TEMPORAL LOSS SIGNATURES FOR DATA PRUNING IN FEDERATED LEARNING WITH NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning enables collaborative model training without sharing raw data, but heterogeneity and label noise can degrade performance. Data pruning is difficult because high loss conflates mislabeled examples with hard but informative ones, while static selections become stale as the global model evolves. We introduce the **Loss Signature Score (LSS)**, a dynamic, client-local criterion that combines sliding-window mean loss with loss gain. Mean loss captures persistent difficulty, whereas gain favors hard examples that show learning progress. Periodic re-scoring, class-aware retention quotas, and FedProx enable adaptive pruning without estimating the noise rate. The framework adds one full-local-data inference pass per round but requires no scoring-related backward pass or sample-level communication. Experiments on CIFAR-10/100 and CIFAR-10N/100N cover non-IID partitions, synthetic and human annotation noise, and nominal pruning rates from 10% to 70%, with the largest benefits under stronger noise and aggressive pruning. On CIFAR-100 with 20% symmetric noise and a 50% pruning target, LSS achieves 65.38% test accuracy, exceeding the next-best baseline by 4.29 percentage points. On CIFAR-100N with 40.2% human-annotated label noise, it outperforms all evaluated data-pruning baselines at every tested pruning rate.

1 INTRODUCTION

Federated learning (FL) enables collaborative model training without sharing raw data McMahan et al. (2017); Wang et al. (2020); Li et al. (2021b), supporting applications in healthcare Teo et al. (2024) and edge intelligence Ale et al. (2021); Wang et al. (2019). In practice, training must contend with both heterogeneous client distributions and unreliable labels. Methods such as FedProx Li et al. (2020), SCAFFOLD Karimireddy et al. (2020), and MOON Li et al. (2021a) mitigate client drift, but do not explicitly identify annotation errors in local data Northcutt et al. (2021). Controlling local optimization therefore does not, by itself, resolve the problem of training on unreliable examples.

Data pruning offers a complementary approach to improving training data quality Paul et al. (2021). In FL, however, selecting useful examples requires addressing two challenges. *Distinguishing noise from difficulty:* both mislabeled and hard-but-clean samples can incur high loss, so pruning by loss alone risks removing informative examples, including rare local classes Li et al. (2024); Pleiss et al. (2020). *Adapting selection over time:* as local training and global aggregation update the model, sample utility can change, making early selection decisions outdated. Building on existing data-selection approaches Hao et al. (2025); Jiang & Zhang (2025), we seek a criterion that addresses both challenges using only client-local training histories.

Our central insight is that loss level and learning progress provide complementary evidence about sample utility. Persistently high loss with little reduction can indicate label noise, whereas high loss accompanied by substantial reduction can identify a difficult yet learnable example. We capture this distinction through the **Loss Signature Score (LSS)**, which combines window-averaged loss with loss gain to guide pruning without prior noise-rate estimation. Rather than treating either statistic as a definitive noise indicator, LSS uses their combination to balance difficulty against learning progress.

We embed LSS in a dynamic federated pruning framework that periodically refreshes scores from sliding-window histories. FedProx constrains local drift, while class-aware allocation accounts for

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.

local class representation during selection. All scoring and pruning remain on clients. Global-model inference passes refresh loss histories without additional scoring-related backward passes or sample-level communication.

Our contributions are:

1. **Trajectory-aware scoring.** LSS combines loss level and loss reduction to guide noise-aware pruning while retaining difficult examples that show learning progress.
2. **Dynamic federated integration.** Sliding-window re-pruning, FedProx, and client-local class-aware allocation jointly support adaptive selection under heterogeneous data without extra scoring-related backward passes or communication.
3. **Empirical validation.** Multi-seed experiments on CIFAR-10/100 and CIFAR-10N/100N cover synthetic and human-annotated noise at nominal pruning rates of 0.1–0.7. Component ablations clarify when loss gain improves on mean loss alone, with the strongest benefits under aggressive pruning.

2 Related Work

Federated optimization. FedAvg (McMahan et al., 2017) alternates local training with model averaging. FedProx (Li et al., 2020) limits client drift with a proximal penalty, SCAFFOLD (Karimireddy et al., 2020) uses control variates, and MOON (Li et al., 2021a) aligns representations contrastively. These methods improve local optimization under heterogeneity but do not select reliable examples. LSS instead curates local training data and uses FedProx as its optimizer.

Dataset pruning. Training-dynamics criteria include forgetting events (Toneva et al., 2019), early gradient norms and prediction errors in GraNd and EL2N (Paul et al., 2021), and BOSS's size-aware balance of diversity and difficulty (Acharya et al., 2024). TDDS measures temporal variability in sample contributions (Zhang et al., 2024), while DUAL combines difficulty and prediction uncertainty (Cho et al., 2025). LSS instead combines mean loss with loss reduction to separate persistent difficulty from learning progress. We compare these criteria under a common federated re-pruning protocol.

Dynamic data pruning. Refreshing training subsets, even randomly, can outperform static pruning at high pruning rates (Raju et al., 2021). InfoBatch dynamically removes low-loss samples and rescales retained gradients to approximate the full-data gradient expectation (Qin et al., 2024). In contrast, LSS targets potentially mislabeled examples and adapts selection through sliding-window, client-local inference.

Federated data selection and label noise. FedCS (Hao et al., 2025) selects coresets using feature-space distance contrast. GCFL (Sivasubramanian et al., 2024) periodically constructs gradient-based coresets using a small server validation set, while FedBSS (Xu et al., 2025) orders local selection by bias to reduce drift, including under noisy labels. FedClean (Jiang & Zhang, 2025) combines clean-sample selection, two-stage label correction, and noise-aware aggregation. LSS integrates dynamic, noise-aware selection through client-local loss level and gain with class-aware pruning, requiring neither server validation data, label correction, nor additional sample-level communication.

3 Methodology

LSS ranks examples using both recent loss level and loss reduction, then periodically refreshes a class-aware training subset on each client. We first define the federated optimization protocol and retention budget, then specify the score, warm-up, and refresh schedule. Finally, we relate retained-label corruption and selection bias to conditional progress on the full-data clean objective. Distinguishing scoring checkpoints from local optimization is essential because subset changes alter both the loss histories and the round-specific training objective.

3.1 Federated Setting and Local Optimization

Consider $K \geq 1$ nonempty clients and $C \geq 2$ classes. Client k holds $n_k \geq 1$ examples $\mathcal{D}_k = \{(x_{ki}, \tilde{y}_{ki}) : i \in I_k\}$, where $I_k = \{1, \ldots, n_k\}$ is a fixed index set, x_{ki} is an input, and $\tilde{y}_{ki} \in \{1, \ldots, C\}$ is its observed label. The latent clean label $y_{ki} \in \{1, \ldots, C\}$ is unavailable to LSS and appears only in the analysis. Let $N = \sum_{k=1}^{K} n_k$ and $\pi_k = n_k/N$, so that $\sum_{k=1}^{K} \pi_k = 1$. These original-size aggregation weights remain fixed after pruning. Unless stated otherwise, client sums range from 1 to K. For a finite set A, $|A|$ denotes its cardinality; $\|\cdot\|$ is the Euclidean norm and $\langle \cdot, \cdot \rangle$ its inner product.

Let $\theta \in \mathbb{R}^d$ denote the model parameters and let the softmax classifier f_θ map each input to a probability vector in the interior of the C-simplex. For any candidate label $y \in \{1, \ldots, C\}$, define $\ell_{ki}(\theta; y) = -\log[f_\theta(x_{ki})]_y$, where $[\cdot]_y$ selects class y. We use the natural logarithm and stable log-softmax arithmetic. Unless explicitly described as a logit gradient, ∇ differentiates with respect to θ.

All clients participate in each round $t \in \{1, \ldots, T\}$, where $T \geq 1$. The server broadcasts θ^t and, after local training, aggregates θ^{t+1}. Federated training begins at the stored checkpoint $\theta^1 = \theta^{\text{init}}$ after the selection-only warm-up in Section 3.4.

Given the retained index set $S_k^t \subseteq I_k$, with $m_k^t = |S_k^t| > 0$, client k initializes at θ^t and optimizes the retained observed-label loss with a FedProx penalty:

$$F_k^t(\theta) = \frac{1}{m_k^t} \sum_{i \in S_k^t} \ell_{ki}(\theta; \tilde{y}_{ki}), \qquad Q_k^t(\theta) = F_k^t(\theta) + \frac{\lambda}{2} \|\theta - \theta^t\|_2^2, \tag{1}$$

Here $\lambda \geq 0$ controls departure from the fixed round-t broadcast model (Li et al., 2020). The corresponding aggregate empirical objective is $F^t(\theta) = \sum_k \pi_k F_k^t(\theta)$. Because F^t depends on the selected sets, refreshing any S_k^t changes the objective. The proximal term affects local training only and is never included in sample scores.

