CRAFT: Conflict-Resolved Aggregation for Federated Training

Ziqi Wang
Friedrich-Alexander-Universität
Erlangen-Nürnberg
ziqi.wang@fau.de

Qiang Liu
School of CIT
Technical University of Munich
qiang7.liu@tum.de

Nils Thuerey
Technical University of Munich
Munich Center for Machine Learning
nils.thuerey@tum.de

Abstract

The aggregation of conflicting client updates remains a fundamental bottleneck in federated learning (FL) under heterogeneous data distributions. Naive averaging can produce a global update that improves the global objective while conflicting with specific clients, causing degradation for those clients. In this work, we propose CRAFT (Conflict-Resolved Aggregation for Federated Training), a new aggregation framework that treats the global update as a geometric correction problem. We formulate aggregation as finding the update closest to a reference direction while satisfying conflict-free alignment constraints. We derive a closed-form expression for the constrained optimization problem, avoiding the computational overhead of iterative solvers. Furthermore, we use a layer-wise adaptation to address conflicts at varying feature granularities. We provide a theoretical analysis showing that CRAFT promotes a common-descent structure and mitigates conflicts through its projection geometry. Extensive experiments on heterogeneous benchmarks demonstrate that CRAFT improves the accuracy of the global model while reducing performance disparity across clients compared with state-of-the-art baselines. The source code for CRAFT is available at https://github.com/tum-pbs/CRAFT.

1 Introduction

Federated learning (FL) [19] enables collaborative training of machine learning models across decentralized devices, or clients, without sharing raw private data. Despite its privacy advantages, FL faces a critical challenge from heterogeneity, where client data are typically not independent and identically distributed (non-IID) and are often imbalanced [9].

In practice, data heterogeneity induces divergent client updates. Conventional aggregation methods, such as FedAvg [19], compute weighted averages of local client updates. However, the averaged direction may benefit some clients while conflicting with others, thereby leading to substantial disparities in per-client accuracy [13]. Consequently, beyond improving the overall mean accuracy, mitigating disparities in per-client performance is also of central importance.

Existing approaches address this challenge through several mechanisms. Heterogeneity-mitigating methods regularize or normalize local training, thereby limiting discrepancies between local and global updates [12, 30]. Meanwhile, fairness-aware methods either modify the federated objective or formulate FL as a multi-objective optimization problem that jointly accounts for client losses,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

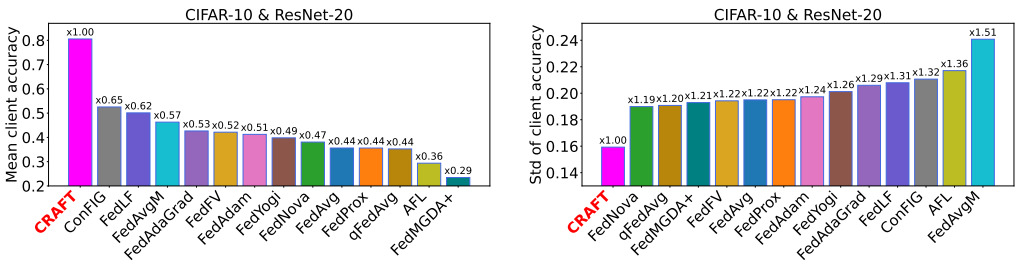

Figure 1: Mean (left, ↑) and standard deviation (right, ↓) of per-client accuracy. CRAFT achieves higher mean client accuracy while simultaneously reducing client-level disparity.

with the goal of improving worst-client performance [7, 20]. In addition, personalized FL methods improve client-specific performance by maintaining individualized models for each client [14, 29, 34]. More recently, conflict-aware methods have directly manipulated the geometry of client updates to reduce conflicts through vector projection or geometric conflict resolution [17, 22, 31]. However, their performance can still be limited by heuristic conflict-resolution rules.

In this paper, we propose CRAFT (**C**onflict-**R**esolved **A**ggregation for **F**ederated **T**raining). Instead of averaging, CRAFT formulates aggregation as a reference-anchored constrained least-squares problem that finds the global update $\mathbf{g}$ closest to a prespecified reference direction while satisfying conflict-free alignment constraints $\langle \mathbf{g}, \mathbf{g}_i \rangle > 0$ for all clients i. Our key innovations are as follows.

- **Conflict-free aggregation by problem reformulation.** We reformulate federated aggregation as a reference-anchored constrained least-squares problem rather than a weighted averaging rule. This formulation directly enforces positive alignment between the global update and every participating client update, yielding a conflict-free solution whenever the conflict-free alignment constraints are feasible. We further derive an explicit closed-form solution based on a Moore–Penrose correction, avoiding iterative projection or quadratic-programming solvers and keeping the server-side computation lightweight.
- **Momentum-like reference and layer-wise adaptation.** We use the previous-round global update as the reference direction, allowing CRAFT to preserve useful temporal information while correcting the components that conflict with the current active clients. We further introduce a layer-wise implementation that resolves conflicts at the granularity of model layers, making the method suitable for deep neural networks. The accompanying theoretical analysis characterizes the projection geometry, establishes its common-descent structure, and provides finite-time convergence guarantees.
- **Strong empirical performance and plug-in extensibility.** Extensive experiments on heterogeneous datasets, deep models, and large client populations show that CRAFT improves mean accuracy and, when mean accuracy is comparable, further enhances client-level fairness by reducing accuracy disparity and improving tail-client performance. CRAFT is designed for training a shared global model rather than client-specific personalized models. Nevertheless, it can serve as a generic aggregation operator that replaces weighted averaging in existing personalized FL pipelines, where this simple substitution already yields substantial performance gains and demonstrates the broad extensibility of CRAFT.

2 Related work

Heterogeneity and fairness in FL. Statistical and system heterogeneity are central challenges in FL [9]. FedAvg often struggles with non-IID data, leading to slow convergence or biased global models. FedProx [12] proposes regularizing local training to keep updates closer to the global model. FedNova [30] proposes normalizing client updates to address objective inconsistency caused by varying local training epochs. Adaptive server optimizers, such as FedAvgM [6] and FedOpt [25], have also been explored to improve convergence under heterogeneity [9]. These methods primarily target global optimization performance, but high mean accuracy does not guarantee reliable accuracy for every client. Client-level fairness aims to reduce performance disparity across clients [13]. To this end, Mohri et al. [20] optimizes a worst-case objective via minimax optimization, and Li et al. [13]

proposes q-FedAvg to reweight the federated objective in favor of lower-performing clients. These approaches address fairness by adjusting the objective function. By contrast, CRAFT tackles the fairness problem directly at the model-update level through conflict-free alignment constraints.

Conflict mitigation. In FL, FedFV [31] identifies gradient conflicts as a key source of unfairness and proposes iteratively removing conflicting components before averaging. FedLF [23] and FedMDFG [22] further develop this direction through layer-wise constraints or fairness-guided gradient adjustment. Similar gradient-conflict issues also arise in multi-task learning (MTL) [27, 33], continual learning (CL) [4, 26], and physics-informed neural networks (PINNs) [8, 17]. For training PINNs, ConFIG [17] formulates conflict resolution as solving a linear system. However, it does not exploit round-wise information to guide the resolution process. To address this limitation, CRAFT formulates conflict resolution as a constrained least-squares problem with a momentum-like reference direction, yielding a more consistent and effective solution.

Multi-objective optimization. In multi-objective formulations of FL, each client loss is treated as an objective rather than being absorbed into a single global objective. This is useful for fairness because improving one client should not come at the cost of degrading others. AFL [20] follows this idea by solving a minimax problem over model parameters and client weights, where the inner maximization selects the loss weighting that emphasizes clients with larger losses. FedMGDA+ [7] makes the multi-objective formulation explicit by treating each client loss as a separate objective and computing a minimum-norm convex combination of client gradients to obtain a common descent direction. These methods provide principled fairness-oriented objectives, but the aggregate direction is reconstructed from the current client gradients and may become small or unstable under strong heterogeneity. In contrast, CRAFT anchors aggregation to a momentum-like reference and applies the minimum correction needed to enforce positive alignment with every participating client.

Relation to personalized FL. Personalization methods handle heterogeneity by maintaining client-specific models alongside a shared model. Per-FedAvg [3] learns an initial shared model that can be quickly adapted to each client through a few local gradient steps. FedRep [2] decomposes the model into a shared backbone and client-specific heads, enabling personalization by locally fine-tuning the client-specific components. Ditto [14], instead of separating model components, learns an additional personalized model for each client by regularizing it toward the shared global model with a proximal term. CRAFT targets training a single global model, where the server outputs one shared model for all clients. Nevertheless, its conflict-free aggregation mechanism can be readily integrated into existing personalized FL frameworks, as demonstrated in the experiments. Therefore, CRAFT provides a more robust foundation that complements, rather than competes with, personalization strategies.

3 Problem setup and preliminaries

3.1 Federated optimization

FL training with N clients can be formulated as minimizing a weighted sum of local objectives:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) \coloneqq \sum_{i=1}^{N} \rho_i f_i(\boldsymbol{\theta}), \tag{3.1}$$

where client $i \in [N] \coloneqq \{1, 2, \ldots, N\}$ holds a local dataset $\mathcal{D}_i$ and minimizes $f_i(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}_i}[\ell(\boldsymbol{\theta};\xi)]$, with $\boldsymbol{\theta} \in \mathbb{R}^d$ denoting the model parameters and ξ denoting a random sample. The weights ρ_i are typically chosen as $\rho_i = |\mathcal{D}_i| / \sum_{j=1}^{N} |\mathcal{D}_j|$.

FL training proceeds in communication rounds. At each round $t = 0, 1, \ldots, T-1$, the server broadcasts the current global model $\boldsymbol{\theta}_t$ to a subset of participating clients $\mathcal{S}_t$ for local training.

Client-side. Each client $i \in \mathcal{S}_t$ initializes its local model as $\boldsymbol{\theta}_i^{t,0} = \boldsymbol{\theta}_t$ and performs K_i steps of local stochastic gradient descent. For $k = 0, \ldots, K_i - 1$, the update is given by

$$\boldsymbol{\theta}_i^{t,k+1} = \boldsymbol{\theta}_i^{t,k} - \eta_l \nabla f_i(\boldsymbol{\theta}_i^{t,k}; \xi_i^{t,k}), \tag{3.2}$$

where η_l is the local learning rate and $\xi_i^{t,k}$ denotes a mini-batch sampled from $\mathcal{D}_i$. After local training, client i sends the following accumulated update to the server for aggregation.

$$\mathbf{g}_i^t \coloneqq \boldsymbol{\theta}_t - \boldsymbol{\theta}_i^{t,K_i}. \tag{3.3}$$

Server-side. Upon receiving updates from participating clients, the server aggregates them to form a global update direction $\mathbf{g}_t$ and updates the global model as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_g \mathbf{g}_t. \tag{3.4}$$

Here, η_g denotes the server learning rate. In FedAvg, the aggregation is given by $\mathbf{g}_t = \sum_{i \in \mathcal{S}_t} \rho_i^t \mathbf{g}_i^t$, where ρ_i^t denotes the data-proportional weights renormalized over $\mathcal{S}_t$.

3.2 Conflict definition and fairness issue

Conflict definition. While the global FL objective (3.1) provides a unified optimization target, it does not explicitly account for conflicts between the aggregate update and individual client updates. Let $\mathbf{g}_i^t$ denote the update of client i at round t. We say that a *conflict* occurs between the aggregate update $\mathbf{g}_t$ and client i if $\langle \mathbf{g}_i^t, \mathbf{g}_t \rangle < 0$. Since $\mathbf{g}_i^t$ approximates a scaled local gradient, the global update changes the local objective by approximately $-\eta_g \langle \nabla f_i(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle$. Thus, a negative inner product indicates that the global update can increase client i's loss to first order and often reduce its test accuracy.

Fairness issue. Conflicts create a fairness issue, since under data heterogeneity, averaging can improve mean accuracy by favoring clients aligned with the aggregate update while degrading clients in conflicting directions. This can lead to a large standard deviation of per-client accuracy. We therefore treat mean accuracy as a prerequisite, since a small standard deviation is uninformative if all clients remain at low accuracy. Given comparable mean accuracy, better fairness means reducing this standard deviation, *i.e.*, narrowing the accuracy gap between head and tail clients.

4 The CRAFT framework

In this section, we introduce the CRAFT framework. We begin by establishing the notation and intuition for conflict-free updates, then present our detailed CRAFT formulation.

4.1 Conflict-free update

Our approach is inspired by recent advances in MTL and PINNs, especially the ConFIG algorithm [17]. Define the normalization operator $\mathcal{U}(\mathbf{v}) := \mathbf{v}/(\|\mathbf{v}\| + \varepsilon)$ with $\varepsilon > 0$. Given m active clients, let $\mathbf{U}_t = [\mathbf{u}_1^t, \ldots, \mathbf{u}_m^t]^\top \in \mathbb{R}^{m \times d}$ be the normalized update matrix, with $\mathbf{u}_i^t = \mathcal{U}(\mathbf{g}_i^t)$ denoting the normalized local update of client i. To ensure positive alignment between the global update $\mathbf{g}$ and all participating client updates, we impose the following inner-product constraints:

$$\mathbf{U}_t \mathbf{g} = \boldsymbol{\rho}_t, \tag{4.1}$$

where $\boldsymbol{\rho}_t = [\rho_1^t, \ldots, \rho_m^t]^\top$ collects the positive round-t targets for active clients. The design choices of targets are elaborated in Appendix B.2. Since $\rho_i^t > 0$ for all i, any feasible solution of (4.1) is conflict-free with respect to all client updates. However, (4.1) does not uniquely determine the update direction. When $\mathbf{U}_t$ has full row rank, the system is feasible for any target $\boldsymbol{\rho}_t \in \mathbb{R}^m$. In federated learning, it is common for the model dimension d to be much larger than the number of participating clients m, *i.e.*, $d \gg m$. Thus, the system is underdetermined and admits infinitely many feasible solutions. We discuss this feasibility condition in Appendix A.1.

