FedCALM: Conflict-aware Layer-wise Mitigation for Selective Aggregation in Deeper Personalized Federated Learning

Supplementary Material

8. Algorithm Details

8.1. Server Aggregation Conflicts

As the number of layers k increases, the cumulative conflict can be expressed as the sum of conflicts across all layers:

$$\Delta\mathcal{L}_j = -\frac{1}{\alpha}\left(\sum_{k=1}^{K}\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} + \sum_{k\neq l}\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(l)}\right) \tag{9}$$

In our analysis, since the parameters Θ are aggregated layer by layer without cross-layer interactions, we can safely ignore the cross-layer inner product terms when considering the change in the loss function. Specifically, the inner product of the total updates simplifies to $\Delta\theta_i \cdot \Delta\theta_j = \sum_{k=1}^{K}\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)}$, as the cross terms $\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(l)}$ for $k \neq l$ are zero due to the absence of cross-layer aggregation. Therefore, the change in the loss function can be expressed as $\Delta\mathcal{L}_j = -\frac{1}{\alpha}\sum_{k=1}^{K}\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)}$, which aligns with the summation over each layer as in Eq.3. This equivalence indicates that when many layers satisfy $\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} < 0$, the negative contributions accumulate, leading to an overall increase in the loss $\mathcal{L}_j$.

Like most methods that use weighted averaging to aggregate all clients on the server (e.g., $\Delta\theta_{\text{G}}^{(k)} \leftarrow \Delta\theta_{\text{avg}}^{(k)} = \sum_{i=1}^{N} p_i\Delta\theta_i^{(k)}$), these existing conflicts have an overall impact on the total loss as follows:

$$\begin{aligned}
\Delta\mathcal{L}_{\text{total}} &= \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\left(-\frac{1}{\alpha}\sum_{k=1}^{K}p_ip_j\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)}\right)\\
&= -\frac{1}{\alpha}\sum_{k=1}^{K}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}p_ip_j\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} - \sum_{i=1}^{N}p_i^2\|\Delta\theta_i^{(k)}\|^2\right)\\
&= -\frac{1}{\alpha}\sum_{k=1}^{K}\left(\|\Delta\theta_{\text{avg}}^{(k)}\|^2 - \sum_{i=1}^{N}p_i^2\|\Delta\theta_i^{(k)}\|^2\right)
\end{aligned} \tag{10}$$

where $\sum_{i=1}^{N}\sum_{j=1}^{N}p_ip_j\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} = \|\Delta\theta_{\text{avg}}^{(k)}\|^2$ represents the squared norm of the average update vector for the k-th layer, and $\sum_{i=1}^{N}p_i^2\|\Delta\theta_i^{(k)}\|^2$ accounts for the weighted sum of the squared norms of individual client updates for the same layer.

Furthermore, to quantify this conflict, we define the layer-wise conflicting client update and client update conflict rate as follows:

Definition 1 (Layer-wise conflicting client updates). For each layer k, the updates $\Delta\theta_i^{(k)}$ and $\Delta\theta_j^{(k)}$ $(i \neq j)$ are said to be conflicting with each other if $\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} < 0$.

Definition 2 (Client updates conflict rate). Given N clients, each with K layers, the total number of layer-wise pairs across clients is $\binom{N}{2}\times K$. Let $\mathcal{P}$ denote the total number of conflict pairs detected across all clients and layers. The conflict rate is then defined as $\mathcal{P}/\frac{N(N-1)}{2}\times K$.

where the value of $\mathcal{P}$ is determined as follows, if $\Delta\theta_i^{(k)}\cdot\Delta\theta_j^{(k)} < 0$, a conflict is recorded for that layer and the conflict count is incremented by 1. This metric reflects the extent of directional inconsistency among client updates during the layer-wise aggregation process. A higher conflict rate indicates a greater level of inconsistency, leading to an amplified negative impact on the global model as the model depth increases.