Each client performs $E \geq 1$ complete local epochs of momentum SGD with learning rate $\eta > 0$, momentum $\nu \in [0, 1)$, and maximum integer batch size $b \geq 1$. Indices are independently reshuffled each epoch, and the final incomplete batch is retained. Thus client k takes $J_k^t = E\lceil m_k^t/b \rceil$ optimizer steps. Let $u_{k,s}^t$ and $v_{k,s}^t$ be its parameters and momentum buffer before step $s \in \{0, \ldots, J_k^t - 1\}$, with $u_{k,0}^t = \theta^t$ and $v_{k,0}^t = 0$. For the nonempty mini-batch $\mathcal{M}_{k,s}^t \subseteq S_k^t$, the update is

$$\begin{aligned} v_{k,s+1}^t &= \nu v_{k,s}^t + \frac{1}{|\mathcal{M}_{k,s}^t|} \sum_{i \in \mathcal{M}_{k,s}^t} \nabla \ell_{ki}(u_{k,s}^t; \tilde{y}_{ki}) + \lambda(u_{k,s}^t - \theta^t), \\ u_{k,s+1}^t &= u_{k,s}^t - \eta v_{k,s+1}^t. \end{aligned} \tag{2}$$

Momentum persists across local steps and epochs within a round, but is reset before the next round. The server computes $\theta^{t+1} = \sum_k \pi_k u_{k,J_k^t}^t$. Consequently, clients with different retained-set sizes can take different numbers of local steps even though all use the same number of epochs.

Heterogeneity and noise. For non-IID partitioning, the allocation proportions of each class across the K clients are sampled from $\text{Dirichlet}(\alpha \mathbf{1}_K)$, where $\mathbf{1}_K \in \mathbb{R}^K$ is the all-ones vector and $\alpha > 0$; smaller α produces more uneven class allocations. We assume the resulting clients are nonempty. For synthetic noise, labels are corrupted independently, with client-specific probability $\varepsilon_k \in [0, 1]$ and transition law

$$\Pr(\tilde{y}_{ki} = c \mid y_{ki}) = \begin{cases} 1 - \varepsilon_k, & c = y_{ki}, \\ \varepsilon_k/(C-1), & c \neq y_{ki}, \end{cases} \qquad 0 \leq \varepsilon_k \leq 1. \tag{3}$$

Thus a corrupted label is uniform over the $C - 1$ incorrect classes. ε_k is the generating probability, not necessarily the realized finite-sample corruption fraction, and human annotation noise need not follow this model. Neither y_{ki} nor ε_k is available to LSS.

3.2 Selection Budget

The client-level pruning rate $\rho \in [0, 1)$ specifies the nominal fraction of local examples to discard and is applied independently within each observed class. For class c on client k, let $I_{kc} = \{i \in I_k :$

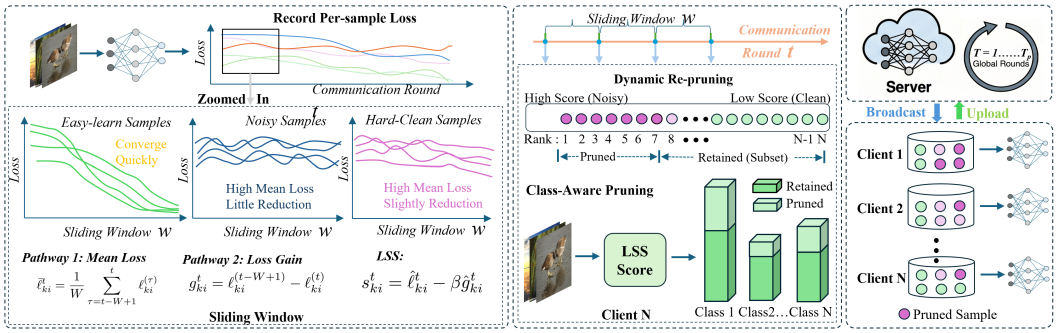

Figure 1: LSS pipeline. At each broadcast-model checkpoint, clients record losses for all local examples. At scheduled rounds with complete history windows, they combine mean loss and loss gain, update the retained set under fixed class-wise quotas, and then perform local training and aggregation.

$\tilde{y}_{ki} = c\}$ and $n_{kc} = |I_{kc}|$. Consistent with the implementation, the number of retained examples is

$$m_{kc} = \begin{cases} 0, & n_{kc} = 0, \\ \max\{1, \lfloor (1-\rho) n_{kc} \rfloor\}, & n_{kc} > 0, \end{cases} \qquad m_k = \sum_{c=1}^{C} m_{kc}, \qquad \rho_k^{\text{eff}} = 1 - \frac{m_k}{n_k}. \tag{4}$$

Thus, the same pruning rate is applied to every nonempty class, while the number of retained examples scales with its class size. The minimum-retention constraint preserves at least one example from each observed class. Because of integer rounding and this constraint, the realized pruning rate ρ_k^{eff} may differ slightly from the nominal rate ρ. Since the observed labels and ρ are fixed, these class-wise quotas remain constant across rounds, although the selected examples may change.

3.3 Loss Signatures and Sample Scoring

Before any selection or local update in round t, client k evaluates every $i \in I_k$ at the broadcast model and records $\ell_{ki}^{(t)} = \ell_{ki}(\theta^t; \tilde{y}_{ki})$. Evaluation uses fixed preprocessing, evaluation mode, and no gradient tracking, so it changes neither model parameters nor state. Using one shared checkpoint for retained and excluded examples avoids mixing global-model losses with losses from different local SGD steps.

Let $W \geq 2$ be the history length in communication rounds. Each client stores the most recent W round-indexed losses for every original example, including the current loss. At round $t \geq W$, a complete window contains a finite value for every $i \in I_k$ and $\tau \in \{t - W + 1, \ldots, t\}$. The window mean and endpoint gain are

$$\bar{\ell}_{ki}^{t} = \frac{1}{W} \sum_{\tau=t-W+1}^{t} \ell_{ki}^{(\tau)}, \qquad g_{ki}^{t} = \ell_{ki}^{(t-W+1)} - \ell_{ki}^{(t)}. \tag{5}$$

The mean measures recent difficulty under the observed label. The unscaled gain is positive for a net loss decrease, zero for no endpoint change, and negative for a net increase. It is not divided by the $W - 1$ transitions and does not imply monotonic improvement, a convergence rate, or learnability: noisy labels may be memorized, while clean examples may have negative gain.

To place the two statistics on a common client-local scale, normalize each over the full candidate set I_k, rather than within classes or only over the current retained set:

$$\hat{\ell}_{ki}^{t} = \frac{\bar{\ell}_{ki}^{t} - \min_{j \in I_k} \bar{\ell}_{kj}^{t}}{\max_{j \in I_k} \bar{\ell}_{kj}^{t} - \min_{j \in I_k} \bar{\ell}_{kj}^{t} + \zeta}, \qquad \hat{g}_{ki}^{t} = \frac{g_{ki}^{t} - \min_{j \in I_k} g_{kj}^{t}}{\max_{j \in I_k} g_{kj}^{t} - \min_{j \in I_k} g_{kj}^{t} + \zeta}, \tag{6}$$

Here $\zeta = 10^{-8}$ prevents division by zero. In exact arithmetic, both normalized values lie in $[0, 1)$, and a constant statistic maps to zero. Because normalization is relative to the client extrema, raw zero gain need not map to zero, and a high normalized gain need not be positive in raw units.

The Loss Signature Score subtracts normalized learning progress from normalized difficulty. At a refresh, LSS retains the lowest-scoring examples within each observed class:

$$s_{ki}^t = \hat{\ell}_{ki}^t - \beta \hat{g}_{ki}^t, \qquad S_k^t = \bigcup_{c=1}^{C} \mathrm{Bottom}_{m_{kc}}\{(s_{ki}^t, i) : i \in I_{kc}\}, \tag{7}$$

For any nonnegative integer m, Bottom_m returns the indices of the m lexicographically smallest score–index pairs: scores are ordered first, and ties are broken by increasing sample index; Bottom_0 is empty. Thus lower mean loss decreases the score, whereas larger gain decreases it by an amount controlled by $\beta \geq 0$. Setting $\beta = 0$ recovers mean-loss ranking. We use $\beta = 2.0$ and report sensitivity in Appendix A.4. Retention still depends on within-class competition and the quotas in equation 4; the score is a ranking heuristic, not a calibrated corruption probability.

3.4 Warm-up and Dynamic Re-Pruning

To initialize selection, each client independently trains θ^{init} for $E_{\mathrm{warm}} \geq 2$ full-data epochs; we use $E_{\mathrm{warm}} = 3$. Training follows equation 2 with $S_k = I_k$, zero initial momentum, the same optimizer hyperparameters, and fixed proximal anchor θ^{init}. No aggregation occurs. After each epoch, the client records evaluation-mode observed-label losses for all local examples at the final checkpoint.

Let $W_{\mathrm{warm}} = \min(W, E_{\mathrm{warm}})$ be the number of epoch snapshots used for initialization. On the last W_{warm} snapshots, compute equation 5 using epoch indices and W_{warm} in place of round indices and W, then apply equation 6–equation 7 to obtain S_k^0, the pre-round-1 selection. At least two complete, finite snapshots are required; otherwise initialization is retried. Clients retain only S_k^0 and discard the warm-up model, optimizer state, and history. The server starts from $\theta^1 = \theta^{\mathrm{init}}$ with empty round-indexed buffers, so warm-up neither changes the federated initialization nor mixes epoch and round histories.

In every round, all-example evaluation precedes subset refresh and local training. With refresh interval $\Delta \geq 1$, client k recomputes S_k^t when $t \geq W$ and $t \bmod \Delta = 0$; otherwise, $S_k^t = S_k^{t-1}$. The first eligible refresh is $\Delta\lceil W/\Delta \rceil$ if it does not exceed T. Thus W controls the scoring history and Δ the refresh frequency; we use $W = \Delta = 10$.

All original examples remain candidates, so excluded examples can return by entering their class quota; positive gain alone is insufficient. If a scheduled window is incomplete, client k retains S_k^{t-1} until a later scheduled refresh with a complete window. Zero imputation and unequal-length comparisons are disallowed. Any nonfinite loss is a numerical failure that must be resolved before scoring or training.