Specifically, ConFIG chooses the feasible direction with the minimum Euclidean norm. Let $\mathbf{A}^\dagger$ denote the Moore–Penrose inverse of a matrix $\mathbf{A}$. For a consistent linear system, the pseudoinverse returns the unique minimum-norm solution. Applying this property to (4.1), ConFIG returns $\mathbf{g}_t = \mathbf{U}_t^\dagger \boldsymbol{\rho}_t$, which is equivalently the minimizer of

$$\min_{\mathbf{g} \in \mathbb{R}^d} \ \frac{1}{2}\|\mathbf{g}\|^2 \quad \text{s.t.} \quad \mathbf{U}_t \mathbf{g} = \boldsymbol{\rho}_t. \tag{4.2}$$

This construction produces a conflict-free direction, but it does not take historical information into account. In FL, consecutive communication rounds are coupled through the evolving global model, and the previous aggregate direction carries useful information about the server's optimization trajectory. Reconstructing the update from scratch at every round can discard this historical signal, attenuate the update norm, and introduce unnecessary variation under partial participation. This motivates a different selection principle that maintains the same conflict-free alignment constraints while selecting the feasible update closest to a stable reference direction.

4.2 Improved conflict resolution via correction

Building on the above observation, CRAFT formulates aggregation as a *correction* problem. Instead of constructing an update direction from the origin, we seek an update $\mathbf{g}$ that remains as close as possible to a reference direction $\hat{\mathbf{g}}_t$ while satisfying the same conflict-free alignment constraints.

$$\min_{\mathbf{g}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{g}-\hat{\mathbf{g}}_t\|^2 \quad \text{s.t.} \quad \mathbf{U}_t\mathbf{g} = \boldsymbol{\rho}_t. \tag{4.3}$$

Let $\boldsymbol{\Delta} \coloneqq \mathbf{g} - \hat{\mathbf{g}}_t$ denote the *correction* applied to the reference direction. Then (4.3) is equivalently written as the following minimum-norm correction problem.

$$\min_{\boldsymbol{\Delta}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{\Delta}\|^2 \quad \text{s.t.} \quad \mathbf{U}_t\boldsymbol{\Delta} = \boldsymbol{\rho}_t - \mathbf{U}_t\hat{\mathbf{g}}_t. \tag{4.4}$$

This is the same pseudoinverse principle applied to the correction variable. When the correction constraint in (4.4) is consistent, the resulting CRAFT update is

$$\mathbf{g}_t = \hat{\mathbf{g}}_t + \boldsymbol{\Delta}_t, \quad \text{with} \quad \boldsymbol{\Delta}_t = \mathbf{U}_t^{\dagger}\big(\boldsymbol{\rho}_t - \mathbf{U}_t\hat{\mathbf{g}}_t\big). \tag{4.5}$$

Thus, CRAFT keeps the reference direction whenever it already satisfies the desired alignments and otherwise adds only the smallest Euclidean correction needed to enforce them. Since this correction lies in the row space of $\mathbf{U}_t$, the update modifies $\hat{\mathbf{g}}_t$ only along directions spanned by the participating client gradients, yielding the closest conflict-free update to the reference direction.

4.3 Choice of reference direction

We propose using the normalized *global update* from the previous round as the reference direction.

$$\hat{\mathbf{g}}_t = \mathcal{U}(\mathbf{g}_{t-1}). \tag{4.6}$$

At the first round, we set $\hat{\mathbf{g}}_0 = \mathbf{0}$, which makes CRAFT reduce to the zero-prior minimum-norm initialization. For later rounds, the reference direction preserves temporal consistency, while the correction step guarantees that the final update is adapted to the current participating clients. Thus, the global model maintains a persistent descent direction that is continuously refined by newly sampled clients. This is particularly valuable when the sampled clients in a round are not representative, as the reference can stabilize the global update against local sampling noise.

Meanwhile, unlike momentum methods [6, 21] that add historical updates directly to the update rule, CRAFT uses history to define a geometric orientation. Rather than simply adding the previous vector, CRAFT searches for the new direction closest to it while satisfying the current conflict-free alignment constraints. See Appendix B.1 for a detailed comparison between CRAFT and momentum methods.

4.4 Layer-wise adaptation

Global conflict resolution treats the entire model parameter vector $\boldsymbol{\theta}$ as a single entity. However, deep neural networks exhibit layer-specific optimization dynamics. Conflicts in shallow layers may signal input distribution shifts, while conflicts in deeper layers may stem from label skew.

To address this, we use a *layer-wise* adaptation. We decompose the problem into Q independent subproblems. For each layer $q \in [Q]$, we construct the normalized update matrix $\mathbf{U}_t^q$ and reference direction $\hat{\mathbf{g}}_t^q$. We then solve the correction problem separately for each layer as

$$\mathbf{g}_t^q = \hat{\mathbf{g}}_t^q + \boldsymbol{\Delta}_t^q, \quad \text{with} \quad \boldsymbol{\Delta}_t^q = \big(\mathbf{U}_t^q\big)^{\dagger}\big(\boldsymbol{\rho}_t - \mathbf{U}_t^q\hat{\mathbf{g}}_t^q\big). \tag{4.7}$$

This layer-wise approach allows for more flexible conflict resolution, preventing severe conflicts in one layer from suppressing useful components in another. The complete procedure of CRAFT, incorporating this layer-wise strategy, is summarized in Algorithm 1.

Layer grouping. The partition depends on the model architecture. In ResNets [5], each trainable convolutional or fully connected layer defines a group containing its weights and biases. The trainable parameters of each normalization layer are assigned to the group of the preceding convolutional layer. For example, ResNet-20 consists of one stem convolution, nine residual blocks with two convolutions each, and one fully connected classifier, yielding $Q = 20$. Setting $Q = 1$ places the entire trainable parameter vector in a single group without layer-wise adaptation.

Algorithm 1 CRAFT framework

Input $\boldsymbol{\theta}_0$, T, η_g, η_l, $\{K_i\}_{i=1}^N$, $\{\rho_i\}_{i=1}^N$
for $t = 0$ **to** $T-1$ **do**
  Server samples client set $\mathcal{S}_t$ and broadcasts $\boldsymbol{\theta}_t$ ▷ *current global model*
  for each client $i \in \mathcal{S}_t$ **in parallel do**
    Client sets $\boldsymbol{\theta}_i^{t,0} \leftarrow \boldsymbol{\theta}_t$
    for $k = 0$ **to** $K_i - 1$ **do**
      $\boldsymbol{\theta}_i^{t,k+1} \leftarrow \boldsymbol{\theta}_i^{t,k} - \eta_l \nabla f_i(\boldsymbol{\theta}_i^{t,k}; \xi_i^{t,k})$ ▷ *local model training, see* (3.2)
    end for
    Client uploads $\mathbf{g}_i^t \leftarrow \boldsymbol{\theta}_t - \boldsymbol{\theta}_i^{t,K_i}$ to the server ▷ *client update, see* (3.3)
  end for
  Server forms $\boldsymbol{\rho}_t \leftarrow [\,\rho_i / \sum_{j\in\mathcal{S}_t} \rho_j\,]_{i\in\mathcal{S}_t}$ ▷ *active-client weights*
  for each parameter group $q \in [Q]$ **do**
    Server constructs $\mathbf{U}_t^q \leftarrow [\mathcal{U}(\mathbf{g}_i^{t,q})]_{i\in\mathcal{S}_t}^\top$ ▷ *normalized local updates*
    $\hat{\mathbf{g}}_t^q \leftarrow \mathbf{0}$ if $t = 0$, else $\mathcal{U}(\mathbf{g}_{t-1}^q)$ ▷ *normalized temporal reference*
    $\mathbf{g}_t^q \leftarrow \hat{\mathbf{g}}_t^q + \big(\mathbf{U}_t^q\big)^\dagger \big(\boldsymbol{\rho}_t - \mathbf{U}_t^q \hat{\mathbf{g}}_t^q\big)$ ▷ *layer-wise projection, see* (4.7)
  end for
  Server sets $\mathbf{g}_t \leftarrow \mathrm{concat}_q(\mathbf{g}_t^q)$ ▷ *global update concatenation*
  Server updates $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_g \mathbf{g}_t$ ▷ *global model update, see* (3.4)
end for

4.5 Communication, memory, and computational cost

No additional communication. Each active client uploads the same d-dimensional model update as in FedAvg, and all conflict-resolution operations are performed on the server.

Server-side memory cost. For m active clients, the server stores the current updates in $\mathcal{O}(dm)$ memory and the reference direction in $\mathcal{O}(d)$ memory. The layer-wise projection only adds $\mathcal{O}(m^2)$ reusable temporary workspace for the pseudoinverse computation.

Server-side computational complexity. For a neural network partitioned into Q parameter groups with dimensions $\{d_q\}_{q=1}^Q$, where $\sum_q d_q = d$, forming the intermediate matrices needed for the layer-wise pseudoinverse computations costs $\mathcal{O}(dm^2)$. Solving the Q corresponding pseudoinverse problems costs $\mathcal{O}(Qm^3)$. Thus, the conservative per-round server complexity is $\mathcal{O}(dm^2 + Qm^3)$, which is typically dominated by $\mathcal{O}(dm^2)$ when $d \gg m$ in neural networks. Meanwhile, the layer-wise computations are naturally parallelizable.

4.6 Theoretical analysis

We provide a formal analysis of the feasibility of the conflict-free alignment constraints, the common-descent structure, and the finite-time convergence behavior of CRAFT in Appendix A. Here, we briefly summarize the main convergence result. For exposition, the main theorem is stated under full client participation with local gradient-descent training. Let $K_{\max}$ denote the maximum number of local steps across clients. Under this setting, Theorem A.4 yields

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\big(\eta_l(K_{\max} - 1)\big) \tag{4.8}$$

under the standard choice $\eta_g = \Theta(1/\sqrt{T})$. The first term is the usual non-convex optimization rate, and the second is the deterministic bias induced by multiple local steps, which vanishes when η_l is also chosen to decay with T. Notably, the analysis does not rely on assumptions about the underlying data distributions and accommodates heterogeneous local update steps K_i across clients.

5 Experiments

In this section, we evaluate whether CRAFT improves the overall training performance of FL, whether the resulting gains are shared across both head and tail clients, and whether CRAFT can strengthen existing personalized FL methods when used as a direct replacement for weighted averaging.

5.1 Experimental setup

Datasets, models, and participation. We use the FEMNIST [1] and CIFAR-10/100 [10] datasets. For FEMNIST, we train a multi-layer perceptron (MLP). For CIFAR-10/100, we use convolutional neural networks (CNNs) and residual networks (ResNet-20/56/110) [5]. FEMNIST uses 1000 clients with 10 active clients per round. For CIFAR, ResNet models use 1000 clients with 100 active clients per round, while CNN models use 100 clients with 10 active clients per round. We simulate non-IID data using highly heterogeneous Dirichlet partitioning [6] with a concentration parameter $\alpha = 0.1$.

Baselines. We compare CRAFT with representative methods covering simple averaging (FedAvg [19]), drift mitigation (FedProx [12], FedNova [30]), server momentum/adaptive optimization (FedAvgM [6], FedAdam/AdaGrad/Yogi [25]), fairness-aware objectives (AFL [20], q-FedAvg [13], FedMGDA+ [7]), and conflict-aware aggregation (FedFV [31], FedLF [23], ConFIG [17]).

Detailed hyperparameter grids and supplementary results are provided in Appendix C.

5.2 Overall performance

Table 1 reports test accuracy metrics for FEMNIST and CIFAR-10, with additional results provided in Appendix C. CRAFT consistently improves mean accuracy, with relative gains of 7.8% over the second-best FedAvgM on FEMNIST and 53.5% over ConFIG on CIFAR-10. Importantly, the improvement is not driven solely by clients that already achieve high accuracy. CRAFT also raises both the best-10% and worst-10% client accuracies, indicating simultaneous gains for head and tail users.

Figure 2 further complements Table 1 by plotting convergence curves on more datasets and deeper model architectures. CRAFT's accuracy rises faster and remains higher across these settings. The advantage is especially stable and pronounced on deeper networks, where several baselines either saturate early or improve slowly under client drift. This behavior supports the central design premise of CRAFT, namely that enforcing client-wise alignment during aggregation makes each server update better able to mitigate conflicts, thereby improving both final accuracy and training stability.

Table 1: Comparative evaluation of test accuracy metrics, including Mean accuracy, Best 10% (mean accuracy of top-10% clients), Worst 10% (mean accuracy of bottom-10% clients), and Standard Deviation (Std). **Bold** and underlined values indicate the best and second-best results, respectively.

	FEMNIST & MLP				CIFAR-10 & ResNet-20			
Algorithm	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)
FedAvg	0.570	0.776	0.305	0.136	0.356	0.727	0.015	0.195
FedProx	0.565	0.770	0.300	0.135	0.356	0.728	0.015	0.195
FedNova	0.589	0.804	0.337	0.135	0.381	0.734	0.040	<u>0.190</u>
FedAvgM	<u>0.655</u>	<u>0.843</u>	<u>0.408</u>	0.125	0.463	0.871	0.040	0.241
FedAdaGrad	0.645	0.828	0.405	<u>0.121</u>	0.427	0.799	0.072	0.206
FedAdam	0.638	0.825	0.390	0.125	0.412	0.790	0.065	0.197
FedYogi	0.639	0.830	0.381	0.130	0.398	0.777	0.047	0.201
AFL	0.548	0.774	0.301	0.137	0.294	0.732	0.000	0.217
FedFV	0.489	0.742	0.246	0.144	0.421	0.780	0.078	0.194
FedMGDA+	0.386	0.658	0.133	0.166	0.235	0.612	0.000	0.193
qFedAvg	0.555	0.774	0.298	0.137	0.352	0.716	0.019	0.191
FedLF	0.426	0.688	0.177	0.154	0.501	0.845	<u>0.149</u>	0.208
ConFIG	0.600	0.801	0.333	0.136	<u>0.525</u>	<u>0.872</u>	0.146	0.211
CRAFT	**0.706**	**0.878**	**0.474**	**0.115**	**0.806**	**1.000**	**0.479**	**0.159**

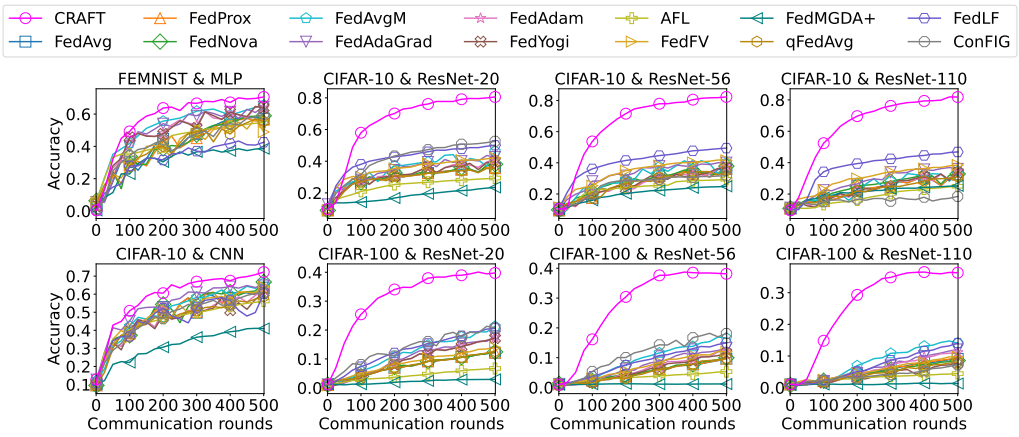

Figure 2: Evolution of mean test accuracy over communication rounds across datasets and model architectures. CRAFT consistently converges faster and reaches higher final accuracy, with especially stable gains on more challenging datasets and deeper neural networks.