8.2. Lagrangian Function and Dual Problem

Since the primal problem is a convex optimization problem and satisfies the Slater condition, strong duality holds. Thus, the optimal solution to the primal problem can be obtained by solving the dual problem. To solve the optimization problem formulated in Section 4.2, we construct the Lagrangian function by introducing Lagrange multipliers $\lambda_i \geq 0$ and $\mu_i \geq 0$ corresponding to the inequality constraints:

$$\begin{aligned}
L(d,\xi,\lambda,\mu) = \frac{1}{2}\|d-\Delta\bar{\theta}_\perp\|^2 + C\sum_{i=1}^{N}\xi_i\\
+\sum_{i=1}^{N}\lambda_i\left(\epsilon\|\Delta\theta_i\| - d^\top\Delta\theta_i - \xi_i\right)\\
+\sum_{i=1}^{N}\mu_i(-\xi_i),
\end{aligned} \tag{11}$$

where d is the adjusted global update we aim to find, $\Delta\bar{\theta}_\perp$ is the global conflict-free guidance vector obtained after the first-stage projection, $\Delta\theta_i$ is the update of client i, $\xi_i \geq 0$ are slack variables allowing for constraint violations, λ_i and μ_i are Lagrange multipliers associated with the inequality constraints.

Taking the partial derivatives of the Lagrangian L with respect to d and ξ_i and setting them to zero yields the fol-

lowing optimality conditions:

$$d = \Delta\overline{\theta}_{\perp} + \sum_{i=1}^{N} \lambda_i \Delta\theta_i. \tag{12}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0. \tag{13}$$

where shows that the optimal global update d is a combination of the global conflict-free update $\Delta\overline{\theta}_{\perp}$ and a weighted sum of the individual client updates $\Delta\theta_i$, where the weights are the Lagrange multipliers λ_i. Since $\mu_i \geq 0$, it follows that $\lambda_i \leq C, \quad \forall i = 1, \ldots, N$. This inequality constrains the Lagrange multipliers λ_i to be less than or equal to the penalty parameter C.

Combining these results, we substitute d back into the Lagrangian to eliminate the primal variables and formulate the dual problem. The dual objective function becomes:

$$\begin{aligned} \max_{\lambda} \quad & \epsilon \sum_{i=1}^{N} \lambda_i \|\Delta\theta_i\| - \sum_{i=1}^{N} \lambda_i \Delta\overline{\theta}_{\perp}^{T} \Delta\theta_i \\ & - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j \Delta\theta_i^T \Delta\theta_j \\ \text{s.t.} \quad & 0 \leq \lambda_i \leq C, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{14}$$

In this dual problem, the term $\epsilon \sum_{i=1}^{N} \lambda_i \|\Delta\theta_i\|$ encourages the gloabl update d to align positively with each client's update $\Delta\theta_i$. The term $\sum_{i=1}^{N} \lambda_i (\Delta\overline{\theta}_{\perp})^T \Delta\theta_i$ accounts for the interaction between the global conflict-free update and each client's update. The quadratic term $\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j (\Delta\theta_i)^{\top} \Delta\theta_j$ captures the pairwise interactions between client updates.

By solving this dual optimization problem, we obtain the optimal Lagrange multipliers λ_i^*. These multipliers are then used to compute the optimal global update $d^* = \Delta\overline{\theta}_{\perp} + \sum_{i=1}^{N} \lambda_i^* \Delta\theta_i$. This adjusted global update d^* balances the need to follow the global conflict-free update while mitigating conflicts between clients, as enforced by the Lagrange multipliers.