Algorithm 1 gives the full sequence. Beyond retained-set training, the protocol adds one full-local-data inference pass per round and the warm-up evaluations, but no scoring-related backward pass. Client k stores $O(n_k W)$ loss scalars; Appendix A.3 quantifies normalization and sorting costs. Loss histories, scores, and class statistics remain local, although this does not provide formal differential privacy.

3.5 Theoretical Analysis of Noise-Aware Pruning

We separate the effects of retained-label corruption, subset selection, and the realized federated update on the full-data clean empirical loss. The analysis gives pointwise identities and a conditional descent bound; it does not interpret LSS scores as noise probabilities. Proofs and counterexamples appear in Appendix A.2.

Logit versus parameter gradients. Fix (x, θ), let $z_\theta(x) \in \mathbb{R}^C$ be the logits, and set $p = f_\theta(x)$. For distinct labels y, y', the corresponding logit gradients are $p - e_y$ and $p - e_{y'}$, where e_c is the cth standard basis vector. If $p_y = \max_c p_c$, then $\langle p - e_y, p - e_{y'} \rangle = \|p\|_2^2 - p_y - p_{y'} \leq 0$ because $\|p\|_2^2 = \sum_c p_c^2 \leq p_y \sum_c p_c = p_y$, so the inner product is at most $-p_{y'}$. Parameter gradients apply the transpose of the $C \times d$ logit Jacobian to these vectors, which need not preserve the inner-product sign (Proposition A.1). Logit-level opposition therefore does not imply improved parameter-gradient alignment after pruning.

Noise removal versus selection bias. Using the same selected indices as F_k^t but replacing observed labels with latent clean labels, define

$$H_k^t(\theta) = \frac{1}{m_k^t} \sum_{i \in S_k^t} \ell_{ki}(\theta; y_{ki}), \qquad H^t(\theta) = \sum_k \pi_k H_k^t(\theta). \tag{8}$$

The fixed reference objective $H(\theta) = \sum_k \pi_k n_k^{-1} \sum_{i \in I_k} \ell_{ki}(\theta; y_{ki})$ is the clean empirical loss on all original data, not a population risk. H_k^t, H^t, and H are analytical references because clean labels are unobserved. At any comparison point θ, the realized sets S_k^t are held fixed; gradients do not differentiate through selection.

The realized retained-set corruption fraction is

$$\varepsilon_{k,S}^t = \frac{1}{m_k^t} \sum_{i \in S_k^t} \mathbf{1}\{\tilde{y}_{ki} \neq y_{ki}\}, \tag{9}$$

where $\mathbf{1}\{A\}$ is the indicator of event A. This realized fraction generally differs from the generating probability ε_k. At the comparison point, assume that a finite $G \geq 0$ bounds every relevant per-example parameter-gradient norm under both labels:

$$\|\nabla \ell_{ki}(\theta; \tilde{y}_{ki})\| \leq G, \qquad \|\nabla \ell_{ki}(\theta; y_{ki})\| \leq G.$$

This is a bound on individual gradient norms, not on losses or gradient differences. Correctly labeled examples contribute zero to $\nabla F_k^t - \nabla H_k^t$; for each mislabeled example, the triangle inequality gives

$$\|\nabla \ell_{ki}(\theta; \tilde{y}_{ki}) - \nabla \ell_{ki}(\theta; y_{ki})\| \leq \|\nabla \ell_{ki}(\theta; \tilde{y}_{ki})\| + \|\nabla \ell_{ki}(\theta; y_{ki})\| \leq G + G = 2G. \tag{10}$$

Because S_k^t contains $m_k^t \varepsilon_{k,S}^t$ such examples, averaging their contributions yields $\|\nabla F_k^t(\theta) - \nabla H_k^t(\theta)\| \leq 2G\varepsilon_{k,S}^t$. Adding and subtracting $\nabla H^t(\theta)$ and using $\pi_k \geq 0$ gives

$$\|\nabla F^t(\theta) - \nabla H(\theta)\| \leq 2G \sum_k \pi_k \varepsilon_{k,S}^t + \|\nabla H^t(\theta) - \nabla H(\theta)\|. \tag{11}$$

The first term is the corruption contribution; its weighted fraction $\sum_k \pi_k \varepsilon_{k,S}^t$ uses original client-size weights and is not, in general, the pooled fraction among all retained examples. The second term is the selection bias: the clean-gradient shift induced by replacing the full data with selected subsets. LSS guarantees neither a reduction in $\varepsilon_{k,S}^t$ nor preservation of clean-label coverage, and loss-dependent selection can destroy initially symmetric corruption. Likewise, for gradient dissimilarity $\sum_k \pi_k \|\nabla F_k^t(\theta) - \nabla F^t(\theta)\|^2$, decomposing each client gradient into clean retained-set and corruption components produces a cross term with unrestricted sign. Independent flips do not generally remove this term (Proposition A.3).

Conditional progress of the actual update. Let $\mathcal{F}_t$ be the sigma-algebra generated by the fixed dataset (including latent labels for analysis) and all model, loss, and selection history through round-t selection, but before drawing that round's local optimizer randomness. Conditional on $\mathcal{F}_t$, both θ^t and S_k^t are fixed. Conditional expectations therefore average only over the remaining randomness, such as mini-batch permutations; including latent labels in $\mathcal{F}_t$ does not make them observable to clients.

For a positive analysis scale γ, define the actual server update direction, its bias relative to the full-data clean gradient, and its conditional variance by

$$D^t = \frac{\theta^t - \theta^{t+1}}{\gamma}, b_t \quad = \mathbb{E}[D^t \mid \mathcal{F}_t] - \nabla H(\theta^t), V_t = \mathbb{E}\left[\|D^t - \mathbb{E}[D^t \mid \mathcal{F}_t]\|^2 \mid \mathcal{F}_t\right]. \tag{12}$$

The identity $\theta^{t+1} = \theta^t - \gamma D^t$ only rescales the realized server update; it does not define a new algorithmic step. The vector b_t is unrelated to mini-batch size b, and V_t is the total squared fluctuation of the update direction, not the variance of sample losses.

Assume that H is differentiable and has an L-Lipschitz gradient, for finite $L > 0$, on a region containing every realized update segment; that is, $\|\nabla H(\theta) - \nabla H(\theta')\| \leq L\|\theta - \theta'\|$. If $\mathbb{E}[\|D^t\|^2 \mid \mathcal{F}_t] < \infty$ and $0 < \gamma \leq 1/L$, then

$$\mathbb{E}[H(\theta^{t+1}) \mid \mathcal{F}_t] \leq H(\theta^t) - \frac{\gamma}{2}\|\nabla H(\theta^t)\|^2 + \frac{\gamma}{2}\|b_t\|^2 + \frac{L\gamma^2}{2} V_t. \tag{13}$$

Here γ is neither the local learning rate η nor an additional step size. The bound separates descent along the clean gradient from update bias and variance. In particular, $\|b_t\|^2 + L\gamma V_t < \|\nabla H(\theta^t)\|^2$ is sufficient for a strict conditional decrease of H. Because b_t also contains the effects of selection, multiple local steps, and momentum, equation 11 alone does not bound it. LSS supplies no noise-rate-vanishing bounds on b_t or V_t; therefore, equation 13 is a conditional descent criterion, not an unconditional convergence, generalization, or optimal-pruning-rate guarantee.

4 Experiments

We evaluate LSS on CIFAR-10/100 with synthetic symmetric label noise and on CIFAR-10N/100N with human annotation noise. After describing the common experimental protocol (Section 4.1), we report results under synthetic noise (Section 4.2) and real-world noise (Section 4.3), followed by component ablations (Section 4.4).

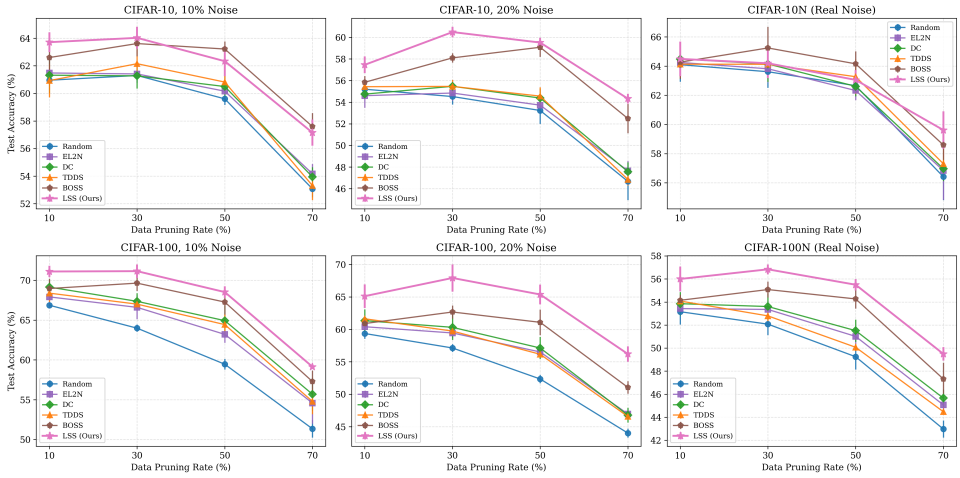

Figure 2: Test accuracy (%) versus data-pruning rate on CIFAR-10 (top row) and CIFAR-100 (bottom row) under synthetic symmetric noise (10%, 20%) and real-world human annotation noise (CIFAR-10N, CIFAR-100N). Results are averaged over four random seeds (mean±std). The panels compare how relative performance changes across noise settings and pruning rates.