5.3 Per-client accuracy and fairness

We next evaluate the full distribution of per-client accuracy. Mean accuracy alone can hide severe disparity because two global models with the same mean may behave very differently across clients.

The results in Table 1 show that CRAFT improves mean accuracy without sacrificing either end of the client distribution. On FEMNIST, relative to the second-best FedAvgM, CRAFT improves best-10% and worst-10% accuracy by 4.2% and 16.2%, respectively. On CIFAR-10 with ResNet-20, relative to the corresponding second-best methods, CRAFT improves best-10% accuracy by 14.7% over ConFIG and worst-10% accuracy by 221.5% over FedLF. These gains indicate that CRAFT does not simply trade tail users for head users or vice versa.

Figure 3 makes this distributional change explicit by plotting the final test accuracy of all clients under FedAvg and CRAFT. The x-axis is test accuracy and the y-axis counts clients, so a distribution shifted to the right means that more clients achieve higher accuracy. On FEMNIST and CIFAR-10, CRAFT moves most of the mass to the right and reduces the low-accuracy tail. On CIFAR-100, the FedAvg distribution can appear concentrated, but this concentration is misleading, as most clients remain near zero accuracy. CRAFT moves the distribution away from this near-zero regime substantially.

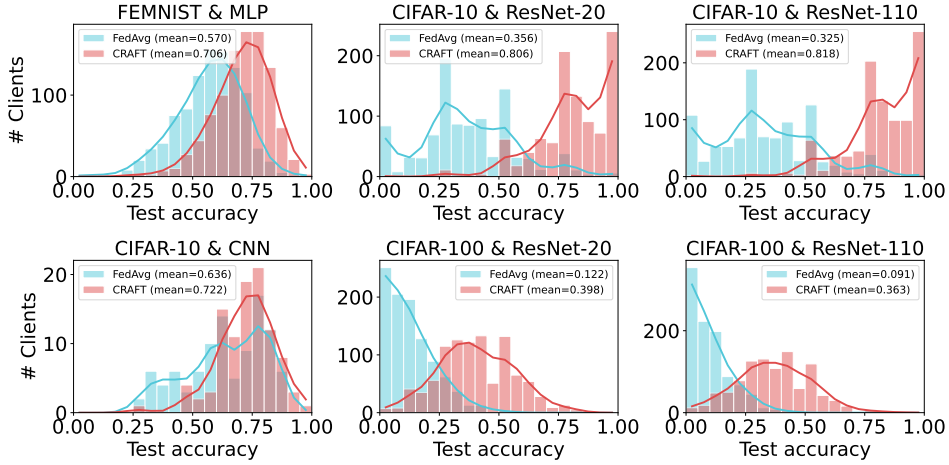

Figure 3: Distribution of per-client test accuracy. A rightward shift indicates higher accuracy for more clients, and CRAFT produces a more favorable distribution with improved mean and tail performance.

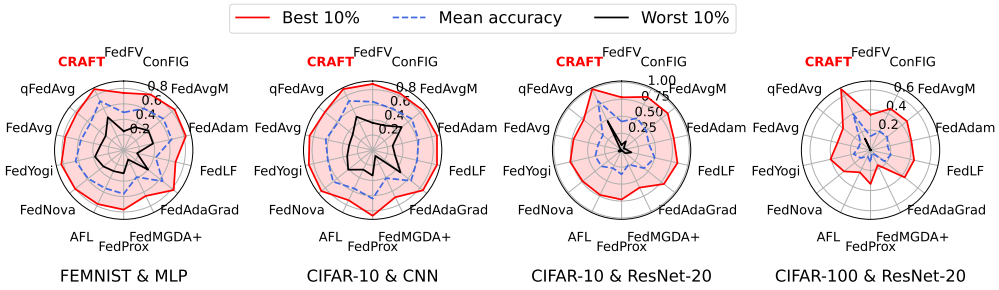

Figure 4: Radar plot of accuracy metrics, with each spoke corresponding to a method. CRAFT improves the worst-client performance while keeping the mean and best performance high.

Moreover, Figure 4 provides a complementary summary of mean, best-10%, and worst-10% accuracy. In this radar plot, each spoke corresponds to an algorithm, and the three curves correspond to best-10%, mean, and worst-10% accuracy, all of which are better when farther from the center. Several baselines obtain reasonable mean accuracy while showing a large gap between the best and worst clients, which reflects a polarized client distribution. CRAFT keeps the mean and best-client performance high while lifting the worst-client curve, showing a much fairer distribution.

5.4 Integration with personalized federated learning

CRAFT provides a new aggregation operator rather than a personalized FL algorithm that maintains a separate model for each client. Nevertheless, personalized FL methods such as Ditto still require an aggregation step to update a shared model or backbone. We therefore test whether CRAFT can improve a representative personalized FL pipeline by replacing the averaging step inside Ditto [14, Alg. 2, line 8] with the CRAFT operator. This keeps Ditto's client-side personalization mechanism unchanged and modifies only the aggregation rule used for the shared update.

Figure 5 reports heatmaps over different server and client learning-rate combinations. Compared with vanilla Ditto, Ditto with CRAFT improves mean test accuracy across the grid. The improvement is especially clear at larger learning rates. These results show that CRAFT has broad plug-in extensibility. It can strengthen personalized FL methods by replacing their internal aggregation step.

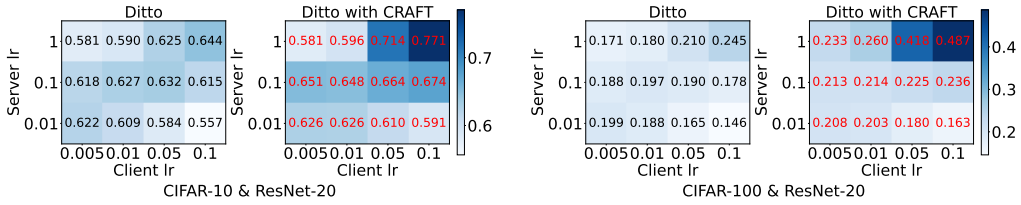

Figure 5: Comparison between vanilla Ditto and Ditto with CRAFT across various settings. Replacing Ditto's average aggregation with the CRAFT operator yields average relative improvements of 6.0% on CIFAR-10 and 32.0% on CIFAR-100, demonstrating CRAFT's strong extensibility.

5.5 Ablation studies on CRAFT components

Layer-wise adaptation and momentum-like reference. We compare four configurations to examine the contributions of layer-wise adaptation and the reference direction. CRAFT uses the momentum-like reference with layer-wise adaptation, while CRAFT-no-Layerwise retains the same reference without layer-wise adaptation. CRAFT-no-Momentum (ConFIG-Layerwise) uses a zero reference with layer-wise adaptation, and ConFIG uses a zero reference without layer-wise adaptation.

We use CIFAR-10 and CIFAR-100 with ResNet-20 and $N = 1000$ clients. Table 2 fixes the Dirichlet parameter at $\alpha = 0.1$ and varies the number of active clients per round, $m \in \{10, 50, 100\}$, reporting test accuracy at round 500. Additional comparisons with baselines across heterogeneity levels at fixed $m = 100$ are provided in Appendix D.4.

Table 2: Test accuracy of different CRAFT component configurations with varying active-client count m at fixed Dirichlet $\alpha = 0.1$ on CIFAR with ResNet-20 and $N = 1000$ clients.

	CIFAR-10			CIFAR-100		
Algorithm	$m = 10$	$m = 50$	$m = 100$	$m = 10$	$m = 50$	$m = 100$
FedAvg	0.260	0.337	0.356	0.092	0.117	0.122
ConFIG	0.300	0.545	0.525	0.090	0.167	0.208
CRAFT-no-Layerwise	<u>0.361</u>	<u>0.613</u>	0.462	<u>0.133</u>	0.189	0.152
CRAFT-no-Momentum	0.297	0.609	<u>0.698</u>	0.098	<u>0.206</u>	<u>0.292</u>
CRAFT	**0.417**	**0.731**	**0.806**	**0.188**	**0.391**	**0.398**

As shown in Table 2, CRAFT achieves the highest accuracy in every setting and consistently outperforms CRAFT-no-Layerwise, with a larger gap as participation increases. The full CRAFT method also outperforms CRAFT-no-Momentum throughout, supporting the effectiveness of combining layer-wise adaptation with the momentum-like reference. With a zero reference, layer-wise adaptation improves upon ConFIG in five of the six settings.

Sensitivity to layer-wise grouping. We further compare different layer partitions $Q \in \{1, 2, 5, 11, 20\}$. The grouping definitions and results across heterogeneity levels are provided in Appendix D.8. As summarised in Table 3, either $Q = 11$ or $Q = 20$ achieves the highest accuracy in all six participation settings. Both consistently outperform the broad $Q = 1$ and $Q = 2$ partitions, supporting conflict resolution at residual-block or layer granularity. The milder-heterogeneity results in Appendix D.8 favor $Q = 11$, showing that accuracy does not improve monotonically with Q. These results support the layer-level default as an effective grouping rule without data-dependent tuning, while indicating that the best granularity can depend on the use case.

Table 3: Test accuracy of different layer-wise grouping strategies with varying active-client count m at fixed Dirichlet $\alpha = 0.1$ on CIFAR with ResNet-20 and $N = 1000$ clients.

	CIFAR-10			CIFAR-100		
Configuration	$m = 10$	$m = 50$	$m = 100$	$m = 10$	$m = 50$	$m = 100$
CRAFT ($Q = 1$)	0.361	0.613	0.462	0.133	0.189	0.152
CRAFT ($Q = 2$)	0.350	0.643	0.530	0.156	0.212	0.312
CRAFT ($Q = 5$)	0.334	0.698	0.754	0.180	0.322	0.394
CRAFT ($Q = 11$)	<u>0.364</u>	<u>0.730</u>	<u>0.798</u>	<u>0.187</u>	<u>0.368</u>	**0.415**
CRAFT ($Q = 20$)	**0.417**	**0.731**	**0.806**	**0.188**	**0.391**	<u>0.398</u>

6 Conclusions and limitations

This work presented CRAFT, a conflict-resolved aggregation framework that treats federated aggregation as a geometric correction problem. By enforcing positive alignment between the global update and each participating client update, CRAFT mitigates conflicts in heterogeneous FL. Our analysis identifies the resulting common-descent structure and establishes convergence guarantees. Experiments show that CRAFT improves mean accuracy while maintaining strong performance for both head and tail clients. CRAFT also transfers directly to personalized FL as a drop-in aggregation module, improving personalization without modifying local adaptation. Overall, CRAFT provides a new perspective on aggregation design and a practical tool for improving FL under heterogeneity.

Meanwhile, several limitations remain. First, CRAFT uses the previous global update as the reference direction, which is simple and empirically effective. Alternatives such as moving averages over multiple previous rounds or adaptive references based on recent client conflicts may provide smoother or more informative guidance. Second, the alignment target is currently chosen according to clients' data proportions. More adaptive targets could reweight clients based on training progress, loss, gradient conflicts, or fairness requirements. Understanding how these reference and alignment choices affect accuracy, stability, and client-level fairness is an important direction for future work.

Acknowledgments

Ziqi Wang acknowledges funding from the European Union's Horizon Europe MSCA project ModConFlex (grant number 101073558). Qiang Liu acknowledges support from the China Scholarship Council (No. 202206120036). We are grateful to Enrique Zuazua for invaluable discussions. We acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is partially funded by the German Research Foundation (DFG).

References

[1] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[2] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2089–2099. PMLR, 18–24 Jul 2021.

[3] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020.

[4] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3762–3773. PMLR, 26–28 Aug 2020.

[5] 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 (CVPR)*, pages 770–778, June 2016.

[6] 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.

[7] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9(4): 2039–2051, 2022.

[8] Youngsik Hwang and Dong-Young Lim. Dual cone gradient descent for training physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 37, pages 98563–98595. Curran Associates, Inc., 2024.

[9] Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

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

[11] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University, 2015. URL `https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf`.

[12] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

[13] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020.

[14] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 18–24 Jul 2021.

[15] 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.

[16] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[17] Qiang Liu, Mengyu Chu, and Nils Thuerey. ConFIG: Towards conflict-free training of physics informed neural networks. In *International Conference on Learning Representations*, pages 59531–59566, 2025.

[18] Yuxiang Lu, Suizhi Huang, Yuwen Yang, Shalayiding Sirejiding, Yue Ding, and Hongtao Lu. FedHCA^2: Towards hetero-client federated multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5599–5609, 2024.

[19] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera 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 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.

[20] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 09–15 Jun 2019.

[21] Yurii E. Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[22] Zibin Pan, Shuyi Wang, Chi Li, Haijin Wang, Xiaoying Tang, and Junhua Zhao. FedMDFG: Federated learning with multi-gradient descent and fair guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9364–9371, 2023.

[23] Zibin Pan, Chi Li, Fangchen Yu, Shuyi Wang, Haijin Wang, Xiaoying Tang, and Junhua Zhao. FedLF: Layer-wise fair federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14527–14535, 2024.

[24] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[25] Sashank J. 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.

[26] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

[27] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[28] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[29] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, volume 33, pages 21394–21405. Curran Associates, Inc., 2020.

[30] 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*, volume 33, pages 7611–7623. Curran Associates, Inc., 2020.

[31] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Chenglu Wen, Cheng Wang, and Rongshan Yu. Federated learning with fair averaging. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1615–1623, 2021.

[32] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5693–5700, 2019.

[33] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.

[34] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. FedALA: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11237–11244, 2023.

