FedPACE: Federated Perturbed Annealing and Conflict Elimination

Abdul Samad [1] **Rumaan Mujtaba** [2] **Muhammad Hamza Habib** [3]

Abstract

Data Heterogeneity in Federated Learning (FL) frequently leads to client divergence where the conflict in gradient update among clients often hinders global model progression. We hypothesize that significant disagreement in gradient directions across clients represents learning of client specific spurious features in a non-IID setting, whereas consensus implies learning of invariant and stable features. To address this, we propose a unified three stage training pipeline. First, we employ a federated variant of GGA (Ballas & Diou, 2025) to bias early optimization towards regions with higher inter-client agreement. Second, we introduce a dampening mechanism, that dynamically scales the update magnitude based on the directional agreement of gradients among clients. Finally, we prune parameters showing persistent conflict during the later stages of the training. We test our approach on CIFAR-10 and PACS datasets and effectively demonstrate that prioritizing gradient agreement effectively filters out noise and minimizes the effects of data heterogeneity leading to better performance in non-IID FL settings.

1. Introduction

In a typical Federated Learning (FL) environment (McMahan et al., 2017), it is common for clients to have non-IID data where the data distribution is not identical across clients. In such settings, data heterogeneity and noisy gradients frequently cause client models to diverge in conflicting directions, negatively affecting global model performance.

We hypothesize that significant disagreement in the gradient direction of a specific parameter across clients implies that the parameter is learning client specific spurious features. Conversely, a parameter direction on which all clients agree implies the learning of some common features among clients. To test our hypothesis, we assign an agreement score between 0 and 1 to each parameter, which scales the update magnitude based on the agreement among clients regarding the training direction for that specific parameter. Furthermore, we prune parameters that continue to disagree towards the later stages of the training. This stage acts as a filter, preserving only the parameters that have learned stable features across clients.

Moreover, we selected FedGH (Yi et al., 2023) along with Gradient-Guided Annealing (Ballas & Diou, 2025) as our baseline, and built on top of GGA. GGA is a lightweight, early training intervention designed to increase inter-client gradient agreement by applying small randomized perturbations iteratively on parameters selecting the ones that maximize the cosine similarity across clients without substantially increasing the loss (controlled by a loss relaxation threshold). At selected annealing rounds, server samples K different perturbations, and evaluates their impacts on client gradients, and loss selecting the one which maximizes average cosine similarity among clients. This mechanism used during an early window of training, biases the optimization strategy towards region where client gradients align better and hence more comfortable for aggregation.

In the following work, we implement a federated variant of GGA as our baseline adapted on CIFAR-10 (Krizhevsky et al., 2009) dataset, and integrate GGA into a unified pipeline that follows annealing with our proposed agreement-weighted dampening update and agreement weighted pruning in the later stages of training.

The codebase for the experiments conducted can be found on GitHub.

2. Methodology

To conduct our experiments, we used CIFAR-10 and PACS dataset to evaluate our framework on both classification and domain generalization tasks. For these experiments, we used a 3-layer CNN having the following architecture.

$$\begin{aligned} Input(3) &\rightarrow Conv2d(32) \rightarrow BN \rightarrow MaxPool \\ &\rightarrow Conv2d(64) \rightarrow BN \rightarrow MaxPool \\ &\rightarrow Conv2d(128) \rightarrow BN \rightarrow MaxPool \\ &\rightarrow AdaptiveAvgPool \rightarrow Linear(10) \end{aligned}$$

Throughout these experiments, the random seed used was 0 unless explicitly stated in some where we used multiple random seeds in some experiments to check robustness.

2.1. Baseline Implementations

Inspired by Ballas & Diou (2025), we implement our own version of Federated Gradient-Guided Annealing (Fed-GGA) to use as our baseline. Our primary focus is on improving the global model's performance on image classification task. The datasets used is CIFAR-10 (60,000 images, and 10 classes) (Krizhevsky et al., 2009). A custom 3-Layer CNN model and Adam optimizer with learning rate set to 1e-3 and weight decay to 1e-4 is utilized in training. We use a 3-Layer CNN model instead of a pre-trained model to make computation feasible.