4.1 Experimental Setup

Datasets, models, and partitioning. For synthetic-noise experiments, we use CIFAR-10 and CIFAR-100 Krizhevsky & Hinton (2009), injecting symmetric label noise zha independently into each client's training data at rates $\varepsilon \in \{0.1, 0.2\}$. For human annotation noise, we use CIFAR-10N and CIFAR-100N Wei et al. (2022). CIFAR-10N uses `aggregate_label`, whose average noise rate is 9.4%; CIFAR-100N uses the fine-grained `noisy_label` set, whose noise rate is 40.2%. We train a randomly initialized ResNet-18 on CIFAR-10 and CIFAR-10N, and an ImageNet-pretrained ResNet-18 on CIFAR-100 and CIFAR-100N. Every dataset is partitioned among $K = 10$ clients using a Dirichlet distribution with $\alpha = 0.1$ to induce non-IID heterogeneity.

Federated training. We use FedProx Li et al. (2020) with proximal coefficient $\lambda = 0.1$ and full client participation. Each client performs $E = 3$ local epochs of SGD per round with momentum $\nu = 0.9$, batch size $b = 64$, and fixed learning rate $\eta = 0.01$. Training lasts $T = 100$ communication rounds.

Pruning protocol and baselines. Algorithm 1 uses $E_{\text{warm}} = 3$ local full-data warm-up epochs, with epoch-end scoring for the initial selection. Federated training then restarts from the stored initial checkpoint with empty round histories. Clients score all examples at the global model every round and re-prune with $W = \Delta = 10$ using the class quotas in equation 4. Reported pruning rates are nominal targets. Retained-count flooring and the minimum-one constraint can make a client's

effective rate larger or smaller than the requested ρ, with the largest discrepancies occurring for small class counts.

We compare LSS with Random, EL2N Paul et al. (2021), DC Hao et al. (2025), TDDS Zhang et al. (2024), and BOSS Acharya et al. (2024). All scoring-based baselines use the same re-scoring and re-pruning schedule every 10 communication rounds. TDDS uses a temporal window of 5 observations and EMA decay 0.9 for variance aggregation. BOSS parameterizes its Beta distribution by $a = 1 + \bar{D} + c_a|S|$ and $b = 2 + c_b|S|$, with $|S| = (1 - \rho)N_c$ and c_a tuned for each subset size. LSS uses $\beta = 2.0$ to balance mean loss and loss gain.

Evaluation. Each experiment uses 4 random seeds. For each seed, we average test accuracy over the final 10 communication rounds, then report the mean and standard deviation across these seed-wise averages. Tables mark the best result in **bold** and the second-best result with underlining.

4.2 Results under Synthetic Label Noise

Table 1: Test accuracy (%) on CIFAR-10 and CIFAR-100 under symmetric label noise with different pruning rates ρ. **Bold**: best; underlined: second-best. All values are mean $\pm$ std over 4 independent runs.

	10% Noise				20% Noise			
Method	ρ=0.1	ρ=0.3	ρ=0.5	ρ=0.7	ρ=0.1	ρ=0.3	ρ=0.5	ρ=0.7
CIFAR-10 (Full-set acc: 61.48±0.83 @ 10% ; 54.49±1.06 @ 20%)								
Random	61.24±0.83	61.59±0.83	59.88±0.75	53.07±0.69	55.20±0.53	54.52±0.85	53.23±1.42	46.67±1.99
EL2N	61.58±1.20	61.62±0.79	60.45±0.87	54.17±0.83	54.61±1.29	54.85±0.81	53.73±0.87	47.65±1.06
DC	61.49±1.33	61.14±0.96	60.71±0.91	53.95±0.82	54.75±0.39	55.47±0.37	54.38±1.01	47.57±1.01
TDDS	61.29±1.45	62.27±1.05	61.16±1.02	52.84±1.48	55.44±0.76	55.46±0.70	54.56±0.95	46.85±0.63
BOSS	62.71±1.17	63.26±1.21	**62.94±0.83**	56.68±2.26	55.84±0.66	58.10±0.50	59.08±1.02	52.49±1.56
LSS (ours)	**63.75±0.72**	**64.10±0.81**	62.38±1.10	**57.15±1.10**	**57.46±0.89**	**60.49±0.56**	**59.52±0.51**	**54.32±0.53**
CIFAR-100 (Full-set acc:67.58±0.80 @ 10%; 60.95±1.90 @ 20%)								
Random	66.86±0.29	63.98±0.45	59.45±0.67	51.34±1.10	59.36±0.95	57.13±0.70	52.34±0.74	44.00±0.79
EL2N	67.91±1.44	66.60±1.49	63.20±1.06	54.57±2.60	60.41±1.57	59.45±0.85	56.51±0.56	46.94±0.81
DC	69.14±0.94	67.36±1.02	64.93±1.90	55.68±2.06	61.34±2.05	60.32±2.22	57.13±1.98	46.77±1.31
TDDS	68.38±1.11	67.00±0.78	64.44±1.38	54.78±1.60	60.98±0.85	59.74±0.33	56.16±0.53	46.56±0.55
BOSS	68.96±1.23	69.64±0.97	67.26±1.59	57.28±1.38	60.90±0.81	62.68±1.17	61.09±2.28	51.05±1.13
LSS (ours)	**71.10±0.73**	**71.14±0.85**	**68.52±0.72**	**59.13±0.11**	**65.12±2.10**	**67.93±2.42**	**65.38±1.75**	**56.20±1.37**

Table 1 compares performance under synthetic symmetric label noise. On CIFAR-10, LSS leads at three of four pruning rates under 10% noise; at $\rho = 0.5$, BOSS retains a 0.56-percentage-point advantage. Under 20% noise, LSS leads at every evaluated pruning rate. Averaged over the four rates, its margin over BOSS increases from 0.45 percentage points at 10% noise to 1.57 percentage points at 20% noise, suggesting that trajectory-aware scoring becomes more beneficial as label corruption grows.

On CIFAR-100 with ImageNet-pretrained ResNet-18, LSS outperforms all baselines at both noise levels. With 20% noise, its advantage over BOSS ranges from 4.22 to 5.25 percentage points, reaching 65.38% accuracy at $\rho = 0.5$ and peaking at 67.93% at $\rho = 0.3$. The larger gains on CIFAR-100 indicate stronger benefits in this setting. However, because CIFAR-100 uses different initialization from CIFAR-10, the contrast cannot be attributed solely to task difficulty.

4.3 Results under Real-World Label Noise

Table 2 evaluates human-annotated noise in CIFAR-10N and CIFAR-100N under the common protocol in Section 4.1, including final-10-round average accuracy, a fixed learning rate, and re-pruning every 10 rounds. CIFAR-10N uses randomly initialized ResNet-18, whereas CIFAR-100N uses ImageNet pre-training.

Table 2: Test accuracy (%) on CIFAR-10N aggre label (9% noise) and CIFAR-100N fine label (40.2% noise) with different pruning rates ρ. Bold: best, underlined: second-best.

Method	CIFAR-10N *(Full-set acc: 64.12±1.37)*				CIFAR-100N *(Full-set acc: 52.84±0.35)*			
	ρ=0.1	ρ=0.3	ρ=0.5	ρ=0.7	ρ=0.1	ρ=0.3	ρ=0.5	ρ=0.7
Random	64.10±1.34	63.61±1.28	62.66±0.59	56.42±1.84	53.15±1.27	52.08±1.10	49.25±1.29	42.97±0.86
EL2N	64.25±1.14	63.82±1.09	62.33±0.76	56.84±2.31	53.43±0.60	53.35±0.86	51.01±1.01	45.06±0.85
DC	64.48±1.21	64.15±1.42	62.61±0.78	56.97±0.99	53.85±1.18	53.61±1.31	51.51±1.12	45.66±0.86
TDDS	64.13±1.09	64.12±0.90	63.27±1.38	57.31±1.18	54.08±0.41	52.79±0.64	50.07±0.36	44.48±0.20
BOSS	64.27±1.33	**65.24**±1.67	**64.15**±0.99	58.60±1.18	54.15±0.56	55.08±0.81	54.27±0.97	47.32±1.63
LSS (ours)	**64.49**±1.37	64.55±1.24	63.05±0.78	**59.62**±1.48	**56.00**±1.24	**56.84**±0.49	**55.50**±0.58	**49.49**±0.68

On CIFAR-100N with 40.2% label noise, LSS leads across all evaluated pruning rates. Its margin over BOSS is 1.23–2.17 percentage points and is largest at $\rho = 0.7$. Results on CIFAR-10N with approximately 9% noise are mixed: BOSS leads at $\rho = 0.3$ and 0.5, whereas LSS leads at $\rho = 0.1$ and 0.7. At $\rho = 0.7$, LSS exceeds BOSS by 1.02 percentage points. Together with the synthetic-noise results, these findings show that LSS is particularly beneficial under stronger corruption and aggressive pruning, rather than uniformly superior in mild-noise settings.

4.4 Ablation Study

Table 3: Ablation study of LSS scoring components (last-10-round average test accuracy %, mean±std over 4 seeds).