The KKT conditions are as follows:

1. Primal Constraints:

$$\begin{aligned} \epsilon \|\Delta\theta_i\| - d^T \Delta\theta_i \leq \xi_i, \quad \forall i = 1, \ldots, N. \\ \xi_i \geq 0, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{15}$$

2. Lagrange Multiplier Conditions:

$$\begin{aligned} \lambda_i \geq 0, \quad \forall i = 1, \ldots, N. \\ \mu_i \geq 0, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{16}$$

3. Dual Feasibility Condition:

$$d = \Delta\overline{\theta}_{\perp} + \sum_{i=1}^{N} \lambda_i \Delta\theta_i \tag{17}$$

4. Complementary Slackness Condition:

$$\begin{aligned} \lambda_i (\epsilon \|\Delta\theta_i\| - d^T \Delta\theta_i - \xi_i) = 0, \quad \forall i = 1, \ldots, N. \\ \mu_i \xi_i = 0, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{18}$$

According to the complementary slackness condition in the KKT conditions, the optimal solution must satisfy $\lambda_i(\epsilon\|\Delta\theta_i\| - d^{\top}\Delta\theta_i - \xi_i) = 0$. When there is a conflict $(\epsilon\|\Delta\theta_i\| - d^{\top}\Delta\theta_i - \xi_i > 0)$ and $\lambda_i > 0$, the constraint is tightly binding, requiring adjustment of d to reduce the conflict. When $\lambda_i = 0$, it indicates that the constraint for the i-th client is inactive, and there is no significant conflict between d and $\Delta\theta_i$. Through the KKT conditions, it is revealed that our method can detect conflicts among clients, focusing selectively on clients where conflicts exist to avoid negative impacts on other clients.

9. Ablation and Additional Orthogonal Studies

9.1. Effect of Two-Stage Strategy in FedCALM

There are two stages in FedCALM: Project Conflicting Client Updates and Conflict-aware Mitigation Strategy. In the first stage, conflicting updates are projected onto the orthogonal plane of the corresponding updates layer by layer, resulting in a conflict-free global update vector. The second stage further optimizes this global conflict-free target vector using a conflict-aware strategy. To evaluate the effectiveness of these stages, we conducted ablation experiments on four datasets: one using only the first stage to generate the global conflict-free target vector (FedCALM_Only Proj) and another utilizing both stages (FedCALM_Two Stage).

As shown in Tables 6 and 7, using only the first stage (FedCALM_Only Proj) achieves the second-best performance across all four datasets. This is because projecting updates effectively mitigates negative transfer caused by parameter conflicts from other clients. However, this approach directly operates on updates without considering the underlying relevance of conflicts to client-specific tasks. To account for this, in the second stage, we balance the trade-off between clients involved in aggregation and the tolerance for conflicts around the global conflict-free target vector, maximizing its effectiveness for all clients. The tables demonstrate that our two-stage approach achieves further performance improvements and highlights its necessity.

9.2. Effect of the Hyperparameters ϵ and C

To determine the optimal update vector within the conflict-free target vector space, two hyperparameters are used: ϵ and C. Here, C is a penalty parameter that regulates the severity of penalties for constraint violations, while ϵ is a small positive constant that defines a minimum threshold for positive contributions from each client.

Tables 8 and 9 evaluate the impact of the hyperparameters ϵ and C on the accuracy of FedCALM, using ResNet4

Methods		Flowers102				CIFAR100			
		ResNet4	ResNet10	ResNet18	ResNet34	ResNet4	ResNet10	ResNet18	ResNet34
FedCP		62.82	70.90	67.15	64.48	58.35	59.63	58.21	57.12
FedPAC		63.55	77.08	74.21	65.79	59.89	62.07	58.62	56.63
FedCALM	Only Proj	<u>63.89</u>	<u>77.69</u>	<u>75.57</u>	<u>69.10</u>	<u>62.11</u>	<u>62.23</u>	<u>58.96</u>	<u>57.52</u>
	Two Stage	**75.57**	**80.24**	**75.80**	**71.14**	**65.20**	**62.59**	**61.13**	**60.79**

Table 6. The ablation experiments for the two stages of FedCALM were conducted on the Flowers102 and CIFAR100 datasets. Only Proj represents the use of only the first stage, Project Conflicting Client Updates, while Two Stage refers to the full two-stage FedCALM approach. The bold numbers indicate the best performance, while the underlined values denote the second-best methods.