Appendix

This appendix provides details about CRAFT on the following topics:

- Appendix A: Technical details of the theoretical analysis.
- Appendix B: Additional discussion and design choices.
- Appendix C: Training details and hyperparameter settings.
- Appendix D: Additional numerical results and visualizations.

A Technical details of the theoretical analysis

In this appendix, we first discuss the feasibility of the conflict-free alignment constraints, then prove the full-vector convergence bound, and finally analyze the layer-wise version and residual alignment constraints as extensions.

A.1 Feasibility and solution existence

It follows from (4.3) that CRAFT's update satisfies the linear system $\mathbf{U}_t\mathbf{g} = \boldsymbol{\rho}_t$ with $\mathbf{U}_t \in \mathbb{R}^{m\times d}$. The system is feasible if and only if $\boldsymbol{\rho}_t \in \mathrm{Range}(\mathbf{U}_t)$, equivalently $\mathrm{rank}(\mathbf{U}_t) = \mathrm{rank}([\mathbf{U}_t \mid \boldsymbol{\rho}_t])$. In the exact statements in this appendix, $\mathbf{U}_t$ denotes the unit-normalized matrix with rows $\mathbf{u}_i^t = \mathbf{g}_i^t/\|\mathbf{g}_i^t\|$ for nonzero uploaded updates. Stabilized normalization with $\varepsilon > 0$ and the handling of near-zero updates are treated as numerical variants in Appendix B.3.

In deep neural networks, the parameter dimension d is typically orders of magnitude larger than the number of sampled clients m. Moreover, under non-IID data distributions, the uploaded local directions arise from distinct local optimization landscapes. An exact linear dependence among the m active-client directions in $\mathbb{R}^d$ is therefore a degenerate event. Thus, in the generic high-dimensional case, $\mathrm{rank}(\mathbf{U}_t) = m$. Then $\mathrm{Range}(\mathbf{U}_t) = \mathbb{R}^m$, so every target $\boldsymbol{\rho}_t \in \mathbb{R}^m$ is attainable. Since $d \gg m$, the linear system is underdetermined and its feasible set contains infinitely many solutions. CRAFT resolves this non-uniqueness by choosing the solution closest to the momentum-like reference $\hat{\mathbf{g}}_t$, which is the Euclidean projection onto the affine set $\{\mathbf{g} \in \mathbb{R}^d \mid \mathbf{U}_t\mathbf{g} = \boldsymbol{\rho}_t\}$ and is given by (4.5).

If the active directions are rank-deficient, exact feasibility is no longer automatic. When $\boldsymbol{\rho}_t \notin \mathrm{Range}(\mathbf{U}_t)$, the Moore–Penrose update in (4.5) should be interpreted as a least-squares projection in the correction variable, *i.e.*, $\mathbf{g}_t$ is the point closest to $\hat{\mathbf{g}}_t$ among all updates that minimize $\|\mathbf{U}_t\mathbf{g} - \boldsymbol{\rho}_t\|_2^2$. The resulting residual, denoted by $\boldsymbol{\delta}_t$, is analyzed in Corollary A.12.

A.2 Convergence analysis for CRAFT

A.2.1 Assumptions

Assumption A.1 (L-Smoothness)**.** Each local loss f_i is L-smooth, *i.e.*, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$,

$$\|\nabla f_i(\boldsymbol{\theta}') - \nabla f_i(\boldsymbol{\theta})\| \le L\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|.$$

Hence, $F(\boldsymbol{\theta}) = \sum_{i=1}^N \rho_i f_i(\boldsymbol{\theta})$ is also L-smooth.

Assumption A.2 (Lower-bounded objective)**.** The global objective is bounded below, *i.e.*, there exists a constant $F^\star \in \mathbb{R}$ such that $F(\boldsymbol{\theta}) \ge F^\star$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$.

Assumption A.3 (Bounded gradients and updates)**.** There exist constants $G_f, G > 0$ such that $\|\nabla f_i(\boldsymbol{\theta})\| \le G_f$ for all i and all $\boldsymbol{\theta}$, and the update direction satisfies $\|\mathbf{g}_t\| \le G$ for every round t.

These are common regularity conditions in non-convex FL analysis. The lower-boundedness assumption defines $F^\star$ as a finite lower bound rather than as an attained global minimizer, and it is used only when telescoping objective values. Bounded gradient assumptions also appear in FL analyses such as [15, 25, 32]. The bounded-update condition can be enforced in practice through norm clipping controls. If such clipping is applied after the projection step, however, it may change the equality constraint and is therefore covered by the projection residual in Corollary A.12. Importantly, our analysis does not impose any distributional assumption on the client datasets and allows heterogeneous local training steps K_i for each client $i \in [N]$.

A.2.2 Main results

We first define the constants that appear in the main bound.

$$K_{\max} := \max_{i\in[N]} K_i, \quad \rho_{\min} := \min_{i\in[N]} \rho_i, \quad c := \frac{\rho_{\min}}{G_f}, \quad B_{\text{drift}} := \frac{LG_f}{2}\left(G + \sum_{i=1}^{N} \rho_i^2\right). \tag{A.1}$$

Theorem A.4 (Convergence Bound for CRAFT)**.** *Consider CRAFT under full client participation and local gradient-descent training in the feasible full-vector alignment-constrained setting. Suppose Assumptions A.1–A.3 hold. Then, for any $T \geq 1$,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 \leq \frac{F(\boldsymbol{\theta}_0) - F^\star}{c\eta_g T} + \frac{L\eta_g G^2}{2c} + \frac{B_{\text{drift}}}{c}\eta_l(K_{\max} - 1). \tag{A.2}$$

Before proving the theorem, we state the standard rate obtained from the chosen step sizes.

Corollary A.5 (CRAFT Rate under Step-Size Schedules)**.** *Under the conditions of Theorem A.4, if the server learning rate is chosen as $\eta_g = \Theta(1/\sqrt{T})$, then*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\big(\eta_l(K_{\max} - 1)\big). \tag{A.3}$$

If, in addition, $K_{\max}$ is fixed and the local learning rate is scheduled as $\eta_l = \mathcal{O}(1/\sqrt{T})$, then the deterministic drift term also decays, and we recover the standard non-convex convergence rate:

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right). \tag{A.4}$$

Sketch of the proof. The proof of Theorem A.4 is deferred until after stating three supporting lemmas. The conflict-free alignment constraint enforces positive alignment between $\mathbf{g}_t$ and each accumulated client update $\mathbf{g}_i^t$, while multi-step local training makes $\mathbf{g}_i^t$ drift from the idealized gradient $\eta_l K_i \nabla f_i(\boldsymbol{\theta}_t)$. Lemma A.6 quantifies this local drift, Lemma A.7 converts the resulting client-wise lower bounds into a global-gradient lower bound, and Lemma A.8 combines them into a common-descent certificate. The theorem then follows from the smoothness descent lemma and telescoping over communication rounds.

In Lemma A.6, we first quantify the drift error incurred by replacing the sum of gradients along the local trajectory with the gradient evaluated at the round-start broadcast model.

Lemma A.6 (Local Drift Bound)**.** *Under Assumptions A.1 and A.3, for each client i and round t,*

$$\mathbf{g}_i^t = \eta_l K_i \nabla f_i(\boldsymbol{\theta}_t) + \mathbf{e}_i^t, \tag{A.5}$$

with

$$\|\mathbf{e}_i^t\| \leq \frac{LG_f}{2}\eta_l^2 K_i(K_i - 1). \tag{A.6}$$

Proof. Adding and subtracting $\nabla f_i(\boldsymbol{\theta}_t)$ in the local client updates gives

$$\mathbf{g}_i^t = \eta_l \sum_{k=0}^{K_i-1} \nabla f_i(\boldsymbol{\theta}_i^{t,k}) = \eta_l K_i \nabla f_i(\boldsymbol{\theta}_t) + \mathbf{e}_i^t, \tag{A.7}$$

where

$$\mathbf{e}_i^t := \eta_l \sum_{k=0}^{K_i-1} \Big(\nabla f_i(\boldsymbol{\theta}_i^{t,k}) - \nabla f_i(\boldsymbol{\theta}_t)\Big). \tag{A.8}$$

For $k = 0$, the difference is zero. For $k \geq 1$, the local recursion gives

$$\boldsymbol{\theta}_i^{t,k} - \boldsymbol{\theta}_t = -\eta_l \sum_{r=0}^{k-1} \nabla f_i(\boldsymbol{\theta}_i^{t,r}). \tag{A.9}$$

Taking norms and applying the triangle inequality gives

$$\begin{aligned}\|\boldsymbol{\theta}_i^{t,k} - \boldsymbol{\theta}_t\| &= \left\| -\eta_l \sum_{r=0}^{k-1} \nabla f_i(\boldsymbol{\theta}_i^{t,r}) \right\| \\ &\le \eta_l \sum_{r=0}^{k-1} \|\nabla f_i(\boldsymbol{\theta}_i^{t,r})\| \\ &\le \eta_l k G_f.\end{aligned} \tag{A.10}$$

We now bound the accumulated drift error.

$$\begin{aligned}\|\mathbf{e}_i^t\| &= \left\| \eta_l \sum_{k=0}^{K_i-1} \Big(\nabla f_i(\boldsymbol{\theta}_i^{t,k}) - \nabla f_i(\boldsymbol{\theta}_t) \Big) \right\| \\ &\le \eta_l \sum_{k=0}^{K_i-1} \left\| \nabla f_i(\boldsymbol{\theta}_i^{t,k}) - \nabla f_i(\boldsymbol{\theta}_t) \right\| \\ &\le \eta_l \sum_{k=0}^{K_i-1} L \|\boldsymbol{\theta}_i^{t,k} - \boldsymbol{\theta}_t\| \\ &\le L G_f \eta_l^2 \sum_{k=0}^{K_i-1} k \\ &= \frac{L G_f}{2} \eta_l^2 K_i (K_i - 1).\end{aligned} \tag{A.11}$$

□

Lemma A.7 (Weighted Gradient Lower Bound)**.** *Under Assumption A.3, for any round t, let*

$$\nabla f_i^t := \nabla f_i(\boldsymbol{\theta}_t), \qquad \nabla F_t := \nabla F(\boldsymbol{\theta}_t). \tag{A.12}$$

Then

$$\sum_{i=1}^N \rho_i^2 \|\nabla f_i^t\| \ge \rho_{\min} \|\nabla F_t\| \ge \frac{\rho_{\min}}{G_f} \|\nabla F_t\|^2. \tag{A.13}$$

Proof. The first step uses only the definition of $\rho_{\min}$ and the probability weights.

$$\begin{aligned}\sum_{i=1}^N \rho_i^2 \|\nabla f_i^t\| &= \sum_{i=1}^N \rho_i \big(\rho_i \|\nabla f_i^t\| \big) \\ &\ge \rho_{\min} \sum_{i=1}^N \rho_i \|\nabla f_i^t\| \\ &\ge \rho_{\min} \left\| \sum_{i=1}^N \rho_i \nabla f_i^t \right\| \\ &= \rho_{\min} \|\nabla F_t\|.\end{aligned} \tag{A.14}$$

For the squared-gradient conversion, bounded local gradients imply

$$\begin{aligned}\|\nabla F_t\| &= \left\| \sum_{i=1}^N \rho_i \nabla f_i^t \right\| \\ &\le \sum_{i=1}^N \rho_i \|\nabla f_i^t\| \le G_f,\end{aligned} \tag{A.15}$$

and therefore

$$\rho_{\min} \|\nabla F_t\| \ge \frac{\rho_{\min}}{G_f} \|\nabla F_t\|^2. \tag{A.16}$$

Combining the inequalities proves the claim. □

Lemma A.8 (Common Descent Certificate)**.** *In the feasible full-vector alignment-constrained setting described above, under Assumptions A.1 and A.3,*

$$\langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle \geq c\|\nabla F(\boldsymbol{\theta}_t)\|^2 - B_{\mathrm{drift}}\, \eta_l (K_{\max} - 1), \tag{A.17}$$

where c and B_{drift} are defined in (A.1)*.*

Proof. Let $\nabla f_i^t := \nabla f_i(\boldsymbol{\theta}_t)$ and $\nabla F_t := \nabla F(\boldsymbol{\theta}_t)$. The exact conflict-free alignment constraint and the definition $\mathbf{u}_i^t = \mathbf{g}_i^t / \|\mathbf{g}_i^t\|$ imply

$$\begin{aligned} \langle \mathbf{g}_i^t, \mathbf{g}_t \rangle &= \|\mathbf{g}_i^t\| \langle \mathbf{u}_i^t, \mathbf{g}_t \rangle \\ &= \rho_i \|\mathbf{g}_i^t\|. \end{aligned} \tag{A.18}$$

Using the drift decomposition (A.5), each client gradient has the following alignment with the projected direction.

$$\begin{aligned} \eta_l K_i \langle \nabla f_i^t, \mathbf{g}_t \rangle &\overset{(A.5)}{=} \langle \mathbf{g}_i^t - \mathbf{e}_i^t, \mathbf{g}_t \rangle \\ &= \langle \mathbf{g}_i^t, \mathbf{g}_t \rangle - \langle \mathbf{e}_i^t, \mathbf{g}_t \rangle \\ &= \rho_i \|\mathbf{g}_i^t\| - \langle \mathbf{e}_i^t, \mathbf{g}_t \rangle \\ &\geq \rho_i \|\mathbf{g}_i^t\| - \|\mathbf{e}_i^t\|\, \|\mathbf{g}_t\|. \end{aligned} \tag{A.19}$$

The uploaded displacement norm is lower bounded by the reverse triangle inequality.

$$\|\mathbf{g}_i^t\| \overset{(A.5)}{=} \|\eta_l K_i \nabla f_i^t + \mathbf{e}_i^t\| \geq \eta_l K_i \|\nabla f_i^t\| - \|\mathbf{e}_i^t\|. \tag{A.20}$$

Substituting this lower bound and dividing by $\eta_l K_i$ gives

$$\begin{aligned} \langle \nabla f_i^t, \mathbf{g}_t \rangle &\geq \rho_i \|\nabla f_i^t\| - \frac{\rho_i + \|\mathbf{g}_t\|}{\eta_l K_i} \|\mathbf{e}_i^t\| \\ &\overset{(A.6)}{\geq} \rho_i \|\nabla f_i^t\| - \frac{L G_f}{2} (\rho_i + G) \eta_l (K_i - 1), \end{aligned} \tag{A.21}$$

where the division is valid because $K_i \geq 1$ and $\eta_l > 0$. Multiplying by ρ_i and summing over clients yields

$$\begin{aligned} \langle \nabla F_t, \mathbf{g}_t \rangle &= \sum_{i=1}^N \rho_i \langle \nabla f_i^t, \mathbf{g}_t \rangle \\ &\geq \sum_{i=1}^N \rho_i^2 \|\nabla f_i^t\| - \frac{L G_f}{2} \eta_l \sum_{i=1}^N \rho_i (\rho_i + G)(K_i - 1) \\ &\overset{(A.1)}{\geq} \sum_{i=1}^N \rho_i^2 \|\nabla f_i^t\| - B_{\mathrm{drift}} \eta_l (K_{\max} - 1) \\ &\geq c\|\nabla F_t\|^2 - B_{\mathrm{drift}} \eta_l (K_{\max} - 1). \end{aligned} \tag{A.22}$$

The last inequality uses Lemma A.7, which gives (A.17). □

With the common-descent bound in place, we can prove the main convergence result.