	CIFAR-10N-worst (40.21% noise)				CIFAR-100N-fine (40.20% noise)			
Mode	ρ=0.3	ρ=0.4	ρ=0.5	ρ=0.7	ρ=0.3	ρ=0.4	ρ=0.5	ρ=0.7
full	**50.64±0.67**	**51.80±1.46**	**51.33±0.91**	**47.25±0.65**	56.84±0.49	56.59±0.52	**55.50±0.58**	**49.49±0.68**
mean_only	50.52±1.20	51.18±1.41	51.01±1.10	44.79±1.43	**57.55±0.18**	**56.81±0.28**	54.80±0.28	48.16±0.35
gain_only	48.42±0.61	48.31±1.00	48.12±0.81	43.65±0.62	54.54±0.99	52.51±0.46	50.95±0.35	44.81±0.64

Table 3 compares the full LSS score ($\beta = 2.0$) with variants using only mean loss or only loss gain. Removing mean loss reduces accuracy in every setting, showing that gain alone is insufficient. The gain term has a more selective effect: mean_only slightly exceeds the full score on CIFAR-100N-worse at $\rho = 0.3$ and 0.4. At $\rho = 0.7$, however, the full score improves over mean_only by 2.46 and 1.33 percentage points on CIFAR-10N-worst and CIFAR-100N-fine, respectively. Thus, learning progress is particularly complementary to mean loss when aggressive pruning risks discarding useful clean examples, but adding gain does not improve every setting.

A sensitivity sweep at $\rho = 0.7$ further supports $\beta = 2.0$ as a competitive default on both CIFAR-10N label sets, although the best value remains label-set dependent (Appendix A.4).

5 Conclusion

LSS enables dynamic, client-local data pruning by combining recent loss level with learning progress. Periodic re-scoring adapts the retained set as the global model evolves, while class-aware quotas maintain observed-label coverage. Experiments show the strongest benefits under severe label noise and aggressive pruning, while the ablations indicate that loss gain is particularly useful at high pruning rates and mean loss alone can remain competitive at lower rates. Scoring requires additional inference and per-sample loss histories, but no scoring-related backward pass or sample-level communication. The current evaluation is limited to ResNet-18, Dirichlet non-IID partitioning with $\alpha = 0.1$, and full client participation; future work should therefore examine other architectures, heterogeneity levels, and partial-participation settings, as well as adaptive schedules that adjust pruning and the contribution of loss gain according to the available data budget.

REPRODUCIBILITY STATEMENT

We provide the implementation and materials needed to reproduce the reported experiments in the anonymous supplementary archive `submit_code.zip`, including training and evaluation scripts, configuration files, and a `README` with instructions for environment setup, data preparation, and experiment execution. Experimental settings and implementation details are described in the main paper and appendix, and results are reported over multiple independent random seeds, with additional per-seed results provided in the supplementary material. The complete source code will be made publicly available on GitHub upon acceptance.

AI USE STATEMENT

Large language models (LLMs) were used to assist with software implementation and code editing, and to improve the grammar, clarity, and readability of the manuscript. The authors reviewed and verified the resulting code and manuscript content and take full responsibility for the correctness of the work.

REFERENCES

Abhinab Acharya, Dayou Yu, Qi Yu, and Xumin Liu. Balancing feature similarity and label variability for optimal size-aware one-shot subset selection. In Forty-first International Conference on Machine Learning, 2024.

Laha Ale, Ning Zhang, Xiaojie Fang, Xianfu Chen, Shaohua Wu, and Longzhuang Li. Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. IEEE Transactions on Cognitive Communications and Networking, 7(3):881–892, 2021. doi: 10.1109/TCCN.2021.3066619.

Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. arXiv preprint arXiv:2502.06905, 2025.

Chenhe Hao, Weiying Xie, Daixun Li, Haonan Qin, Hangyu Ye, Leyuan Fang, and Yunsong Li. Fedcs: Coreset selection for federated learning. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 15434–15443, 2025.

Xiaoqian Jiang and Jing Zhang. FedClean: A general robust label noise correction for federated learning. In Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 27772–27792, 2025.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In International conference on machine learning, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Anran Li, Guangjing Wang, Ming Hu, Jianfei Sun, Lan Zhang, Luu Anh Tuan, and Han Yu. Joint client-and-sample selection for federated learning via bi-level optimization. IEEE Transactions on Mobile Computing, 23(12):15196–15209, December 2024. ISSN 1536-1233. doi: 10.1109/TMC.2024.3455331. URL https://doi.org/10.1109/TMC.2024.3455331.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10713–10722, 2021a.

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.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. arXiv preprint arXiv:2102.07623, 2021b.

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.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. arXiv preprint arXiv:2103.14749, 2021.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. Advances in neural information processing systems, 34:20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. Advances in Neural Information Processing Systems, 33:17044–17056, 2020.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Zhou Daquan, Lei Shang, Baigui Sun, Xuansong Xie, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In International Conference on Learning Representations, volume 2024, pp. 39723–39744, 2024.

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. arXiv preprint arXiv:2111.12621, 2021.

Durga Sivasubramanian, Lokesh Nagalapatti, Rishabh Iyer, and Ganesh Ramakrishnan. Gradient coreset for federated learning. In 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2648–2657. IEEE, 2024.

Zhen Ling Teo, Liyuan Jin, Nan Liu, Siqi Li, Di Miao, Xiaoman Zhang, Wei Yan Ng, Ting Fang Tan, Deborah Meixuan Lee, Kai Jie Chua, et al. Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture. Cell Reports Medicine, 5 (2), 2024.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. Advances in neural information processing systems, 33:7611–7623, 2020.

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, 37(6):1205–1221, 2019. doi: 10.1109/JSAC.2019.2904348.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Clayton Scott. Learning with noisy labels revisited: A study using real-world human label noise. In International Conference on Learning Representations, 2022.

Haoran Xu, Jiaze Li, Wanyi Wu, and Hao Ren. Federated learning with sample-level client drift mitigation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pp. 21752–21760, 2025.

Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26223–26232, 2024.

A Appendix

A.1 Executable Training Protocol

Local optimizer. Algorithm 1 uses the variables and update in equation 2. Each round comprises $J_k^t = E\lceil m_k^t/b \rceil$ steps over E independently shuffled epochs, including the final incomplete batch. Momentum persists within a round and is reset across rounds. Warm-up uses the same update on I_k for E_{warm} epochs, with fixed anchor θ^{init}; the round-t anchor is θ^t. The proximal term never enters the sample score. The server aggregates final local parameters using the fixed original-size weights π_k and receives neither class counts nor retained set sizes.

Algorithm 1 LSS with warm-up and dynamic class-aware pruning

Require: $\{\mathcal{D}_k\}_{k=1}^K$, θ^{init}; $T, E, b, \Delta \in \mathbb{N}_{>0}$; $E_{\text{warm}}, W \in \mathbb{N}_{\geq 2}$; $0 \leq \rho < 1$, $\beta, \lambda \geq 0$, $\eta > 0$, $0 \leq \nu < 1$

Ensure: θ^{T+1} and client-local retained sets $\{S_k^T\}_{k=1}^K$

Fix $\pi_k = n_k / \sum_{j=1}^K n_j$ for $k = 1, \ldots, K$
Selection warm-up
for each client k in parallel **do**
  Form each I_{kc} and compute the fixed quotas m_{kc} by the floor-and-minimum-one rule in equation 4; set $m_k = \sum_c m_{kc}$
  Initialize a warm-up model at θ^{init} with zero momentum
  for epoch $e = 1, \ldots, E_{\text{warm}}$ **do**
    Train one shuffled epoch on I_k using equation 2 with anchor θ^{init}
    At the epoch-final model, evaluate all $i \in I_k$ with fixed preprocessing, in evaluation mode, and without gradients; store their observed-label losses
  end for
  Set $W_{\text{warm}} = \min(W, E_{\text{warm}})$; if the last W_{warm} snapshots are incomplete or nonfinite, resolve the failure and retry
  Using these snapshots, apply equation 5 with epoch indices and W_{warm}, followed by equation 6–equation 7, to obtain S_k^0
  Discard the warm-up model, momentum, and histories; set $\mathcal{B}_{ki} = \emptyset$ for each $i \in I_k$
end for
Server sets $\theta^1 = \theta^{\text{init}}$
Federated training
for round $t = 1, \ldots, T$ **do**
  Broadcast θ^t to all clients
  for each client k in parallel **do**
    In evaluation mode with fixed preprocessing and no gradients, compute $\ell_{ki}^{(t)} = \ell_{ki}(\theta^t; \tilde{y}_{ki})$ for all $i \in I_k$
    Resolve nonfinite losses; append $(t, \ell_{ki}^{(t)})$ to $\mathcal{B}_{ki}$ and delete records with $\tau < t - W + 1$
    Set $S_k^t = S_k^{t-1}$
    if $t \geq W$, $t \bmod \Delta = 0$, and every $\mathcal{B}_{ki}$ contains all losses for $\tau = t - W + 1, \ldots, t$ **then**
      Compute equation 5, normalize over I_k by equation 6, and update S_k^t by equation 7
    end if
    Set $m_k^t = |S_k^t|$, $J_k^t = E\lceil m_k^t/b \rceil$, $u_{k,0}^t = \theta^t$, and $v_{k,0}^t = 0$
    Train for E shuffled epochs on S_k^t using equation 2, including each final incomplete batch
    Upload $u_{k,J_k^t}^t$ only; retain all sample-level state locally
  end for
  Set $\theta^{t+1} = \sum_{k=1}^K \pi_k u_{k,J_k^t}^t$
end for

Boundary conditions. Clients with no original data are excluded before defining K and π_k. At $\rho = 0$, all examples are retained; if every observed class is a singleton, the same holds for any $\rho < 1$. More generally, with $C_k = |\{c : n_{kc} > 0\}|$, class-wise rounding gives $|m_k - (1-\rho)n_k| < C_k$, or equivalently $|\rho_k^{\text{eff}} - \rho| < C_k/n_k$. Flooring can increase the effective rate, while enforcing one retained example per nonempty class can decrease it. Increasing sample index breaks score

ties. Constant gain vectors normalize to zero, and negative gains are not clipped. Unavailable observations defer a scheduled refresh without imputation; nonfinite losses are numerical errors that must be resolved. Full participation and all-example scoring provide complete windows. Partial participation would require an additional observation-time policy and is not covered here.