Training runs for a total of $R = 50$ rounds, where GGA is active between rounds $R_s = 2$ and $R_e = 15$. The range is consistent with the sensitivity analysis results of Ballas & Diou (2025) that gradient annealing during early training stages leads to increased model performance. We set the number of clients $N = 3$, and for each client, we perform normal FedAvg training (McMahan et al., 2017), where a reference gradient $\{g_i\}$ and loss $\mathcal{L}$ is computed. If the round r during training falls between R_s and R_e, we sample $K = 8$ perturbations from a uniform distribution with range $\rho = 1e-5,$. The choice for number of clients and perturbation samples K is to make computation feasible, while value of ρ is consistent with what's used in Ballas & Diou (2025).

The perturbation which leads to a parameter update that satisfies the loss relaxation $\delta = 0.05$ and $\beta = 0.3$ constraint and has the highest average cosine similarity across clients is selected to update the global parameter (see Eq. (1)). This GGA algorithm is presented in **Algorithm** 1. For domain generalization task, we perform the same experiment on the PACS dataset (9,991 images, 7 classes, and 4 domains) (Li et al., 2017). Training runs on a single seed and held out domain strategy is used during training to evaluate the OOD generalizability performance of the model.

$$(\text{sim}_k > \text{sim} + \beta) \wedge (\mathcal{L}_k - \mathcal{L} < \delta) \tag{1}$$

Moreover, the training is performed under two settings. In one case, it proceeds on a single seed with value 0, while in another case, it is repeated over three different seeds with values 0, 1, 2. This is to demonstrate the reproducibility and robustness of our implementation.

To further test our approach robustly, we also used Federated Gradient Harmonization, FedGH, (Yi et al., 2023) as our baseline. We compute a pairwise cosine similarity matrix between all client gradients, where each entry (i,j) represents the alignment between client i's and client j's gradient vectors. The harmonization weight for each client is computed as the sum of its similarities with all other clients, including the self-similarity term which equals 1.0. This produces weight for client i given as

$$W_i \propto \sum_j \cos(g_i, g_j) \tag{2}$$

where g_i represents the gradient for i^{th} client and g_j represents the gradient for j^{th} client. Then the global model's state is calculated by taking a weighted average of client gradients and normalizing it by harmonization weights. This approach differs from ours as we assign a parameter wise weight during the aggregation while FedGH assigns a client wise weight during aggregation.

2.2. FedPACE Implementation

The training in FedPACE happens in a 3-stage pipeline which progresses as follows:

$$\begin{aligned} &FedAvgRounds \rightarrow Annealing \rightarrow FedAvgRounds \\ &\quad \rightarrow Dampening \rightarrow Pruning \end{aligned}$$

FedAvg rounds before annealing and dampening ensure that the parameters are at a better optimal state where we can find some agreement among clients.

The baseline builds on top of the baseline Fed-GGA implementation. The only change is that we scale the perturbation vector by its norm to ensure that its relative magnitude remains consistent throughout training. This is consistent for both our classification and OOD generalization task.

Algorithm 1 Implementing GGA in Training

```
Require: Global model $\theta_0$, clients $\{C_i\}_{i=1}^N$ with data domains $\{D_i\}$, learning rate $\eta$, perturbation scale $\rho$, loss relaxation $\delta$, similarity relaxation $\beta$,total rounds $R$, annealing rounds $R_s$ to $R_e$, number of perturbation samples $K$
for $r = 1$ to $R$ do                                  ▷ training rounds
    Sample clients and perform local update to obtain gradients $\{g_i\}$
    if $R_s \leq r \leq R_e$ then                      ▷ Gradient-Guided Annealing phase
        Compute similarity: sim $= \min_{i \neq j} \left( \frac{g_i^\top g_j}{\|g_i\| \|g_j\|} \right)$
        $\theta' \leftarrow \theta_r$
        for $k = 1$ to $K$ do
            Sample perturbed parameters: $\theta_k \leftarrow \theta_r + \mathcal{U}(-\rho, \rho)$
            Get new gradients $\{g_i^k\}$ and similarity sim$_k$
            Compute loss $\mathcal{L}_k$ on aggregated client data
            if $(\text{sim}_k > \text{sim} + \beta) \wedge (\mathcal{L}_k - \mathcal{L} < \delta)$ then
                $\theta' \leftarrow \theta_k, \ \text{sim} \leftarrow \text{sim}_k$
            end if
        end for
    else
        $\theta' \leftarrow \theta_r$
    end if
    Update global model: $\theta_{r+1} \leftarrow \theta' - \eta \cdot \sum_i g_i$
end for
```

2.2.1. Sign Disagreement Dampening:

To minimize the effects of client drift introduced by non-IID data across clients, we introduce a per parameter regularization mechanism during the aggregation phase. This regularizer applies a per parameter penalty based on the level of disagreements in gradient directions among clients. The training loop follows a standard FL cycle where the server first sends the current global model θ_{old} to all N clients. Each client i performs a backward pass on the local model θ_i to compute the gradient vector g_i. These local gradients are then transmitted back to server for the aggregation.