Proof of Theorem A.4. Applying L-smoothness of F to $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_g \mathbf{g}_t$, we have

$$\begin{aligned} F(\boldsymbol{\theta}_{t+1}) &\leq F(\boldsymbol{\theta}_t) - \eta_g \langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle + \frac{L \eta_g^2}{2} \|\mathbf{g}_t\|^2 \\ &\leq F(\boldsymbol{\theta}_t) - c\eta_g \|\nabla F(\boldsymbol{\theta}_t)\|^2 + \eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L \eta_g^2}{2} G^2. \end{aligned} \tag{A.23}$$

The second inequality above uses Lemma A.8. Rearranging this one-step descent inequality gives

$$c\eta_g \|\nabla F(\boldsymbol{\theta}_t)\|^2 \leq F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_{t+1}) + \eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L \eta_g^2}{2} G^2. \tag{A.24}$$

Summing from $t = 0$ to $T - 1$, the objective differences telescope.

$$\begin{aligned} c\eta_g \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 &\le \sum_{t=0}^{T-1} \big[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_{t+1})\big] + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} TG^2 \\ &= F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}_T) + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} TG^2 \\ &\le F(\boldsymbol{\theta}_0) - F^\star + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} TG^2. \end{aligned} \tag{A.25}$$

Dividing both sides by $c\eta_g T$ gives (A.2). □

Interpretation of the bound. When $K_{\max} = 1$, the drift term vanishes and the theorem reduces to the one-step bound. For $K_{\max} > 1$, local computation introduces a stationarity neighborhood of size $\mathcal{O}(\eta_l(K_{\max} - 1))$. By scheduling η_l to decay with T, this deterministic bias can be forced to zero.

A.2.3 Average gradient-norm bound

The squared-gradient result above uses the bounded-gradient assumption to convert a linear descent certificate into a squared stationarity measure. If the desired stationarity measure is instead the average gradient norm, the same argument yields a direct bound with alignment constant $\rho_{\min}$.

Lemma A.9 (Linear Common-Descent Bound)**.** *In the feasible full-vector alignment-constrained setting described above, under Assumptions A.1 and A.3,*

$$\langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle \ge \rho_{\min} \|\nabla F(\boldsymbol{\theta}_t)\| - B_{\mathrm{drift}}\, \eta_l (K_{\max} - 1), \tag{A.26}$$

where B_{drift} is defined in (A.1)*.*

Proof. The proof of Lemma A.8 first establishes the pre-conversion bound

$$\begin{aligned} \langle \nabla F_t, \mathbf{g}_t \rangle &\ge \sum_{i=1}^{N} \rho_i^2 \|\nabla f_i^t\| - B_{\mathrm{drift}} \eta_l (K_{\max} - 1) \\ &\ge \rho_{\min} \|\nabla F_t\| - B_{\mathrm{drift}} \eta_l (K_{\max} - 1). \end{aligned} \tag{A.27}$$

The second inequality uses Lemma A.7. Equivalently, this uses only

$$\sum_{i=1}^{N} \rho_i^2 \|\nabla f_i^t\| \ge \rho_{\min} \|\nabla F_t\|, \tag{A.28}$$

and does not convert $\|\nabla F_t\|$ into $\|\nabla F_t\|^2 / G_f$. This proves (A.26). □

Corollary A.10 (Average Gradient-Norm Bound)**.** *Under the conditions of Theorem A.4, for any $T \ge 1$,*

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\| \le \frac{F(\boldsymbol{\theta}_0) - F^\star}{\rho_{\min} \eta_g T} + \frac{L\eta_g G^2}{2\rho_{\min}} + \frac{B_{\mathrm{drift}}}{\rho_{\min}} \eta_l (K_{\max} - 1). \tag{A.29}$$

Consequently, choosing $\eta_g = \Theta(1/\sqrt{T})$ gives

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\| = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\big(\eta_l (K_{\max} - 1)\big). \tag{A.30}$$

If $K_{\max}$ is fixed and $\eta_l = \mathcal{O}(1/\sqrt{T})$, then the drift term also decays and

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\| = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right). \tag{A.31}$$

Proof. By L-smoothness of F and the update rule $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_g \mathbf{g}_t$, we have

$$\begin{aligned} F(\boldsymbol{\theta}_{t+1}) &\le F(\boldsymbol{\theta}_t) - \eta_g \langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle + \frac{L\eta_g^2}{2} \|\mathbf{g}_t\|^2 \\ &\le F(\boldsymbol{\theta}_t) - \rho_{\min} \eta_g \|\nabla F(\boldsymbol{\theta}_t)\| + \eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} G^2. \end{aligned} \tag{A.32}$$

The second inequality above uses Lemma A.9. Rearranging, summing, and telescoping gives

$$\begin{aligned} \rho_{\min} \eta_g \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\| &\le \sum_{t=0}^{T-1} \big[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_{t+1})\big] + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} T G^2 \\ &= F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}_T) + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} T G^2 \\ &\le F(\boldsymbol{\theta}_0) - F^\star + T\eta_g B_{\mathrm{drift}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} T G^2. \end{aligned} \tag{A.33}$$

Dividing by $\rho_{\min} \eta_g T$ gives (A.29). The step-size statements follow by substitution. □

A.3 Layer-wise multi-step projection extension

The layer-wise analysis follows the full-vector argument block by block. The only change is that the drift constant and update-norm bound are accumulated over layers. We use

$$B_q := \frac{L_q G_f}{2} \left(G^q + \sum_{i=1}^{N} \rho_i^2 \right), \qquad B_{\mathrm{layer}} := \sum_{q=1}^{Q} B_q, \qquad G_{\mathrm{layer}}^2 := \sum_{q=1}^{Q} \big(G^q\big)^2. \tag{A.34}$$

Corollary A.11 (Layer-Wise Multi-Step Projection)**.** *Consider the full-participation layer-wise CRAFT update* (4.7) *with Q layers, indexed by q. Under Assumptions A.1–A.3, suppose that, for every client i and layer q, $\nabla^q f_i$ is L_q-Lipschitz with respect to the full parameter norm, and that $\|\mathbf{g}_t^q\| \le G^q$ for all t, q. Let $c = \rho_{\min}/G_f$ as in* (A.1)*. Then, for any $T \ge 1$,*

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 \le \frac{F(\boldsymbol{\theta}_0) - F^\star}{c\eta_g T} + \frac{L\eta_g G_{\mathrm{layer}}^2}{2c} + \frac{B_{\mathrm{layer}}}{c} \eta_l (K_{\max} - 1). \tag{A.35}$$

Consequently, choosing $\eta_g = \Theta(1/\sqrt{T})$ gives the same asymptotic form as Corollary A.5. If $K_{\max}$ is fixed and $\eta_l = \mathcal{O}(1/\sqrt{T})$, the deterministic drift term also decays.

Proof. Fix a layer q. The proof follows the full-vector argument applied to the coordinates of this block, with L_q replacing the global smoothness constant in the drift term. Define

$$\mathbf{e}_i^{t,q} := \eta_l \sum_{k=0}^{K_i - 1} \left(\nabla^q f_i(\boldsymbol{\theta}_i^{t,k}) - \nabla^q f_i(\boldsymbol{\theta}_t) \right). \tag{A.36}$$

As in Lemma A.6, we have

$$\|\boldsymbol{\theta}_i^{t,k} - \boldsymbol{\theta}_t\| \le \eta_l k G_f \tag{A.37}$$

and the block Lipschitz condition gives

$$\begin{aligned} \mathbf{g}_i^{t,q} &= \eta_l K_i \nabla^q f_i(\boldsymbol{\theta}_t) + \mathbf{e}_i^{t,q}, \\ \|\mathbf{e}_i^{t,q}\| &\le \frac{L_q G_f}{2} \eta_l^2 K_i (K_i - 1). \end{aligned} \tag{A.38}$$

The exact layer-wise constraint gives, for every client i,

$$\big\langle \mathbf{g}_i^{t,q}, \mathbf{g}_t^q \big\rangle = \rho_i \|\mathbf{g}_i^{t,q}\|. \tag{A.39}$$

Therefore,

$$\begin{aligned} \eta_l K_i \langle \nabla^q f_i(\boldsymbol{\theta}_t), \mathbf{g}_t^q \rangle &= \big\langle \mathbf{g}_i^{t,q} - \mathbf{e}_i^{t,q}, \mathbf{g}_t^q \big\rangle \\ &\ge \rho_i \|\mathbf{g}_i^{t,q}\| - \|\mathbf{e}_i^{t,q}\| \, \|\mathbf{g}_t^q\|. \end{aligned} \tag{A.40}$$

The reverse triangle inequality gives

$$\|\mathbf{g}_i^{t,q}\| \geq \eta_l K_i \|\nabla^q f_i(\boldsymbol{\theta}_t)\| - \|\mathbf{e}_i^{t,q}\|. \tag{A.41}$$

Dividing by $\eta_l K_i$, using $\|\mathbf{g}_t^q\| \leq G^q$, and substituting the block drift bound gives

$$\langle \nabla^q f_i(\boldsymbol{\theta}_t), \mathbf{g}_t^q \rangle \geq \rho_i \|\nabla^q f_i(\boldsymbol{\theta}_t)\| - \frac{L_q G_f}{2}(\rho_i + G^q)\eta_l(K_i - 1). \tag{A.42}$$

Multiplying by ρ_i, summing over clients, using $K_i - 1 \leq K_{\max} - 1$, and using $\sum_i \rho_i = 1$ yields

$$\begin{aligned} \langle \nabla^q F(\boldsymbol{\theta}_t), \mathbf{g}_t^q \rangle &\geq \sum_{i=1}^{N} \rho_i^2 \|\nabla^q f_i(\boldsymbol{\theta}_t)\| - B_q \eta_l (K_{\max} - 1) \\ &\geq c \left\|\nabla^q F(\boldsymbol{\theta}_t)\right\|^2 - B_q \eta_l (K_{\max} - 1). \end{aligned} \tag{A.43}$$

The second inequality is the block version of Lemma A.7. It is valid because $\|\nabla^q F(\boldsymbol{\theta}_t)\| \leq \|\nabla F(\boldsymbol{\theta}_t)\| \leq G_f$. Summing over layers gives

$$\begin{aligned} \langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle &= \sum_{q=1}^{Q} \langle \nabla^q F(\boldsymbol{\theta}_t), \mathbf{g}_t^q \rangle \\ &\geq c \sum_{q=1}^{Q} \left\|\nabla^q F(\boldsymbol{\theta}_t)\right\|^2 - B_{\text{layer}} \eta_l (K_{\max} - 1) \\ &= c \|\nabla F(\boldsymbol{\theta}_t)\|^2 - B_{\text{layer}} \eta_l (K_{\max} - 1). \end{aligned} \tag{A.44}$$

The layer-wise update norm satisfies

$$\|\mathbf{g}_t\|^2 = \sum_{q=1}^{Q} \|\mathbf{g}_t^q\|^2 \leq G_{\text{layer}}^2. \tag{A.45}$$

Applying L-smoothness of F to the update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_g \mathbf{g}_t$, we obtain

$$\begin{aligned} F(\boldsymbol{\theta}_{t+1}) &\leq F(\boldsymbol{\theta}_t) - \eta_g \langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t \rangle + \frac{L\eta_g^2}{2} \|\mathbf{g}_t\|^2 \\ &\leq F(\boldsymbol{\theta}_t) - c\eta_g \|\nabla F(\boldsymbol{\theta}_t)\|^2 + \eta_g B_{\text{layer}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} G_{\text{layer}}^2. \end{aligned} \tag{A.46}$$

Rearranging and summing over $t = 0, \ldots, T-1$ yields

$$\begin{aligned} c\eta_g \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\theta}_t)\|^2 &\leq \sum_{t=0}^{T-1} \left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}_{t+1})\right] + T\eta_g B_{\text{layer}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} T G_{\text{layer}}^2 \\ &\leq F(\boldsymbol{\theta}_0) - F^\star + T\eta_g B_{\text{layer}} \eta_l (K_{\max} - 1) + \frac{L\eta_g^2}{2} T G_{\text{layer}}^2. \end{aligned} \tag{A.47}$$

Dividing by $c\eta_g T$ gives (A.35). □

Interpretation. Layer-wise projection is therefore a direct block-wise generalization of Theorem A.4. The convergence rate keeps the same asymptotic form, while the drift constant becomes a sum of layer-specific terms. When some block-gradient smoothness constants L_q are smaller than the global constant L, the resulting B_q terms give a sharper drift bound.

A.4 Approximate alignment under residual constraints

The preceding results use the exact conflict-free alignment constraint $\mathbf{U}_t \mathbf{g}_t = \boldsymbol{\rho}$. In implementation, the update may satisfy this relation only approximately. This can happen when the linear system is solved in a least-squares sense, when the exact constraint is infeasible, or when the final update is modified by practical post-processing such as clipping. The next corollary isolates the effect of such deviations through an explicit residual term. It requires only that the unit-normalized alignment relation hold up to a residual vector. When the projection is exact, this residual is zero and the bound reduces to Theorem A.4. Throughout the statement, the uploaded client updates used to form the unit directions are assumed to be nonzero.

