{24}$$

Thus the dependence on stopping is retained through the random ratios $\bar{H}_t/H_i$, without assuming independence between the counts and updates.

Table 3: Main-benchmark configurations.

Setting	FEMNIST	CIFAR-100	StackOverflow
Training clients	3,597	500	342,477
Clients per round	10	10	10
Batch size	20	10	30
Work cap	100,000	100,000	100,000
Adaptive local-step range	$[10, 100]$	$[10, 100]$	$[10, 100]$

B Experimental Protocol and Additional MNIST Results

This appendix gives the experimental protocol used for the main benchmark results in Table 2.

B.1 Federated learning configuration

One unit of work is one client minibatch optimizer update. For realized local counts $H_{i,t}$, the completed work before round t is

$$W_t = \sum_{r<t} \sum_{i \in \mathcal{S}_r} H_{i,r}.$$

Every reported main-benchmark run uses the same cap $W_{\max} = 100{,}000$ within its dataset and configuration. This cap is not an equality constraint: we report W_{final} and $W_{\max} - W_{\text{final}}$, without dummy updates or extrapolation. Fixed-count methods start a round only when all participants can complete their prescribed count. Adaptive methods use a common residual-budget cap, at most $\lfloor (W_{\max} - W_t)/|\mathcal{S}_t| \rfloor$, and stop when fewer than ten updates per participant remain. After the base-only first round, TCS applies its stopping rule with the cap $\min\{H_{\max}, \lfloor (W_{\max} - W_t)/|\mathcal{S}_t| \rfloor\}$. Participation is not reduced to exhaust the budget. All main-benchmark runs sample 10 clients per round and use SGD with zero momentum. No weight decay or gradient clipping were used All partition, initialization, and participation streams are paired across methods using seeds 4286317001, 543553321, and 1610997163, generated from `numpy.random.SeedSequence(5959)`.

For CIFAR-100, the reported runs use initial learning rate 0.1. For FEMNIST, 0.01 and for StackOverflow, 0.31622. These learning rates are tuned with separate pilot seeds $(5101, 5102, 5103)$ and the learning rate is decayed by multiplying 0.3 after 50% and 75% of the work cap has been completed.

B.2 Datasets, Models, and Batch Calibration

Table 3 summarizes the benchmark configurations. The protocol uses FP32, one GPU per run, and at most four concurrent independent runs.

Epoch-scale calibration. We choose a shared batch size within each dataset and keep it fixed across methods and local step counts. Ten local steps process roughly as many examples as the mean client training set. Note that the per-client ratio $10B/n_i$ can differ substantially from one, so this calibration does not imply an equal number of local epochs for every client. The local-epoch formulation in FedAvg (McMahan et al., 2017) motivates this choice; it does not imply that ten steps or one epoch is a universal optimum.

Models. For FEMNIST, the model is the two-convolution neural networks with hidden-size setting 64 and 62 outputs (Caldas et al., 2018). The primary metric is pooled sample-level, federated test accuracy over the fixed writer holdout sets. We additionally report client-micro accuracy and NLL, with empty evaluation clients excluded and counted.

For CIFAR-100, we use a ResNet-18 (He et al., 2016) trained from scratch, with a CIFAR-size convolutional stem and two-group group normalization layer in place of the original batch normalization layer.

For StackOverflow, we use a causal Transformer with three layers, width 160, four attention heads, feed-forward width 640, dropout 0.1, context length 20, and 10,000 vocabulary words plus four

special symbols. The model has about 4.1 million trainable parameters. Each sequence predicts the next token at every non-padding target position.

B.3 Baseline Adaptations and Stopping Controls

The short names in the main text denote our documented FL implementations. They do not imply that the original papers used this work budget, step range, optimizer, or data partition. For MILLS, ADACOMM, and QSR, the main action set contains every integer from 10 through 100.

FIXED. We run counts 10, 20, 50, and 100 from the first round. All four use the declared common learning-rate schedule.

MILLS. The reported Mills baseline uses the training-progress rule of Mills et al. (2023). It starts from $H_0 = 100$ and computes

$$H_t = \left\lceil \left(\overline{L}_t / L_0\right)^{1/3} H_0 \right\rceil ,$$

where L_0 is the first observed weighted mean client pre-training loss and $\overline{L}_t$ is the latest weighted mean pre-training loss from completed rounds. The proposal is clipped to the common integer action set $[10, 100]$.