After receiving the updated parameters from clients, the server then calculates new global weights in a series of steps. First the server calculates the arithmetic means of the gradients across all clients. The server then calculates new weights in a series of steps, representing the baseline update direction used in algorithms like FedSGD (McMahan et al., 2017).

$$g_{avg} = \frac{1}{N} \sum_{i=1}^{N} g_i \tag{3}$$

Next, to quantify the agreement among clients regarding the optimization trajectory, the server calculates an element wise agreement score W_j for each parameter index j. The score relies on the sign of gradients to focus purely on directional alignment.

$$W_j = \frac{\left| \sum_{i=1}^{N} \text{sign}\left(g_i\right)_j \right|}{N} \tag{4}$$

where N is the total number of clients, $(g_i)_j$ is the scaler gradient for parameter j of client i, and sign$(\cdot)$ is the sign function. The resulting score W_j is a scaler value in range $[0, 1]$. A value of 1 represents maximum directional agreement among clients and a value of 0 represents maximum conflict among clients (an even split between positive and negative gradient update).

We then use this agreement score as a dynamic, per-parameter scaling factor. The server then uses this factor as follows:

$$\theta_{new} = \theta_{old} - \eta \cdot (g_{avg} \odot W_j) \tag{5}$$

where η is the global learning rate. In this calculation, W_j acts as a dampener. Parameters showing higher agreement across clients are updated with a magnitude close to average gradient, while parameters with high disagreement are suppressed. This score ensures that parameters with high agreement among clients are updated faster while keeping the parameters with low agreement from diverging too far.

For this specific implementation, we started dampening at the 20^{th} round of the training with a learning rate (η) of 0.01 to make up for the slower learning rate caused by penalties which leads to slower training. This ensures that the global model learns features that are stable across clients while preventing divergence into some client-specific minima.

For this specific implementation, we perform annealing during round R_e 2 to 15, which are then followed by 5 warmup rounds for the model to learn general features. We then activate the dampening mechanism at round 20 after warmup rounds. Moreover, since the agreement score for parameters among clients is always < 1, it decreases the step size in each training round. To compensate for this, we increase the learning rate to 0.01 during dampening phase.

2.2.2. Sign Disagreement Pruning:

This stage acts as the last filter in training. Here, we prune parameters that have an agreement score (as calculated above in Equation 4) below a certain threshold t_p for a specific number of epochs e_p. By pruning weights whose gradients conflict significantly across clients, the final model retains only the parameters that learn stable features across all the clients. In this specific implementation, we enable pruning from round 42 to 50 with an agreement threshold $t_p = 0.2$, and $e_p = 1$.

$$\theta_j^{new} = \begin{cases} 0 & \text{if } W_j < t_p \\ \theta_j^{new} \cdot W_j & \text{otherwise} \end{cases} \tag{6}$$

2.3. Experiments Conducted

To assess the individual and combined contributions of the components - annealing, dampening, and pruning - of FedPACE on model performance and convergence, we perform an ablation study, comparing the entire pipeline against its constituent combinations. Specifically, dampening with pruning, and annealing with dampening.

3. Results

We first tested our proposed framework on a CIFAR-10 dataset simulating a highly heterogeneous environment. The data was distributed among three clients using a Dirichlet distribution with concentration parameter $\alpha = 0.1$. To ensure the fairness, statistical significance and reproducibility of the experiments conducted, we executed the same training loop over three distinct random seeds $\in [0, 1, 2]$. FedPACE outperformed FedGGA across all 3 runs. FedGGA achieved an average accuracy of 36.13% in 50 communication rounds across all 3 seeds while FedPACE outperformed FedGGA by a margin of 2.5% achieving an accuracy of 38.63% as shown in Table 1. This consistent gain validates that the orthogonal implementation of FedPACE on top of FedGGA architecture effectively minimizes the adverse effects of client drift in highly non-IID settings.