Corollary A.12 (Approximate Alignment under Residual Constraints)**.** *Consider the full-vector, full-participation setting of Theorem A.4 under Assumptions A.1–A.3, and suppose all uploaded updates are nonzero. Let $\mathbf{U}_t$ denote the unit-normalized matrix. Suppose that, instead of the exact relation $\mathbf{U}_t\mathbf{g}_t = \boldsymbol{\rho}$, the computed update satisfies*

$$\mathbf{U}_t\mathbf{g}_t = \boldsymbol{\rho} + \boldsymbol{\delta}_t. \tag{A.48}$$

Define

$$R_t := \left(G_f + \frac{LG_f}{2}\eta_l(K_{\max} - 1)\right)\sum_{i=1}^{N}\rho_i|\delta_{t,i}|. \tag{A.49}$$

Then, for any $T \geq 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{\theta}_t)\|^2 \leq \frac{F(\boldsymbol{\theta}_0) - F^\star}{c\eta_g T} + \frac{L\eta_g G^2}{2c} + \frac{B_{\text{drift}}}{c}\eta_l(K_{\max} - 1) + \frac{1}{cT}\sum_{t=0}^{T-1}R_t. \tag{A.50}$$

Proof. The residual relation gives the perturbed client-alignment identity

$$\begin{aligned}\langle \mathbf{g}_i^t, \mathbf{g}_t\rangle &= \|\mathbf{g}_i^t\|\langle \mathbf{u}_i^t, \mathbf{g}_t\rangle \\ &= (\rho_i + \delta_{t,i})\|\mathbf{g}_i^t\| \\ &\geq \rho_i\|\mathbf{g}_i^t\| - |\delta_{t,i}|\|\mathbf{g}_i^t\|.\end{aligned} \tag{A.51}$$

Relative to the exact proof, the only new term is the residual contribution. For each client,

$$\begin{aligned}\frac{\|\mathbf{g}_i^t\|}{\eta_l K_i} &\overset{(A.5)}{\leq} \|\nabla f_i^t\| + \frac{\|\mathbf{e}_i^t\|}{\eta_l K_i} \\ &\overset{(A.6)}{\leq} G_f + \frac{LG_f}{2}\eta_l(K_i - 1) \\ &\leq G_f + \frac{LG_f}{2}\eta_l(K_{\max} - 1).\end{aligned} \tag{A.52}$$

Combining this upper bound with the same Cauchy–Schwarz and reverse-triangle steps used in Lemma A.8, the client-wise lower bound becomes

$$\begin{aligned}\langle \nabla f_i^t, \mathbf{g}_t\rangle \geq& \rho_i\|\nabla f_i^t\| - \frac{LG_f}{2}(\rho_i + G)\eta_l(K_i - 1) \\ &- |\delta_{t,i}|\left(G_f + \frac{LG_f}{2}\eta_l(K_{\max} - 1)\right),\end{aligned} \tag{A.53}$$

where the first two terms are exactly the drift terms from Lemma A.8. Multiplying by ρ_i, summing over clients, applying Lemma A.7, and using the definition of R_t gives

$$\langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t\rangle \geq c\|\nabla F(\boldsymbol{\theta}_t)\|^2 - B_{\text{drift}}\eta_l(K_{\max} - 1) - R_t. \tag{A.54}$$

Substituting this inequality into the L-smooth descent bound gives

$$\begin{aligned}F(\boldsymbol{\theta}_{t+1}) &\leq F(\boldsymbol{\theta}_t) - \eta_g\langle \nabla F(\boldsymbol{\theta}_t), \mathbf{g}_t\rangle + \frac{L\eta_g^2}{2}\|\mathbf{g}_t\|^2 \\ &\leq F(\boldsymbol{\theta}_t) - c\eta_g\|\nabla F(\boldsymbol{\theta}_t)\|^2 + \eta_g B_{\text{drift}}\eta_l(K_{\max} - 1) + \eta_g R_t + \frac{L\eta_g^2}{2}G^2.\end{aligned} \tag{A.55}$$

Rearranging, summing, using $F(\boldsymbol{\theta}_T) \geq F^\star$, and dividing by $c\eta_g T$ gives (A.50). □

B Additional discussion and design choices

B.1 Comparison with momentum-based and adaptive server optimizers

CRAFT is not a momentum variant or a naive combination of ConFIG [17] and momentum methods [21, 24, 28]. Momentum-based server optimizers smooth a raw aggregate, for example

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\mathbf{a}_t, \tag{B.1}$$

where $\mathbf{a}_t$ is the aggregated current-round update. By contrast, CRAFT solves the reference-anchored projection problem

$$\mathbf{g}_t = \arg\min_{\mathbf{g}} \frac{1}{2}\|\mathbf{g} - \hat{\mathbf{g}}_t\|^2 \quad \text{s.t.} \quad \mathbf{U}_t\mathbf{g} = \boldsymbol{\rho}_t, \tag{B.2}$$

where $\hat{\mathbf{g}}_t$ is the reference direction constructed from the previous global update. The resulting update is the projection of this reference onto the affine set defined by the current conflict-free alignment constraints, not additive smoothing of the averaged update.

This distinction matters because CRAFT depends on the geometry of the active client updates. Two active-client sets can have the same averaged update but different pairwise angles or inner products. Momentum and adaptive server optimizers that operate on the average would produce the same raw aggregate in such cases, whereas CRAFT also enforces the client-wise alignment constraints and can therefore produce different updates. Thus, our contribution reformulates aggregation from zero-prior reconstruction, as in ConFIG, or temporal smoothing, as in FedAvgM methods, into a reference-anchored constrained projection that explicitly avoids conflicts with each active client.

B.2 Choice of the alignment target $\boldsymbol{\rho}_t$

In our implementation, the alignment target $\boldsymbol{\rho}_t$ in (4.1)–(4.5) is computed from the dataset sizes of the active clients. At round t, for each $i \in \mathcal{S}_t$, we set

$$\rho_i^t = \frac{|\mathcal{D}_i|}{\sum_{j\in\mathcal{S}_t} |\mathcal{D}_j|}. \tag{B.3}$$

The vector $\boldsymbol{\rho}_t = [\rho_i^t]_{i\in\mathcal{S}_t}$ is ordered consistently with the rows of $\mathbf{U}_t$. This choice matches the FL objective (3.1), where clients are weighted by their data proportions. Under partial participation, FedAvg uses the same renormalized dataset-size weights when averaging the updates from the sampled clients. Thus, $\boldsymbol{\rho}_t$ is not an extra heuristic parameter. It is the round-wise counterpart of the data-proportional weights used in federated optimization.

At the same time, $\boldsymbol{\rho}_t$ plays a structural role in CRAFT. It specifies the target alignment levels in $\mathbf{U}_t\mathbf{g}_t = \boldsymbol{\rho}_t$. A larger ρ_i^t requires stronger alignment with client i's normalized update direction and therefore gives that client greater influence in the aggregated update. In this sense, CRAFT keeps the same relative weighting principle as FedAvg, but changes the aggregation geometry so that the final update remains positively aligned with every active client.

More generally, different choices of $\boldsymbol{\rho}_t$ encode different aggregation priorities. The default data-size-based target is natural when the goal is consistency with the global FL objective. A uniform $\boldsymbol{\rho}_t$, with $\rho_i^t = 1/m$ for each of the m active clients, treats active clients equally regardless of local sample size. Appendix D.7 compares this equal target with the data-proportional target. Adaptive targets based on client reliability, heterogeneity, training progress, or fairness requirements are also possible and remain an interesting direction for future work.

B.3 Numerical active set for near-zero updates

The equality constraint $\mathbf{U}_t\mathbf{g}_t = \boldsymbol{\rho}_t$ is intended to enforce positive alignment with informative clients. It should not assign a fixed positive target to a vanishing update. Indeed, for a stabilized row

$$\widetilde{\mathbf{u}}_i^t = \frac{\mathbf{g}_i^t}{\|\mathbf{g}_i^t\| + \varepsilon}, \tag{B.4}$$

the constraint $\langle \widetilde{\mathbf{u}}_i^t, \mathbf{g}_t \rangle = \rho_i^t > 0$ implies

$$\|\mathbf{g}_t\| \geq \rho_i^t \frac{\|\mathbf{g}_i^t\| + \varepsilon}{\|\mathbf{g}_i^t\|}. \tag{B.5}$$

Thus, if $\|\mathbf{g}_i^t\|$ is close to zero while the target $\rho_i^t > 0$ and the stabilizer $\varepsilon > 0$ remain fixed, the factor $\varepsilon/\|\mathbf{g}_i^t\|$ can make the constraint numerically ill conditioned. For this reason, a more stable numerical implementation uses an active set for projection. In the layer-wise version, for each layer q, we define

$$\mathcal{A}_t^q = \{i \in \mathcal{S}_t \mid \|\mathbf{g}_i^{t,q}\| \geq \tau_q\}, \tag{B.6}$$

where $\tau_q > 0$ is a numerical tolerance, *e.g.*, $\tau_q = 10^{-6}$. The alignment matrix is then built as

$$\mathbf{U}_{t,\mathcal{A}}^q = \left[\mathcal{U}(\mathbf{g}_i^{t,q})\right]_{i\in\mathcal{A}_t^q}^\top. \tag{B.7}$$

Equivalently, the alignment target is magnitude-gated.

$$\bar{\rho}_i^{t,q} = \begin{cases} \dfrac{\rho_i^t}{\sum_{j\in\mathcal{A}_t^q}\rho_j^t}, & i \in \mathcal{A}_t^q, \\ 0, & i \notin \mathcal{A}_t^q, \end{cases} \tag{B.8}$$

whenever $\mathcal{A}_t^q \neq \emptyset$. Here $\bar{\boldsymbol{\rho}}_{t,\mathcal{A}}^q := [\bar{\rho}_i^{t,q}]_{i\in\mathcal{A}_t^q}$ denotes the vector of active entries, indexed in the same order as the rows of $\mathbf{U}_{t,\mathcal{A}}^q$. CRAFT then solves

$$\mathbf{g}_t^q = \hat{\mathbf{g}}_t^q + \left(\mathbf{U}_{t,\mathcal{A}}^q\right)^\dagger\left(\bar{\boldsymbol{\rho}}_{t,\mathcal{A}}^q - \mathbf{U}_{t,\mathcal{A}}^q\hat{\mathbf{g}}_t^q\right). \tag{B.9}$$

If $\mathcal{A}_t^q = \emptyset$, the projection for that layer is skipped and we use the ordinary weighted average update for that layer, which is already small under the same threshold criterion.

Note that the above active-set rule is not a client-sampling change. At each round, every selected client still performs local training and uploads its update. The rule only decides whether an update is sufficiently large to define a reliable row of the conflict-free alignment constraint. A near-zero update is therefore assigned a zero alignment target for numerical stability, and it automatically re-enters the conflict-free alignment set once its norm exceeds the tolerance.

B.4 Initialization and early-stage robustness

CRAFT is not sensitive to the initial direction because the reference is used as a soft prior, not as a hard constraint. At $t = 0$, we set the initial reference to zero. The first update therefore reduces to the minimum-norm feasible solution with a zero reference, namely the ConFIG-style update. This provides a neutral initialization and avoids injecting arbitrary directional bias at the start of training.

For $t > 0$, the previous update serves only as a reference orientation. At each round, it is reprojected onto the conflict-free feasible set defined by the newly sampled active clients. Consequently, even if the early active-client samples are skewed, the induced bias cannot be carried forward unchanged. Any component inherited from the previous reference that is inconsistent with the current client geometry is removed by construction through the projection. CRAFT therefore has an intrinsic self-correction mechanism that mitigates early-stage sampling bias and prevents persistent error propagation across rounds.

C Hyperparameter settings

This appendix provides the detailed experimental setup, including dataset descriptions, model architectures, scalability settings, baselines, and hyperparameter choices used in Section 5.

Datasets. We evaluate on three widely used benchmarks, FEMNIST [1], CIFAR-10, and CIFAR-100 [10]. FEMNIST is a naturally non-IID dataset containing handwritten characters from 62 classes. For CIFAR-10 and CIFAR-100, we construct heterogeneous client datasets using Dirichlet partitions [6] with a concentration parameter $\alpha = 0.1$, where smaller α indicates stronger heterogeneity. We manually set the minimum number of samples per client to 20. For each experiment, every client locally splits its assigned data into 80% training and 20% test samples. We also provide an illustration of the heterogeneous data distribution in Figure 6.

Models. For FEMNIST, we use a multi-layer perceptron (MLP) with two hidden layers of 200 units each. For CIFAR-10/100, we evaluate both convolutional neural networks (CNNs) and residual networks (ResNets [5]). The CNN consists of two convolutional layers followed by three fully connected layers, as in [23]. For ResNets, we use ResNet-20/56/110. Following the setup of [25], we replace batch normalization layers with group normalization layers using a group size of one.

Hyperparameters. For all experiments, we perform a grid search of learning rates and report the best results. We search over client learning rates $\eta_l \in \{0.005, 0.01, 0.05, 0.1\}$ and server learning rates $\eta_g \in \{0.01, 0.1, 1\}$. For client training, we apply a per-round learning rate decay of 0.999. For methods with specific hyperparameters, such as q for q-FedAvg, we use the default values recommended in their original papers. We set the batch size to 50 for MLP and CNN experiments and 20 for ResNet experiments. Each client performs one local epoch per communication round.

Baselines. We compare CRAFT with representative methods covering simple averaging (FedAvg [19]), drift mitigation (FedProx [12], FedNova [30]), server momentum/adaptive optimization (FedAvgM [6], FedAdam/AdaGrad/Yogi [25]), fairness-aware objectives (AFL [20], q-FedAvg [13], FedMGDA+ [7]), and conflict-aware aggregation (FedFV [31], FedLF [23], ConFIG [17]). We omit the history-based adaptation, as explained in Appendix D.1.

Scalability. To assess scalability, we use a large-scale setting with $N = 1000$ clients for FEMNIST and CIFAR experiments with ResNet models. To also evaluate medium-scale cross-silo settings, we conduct CIFAR experiments with CNNs using $N = 100$ clients. At each global round, we sample 10 clients for MLP and CNN experiments, and 100 clients for ResNet experiments.