A.2 Proofs and Limits of the Analysis

Unless an expectation is stated explicitly, the results below are pointwise in the realized dataset, selected sets, and comparison parameters. Latent clean labels are used only to define analytical reference quantities. These results establish identities and conditional bounds; they do not prove that LSS reduces corruption, selection bias, or optimization error.

A.2.1 Logit Gradients Do Not Determine Parameter Alignment

Proposition A.1 (Cross-entropy logit gradients)**.** *Let $p = \mathrm{softmax}(z)$ for finite logits $z \in \mathbb{R}^C$, let $\ell(z; y) = -\log[\mathrm{softmax}(z)]_y$, and let $y \neq y'$ be two candidate labels. Let e_c denote the cth standard basis vector and $\mathbf{1}_C$ the all-ones vector in $\mathbb{R}^C$. Then*

$$\langle p - e_y, p - e_{y'} \rangle = \|p\|^2 - p_y - p_{y'}. \tag{14}$$

If $p_y = \max_c p_c$, this quantity is at most $-p_{y'} < 0$. Moreover, suppose y is the clean label, $\tilde{y}$ follows the symmetric corruption model equation 3 with rate ε, and z is fixed independently of this fresh label flip. Then

$$\mathbb{E}[\nabla_z \ell(z; \tilde{y}) \mid z, y] = p - \left(1 - \frac{C\varepsilon}{C-1}\right) e_y - \frac{\varepsilon}{C-1} \mathbf{1}_C. \tag{15}$$

Neither result determines the sign of the corresponding parameter-gradient inner product. Nor does either result imply monotone improvement after loss-dependent selection.

Proof. For cross-entropy, $\nabla_z \ell(z; y) = p - e_y$. Since $y \neq y'$, $e_y^\top e_{y'} = 0$, and expansion gives equation 14. If p_y is maximal, then

$$\|p\|^2 = \sum_c p_c^2 \leq p_y \sum_c p_c = p_y,$$

which proves the stated upper bound. The inner product has no uniform negative margin, because it approaches zero as $p_y \to 1$; it equals $-1/2$ for balanced binary probabilities.

Conditioning on the fixed (z, y), the corruption model gives

$$\begin{aligned} \mathbb{E}[e_{\tilde{y}} \mid z, y] &= (1 - \varepsilon) e_y + \frac{\varepsilon}{C-1}(\mathbf{1}_C - e_y) \\ &= \left(1 - \frac{C\varepsilon}{C-1}\right) e_y + \frac{\varepsilon}{C-1} \mathbf{1}_C. \end{aligned}$$

Subtracting this vector from fixed p yields equation 15; at $\varepsilon = (C-1)/C$, the expected target is uniform. If z was fitted using the same realized corrupted label, the required independence is absent and this fresh-flip expectation does not describe that dependence.

Finally, let $J_\theta = \partial z / \partial \theta \in \mathbb{R}^{C \times d}$. The parameter gradients are $J_\theta^\top (p - e_y)$ and $J_\theta^\top (p - e_{y'})$, whose inner product is $(p - e_y)^\top J_\theta J_\theta^\top (p - e_{y'})$. Although $J_\theta J_\theta^\top$ is positive semidefinite, it need not preserve the sign of a cross inner product. For example, with $d = 1$, $p = (0.6, 0.3, 0.1)$, $y = 1$, $y' = 2$, and $J_\theta = (0, 0, 1)^\top$, the logit-gradient inner product is -0.44, whereas the parameter-gradient inner product is 0.01. □

A.2.2 Retained-Set Corruption and Selection Bias

Proposition A.2 (Gradient error at a fixed comparison point)**.** *Fix round t, parameters θ, and the realized sets S_k^t, which are held constant when differentiating. Suppose that, for every k and $i \in S_k^t$,*

$$\|\nabla \ell_{ki}(\theta; \tilde{y}_{ki})\| \leq G, \qquad \|\nabla \ell_{ki}(\theta; y_{ki})\| \leq G$$

for a finite $G \geq 0$. Then equation 11 holds, even if the realized selection depends on observed labels and prior loss histories.

Proof. Subtracting the retained observed- and clean-label gradients gives

$$\nabla F_k^t(\theta) - \nabla H_k^t(\theta) = \frac{1}{m_k^t} \sum_{i \in S_k^t} \big[\nabla \ell_{ki}(\theta; \tilde{y}_{ki}) - \nabla \ell_{ki}(\theta; y_{ki})\big].$$

The bracket is zero when $\tilde{y}_{ki} = y_{ki}$ and has norm at most $2G$ otherwise, by equation 10. Therefore,

$$\begin{aligned}\|\nabla F_k^t(\theta) - \nabla H_k^t(\theta)\| &\le \frac{2G}{m_k^t} \sum_{i \in S_k^t} \mathbf{1}\{\tilde{y}_{ki} \neq y_{ki}\} \\ &= 2G\varepsilon_{k,S}^t.\end{aligned}$$

Using $\nabla F^t(\theta) = \sum_k \pi_k \nabla F_k^t(\theta)$ and $\nabla H^t(\theta) = \sum_k \pi_k \nabla H_k^t(\theta)$, the aggregate error decomposes exactly as

$$\begin{aligned}\nabla F^t(\theta) - \nabla H(\theta) = &\sum_k \pi_k \big[\nabla F_k^t(\theta) - \nabla H_k^t(\theta)\big] \\ &+ \nabla H^t(\theta) - \nabla H(\theta).\end{aligned}$$

The triangle inequality and $\pi_k \ge 0$ then give

$$\begin{aligned}\|\nabla F^t(\theta) - \nabla H(\theta)\| &\le \sum_k \pi_k \|\nabla F_k^t(\theta) - \nabla H_k^t(\theta)\| \\ &\quad + \|\nabla H^t(\theta) - \nabla H(\theta)\| \\ &\le 2G \sum_k \pi_k \varepsilon_{k,S}^t + \|\nabla H^t(\theta) - \nabla H(\theta)\|.\end{aligned}$$

This argument is deterministic conditional on the realized data and sets; it requires no independence between selection and corruption. The bound does not compare the actual gradient error before and after pruning, because both the retained corruption fractions and the selection-bias term can change. □

Tightness and scope. The factor 2 is unavoidable under only the stated norm bounds. Consider a single binary example with scalar parameter θ and logits $z_\theta = (\theta, 0)$. At $\theta = 0$, the parameter gradients for labels 1 and 2 are $-1/2$ and $1/2$. Both have norm $G = 1/2$, but their difference has norm $1 = 2G$. If one client has only this example and its observed label is incorrect, then $S_1^t = I_1$, the selection-bias term is zero, and equation 11 holds with equality. Thus $2G$ is a worst-case constant, not the error contributed by every mislabeled example. The assumption is pointwise in θ; a uniform-in-training bound would require the same gradient bound at every parameter value visited.

A.2.3 Dissimilarity Includes a Cross Term

Proposition A.3 (Exact weighted dissimilarity decomposition)**.** *Fix round t and comparison point θ, and write $a_k = \nabla H_k^t(\theta)$, $\delta_k = \nabla F_k^t(\theta) - a_k$, $\bar{a} = \sum_k \pi_k a_k$, and $\bar{\delta} = \sum_k \pi_k \delta_k$. Define the nonnegative quantities B_{clean} and B_δ through $B_{\text{clean}}^2 = \sum_k \pi_k \|a_k - \bar{a}\|^2$ and $B_\delta^2 = \sum_k \pi_k \|\delta_k - \bar{\delta}\|^2$. Then*

$$B_{\text{obs}}^2 := \sum_k \pi_k \|\nabla F_k^t(\theta) - \nabla F^t(\theta)\|^2 = B_{\text{clean}}^2 + B_\delta^2 + 2\sum_k \pi_k \langle a_k - \bar{a}, \delta_k - \bar{\delta}\rangle, \tag{16}$$

where B_{obs} is the nonnegative square root. In particular, $B_{\text{obs}} \le B_{\text{clean}} + B_\delta$, but $B_{\text{obs}}^2 \ge B_{\text{clean}}^2$ need not hold.

Proof. Because $\nabla F^t(\theta) = \bar{a} + \bar{\delta}$, each centered observed gradient is

$$\nabla F_k^t(\theta) - \nabla F^t(\theta) = (a_k - \bar{a}) + (\delta_k - \bar{\delta}).$$

Expanding the weighted sum of squared norms gives equation 16. Viewing the centered client vectors as elements of the weighted product space with norm $\|(x_k)_k\|_\pi = (\sum_k \pi_k \|x_k\|^2)^{1/2}$, Minkowski's inequality gives $B_{\text{obs}} \le B_{\text{clean}} + B_\delta$.