Methods		CIFAR10				CINIC10			
		ResNet4	ResNet10	ResNet18	ResNet34	ResNet4	ResNet10	ResNet18	ResNet34
FedCP		91.13	90.08	89.91	89.72	87.91	85.90	85.78	85.52
FedPAC		89.41	90.53	89.40	87.57	88.22	86.94	86.70	85.84
FedCALM	Only Proj	<u>92.14</u>	<u>91.03</u>	<u>90.40</u>	<u>90.04</u>	<u>89.39</u>	<u>87.08</u>	<u>86.95</u>	<u>86.90</u>
	Two Stage	**93.00**	**92.26**	**91.68**	**91.55**	**89.69**	**87.18**	**87.09**	**87.25**

Table 7. The ablation experiments for the two stages of FedCALM were conducted on the CIFAR10 and CINIC10 datasets. Only Proj represents the use of only the first stage, Project Conflicting Client Updates, while Two Stage refers to the full two-stage FedCALM approach. The bold numbers indicate the best performance, while the underlined values denote the second-best methods.

ϵ \ C	0.1	1	10	50	100
0.01	62.39	63.17	63.25	63.17	63.15
0.1	62.64	63.89	**65.20**	64.88	64.76
1	62.61	61.58	57.08	57.55	57.97

Table 8. The ablation experiments for different hyperparameters ϵ and C using ResNet4 as the base model on CIFAR100 dataset.

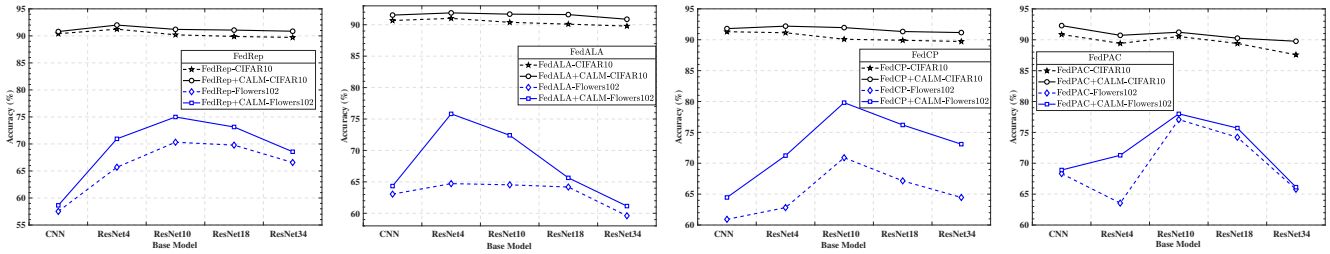
ϵ \ C	0.1	1	10	50	100
0.01	62.00	62.18	62.15	61.87	61.82
0.1	62.48	62.51	**62.67**	61.61	61.15
1	61.92	62.31	62.59	60.63	60.13

Table 9. The ablation experiments for different hyperparameters ϵ and C using ResNet10 as the base model on CIFAR100 dataset.

and ResNet10 as the base models on the CIFAR100 dataset. The results show that the highest accuracies of 65.20% and 62.67% are achieved when $\epsilon = 0.1$ and $C = 10$, respectively. The hyperparameter ϵ determines the degree of positive contribution from the global update to each client. With ϵ fixed, accuracy initially increases and then decreases as C increases, indicating the existence of an optimal balance point. At this balance point, the strict enforcement of constraints and the minimization of the objective function are optimally aligned, effectively resolving conflicts and promoting client collaboration. However, when C exceeds this balance point, overly large penalties for constraint violations lead the optimization process to focus excessively on satisfying constraints, ultimately resulting in a decline in model performance.

The optimal hyperparameters of FedCALM using different base models across the four datasets are as follows: On the CIFAR10 dataset, the optimal hyperparameters for the CNN base model are $\epsilon = 0.1$ and $C = 0.1$, while for ResNet4, the optimal hyperparameters are $\epsilon = 0.1$ and $C = 1$. For ResNet10, ResNet18, and ResNet34, the optimal hyperparameters are $\epsilon = 0.1$ and $C = 10$. On the CIFAR100 dataset, the optimal hyperparameters for all base models are $\epsilon = 0.1$ and $C = 10$. On the Flowers102 dataset, the optimal hyperparameters for the CNN and ResNet4 base models are $\epsilon = 0.1$ and $C = 100$, while for ResNet10, ResNet18, and ResNet34, the optimal hyperparameters are $\epsilon = 0.1$ and $C = 10$. On the CINIC10 dataset, the optimal hyperparameters for all base models are $\epsilon = 0.1$ and $C = 1$.