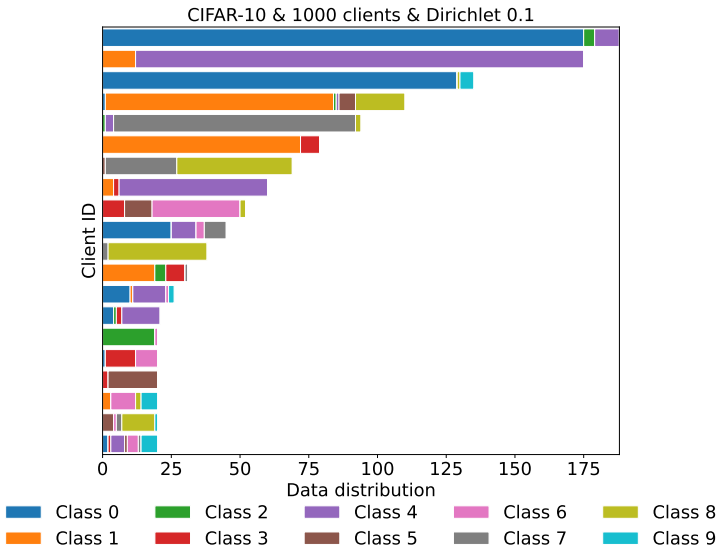

Figure 6: Data distribution across clients for CIFAR-10 with 1000 clients under the Dirichlet non-IID setting with $\alpha = 0.1$. Each row corresponds to an example client, and the color intensity indicates the proportion of samples from each class. The strong heterogeneity is evident from the concentration of classes within individual clients and from imbalanced data proportions.

D Additional results

D.1 Comparison with history-based conflict mitigation

Methods such as FedLF [23] mitigate conflicts by explicitly storing and reusing historical client gradients. In contrast, CRAFT avoids maintaining such memory-intensive history buffers. Instead, it carries temporal information through the momentum-guided reference direction $\hat{\mathbf{g}}$, which is reprojected onto the current conflict-free feasible set at each round. Figure 7 shows that CRAFT remains competitive with history-based conflict mitigation while avoiding stale-gradient storage. Therefore, for computational efficiency, we omit the history-based adaptation in all other experiments.

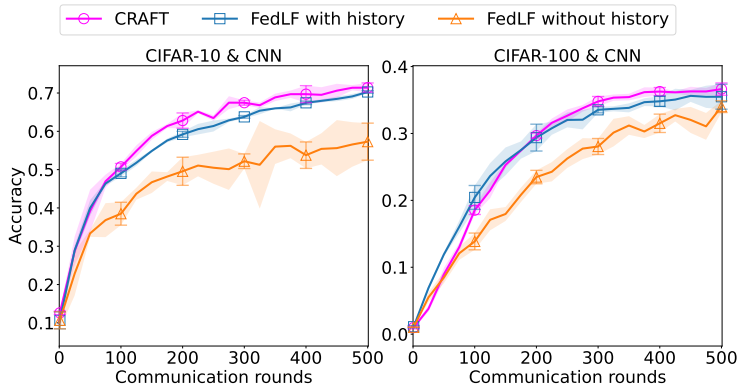

Figure 7: Comparison with history-based conflict mitigation methods.

D.2 Multiple-seed tests on CIFAR with CNN

We conduct multiple-seed tests to evaluate the stability and robustness of CRAFT compared with baselines. We run the tests on CIFAR-10/100 using CNN models with three random seeds. As shown in Figure 8, CRAFT consistently outperforms baselines across all seeds and shows lower variance in the performance metrics. This indicates that CRAFT's improvements are not due to a particular random seed or data split, but reflect a robust enhancement in handling client heterogeneity.

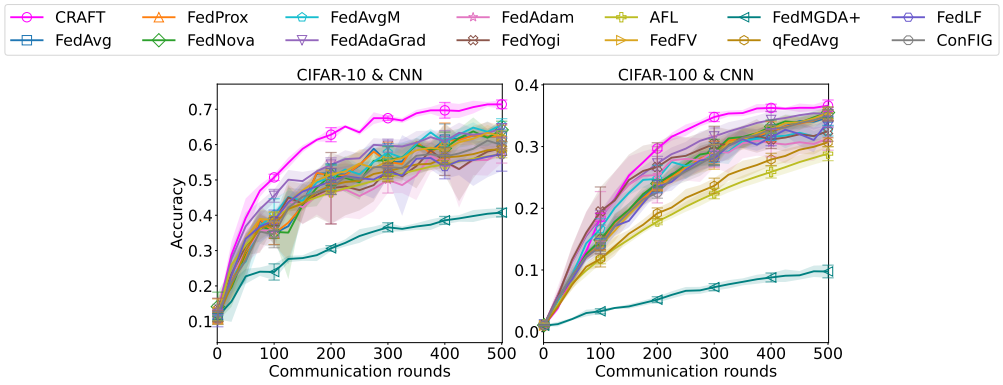

Figure 8: Evolution of mean test accuracy over three random seeds. The error bars indicate the standard deviation across seeds. CRAFT consistently performs better.

D.3 Detailed FEMNIST/CIFAR-10/CIFAR-100 results

We provide more detailed FEMNIST/CIFAR-10/CIFAR-100 results in Table 4–6 and Figure 10. In most settings, CRAFT achieves the strongest overall performance across these metrics.

Figure 9 compares the standard deviation of per-client accuracy on CIFAR-10 with ResNet-56 and ResNet-110. CRAFT achieves the lowest standard deviation in both settings, indicating a tighter distribution of client accuracy.

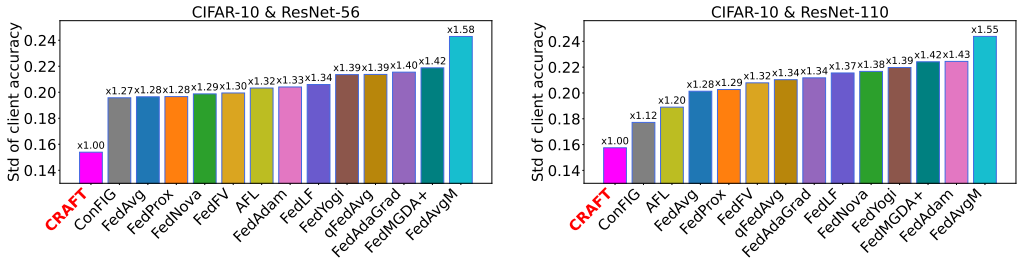

Figure 9: Client-level fairness with deeper ResNets, measured by the standard deviation of per-client accuracy. Lower values indicate a tighter distribution and therefore better fairness.

One exception is the CIFAR-100 setting, where CRAFT attains the highest standard deviation across clients. However, this does not indicate weaker robustness. Note that most baselines produce per-client accuracies that are heavily concentrated near zero. In such cases, a low standard deviation can only reflect *uniformly poor* client performance rather than fairness.

In contrast, CRAFT improves accuracy for a larger fraction of clients, thereby shifting the distribution toward higher accuracies while also increasing its spread. This behavior is also illustrated in the last column of Figure 10. CRAFT produces a more favorable per-client accuracy distribution, with a clear rightward shift and improved tail performance compared with the baselines.

Table 4: Evaluation on CIFAR-10 with different ResNets.

	CIFAR-10 & ResNet-20				CIFAR-10 & ResNet-56				CIFAR-10 & ResNet-110			
Algorithm	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)
FedAvg	0.356	0.727	0.015	0.195	0.362	0.728	0.016	0.196	0.325	0.712	0.000	0.201
FedProx	0.356	0.728	0.015	0.195	0.363	0.721	0.012	0.197	0.325	0.713	0.000	0.203
FedNova	0.381	0.734	0.040	0.190	0.378	0.737	0.022	0.199	0.329	0.741	0.000	0.217
FedAvgM	0.463	0.871	0.040	0.241	0.399	0.826	0.009	0.243	0.344	0.811	0.000	0.244
FedAdaGrad	0.427	0.799	0.072	0.206	0.401	0.792	0.026	0.215	0.361	0.767	0.005	0.212
FedAdam	0.412	0.790	0.065	0.197	0.376	0.760	0.020	0.204	0.348	0.785	0.001	0.224
FedYogi	0.398	0.777	0.047	0.201	0.368	0.771	0.010	0.214	0.336	0.760	0.000	0.220
AFL	0.294	0.732	0.000	0.217	0.294	0.701	0.000	0.203	0.250	0.597	0.000	0.189
FedFV	0.421	0.780	0.078	0.194	0.416	0.793	0.076	0.199	0.385	0.779	0.029	0.208
FedMGDA+	0.235	0.612	0.000	0.193	0.249	0.693	0.000	0.219	0.252	0.693	0.000	0.224
qFedAvg	0.352	0.716	0.019	0.191	0.350	0.760	0.006	0.214	0.310	0.712	0.000	0.210
FedLF	0.501	0.845	0.149	0.208	0.494	0.831	0.132	0.206	0.468	0.822	0.082	0.215
ConFIG	0.525	0.872	0.146	0.211	0.317	0.694	0.000	0.196	0.185	0.556	0.000	0.177
CRAFT	**0.806**	**1.000**	**0.479**	**0.159**	**0.823**	**1.000**	**0.500**	**0.154**	**0.818**	**1.000**	**0.486**	**0.158**

Table 5: Evaluation on CIFAR-100 with different ResNets.

	CIFAR-100 & ResNet-20				CIFAR-100 & ResNet-56				CIFAR-100 & ResNet-110			
Algorithm	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)
FedAvg	0.122	0.336	0.000	0.104	0.097	0.297	0.000	0.097	0.091	0.273	0.000	0.088
FedProx	0.122	0.336	0.000	0.104	0.097	0.298	0.000	0.097	0.093	0.281	0.000	0.090
FedNova	0.124	0.337	0.000	0.105	0.100	0.298	0.000	0.097	0.086	0.281	0.000	0.091
FedAvgM	0.213	0.462	0.008	0.130	0.166	0.398	0.000	0.117	0.141	0.372	0.000	0.114
FedAdaGrad	0.178	0.433	0.000	0.127	0.132	0.352	0.000	0.108	0.121	0.328	0.000	0.103
FedAdam	0.174	0.415	0.000	0.121	0.123	0.335	0.000	0.105	0.111	0.314	0.000	0.101
FedYogi	0.172	0.408	0.000	0.120	0.116	0.327	0.000	0.104	0.078	0.265	0.000	0.087
AFL	0.067	0.244	0.000	0.079	0.053	0.213	0.000	0.072	0.044	0.194	0.000	0.067
FedFV	0.139	0.352	0.000	0.110	0.120	0.329	0.000	0.104	0.097	0.295	0.000	0.095
FedMGDA+	0.029	0.164	0.000	**0.057**	0.011	0.098	0.000	**0.032**	0.013	0.114	0.000	**0.040**
qFedAvg	0.126	0.349	0.000	0.108	0.098	0.291	0.000	0.095	0.081	0.259	0.000	0.085
FedLF	0.206	0.443	0.000	0.127	0.149	0.383	0.000	0.113	0.141	0.360	0.000	0.110
ConFIG	0.208	0.458	0.000	0.131	0.181	0.418	0.000	0.121	0.072	0.250	0.000	0.084
CRAFT	**0.398**	**0.673**	**0.137**	0.154	**0.381**	**0.661**	**0.112**	0.156	**0.363**	**0.629**	**0.110**	0.149

Table 6: Evaluation on FEMNIST and CIFAR-10 with different networks.

	FEMNIST & MLP				CIFAR-10 & CNN				CIFAR-100 & CNN			
Algorithm	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)	Mean (↑)	Best (↑)	Worst (↑)	Std (↓)
FedAvg	0.570	0.776	0.305	0.136	0.636	0.856	0.318	0.167	0.335	0.474	0.209	0.079
FedProx	0.565	0.770	0.300	0.135	0.640	0.867	0.335	0.162	0.342	0.485	0.223	0.079
FedNova	0.589	0.804	0.337	0.135	0.667	0.867	0.413	0.138	0.345	**0.490**	0.229	0.078
FedAvgM	0.655	0.843	0.408	0.125	0.662	0.846	0.491	**0.108**	0.343	0.446	0.201	0.070
FedAdaGrad	0.645	0.828	0.405	0.121	0.648	0.850	0.472	0.116	0.340	0.464	0.236	0.068
FedAdam	0.638	0.825	0.390	0.125	0.621	0.872	0.272	0.173	0.314	0.456	0.207	0.069
FedYogi	0.639	0.830	0.381	0.130	0.607	0.865	0.345	0.152	0.333	0.453	0.237	0.063
AFL	0.548	0.774	0.301	0.137	0.562	0.733	0.285	0.134	0.276	0.394	0.172	0.062
FedFV	0.489	0.742	0.246	0.144	0.641	0.875	0.361	0.150	0.346	0.474	0.225	0.074
FedMGDA+	0.386	0.658	0.133	0.166	0.410	0.690	0.083	0.182	0.090	0.197	0.017	**0.051**
qFedAvg	0.555	0.774	0.298	0.137	0.586	0.808	0.336	0.143	0.299	0.420	0.197	0.065
FedLF	0.426	0.688	0.177	0.154	0.603	0.872	0.303	0.165	0.332	0.482	0.212	0.075
ConFIG	0.600	0.801	0.333	0.136	0.609	0.834	0.362	0.142	0.341	0.472	0.218	0.077
CRAFT	**0.706**	**0.878**	**0.474**	**0.115**	**0.722**	**0.896**	**0.491**	0.116	**0.359**	0.484	**0.255**	0.069