Method	**Dataset**	**Final Global Accuracy (%)**
FedGGA	CIFAR-10	36.13
FedPACE	CIFAR-10	38.63

Table 1. Performance comparison over 50 rounds. CIFAR-10 accuracy is reported as a 3-seed average.

Method	**Dataset**	**Final Global Accuracy (%)**
FedGGA	PACS	16.93
FedPACE	PACS	20.82

Table 2. Performance comparison over 50 rounds. PACS accuracy is reported as average on Leave-One-Domain-Ouf protocol for all 4 domain.

Moreover, to test model's performance on Domain Generalization (DG) tasks, we extended our experiments to PACS benchmark dataset (Li et al., 2017). In this setting, FedPACE demonstrated significant robustness by outperforming FedGGA by a significant margin of 3.89% on PACS dataset. The results reported in Table **2** are derived from averaged Leave-One-Domain-Out experiments, a rigorous protocol where the model is trained on three source domains and evaluated on a strictly unseen target domain. Although the baseline struggles to generalize on PACS effectively, as depicted in Figure **1**, FedPACE leverages the conflict aware dampening to escape the local minima, leading to much higher accuracy and better convergence over the course of 50 rounds.

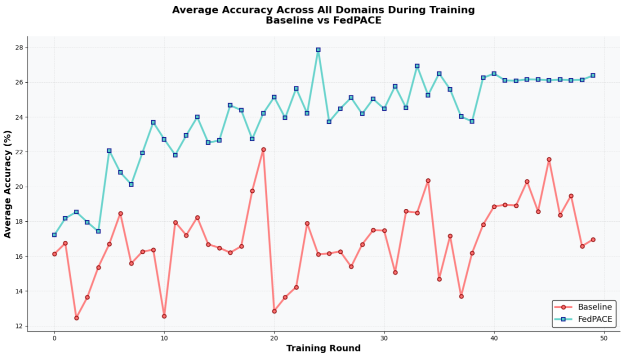

Figure 1. Training accuracy over 50 communication rounds

Throughout the training loop, we also performed analysis of pairwise similarity among clients to understand the internal dynamics of the optimization. Critically, we observed a distinct and perfectly inverse relationship between pairwise client similarity and the average global accuracy, as demonstrated in Figures **2** and **3**. This offers a compelling insight into how our training loop works, as FedPACE actively dampens conflicting gradients, the cosine similarity between client updates naturally decreases. However, this reduction in similarity correlates with an increase in global accuracy, suggesting that by reducing the interference between divergent client objectives, the global model is able to aggregate more diverse and meaningful features.

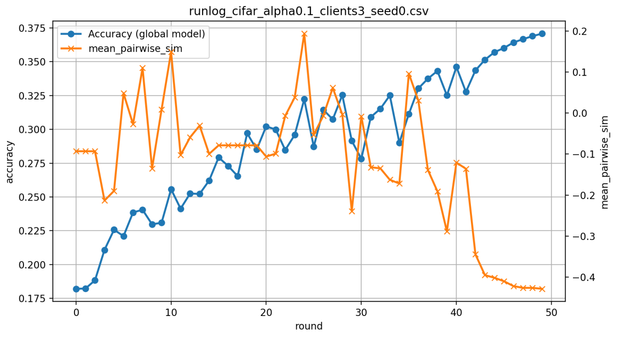

Figure 2. Global Accuracy vs. Pairwise Cosine Similarity for Seed 0 during FedPACE

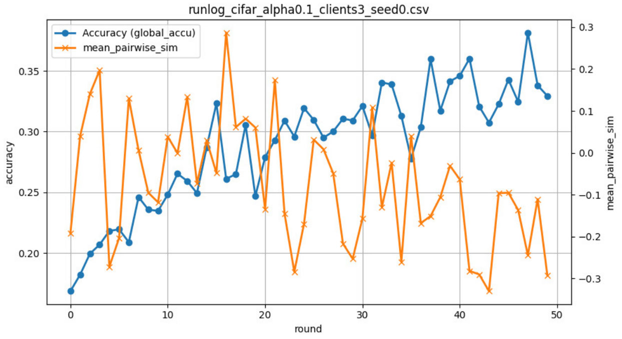

Figure 3. Global Accuracy vs. Pairwise Cosine Similarity for Seed 0 during baseline FedGGA run

To assess the impact of different hyperparameters on global model accuracy and mean similarity, we experiment on them with different values and one seed with value 0. We experiment with $K \in \{3, 8, 16, 32\}$, and $\beta \in \{0.1, 0.3, 1.0\}$ separately, while keeping all other hyperparameters as described in subsection **2.1**. The results are shown in Figure **6**.