ADACOMM. We adapt Eq. (20) of Wang & Joshi (2019), starting from 100 updates. Its shared proposal is proportional to $\sqrt{(\eta_0/\eta_t)(\overline{L}_t/L_0)}$, where the losses are training observations from completed rounds, never test or validation feedback. The implementation applies integer rounding and the common bounds.

QSR. Following Gu et al. (2024), the proposal grows in proportion to $1/\eta_t^2$ as: $H_t = \min\{100, \max\{10, \lfloor(\eta_0/\eta_t)^2\rfloor\}\}$. Growth is tied to the work-indexed learning-rate schedule, not a separately tuned communication-round schedule.

ALT. We compare local and received-global feature representations on the current client mini-batch, as in Shenaj et al. (2025), and halt when their cosine similarity falls below the threshold. The current implementation uses paper-guided threshold tuning scheme and the common step bounds. Its additional global-model forward evaluations are outside the optimizer-step budget.

B.4 MNIST Diagnostics

The figures use MNIST with 100 clients, 10 participants per round, batch size 60, and a 784–200–200–10 network. Client labels follow Dirichlet concentrations $\{0.1, 10.0\}$ (Hsu et al., 2019). Local SGD uses rate 0.1, multiplied by 0.3 at 50% and 75% of the 100,000-update cap, with zero momentum and weight decay. Global test accuracy is evaluated after every round. We use a 92% target for the retrospective MNIST comparison in Table 5.

We report all $16 \times 2 \times 3 = 96$ completed runs: four fixed counts $\{10, 20, 50, 100\}$, eight TCS settings $\kappa \in \{0, .1, .25, .5, .75, .9, .95, .99\}$, and MILLS, ADACOMM, QSR, and ALT, at two concentrations and three paired seeds. Each seed uses the same data partition and initial model across methods. TCS uses $H_{\min} = 10$, a base-only first round, and a cap of 100 thereafter. The remaining adaptive methods follow their own first-round schedules; MILLS, ADACOMM, and QSR choose any integer in $[10, 100]$.

Figure 3 shows four fixed counts and the same two TCS settings in both partitions; the complete table retains all 16 methods and settings. The sensitivity sweep holds $H_{\min} = 10$ fixed and reports every tested κ, without selecting a best test-set configuration.

Connection to the quadratic interpretation. In the deterministic isotropic model, with small learning rate and a nonbinding cap, the theory predicts $\eta_t \lambda_i H_i \to x_\kappa$, with $(1 - e^{-x_\kappa})/x_\kappa = (1+\kappa)/2$. This predicts shorter trajectories at larger curvature or larger κ within that model. Indeed,

Table 4: MNIST outcomes under a 100,000-update cap. Accuracy (percent) and rounds are means $\pm$ sample standard deviation over three paired seeds. Fewer rounds reports the percentage reduction relative to FIXED-10. All methods use nearly the same work and share work-normalized aggregation.

	$\alpha = 0.1$			$\alpha = 10.0$		
Method	Accuracy	Rounds	Fewer rounds	Accuracy	Rounds	Fewer rounds
FIXED-10	97.04 ± 0.13	1000.0 ± 0.0	0.0%	97.84 ± 0.04	1000.0 ± 0.0	0.0%
FIXED-20	96.60 ± 0.22	500.0 ± 0.0	50.0%	97.75 ± 0.01	500.0 ± 0.0	50.0%
FIXED-50	95.38 ± 0.64	200.0 ± 0.0	80.0%	97.63 ± 0.06	200.0 ± 0.0	80.0%
FIXED-100	93.98 ± 1.13	100.0 ± 0.0	90.0%	97.25 ± 0.01	100.0 ± 0.0	90.0%
TCS ($\kappa = .5$)	96.78 ± 0.20	631.3 ± 14.8	**36.9%**	97.85 ± 0.03	726.0 ± 1.0	**27.4%**

Table 5: MNIST work and communication to the first observed 92% test accuracy. Entries are means $\pm$ sample standard deviation over three paired seeds; every listed setting reaches the target in all three seeds. Evaluation occurs after every round, with no interpolation. All methods share work-normalized aggregation.