In summary, FedCALM demonstrates robust performance without requiring complex hyperparameter tuning. Across all datasets, the optimal value for ϵ consistently remains $\epsilon = 0.1$, while the optimal value for C varies within a manageable range of $\{0.1, 1, 10, 100\}$. This simplicity in parameter selection highlights the adaptability and practical



(a) FedRep combined with CALM. (b) FedALA combined with CALM. (c) FedCP combined with CALM. (d) FedPAC combined with CALM.

Figure 5. Average test accuracies (%) were reported for SOTA methods with the CALM aggregation strategy, under increasing base model depth on CIFAR10 and Flowers102 datasets. Black represents the CIFAR10 dataset, while blue represents the Flowers102 dataset. Dashed lines correspond to the original methods, and solid lines represent methods combined with our CALM aggregation strategy.

applicability of FedCALM in diverse settings.

9.3. Additional Orthogonal Experiments

The consistent improvement across all SOTA methods highlights CALM's flexibility and compatibility. For instance, in FedCP combined with CALM (Fig.5c), the accuracy on Flowers102 improves significantly for deeper base models, with ResNet10 showing a considerable gain compared to the original method. On the CIFAR10 dataset (black lines), the accuracy remains relatively stable with increasing base model depth, and CALM provides modest improvements. This is because CIFAR10 exhibits less heterogeneity, resulting in fewer conflicts to mitigate. From Fig.5d, the FedPAC method with ResNet4 on the Flowers102 dataset (blue dashed line) exhibits a significant drop in accuracy compared to other base models, indicating a pronounced negative impact of client conflicts in this scenario. However, after integrating CALM (blue solid line), this downward trend is reversed, resulting in a substantial improvement in accuracy. This highlights CALM's ability to effectively mitigate aggregation conflicts.

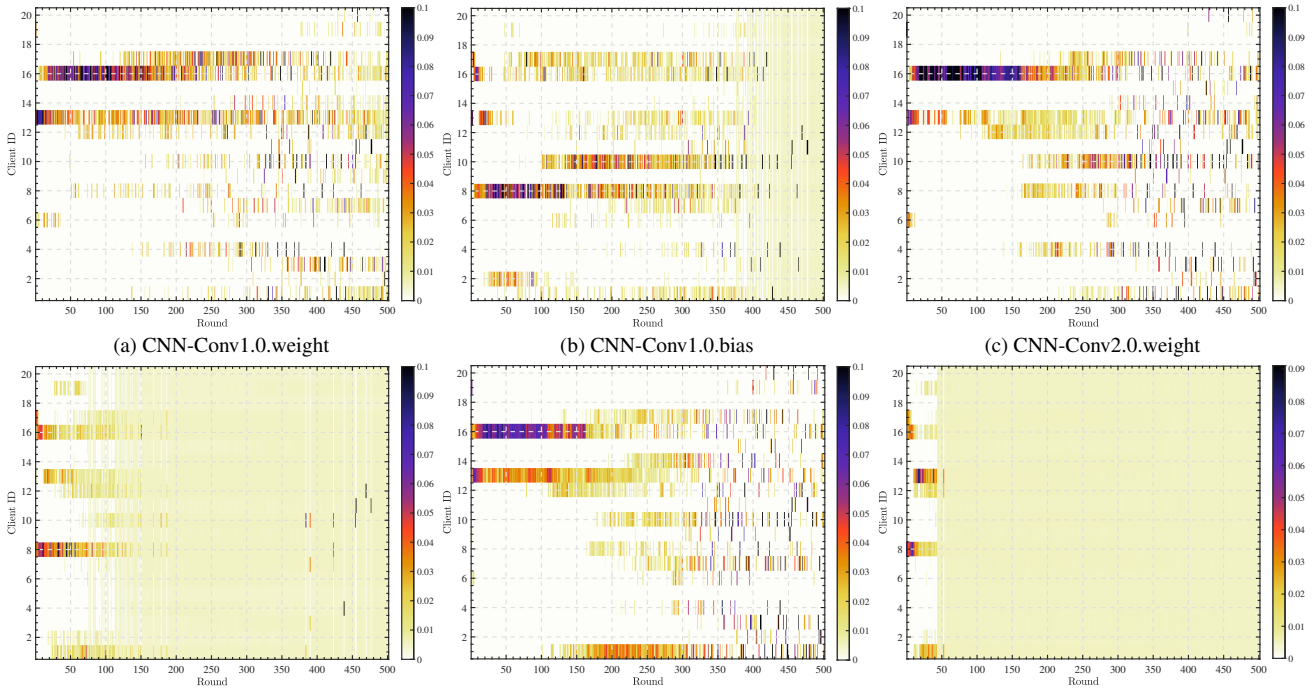
10. Visualization of Selective Aggregation