Method	Final Global Accuracy (%)	Mean Gradient Similarity	AUC
FedGGA	31.94	-0.073	0.287
FedPACE	37.08	-0.121	0.292

Table 3. Performance metrics on CIFAR-10 dataset and seed 0

For our ablation study, described in subsection **2.3**, the global model accuracy and mean similarity during training for three combinations of components is shown in Figure **4**.

4. Discussion

Across the datasets CIFAR-10 and PACS, our full pipeline (FedPACE) demonstrates higher global accuracy, hence better performance and OOD generalization. Although annealing tries to enforce cross-client alignment during early stages of training, it only controls the relative importance of alignment. So when clients vary in feature or data distribution, the uncontrolled large local updates can suppress global consensus. Dampening complements this by helping to normalize update influence, which prevents dominant or noisy clients from destabilizing convergence after annealing. Finally, during later stages of training, pruning allows us to identify consistently unhelpful parameters to eliminate and enhance generalization. This is demonstrated by a higher final global accuracy of FedPACE in Figure **2** when compared to Figure **3**.

Interpreting Table **3**, it should be noted that pairwise cosine similarity is not a direct measure of client agreement. Although negative values indicate that clients tend to push gradients in opposite directions on average, it doesn't necessarily mean that learning is hindered. Our results in Table **3** is proof of this. FedPACE having a more negative mean gradient similarity (-0.121 vs. -0.073) suggests that it is better at amplifying useful aligned components and suppressing harmful directions such that the net effect is positive, even if the whole-parameter similarity becomes worse. Hence, similarity is more informative and less predictive of overall accuracy.

Looking at the training curves in Figure 2 and 3, we can see that pairwise similarity and accuracy has almost perfectly inverse relationship where high pairwise similarity among clients means low accuracy and vice verse. During the early stages of training, annealing tries to maximize the inter-client similarity and as a result we can see a high pairwise similarity among clients. As the training progresses, we can see a trend in accuracy and similarity where accuracy increases progressively with decrease in mean similarity across clients. This phenomenon captures the fundamental tension between local specialization and global generalization in highly heterogeneous federated learning settings. Our experimental setup uses Dirichlet with $\alpha = 0.1$ where each client has only two to three classes out of 10 classes. Furthermore, this trend is less prominent in a DG setting where all clients have the same set of classes and clients can learn common causal featuers of the classes. This leads to clients agreeing to a local minima while performing robustly on their own local dataset distributions. This phenomenon can be observed from Figure 5 where client models do manage to maintain a certain degree of agreebility while improving on accuracy.

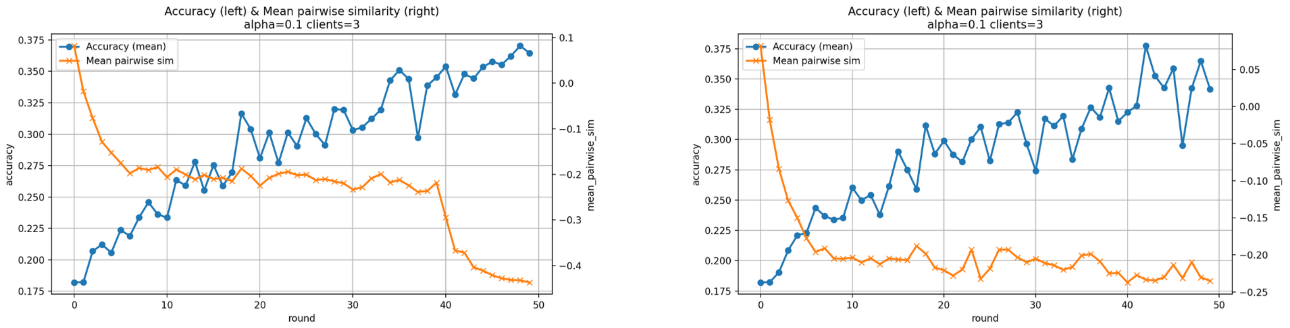

Figure 4. The accuracy and mean similarity during training for our ablation study. The plot on the right is a combination of annealing and dampening, while the left one is dampening and pruning