	$\alpha = 0.1$		$\alpha = 10.0$	
Method	Updates (10^3)	Rounds	Updates (10^3)	Rounds
FIXED $H = 10$	13.37 ± 1.88	133.7 ± 18.8	5.20 ± 0.00	52.0 ± 0.0
FIXED $H = 20$	15.67 ± 2.73	78.3 ± 13.7	5.40 ± 0.20	27.0 ± 1.0
FIXED $H = 50$	21.67 ± 3.33	43.3 ± 6.7	6.00 ± 0.00	12.0 ± 0.0
FIXED $H = 100$	39.00 ± 5.20	39.0 ± 5.2	7.00 ± 0.00	7.0 ± 0.0
MILLS	19.10 ± 1.40	47.7 ± 4.9	5.96 ± 0.27	9.3 ± 0.6
ADACOMM	21.84 ± 3.72	42.7 ± 7.8	5.70 ± 0.37	11.0 ± 1.0
QSR	13.37 ± 1.88	133.7 ± 18.8	5.20 ± 0.00	52.0 ± 0.0
ALT	32.59 ± 5.54	43.7 ± 11.6	7.28 ± 0.03	8.0 ± 0.0
TCS ($\kappa = 0.50$)	14.96 ± 2.25	103.3 ± 15.0	5.09 ± 0.21	36.3 ± 1.5
TCS ($\kappa = 0.75$)	13.04 ± 0.85	109.7 ± 6.8	5.14 ± 0.26	44.7 ± 2.1
TCS ($\kappa = 0.90$)	12.54 ± 1.45	114.0 ± 13.1	5.25 ± 0.07	47.7 ± 0.6
TCS ($\kappa = 0.95$)	13.41 ± 1.96	122.0 ± 17.8	5.27 ± 0.11	48.0 ± 1.0

equal-curvature quadratics with different centers can have the same stopping lengths despite different gradient dissimilarity. We therefore interpret the neural-network results as trajectory-dependent resource trade-offs.

B.5 COMPLETE GRIDS AND FIGURE CONSTRUCTION

Table 4 compares final accuracy and communication with FIXED-10 under the common work cap. Table 5 reports the first observed 92% accuracy costs, a separate retrospective comparison.

Table 6 reports all 32 method–partition settings, including those with lower accuracy. Figures 6 and 7 show accuracy/loss trajectories and stopping behavior.

B.6 SENSITIVITY TO CONSERVATISM

We vary all eight tested κ values while holding $H_{\min} = 10$ and the work cap fixed. Figure 5 shows how this choice affects accuracy and communication. At $\alpha = 0.1$, increasing κ from zero to 0.99 raises mean accuracy from 96.28% to 96.96%, while mean rounds increase from 353.7 to 908.7. At $\alpha = 10.0$, mean accuracy stays between 97.7% and 97.9% across the grid, while mean rounds range from 352.7 to 909.0. At large κ, local training usually stops immediately after the first additional update beyond the ten base steps. We use the same grid in both partitions and report final checkpoints rather than selecting the best test-set configuration.

Table 6: Complete MNIST results at $\alpha \in \{0.1, 10.0\}$ and a 100,000-update cap. Accuracy (percent), NLL, and rounds report mean $\pm$ sample SD over three paired seeds; work (thousands) and local steps report seed means. All eight TCS settings use $H_{\min} = 10$.