For a better understanding on the selective aggregation strategy of FedCALM, we visualized the aggregation weight dynamics of each client across different layers in detail. As shown in Fig.6 and 7, the visualizations illustrate the training process over 500 rounds on CIFAR10 for 20 clients, using CNN ($\epsilon = 0.1, C = 0.1$) and ResNet10 ($\epsilon = 0.1, C = 10$) as the base models, respectively. The CNN visualization includes all layers, while the ResNet10 visualization focuses on the first three and the last three layers.

From the subplots (a, c, e) in Fig.6, we observe that the aggregation weights in the weight layers exhibit significant sparsity during each training round: only a small number of clients have large weights (indicated by darker regions), while the weights of the remaining clients are zero. This behavior arises because the hyperparameters $\epsilon = 0.1$ and $C = 0.1$ are set to relatively small values, emphasizing the strict enforcement of the client update conflict constraint ($\epsilon\|g_i\| - d^\top g_i \leq \xi_i$). The clients with larger weights are those with the greatest conflict with the update direction d. FedCALM adjusts d to ensure that these conflicting clients directly contribute to modifying d, thereby alleviating the conflict.

In contrast, as shown in the subplots (b, d, f) of Fig.6, the bias layers demonstrate a different pattern. As training progresses, the aggregation weights for all clients gradually stabilize and tend toward a uniform distribution, represented by the consistent light yellow coloring. This phenomenon occurs because bias parameters typically have smaller magnitudes, leading to weaker conflicts under FedCALM's strict constraints. As training continues, the update direction d for the bias parameters tends to align globally, causing the contributions of all clients to d to become evenly distributed. In the bias layers, FedCALM focuses more on global collaboration among all clients and is less sensitive to update conflicts, resulting in a balanced aggregation weight distribution in the final state.