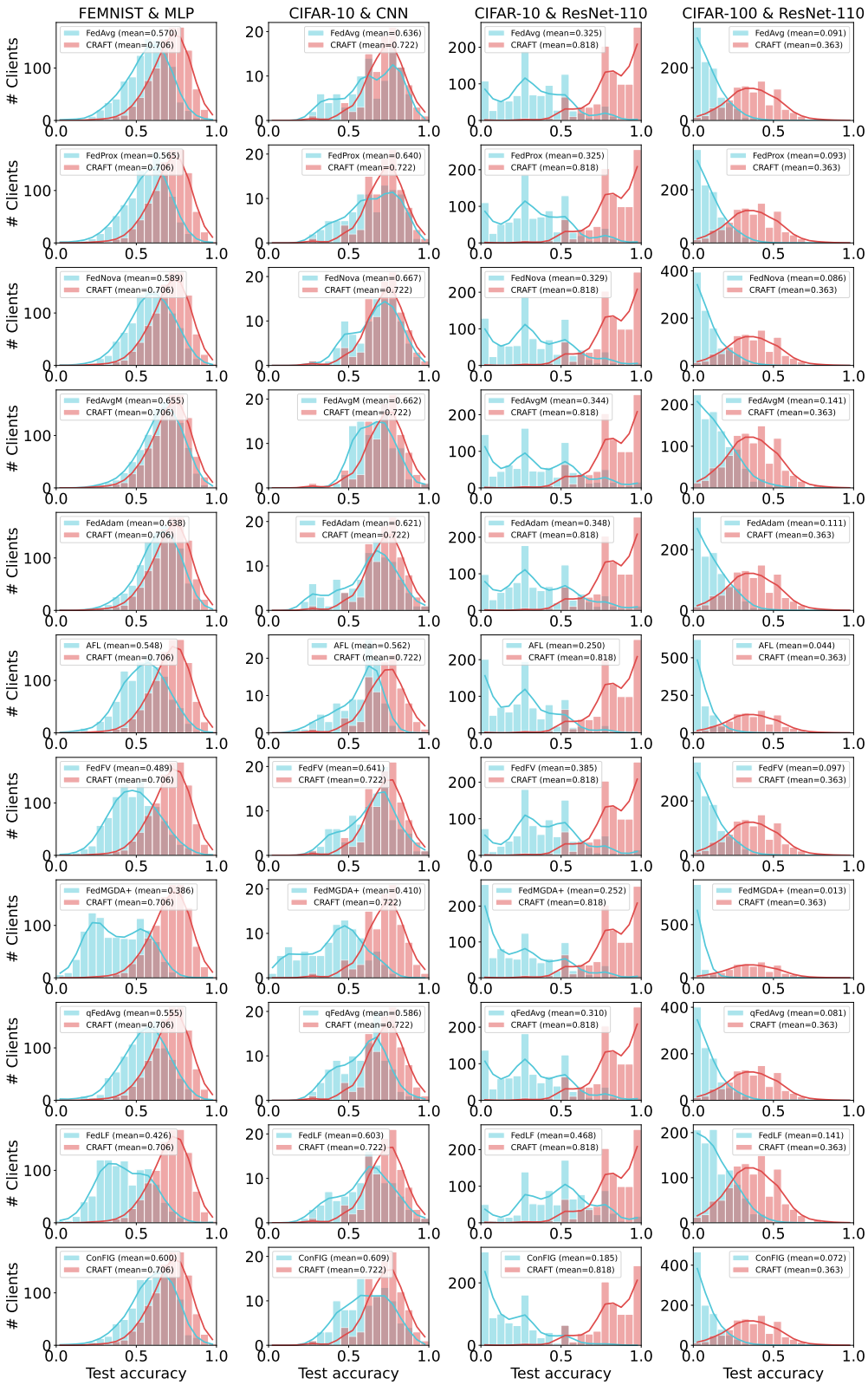

Figure 10: Distribution of per-client test accuracy. Each row corresponds to a baseline, and each column corresponds to a test setting. A rightward shift indicates higher accuracy for more clients, and CRAFT produces a more favorable distribution with improved mean and tail performance.

D.4 Robustness to intermediate data heterogeneity

To assess robustness across different non-IID distributions, we extend the evaluation from the highly heterogeneous Dirichlet setting with $\alpha = 0.1$ to milder heterogeneity levels with $\alpha \in \{0.5, 1\}$. A larger concentration parameter yields more similar class distributions across clients. We use CIFAR-10 and CIFAR-100 with ResNet-20, $N = 1000$ total clients, and $m = 100$ active clients per communication round. All methods follow the same evaluation protocol, with the remaining hyperparameters specified in Appendix C.

Table 7 reports test accuracy at round 500 for CRAFT and four baselines. CRAFT ranks first in all six settings. ConFIG achieves the second-best result in five settings, while FedAvgM ranks second on CIFAR-100 at $\alpha = 0.1$. At the milder heterogeneity levels $\alpha = 0.5$ and $\alpha = 1$, CRAFT maintains an accuracy of 0.812 on CIFAR-10 and achieves 0.418 and 0.421 on CIFAR-100, respectively. These results show that CRAFT's advantage over the baselines persists across the tested non-IID distributions and is not confined to the most heterogeneous regime.

Table 7: Test accuracy under different Dirichlet heterogeneity levels. **Bold** and underlined values indicate the best and second-best results in each column, respectively.

	CIFAR-10			CIFAR-100		
Algorithm	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$
FedAvg	0.356	0.375	0.390	0.122	0.135	0.136
FedNova	0.381	0.376	0.385	0.124	0.136	0.137
FedAvgM	0.463	0.351	0.366	<u>0.213</u>	0.103	0.099
ConFIG	<u>0.525</u>	<u>0.591</u>	<u>0.625</u>	0.208	<u>0.222</u>	<u>0.248</u>
CRAFT	**0.806**	**0.812**	**0.812**	**0.398**	**0.418**	**0.421**

D.5 Larger-scale image benchmark: Tiny-ImageNet

We further evaluate CRAFT on Tiny-ImageNet [11], a larger and more challenging natural-image benchmark. We use ResNet-20 and partition the data among $N = 1000$ clients using a Dirichlet distribution with concentration parameter $\alpha = 0.1$. At each communication round, we sample $m = 100$ active clients. All other hyperparameters follow the setup in Appendix C.

Table 8 reports test accuracy after 1000 communication rounds for CRAFT, two component variants, and four baselines. CRAFT-no-Layerwise uses the same momentum-like reference without layer-wise adaptation, while CRAFT-no-Momentum uses a zero reference with layer-wise adaptation (ConFIG-Layerwise). CAGrad [16] is applied to client updates for server-side aggregation.

CRAFT achieves the highest test accuracy, exceeding the second-best CRAFT-no-Momentum by 10.4 percentage points. Removing layer-wise adaptation reduces accuracy from 0.325 to 0.061. With a zero reference, layer-wise adaptation also improves accuracy from 0.094 for ConFIG to 0.221 for CRAFT-no-Momentum. These results support the importance of layer-wise adaptation on this larger benchmark and the effectiveness of its combination with the momentum-like reference. Overall, CRAFT remains effective beyond the smaller image benchmarks evaluated in the main experiments.

Table 8: Test accuracy on Tiny-ImageNet with ResNet-20 after 1000 communication rounds. **Bold** and underlined values indicate the best and second-best results, respectively.

Algorithm	Test accuracy (↑)
FedAvg	0.073
FedAvgM	0.027
ConFIG	0.094
CAGrad	0.085
CRAFT-no-Layerwise	0.061
CRAFT-no-Momentum	<u>0.221</u>
CRAFT	**0.325**

D.6 Comparison with additional conflict-aware methods

FedHCA^2 [18] is a federated multi-task learning framework. Its Hyper Conflict-Averse Aggregation (HCA) module adapts CAGrad's conflict-averse objective [16] from task gradients to client updates and computes a shared encoder update. The full framework combines this update with personalization mechanisms to produce client-specific models. We therefore evaluate CRAFT both as an aggregator for a single shared global model and as a replacement aggregation module within an adapted personalized pipeline, similar to the integration of CRAFT with Ditto in Section 5.4.

Both comparisons use CIFAR-10 and CIFAR-100 with ResNet-20, $N = 1000$ clients, Dirichlet $\alpha = 0.1$, and $m \in \{10, 50, 100\}$ active clients per round. The tables report accuracy at round 500.

Shared global-model comparison. We adapt CAGrad for server-side aggregation by treating active-client updates as objective-specific directions, following the update-space formulation of HCA. This comparison trains one shared model without personalization. Table 9 shows that CRAFT achieves the highest accuracy on both datasets at every participation level, with larger gains over CAGrad at moderate and high participation.

Table 9: Shared global-model test accuracy: CRAFT, CAGrad applied to client updates, and FedAvg.

	CIFAR-10			CIFAR-100		
Algorithm	$m = 10$	$m = 50$	$m = 100$	$m = 10$	$m = 50$	$m = 100$
FedAvg	0.260	0.337	<u>0.356</u>	<u>0.092</u>	<u>0.117</u>	0.122
CAGrad	<u>0.291</u>	<u>0.346</u>	0.350	0.088	<u>0.117</u>	<u>0.129</u>
CRAFT	**0.417**	**0.731**	**0.806**	**0.188**	**0.391**	**0.398**

Personalized comparison within adapted FedHCA^2. In our CIFAR setting, all clients perform the same classification task with ResNet-20. The decoder cross-attention mechanism for incongruent task-specific decoders in the full FedHCA^2 framework is therefore not applicable. We construct an adapted FedHCA^2 pipeline that retains client-specific models and Hyper Aggregation Weights. We replace only the HCA aggregation rule with CRAFT, keeping all applicable personalization components fixed.

Table 10: Personalized test accuracy within the same adapted FedHCA^2 pipeline, using either HCA or CRAFT as the aggregation module.

	CIFAR-10			CIFAR-100		
Aggregation module	$m = 10$	$m = 50$	$m = 100$	$m = 10$	$m = 50$	$m = 100$
HCA	<u>0.566</u>	<u>0.608</u>	<u>0.613</u>	<u>0.117</u>	<u>0.160</u>	<u>0.181</u>
CRAFT	**0.576**	**0.723**	**0.762**	**0.132**	**0.261**	**0.297**

Table 10 shows that replacing HCA with CRAFT improves personalized accuracy in every setting, with larger gains as participation increases. Together, these comparisons support CRAFT's effectiveness for shared global-model training and its use as an aggregation module in this adapted personalized pipeline, complementing the Ditto integration in Section 5.4.

D.7 Sensitivity to the alignment target

We compare the data-proportional alignment target discussed in Appendix B.2 with the equal target $\rho_i^t = 1/m$ for each active client. CRAFT denotes the original data-proportional variant, while CRAFT-Equal uses the equal target. Both variants use CIFAR-10 and CIFAR-100 with ResNet-20, $N = 1000$ total clients, Dirichlet heterogeneity $\alpha = 0.1$, and $m \in \{10, 50, 100\}$ active clients per round. Table 11 reports test accuracy at round 500 using the hyperparameter selection protocol in Appendix C.

The equal target achieves higher accuracy in five of the six settings. On CIFAR-10, the improvement is largest under sparse participation ($m = 10$), increasing accuracy from 0.417 to 0.448. At $m = 100$,

Table 11: Round-500 test accuracy with equal and data-proportional alignment targets. **Bold** and underlined values indicate the best and second-best results in each column, respectively.

	CIFAR-10			CIFAR-100		
Algorithm	$m = 10$	$m = 50$	$m = 100$	$m = 10$	$m = 50$	$m = 100$
CRAFT-Equal	**0.448**	**0.753**	<u>0.805</u>	**0.195**	**0.394**	**0.418**
CRAFT	<u>0.417</u>	<u>0.731</u>	**0.806**	<u>0.188</u>	<u>0.391</u>	<u>0.398</u>

the two targets perform nearly identically, with 0.805 for CRAFT-Equal and 0.806 for CRAFT. On CIFAR-100, the equal target improves accuracy at every participation level. These results support equal targeting as a useful alternative, while the original data-proportional CRAFT already outperforms the baselines in Table 9.

For positive round-wise targets satisfying $\sum_{i\in\mathcal{S}_t} \rho_i^t = 1$, the minimum target $\rho_{\min}^t := \min_{i\in\mathcal{S}_t} \rho_i^t$ is at most $1/m$, with equality only for equal targets. This provides motivation for balanced target allocation in view of the inverse dependence on $\rho_{\min}$ in Theorem A.4.

D.8 Layer-wise grouping ablation

We isolate the effect of grouping granularity in CRAFT on CIFAR-10 and CIFAR-100 using ResNet-20 with $N = 1000$ clients. All configurations retain the same momentum-like reference and follow the same training and evaluation protocol, with the remaining settings specified in Appendix C.

We compare five architecture-aligned partitions. With $Q = 1$, the full trainable parameter vector forms one group (CRAFT-no-Layerwise). With $Q = 2$, the feature extractor and the final classifier form separate groups. With $Q = 5$, the stem, the three residual stages, and the classifier each form one group. With $Q = 11$, the stem, each of the nine residual blocks, and the classifier form separate groups, with the two convolutions within each block grouped together. The default $Q = 20$ separates the stem convolution, the two convolutions in each residual block, and the classifier. In every partition, normalization parameters remain with their associated convolution, as specified in Section 4.4.

The results of varying client participation are shown in Table 3 and discussed in Section 5.5. In this part, we focus on the effect of varying data heterogeneity at fixed $m = 100$. Table 12 shows that $Q = 11$ performs best in all four settings with milder heterogeneity ($\alpha \in \{0.5, 1\}$). The default $Q = 20$ outperforms $Q = 1$ throughout, while either $Q = 11$ or $Q = 20$ achieves the highest accuracy in each column. Together with Table 3, these results show that useful grouping granularity depends on participation and heterogeneity, and that accuracy does not increase monotonically with the number of groups.

We retain the architecture-derived layer-level partition as the default because it provides a simple, deterministic rule without data-dependent tuning. Adaptive grouping would introduce an additional design choice and is left for future work.

Table 12: Layer-wise grouping ablation with varying Dirichlet heterogeneity α at fixed $m = 100$: test accuracy at round 500 on CIFAR-10 and CIFAR-100 with ResNet-20 and $N = 1000$ clients. All variants retain the momentum-like reference. **Bold** and underlined values indicate the best and second-best results in each column, respectively.

	CIFAR-10			CIFAR-100		
Configuration	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$
CRAFT ($Q = 1$)	0.462	0.712	0.724	0.152	0.257	0.283
CRAFT ($Q = 2$)	0.530	0.722	0.707	0.312	0.345	0.350
CRAFT ($Q = 5$)	0.754	0.791	0.796	0.394	<u>0.421</u>	<u>0.425</u>
CRAFT ($Q = 11$)	<u>0.798</u>	**0.833**	**0.825**	**0.415**	**0.424**	**0.429**
CRAFT ($Q = 20$)	**0.806**	<u>0.812</u>	<u>0.812</u>	<u>0.398</u>	0.418	0.421

NeurIPS paper checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the assumptions and complete proofs in Appendix A.2

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided all the technical details needed to reproduce the results in Section 5.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code for CRAFT is available at `https://github.com/tum-pbs/CRAFT`. We use datasets from other research that are already publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided all the experimental details needed to reproduce the results in Section 5.1 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have provided error bars for the moderate-scale experiments in Appendix D.2. For the large-scale experiments, we do not report error bars due to the substantial computational cost. Nevertheless, the performance gaps are sufficiently large and consistent across settings, making it unlikely that the observed improvements are caused by random variation alone.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The training cost of the current study is discussed in Section 4.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research reported in the paper complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We don't foresee any significant societal impacts directly associated with our work to be highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The current paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The current paper uses datasets from other research work, which are correctly cited in Section 5.1.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The current submission doesn't release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.