α	Method	Accuracy (%)	NLL	Rounds	Work (10^3)	Mean H_i
0.1	FIXED-10	97.04 ± 0.13	0.097 ± 0.006	1000.0 ± 0.0	100.000	10.00
0.1	FIXED-20	96.60 ± 0.22	0.112 ± 0.005	500.0 ± 0.0	100.000	20.00
0.1	FIXED-50	95.38 ± 0.64	0.146 ± 0.019	200.0 ± 0.0	100.000	50.00
0.1	FIXED-100	93.98 ± 1.13	0.195 ± 0.035	100.0 ± 0.0	100.000	100.00
0.1	MILLS	96.65 ± 0.31	0.109 ± 0.008	530.0 ± 0.0	99.920	18.85
0.1	ADACOMM	95.82 ± 0.42	0.134 ± 0.012	218.3 ± 12.1	100.000	45.89
0.1	QSR	96.50 ± 0.21	0.114 ± 0.008	753.0 ± 0.0	100.000	13.28
0.1	ALT	93.83 ± 1.93	0.190 ± 0.049	111.3 ± 6.7	99.995	90.02
0.1	TCS ($\kappa = 0$)	96.28 ± 0.29	0.122 ± 0.013	353.7 ± 2.5	99.963	28.27
0.1	TCS ($\kappa = 0.1$)	96.37 ± 0.36	0.116 ± 0.010	403.3 ± 4.5	99.990	24.79
0.1	TCS ($\kappa = 0.25$)	96.47 ± 0.32	0.115 ± 0.012	482.7 ± 9.3	99.961	20.72
0.1	TCS ($\kappa = 0.5$)	96.78 ± 0.20	0.105 ± 0.007	631.3 ± 14.8	99.981	15.84
0.1	TCS ($\kappa = 0.75$)	96.82 ± 0.35	0.103 ± 0.012	795.7 ± 16.7	99.943	12.56
0.1	TCS ($\kappa = 0.9$)	96.94 ± 0.21	0.098 ± 0.007	896.3 ± 7.4	99.963	11.15
0.1	TCS ($\kappa = 0.95$)	97.03 ± 0.15	0.097 ± 0.007	905.7 ± 3.2	99.941	11.04
0.1	TCS ($\kappa = 0.99$)	96.96 ± 0.15	0.098 ± 0.005	908.7 ± 0.6	99.970	11.00
10.0	FIXED-10	97.84 ± 0.04	0.070 ± 0.001	1000.0 ± 0.0	100.000	10.00
10.0	FIXED-20	97.75 ± 0.01	0.072 ± 0.001	500.0 ± 0.0	100.000	20.00
10.0	FIXED-50	97.63 ± 0.06	0.078 ± 0.000	200.0 ± 0.0	100.000	50.00
10.0	FIXED-100	97.25 ± 0.01	0.088 ± 0.001	100.0 ± 0.0	100.000	100.00
10.0	MILLS	97.82 ± 0.06	0.071 ± 0.000	530.0 ± 0.0	99.920	18.85
10.0	ADACOMM	97.79 ± 0.05	0.071 ± 0.000	375.3 ± 2.1	99.963	26.63
10.0	QSR	97.80 ± 0.06	0.071 ± 0.001	753.0 ± 0.0	100.000	13.28
10.0	ALT	97.21 ± 0.04	0.089 ± 0.001	101.0 ± 0.0	99.992	99.00
10.0	TCS ($\kappa = 0$)	97.74 ± 0.04	0.073 ± 0.001	352.7 ± 2.1	99.992	28.35
10.0	TCS ($\kappa = 0.1$)	97.73 ± 0.08	0.073 ± 0.001	421.7 ± 1.5	99.986	23.71
10.0	TCS ($\kappa = 0.25$)	97.77 ± 0.06	0.071 ± 0.001	536.3 ± 0.6	99.971	18.64
10.0	TCS ($\kappa = 0.5$)	97.85 ± 0.03	0.071 ± 0.002	726.0 ± 1.0	99.953	13.77
10.0	TCS ($\kappa = 0.75$)	97.82 ± 0.07	0.070 ± 0.001	868.0 ± 0.0	99.938	11.51
10.0	TCS ($\kappa = 0.9$)	97.86 ± 0.06	0.070 ± 0.001	908.0 ± 0.0	99.925	11.00
10.0	TCS ($\kappa = 0.95$)	97.82 ± 0.06	0.070 ± 0.001	909.0 ± 0.0	99.983	11.00
10.0	TCS ($\kappa = 0.99$)	97.82 ± 0.06	0.070 ± 0.001	909.0 ± 0.0	99.980	11.00