The cross term need not be nonnegative, even under independent corruption. For a counterexample, consider two equally weighted binary logistic clients with class-2 probability $(1 + e^{-\theta})^{-1}$. Each has n identical unit inputs; client 1 has only clean label 1 and client 2 only clean label 2. At $\theta = 0$, their clean mean gradients are $1/2$ and $-1/2$, so $B^2_{\text{clean}} = 1/4$. Independently flip each label with probability $1/2$. If G_k denotes client k's observed mean gradient, then $\mathbb{E}[G_k] = 0$ and $\text{Var}(G_k) = 1/(4n)$. Since the clients are independent and equally weighted,

$$B^2_{\text{obs}} = \frac{(G_1 - G_2)^2}{4}, \qquad \mathbb{E}[B^2_{\text{obs}}] = \frac{\text{Var}(G_1) + \text{Var}(G_2)}{4} = \frac{1}{8n} < \frac{1}{4}.$$

Therefore $B^2_{\text{obs}} \geq B^2_{\text{clean}}$ cannot hold for every realization. For even n, the event that each client observes equal numbers of the two labels has positive probability and yields $B_{\text{obs}} = 0$. This example uses neither pruning nor label-dependent fitting. It also shows why independent flips do not imply $\mathbb{E}[\delta_k] = 0$ or eliminate the cross term. □

A.2.4 Conditional Descent and the Missing Drift Term

Proposition A.4 (Conditional descent of the realized server update)**.** *Under the smoothness and finite-second-moment conditions stated in Section 3.5, for any analysis scale $0 < \gamma \leq 1/L$ used in equation 12, the actual update of Algorithm 1 satisfies equation 13.*

Proof. Let $h = \nabla H(\theta^t)$. Conditional on $\mathcal{F}_t$, θ^t is fixed, hence so are $H(\theta^t)$ and h, while $\theta^{t+1} = \theta^t - \gamma D^t$. Smoothness along each realized update segment gives, almost surely,

$$H(\theta^{t+1}) \leq H(\theta^t) - \gamma\langle h, D^t\rangle + \frac{L\gamma^2}{2}\|D^t\|^2.$$

The finite-second-moment assumption permits conditional expectation. By equation 12 and the conditional bias–variance identity,

$$\mathbb{E}[D^t \mid \mathcal{F}_t] = h + b_t, \qquad \mathbb{E}[\|D^t\|^2 \mid \mathcal{F}_t] = \|h + b_t\|^2 + V_t.$$

Substitution yields

$$\begin{aligned}\mathbb{E}[H(\theta^{t+1}) \mid \mathcal{F}_t] &\leq H(\theta^t) - \gamma\langle h, h + b_t\rangle + \frac{L\gamma^2}{2}\big(\|h + b_t\|^2 + V_t\big) \\ &\leq H(\theta^t) - \frac{\gamma}{2}\|h\|^2 + \frac{\gamma}{2}\|b_t\|^2 + \frac{L\gamma^2}{2}V_t.\end{aligned}$$

The second inequality follows from the exact identity

$$\begin{aligned}-\gamma\langle h, h + b_t\rangle + \frac{L\gamma^2}{2}\|h + b_t\|^2 = -\frac{\gamma}{2}\|h\|^2 + \frac{\gamma}{2}\|b_t\|^2 \\ - \frac{\gamma}{2}(1 - L\gamma)\|h + b_t\|^2.\end{aligned}$$

Since $L\gamma \leq 1$, the last term is nonpositive and may be dropped, proving equation 13. The argument uses neither unbiased local SGD, equal client step counts, independence of selected labels, nor a PL condition. □

Interpretation and limitations. Rearranging equation 13 gives

$$\mathbb{E}[H(\theta^{t+1}) \mid \mathcal{F}_t] - H(\theta^t) \leq -\frac{\gamma}{2}\big(\|\nabla H(\theta^t)\|^2 - \|b_t\|^2 - L\gamma V_t\big).$$

Thus $\|b_t\|^2 + L\gamma V_t < \|\nabla H(\theta^t)\|^2$ is sufficient, but not necessary, for strict conditional descent. Both b_t and V_t depend on the analysis scale γ through D^t; γ is not an additional algorithmic step size. The bias includes corruption, selection, multiple local steps, momentum, and client drift, so reducing the retained corruption fraction alone does not ensure this condition.

Client dissimilarity and stochastic variance also cannot by themselves bound displacement from the broadcast model. Let every client have the noiseless objective $F_k(\theta) = \|\theta\|^2/2$, use zero initial momentum, and take one local step from $\theta^t \neq 0$. The gradient of the proximal term is initially zero, so

$$u^t_{k,1} - \theta^t = -\eta\theta^t, \qquad \|u^t_{k,1} - \theta^t\|^2 = \eta^2\|\theta^t\|^2 > 0.$$

All clients have identical gradients and the update is deterministic; hence both gradient dissimilarity and stochastic variance are zero. A valid drift bound must therefore also account for the common optimization direction.

Finally, mini-batch averaging does not generally produce variance proportional to $1/n_k$; a $1/b$ scaling requires an appropriate sampling-independence assumption and changes under sampling without replacement. Together with the round-varying objectives F^t, these facts prevent directly importing a fixed-objective FedProx or SCAFFOLD convergence guarantee (Li et al., 2020; Karimireddy et al., 2020) without additional assumptions.

A.3 Overall Pipeline and Complexity Analysis

The protocol has one client-local initialization phase followed by the T federated rounds in Algorithm 1. Once, client k forms its observed-class index sets and fixed quotas. It then performs the full-data warm-up, evaluates all n_k examples after each warm-up epoch, constructs S_k^0, and discards the warm-up model and histories. In each federated round, the client (i) evaluates all original examples at the broadcast model, (ii) updates the round-indexed loss buffer and, only at an eligible refresh, recomputes scores and S_k^t, (iii) trains on the retained set, and (iv) uploads only its final local model. The server then aggregates with the fixed weights π_k. Thus changing the refresh interval affects score construction and selection, but not the full-data inference pass performed in every round.

We now separate setup, recurring, and periodic costs. Let $C_k = |\{c : n_{kc} > 0\}|$, let d be the number of model parameters, and let c_{f} and c_{b} denote the per-example costs of a forward pass and backward pass, respectively; these are arithmetic costs rather than a claim of linear wall-clock scaling under batching. The following counts are per client; summing them over k gives total client work, whereas parallel wall-clock time depends on the slowest participating client. Class-index construction and quota calculation take $\Theta(n_k + C_k)$ time and $O(n_k + C_k)$ index and counter storage if the class lists are materialized. Since the quotas are fixed, $m_k^t = m_k$ and the number of optimizer steps per round is $J_k = E\lceil m_k/b \rceil$. Similarly, the warm-up uses $J_{k,\mathrm{warm}} = E_{\mathrm{warm}}\lceil n_k/b \rceil$ steps and $W_{\mathrm{warm}} = \min(W, E_{\mathrm{warm}})$ snapshots. Its model-pass cost is

$$E_{\mathrm{warm}} n_k (c_{\mathrm{f}} + c_{\mathrm{b}}) + E_{\mathrm{warm}} n_k c_{\mathrm{f}},$$

for full-data training and epoch-end evaluation. In addition, parameter, momentum, and proximal updates cost $O(J_{k,\mathrm{warm}} d)$; recording the snapshots costs $O(E_{\mathrm{warm}} n_k)$ scalar operations; and the one-time initial selection costs $O(n_k W_{\mathrm{warm}} + \sum_{c:n_{kc}>0} n_{kc} \log n_{kc})$ with direct window summation and within-class sorting. The warm-up performs no model aggregation, and its snapshot storage is released before round 1.

Let $\mathcal{R}_k$ be the rounds in which client k actually completes a refresh. With direct evaluation of equation 5, its round-t work is

$$n_k c_{\mathrm{f}} + E m_k (c_{\mathrm{f}} + c_{\mathrm{b}}) + O(n_k + J_k d) + \mathbf{1}\{t \in \mathcal{R}_k\} O\left(n_k W + \sum_{c:n_{kc}>0} n_{kc} \log n_{kc} \right). \tag{17}$$

The first term is the mandatory all-example scoring pass; the second is retained-set training. The $O(n_k)$ term covers insertion and expiration in fixed-size loss buffers, while $O(J_k d)$ covers the momentum, proximal, and parameter updates. At a refresh, forming all window means directly costs $O(n_k W)$; endpoint gains, extrema, normalization, and score construction add $O(n_k)$. Sorting score–index pairs within observed classes costs $O(\sum_{c:n_{kc}>0} n_{kc} \log n_{kc})$, which is at most $O(n_k \log n_k)$ because the class sets partition I_k. A semantics-equivalent implementation may maintain one rolling sum per example while updating the buffers. This keeps the per-round buffer work at $O(n_k)$ and reduces the mean computation at a refresh from $O(n_k W)$ to $O(n_k)$, without changing the $O(n_k W)$ history storage.

Under the specified full-participation protocol, absent a deferred incomplete window, the first refresh is $t_0 = \Delta\lceil W/\Delta \rceil$ and the number of scheduled refreshes is $R = 0$ if $t_0 > T$, and otherwise $R = 1 + \lfloor (T - t_0)/\Delta \rfloor$. Hence the cumulative federated client cost is the first three terms of equation 17 multiplied by T, plus the refresh term multiplied by $|\mathcal{R}_k| \le R$. In comparison, full-data FedProx uses approximately $E n_k (c_{\mathrm{f}} + c_{\mathrm{b}})$ model-pass work per round. Ignoring optimizer and selection

bookkeeping, LSS reduces this work only when

$$E(n_k - m_k)(c_\mathrm{f} + c_\mathrm{b}) > n_k c_\mathrm{f};$$

the periodic scoring and sorting costs make the actual break-even condition stricter. Computational savings are therefore not guaranteed, particularly at low effective pruning rates.