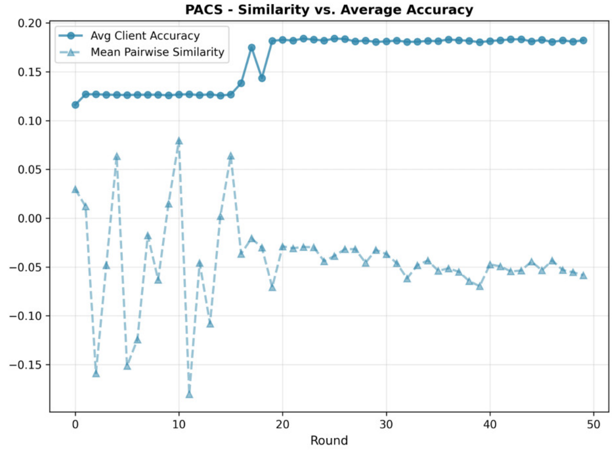

Figure 5. Accuracy vs. Similarity graph for a training run on PACS with art painting as a held out domain

This reveals that agreement weighted score does not act as a similarity regularizer but rather as a selective amplifier. Notably, it operates after clients compute their local gradients on their own datasets. Only during server aggregation, dampening distinguishes between parameters and scales them based on agreement. The above plots show that that the productive client specialization requires divergence. Agreement weighted aggregation enables this safe divergence by selectively applying consensus updates while dampening high conflicts.

We hypothesize that for $\alpha = 0.1$, clients has to specialize on their local data and cater to specific classes they have in their local datasets. This requires clients to diverge from the global model and specialize on their own training data instead of converging to a unified point. To test our hypothesis, we tested FedGH (Yi et al., 2023)on the same dataset with Dirichlet distribution having $\alpha = 0.1$. As shown in Figure 7, it can be observed that FedGH collapses when $\alpha = 0.1$ on this distribution. This is because of the above mentioned reason that the model cannot find a common minima between all the clients and performs worse if we try to align all clients to a common minima. In a setting where each client has only 2 or 3 classes out of 10 total classes, it's hard for clients to agree on some common features when they all have different classes. The opposite of this phenomenon, where clients have same set of classes and they can find a common minima, can also be observed from Figure 5 where clients do manage to find some agreement while slowly improving their accuracy. While FedGH might perform robustly on a DG setting, it fails to adapt to this specific scenario where clients have different classes distributed among them and there is no common minima for them to agree on. Our proposed method FedPACE deals with this issue in a better way by only selectively suppressing weights while amplifying parameters with agreeing gradient.

Dampening helps stabilize server updates by controlling parameter updates when sing-consensus is low. This is evident from Figure **2** having a more stable accuracy updates during training than Figure **3**. Also, Table **3** shows that FedPACE has a higher AUC score (+0.005) or better convergence which is reflective of its ability to result in parameter updates that improve convergence.

Although using pruning as a last-resort refinement improves global model accuracy, but the sinking mean similarity, shown in Figure **2**, needs to be addressed. This can be attributed to the hypothesis that pruning started while some disagreement persisted, temporarily lowering similarity, or the fact that pruning selectively reduces magnitudes, which can alter cosine statistics. This demonstrates another limitation of our implementation: computation infeasibility. The total number of rounds weren't enough for annealing and dampening to create a scenario where the sufficient benefit of pruning could be observed.