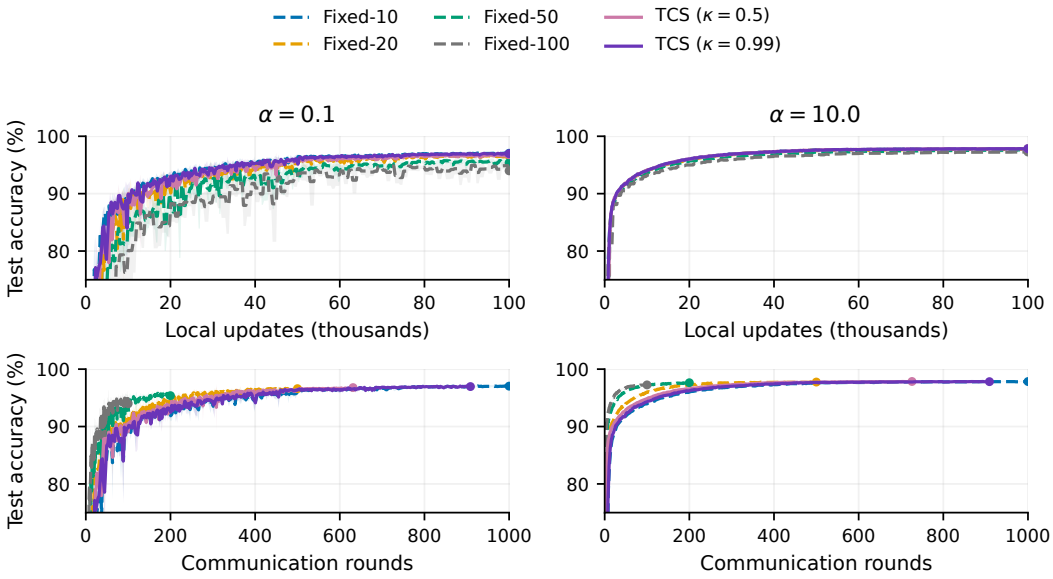

Figure 3: MNIST at $\alpha \in \{0.1, 10.0\}$ and a 100,000-update cap: the same runs against work (top) and communication (bottom). We compare four fixed counts with TCS at $H_{\min} = 10$, using the same two κ values in both partitions. Curves show three-seed means, bands show sample standard deviation, and dots mark mean final checkpoints. These MNIST runs share work-normalized aggregation. Accuracy axes are truncated for readability.

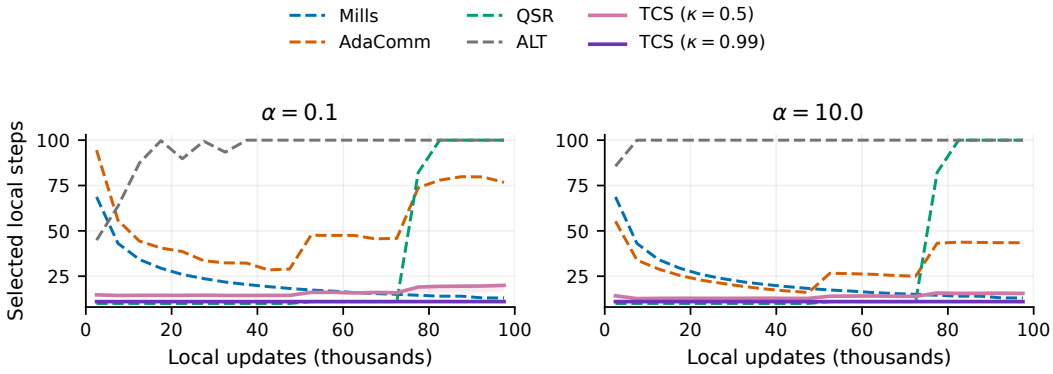

Figure 4: Additional MNIST schedule comparison in 5,000-update bins. Lines average per-seed mean local steps. TCS bands show quartiles of client participation pooled within each bin, then averaged over seeds. These bands combine within-round and between-round variation. The main mechanism figure instead computes quartiles separately within each round.

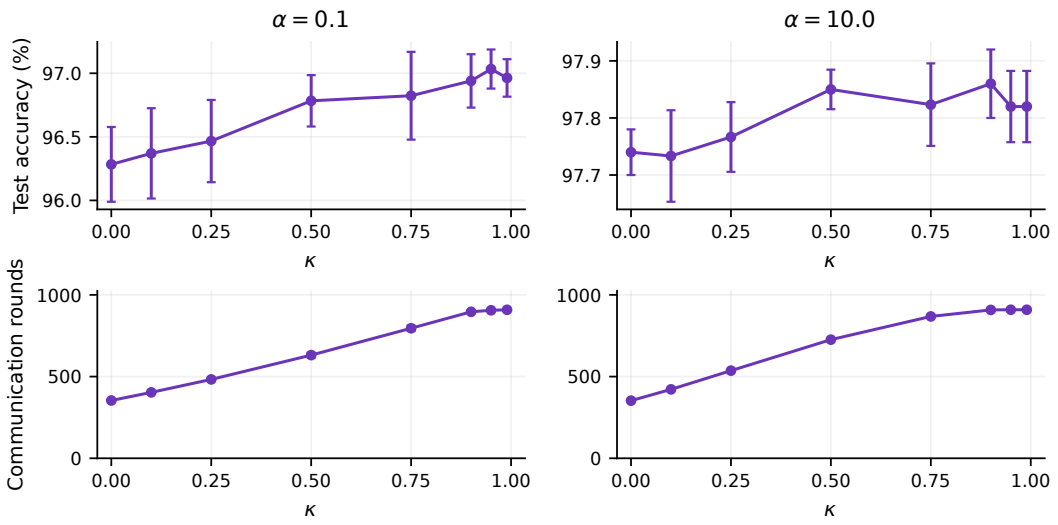

Figure 5: MNIST sensitivity to κ at $\alpha \in \{0.1, 10.0\}$. We hold $H_{\min} = 10$ and the 100,000-update cap fixed. Top: final test accuracy. Bottom: total communication rounds. Each point reports the mean over three paired seeds, with sample SD error bars; all eight tested κ values are shown. Accuracy axes differ across partitions to expose small changes.