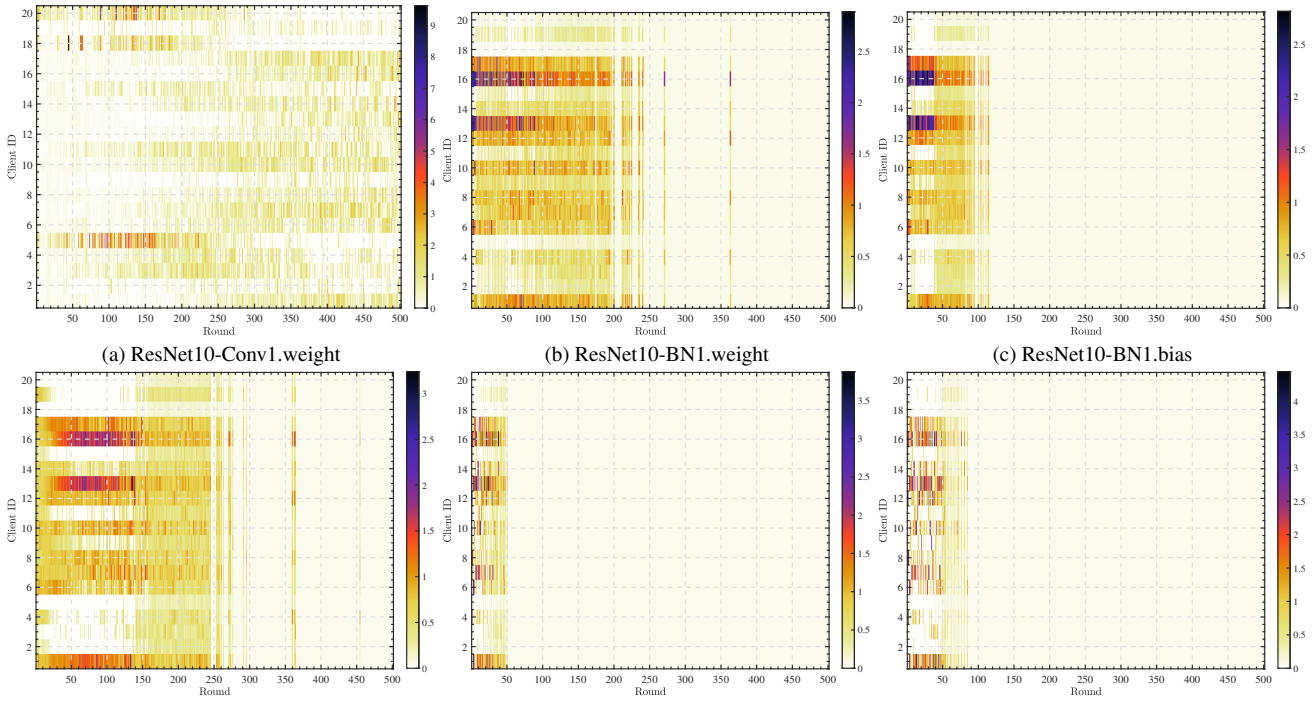
Compared to the CNN experiment results, Fig.7 uses ResNet10 as the base model with $\epsilon = 0.1$ and $C = 10$. Due to the larger C, which relaxes the constraint on the slack variable ξ_i, more clients participate in the aggregation process during each training round. This is evident from the broader range of darker regions in the heatmaps across all subplots. Although a higher C reduces the emphasis on resolving conflicts for specific clients, allowing some clients whose updates are not fully aligned with the global update direction d to participate, it also enables the global model to benefit from a broader range of clients. Similar to the CNN results, the weight layers still exhibit some sparsity. However, as training progresses, the weight distribution among different clients in all layers (except for the Conv1.weight layer) gradually stabilizes. The results from Fig.6 and 7 demonstrate that FedCALM's selective aggregation strategy is highly flexible, showcasing its ability to balance conflict resolution and global collaboration effectively.



(a) CNN-Conv1.0.weight (b) CNN-Conv1.0.bias (c) CNN-Conv2.0.weight

(d) CNN-Conv2.0.bias (e) CNN-FC1.0.weight (f) CNN-FC1.0.bias

Figure 6. Visualization of the proposed selective aggregation strategy in FedCALM, using a **CNN** as the base model on the CIFAR10 dataset. The six subplots correspond to different layers of the CNN (Conv1.weight, Conv1.bias, Conv2.weight, Conv2.bias, FC1.weight, FC1.bias), showing the aggregation weight dynamics across 20 clients over 500 rounds.



(a) ResNet10-Conv1.weight (b) ResNet10-BN1.weight (c) ResNet10-BN1.bias

(d) ResNet10-Layer3.downsample.0.weight (e) ResNet10-Layer3.downsample.1.weight (f) ResNet10-Layer3.downsample.1.bias

Figure 7. Visualization of the proposed selective aggregation strategy (FedCALM), using a **ResNet10** as the base model on the CIFAR10. The six subplots correspond to the first three layers and the last three layers of the ResNet10 (Conv1.weight, BN1.weight, BN1.bias, Layer3.DS0.weight, Layer3.DS1.weight, Layer3.DS1.bias), showing the aggregation weight dynamics across 20 clients over 500 rounds.