During federated training, the pruning-specific persistent client state is W losses (and their round indices) for each original example, requiring $\Theta(n_k W)$ scalars. Retained and class indices, quotas, rolling sums if used, and temporary means, gains, and scores require $O(m_k + C_k + n_k) = O(n_k)$ additional storage. Only the last W_warm epoch snapshots are needed for initialization, so a warm-up ring buffer uses $\Theta(n_k W_\mathrm{warm})$ transient storage; retaining all epoch snapshots instead would use $\Theta(n_k E_\mathrm{warm})$. These costs are in addition to the usual model, anchor, momentum, and gradient state, which is $O(d)$, and architecture-dependent mini-batch activations. No sample-level state is stored on the server.

Communication is unchanged from the corresponding full-participation FedProx protocol: each round sends one d-parameter broadcast to, and receives one d-parameter model from, every client, totaling $2Kd = \Theta(Kd)$ transmitted scalars and $\Theta(Kd)$ server aggregation work per round, or $2TKd$ model scalars over T rounds. Establishing the fixed original-size weights requires at most one client-size scalar per client; class counts, retained-set sizes, losses, and scores are never transmitted. Including the K weights, the server can aggregate with $O(d+K)$ memory using a streaming accumulator (or $O(Kd + K)$ if all uploads are buffered), and LSS adds no server-side state. These communication properties do not constitute a formal privacy guarantee, since the uploaded models remain data dependent.

A.4 Sensitivity to the Loss-Gain Weight

Table 4 varies $\beta \in \{0.5, 1.0, 2.0, 4.0, 8.0\}$ at the most aggressive evaluated pruning rate, $\rho = 0.7$. On CIFAR-10N-aggre, mean accuracy increases from 55.92% at $\beta = 0.5$ to a maximum of 59.62% at the default $\beta = 2.0$. Increasing β to 4.0 or 8.0 lowers the mean by only 0.20 and 0.50 percentage points, respectively. On CIFAR-10N-worst, $\beta = 1.0$ gives the highest mean accuracy (47.28%), but exceeds $\beta = 2.0$ by only 0.03 points, which is small relative to the corresponding standard deviations. In contrast, larger values reduce the mean to 46.33% and 45.21%, indicating that heavily weighting short-window loss gain is less effective under this noise setting. Overall, $\beta = 2.0$ is a robust default across the two label sets, although the best value can depend on the noise regime.

Table 4: Sensitivity to the loss-gain weight β on CIFAR-10N at $\rho = 0.7$. Entries are last-10-round average test accuracy (%), reported as mean $\pm$ standard deviation over four seeds.

Label set	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.0$ (default)	$\beta = 4.0$	$\beta = 8.0$
CIFAR-10N-aggre ($\approx 9\%$ noise)	55.92±1.25	58.26±1.27	**59.62±1.28**	59.42±1.40	59.12±1.56
CIFAR-10N-worst ($\approx 40\%$ noise)	46.16±0.53	**47.28±0.65**	47.25±0.56	46.33±0.38	45.21±0.54

A.5 Training Dynamics and Convergence

Figures 3 and 4 complement the final-round summaries by showing how performance evolves on CIFAR-10 with 20% symmetric label noise. Across all criteria, most of the initial accuracy gain and loss reduction occurs during approximately the first 10–20 rounds, followed by slower improvement. The trajectories for $\rho \in \{0.1, 0.3, 0.5\}$ are comparatively close and often overlap; moderate pruning can match or exceed $\rho = 0.1$, so performance does not degrade monotonically throughout this range. In contrast, $\rho = 0.7$ separates early, makes less subsequent progress, and ends with lower accuracy and higher loss for every criterion. Its uncertainty bands are also generally wider, showing greater sensitivity to the random seed when little data is retained.

Differences among criteria are modest under mild and moderate pruning but become pronounced at $\rho = 0.7$. LSS continues to improve through later rounds and maintains the highest mean accuracy and lowest mean loss among the aggressively pruned trajectories; BOSS is the closest competitor, while DC, EL2N, TDDS, and Random level off lower. This separation develops after the shared initial learning phase rather than from a faster start. The curves therefore support the aggregate

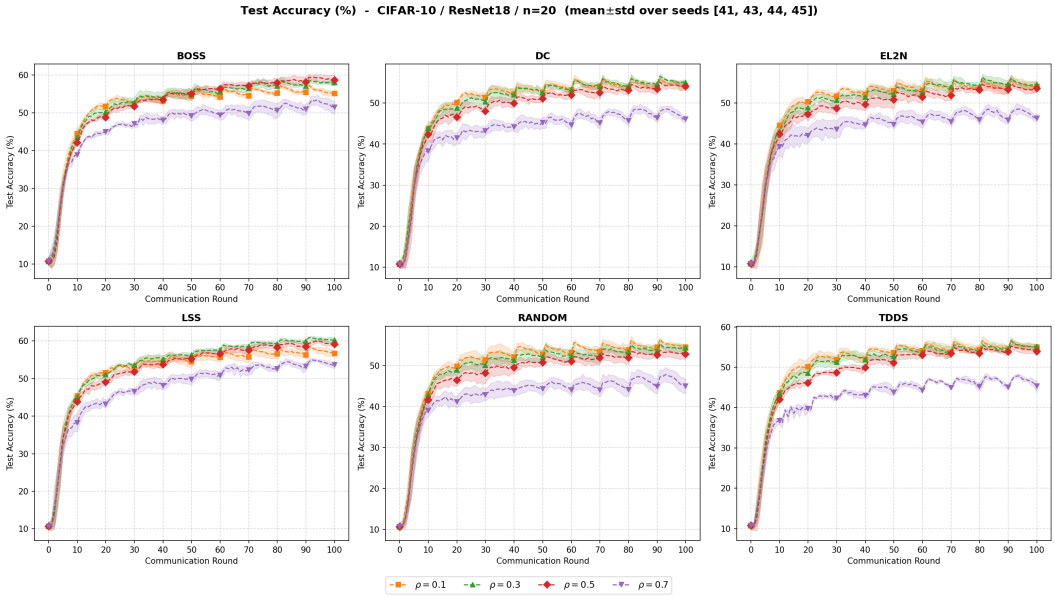

Figure 3: Test accuracy over communication rounds on CIFAR-10 with 20% symmetric label noise using ResNet-18. Each panel shows one pruning criterion; curves are means over four seeds and shaded bands indicate $\pm$ one standard deviation. The nominal client-level rate ρ is applied within each observed class before integer rounding and minimum-one retention.

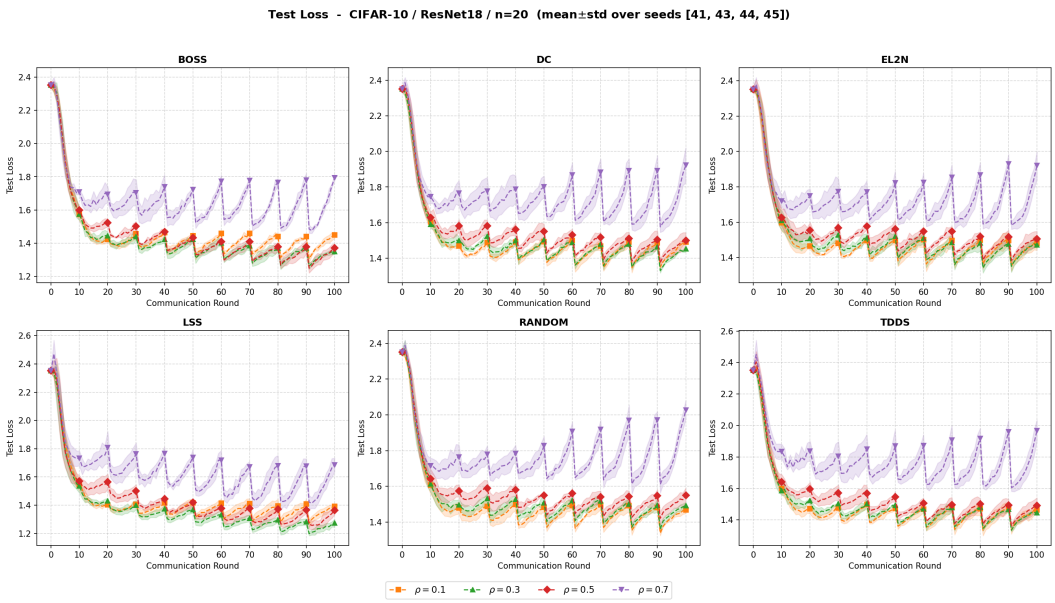

Figure 4: Test loss over communication rounds on CIFAR-10 with 20% symmetric label noise using ResNet-18. Each panel shows one pruning criterion; curves are means over four seeds and shaded bands indicate $\pm$ one standard deviation. The nominal client-level rate ρ is applied within each observed class before integer rounding and minimum-one retention.

finding that LSS is most beneficial under aggressive pruning and are consistent with the ablation result that loss gain is more useful when the retained-data budget is small. At lower pruning rates, the overlapping bands and changing within-panel ordering qualify any claim of uniform superiority.

The loss curves exhibit an approximately 10-round sawtooth pattern aligned with the re-pruning interval, especially at $\rho = 0.7$, whereas the accuracy curves are smoother. Thus, short-term loss changes around subset refreshes do not translate one-for-one into changes in classification accuracy. The broad inverse relationship still holds—trajectories with persistently lower loss generally attain higher accuracy—but the ordering of the mild-pruning curves is not identical at every round. These periodic fluctuations also motivate the last-10-round averaging used for the aggregate results rather than relying on a single terminal checkpoint. Because the round-specific training objective changes when the retained subset is refreshed, these plots characterize empirical stabilization of test metrics, not convergence to a fixed objective.