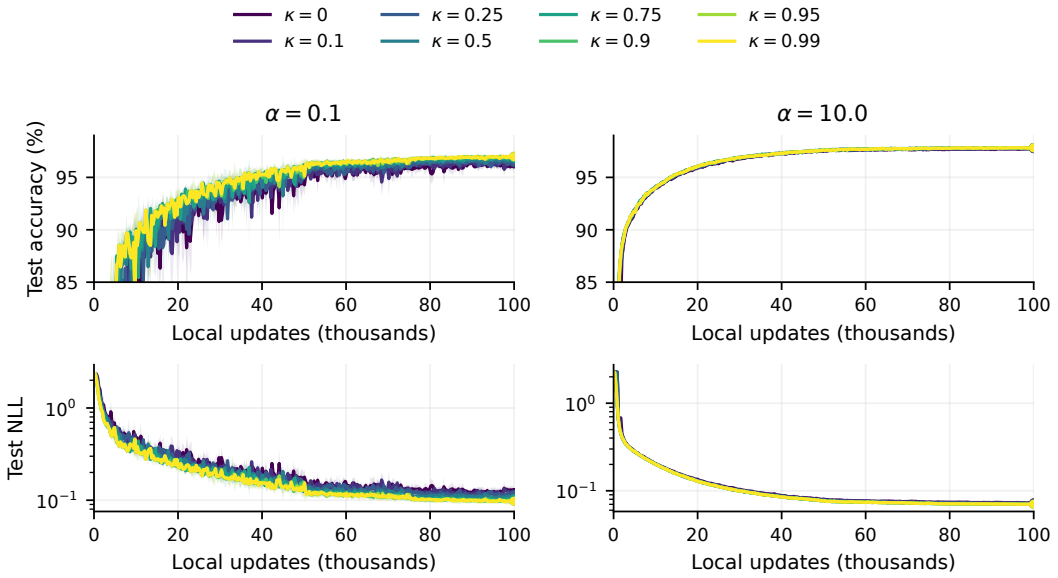

Figure 6: All tested κ values at $H_{\min} = 10$ on MNIST with $\alpha \in \{0.1, 10.0\}$. Top: test accuracy; bottom: NLL on a logarithmic scale. Bands show sample SD over three paired seeds. Accuracy axes are truncated for readability.

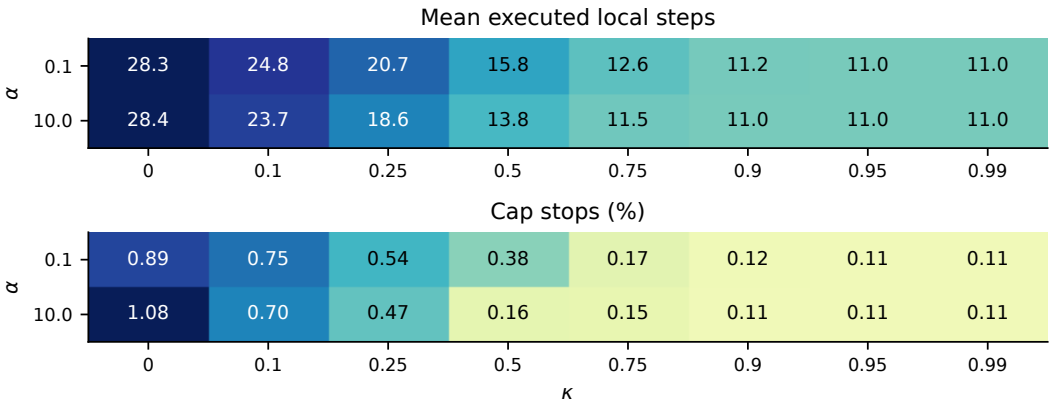

Figure 7: MNIST stopping diagnostics at $\alpha \in \{0.1, 10.0\}$ and $H_{\min} = 10$. Top: mean executed local steps. Bottom: cap-stop frequency in percent. Values average the per-run statistics over three seeds and include bootstrap and residual-budget rounds.