Method	CIF10	w/ CALM	CIF100	w/ CALM	FL102	w/ CALM
FedRep	87.01	87.35	49.17	50.03	55.91	56.35
FedALA	87.12	87.43	53.00	53.63	60.24	61.21
FedCP	87.00	87.30	50.51	51.93	57.08	57.62
FedPAC	86.95	87.28	52.13	52.73	62.00	62.87
FedCALM	**87.82**	↑ **0.32**	**54.29**	↑ **1.00**	**63.48**	↑ **0.71**

Table 10. Test accuracy of FedCALM and SOTA methods with ViT as base model on various datasets (w/ CALM indicates performance with CALM enhancement, ↑ indicates the average gain).

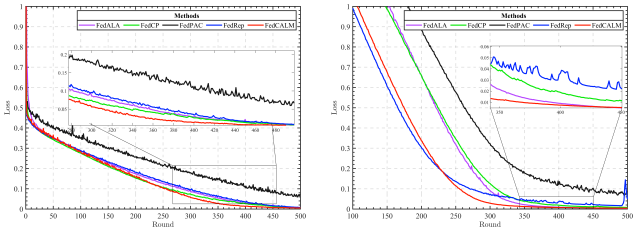


Figure 8. Average training loss of FedCALM and SOTA methods with ViT as base model (CIFAR-10 Left, CIFAR-100 Right).

11. Additional Experiments

FedCALM is not limited to CNN architectures. Due to resource constraints, we evaluate the performance of FedCALM using a non-pretrained ViT-Tiny-Patch16-224 as the base model. As shown in Tab.10, FedCALM consistently achieves the best performance, and all SOTA methods exhibit a significant improvement when combined with CALM. Additionally, we observe that using ViT as the base model yields performance comparable to CNNs but inferior to larger ResNet architectures. This observation aligns with the findings in the ViT paper, which explicitly states that training ViT from scratch on small datasets (e.g., ImageNet or smaller) results in lower performance compared to CNNs [51]. Nevertheless, our method still outperforms existing ViT-based PFL approaches [52].

We provide the average training loss curves (as shown in fig.8) for the newly added ViT base model experiments, illustrating the convergence behavior of FedCALM. As the results show, our method effectively mitigates conflicts during global aggregation, leading to faster and more stable convergence compared to other methods. Additionally, convergence curves for the original scenarios, which converge faster than ViT in our experiments.

We provide boxplots to better illustrate the distribution of client accuracy across different methods. As shown in Fig.9, with ResNet4, FedCALM achieves a median accuracy significantly higher than the upper quartile (75th percentile) of all other methods, demonstrating its superior overall performance. With ResNet10, FedCALM's minimum accuracy surpasses the median accuracy of other methods, indicating that even the worst-performing clients

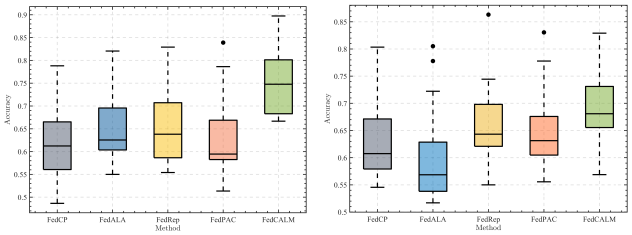


Figure 9. Client accuracy distribution of FedCALM and SOTA methods on Flowers102 dataset (ResNet4 Left, ResNet34 Right).

in FedCALM outperform at least 50% of clients trained with other approaches. Furthermore, the fewer extreme outliers (black dots) indicate that FedCALM achieves more stable performance.

[51] ICLR. 21. An image is worth 16x16 words: Transformers for image recognition at scale.

[52] CVPR. 23. Fedperfix: Towards partial model personalization of vision transformers in federated learning.