The proposed FedPACE also demonstrates higher fi-

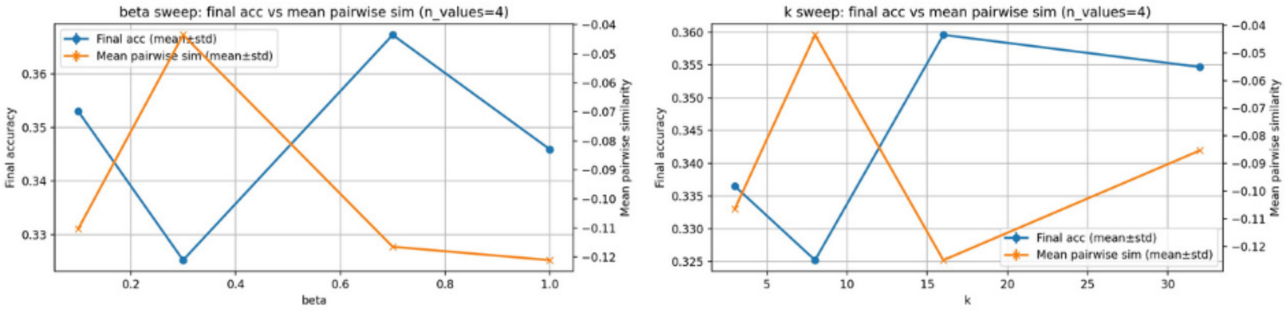

Figure 6. K, B and Warmup sweep ablation studies. Results of varying these two parameters while keeping rest of the hyper-parameters constant.

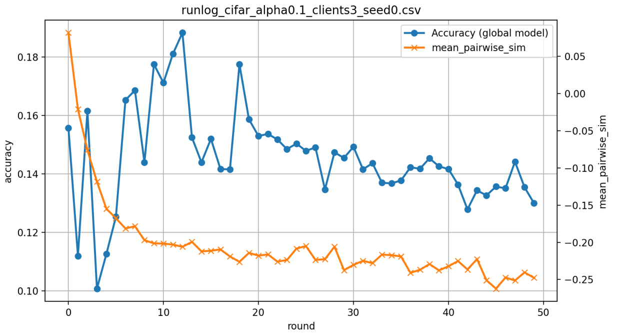

Figure 7. Federated Gradient Harmonization's (FedGH) performance on Dirichlet distribution of CIFAR-10 with $\alpha = 0.1$

.

nal test accuracy and more stable training than baseline FedGGA, for seeds ∈ [1,2,3]. On CIFAR-10, it has an accuracy gain (+2.5%), shown in Table 1, demonstrating the robustness of our proposed implementation. Moreover, on PACS dataset, with seed value 0 only, Figure 1 shows FedPACE having a smoother and less volatile training curves: smaller accuracy fluctuations per-round compared to FedGGA. The final test accuracy improvement (+3.89%) is visible from Table 2, and is a sufficient metric to conclude better generalizability of our implementation. These findings show how dampening, and late pruning combined with annealing, even on a low-setting, speed up beneficial advancement and lessen detrimental inter-client interference.

For Figure 6, we wanted to understand the impact of some important hyperparameters on the performance of our implementation. The trends observed in all three hyperparameter experiments is counterintuitive but valid. Gradient similarity decreases as test accuracy increases, because in non-IID scenarios, clients have to diverge on their particular data distributions instead of maintaining a surface-level consensus. The plot of β, which represents the strength of search radius in parameter space, is consistent with this rationale. Because of aggressive search, higher β values allow the model to escape local minima which improves accuracy, but decreases similarity. This is because the global model trajectory deviates from what any one client would prefer in isolation.

For our ablation study, it is clear from Figure 4 that both of these combinations global model accuracy does not exceed our FedPACE's implementation, proving our implementation more effective. Secondly, it should be notes that although overall mean similarity is positive, it does not increase and even decreases during training, whereas global model accuracy increases. The reason is that on CIFAR-10, each client is getting 3-4 classes, and because they have separate classes, they cannot agree on the same thing. This rules out the possibility of a common minima, and in an attempt by clients to cater to their own local classes, they diverge from each other. This is also validated by our FedGH implementation even, shown in Figure 7, as it's mean similarity also decreases along training. Thus, this validates the rationale we presented behind why mean similarity decreases while global model accuracy increases during training on CIFAR-10.

The plot of K, which represents the number of perturbations we sample, is able to find better perturbations aligning with multiple client's objectives with a higher value. This reduces similarity because optimal updates for heterogeneous clients necessitates divergence rather than simple averaging. Another important aspect of our implementation is the warm-up rounds we have between when annealing ends and dampening starts for stability. A plausible reason of this is that near the end of annealing phase, the model might still be adjusting to a change in optimization dynamics. If dampening is applied immediately, the aggregation process will undergo two sudden changes in a row.

4.1. Limitations and Future Prospects

While this specific implementation focuses mainly on CIFAR-10, the robustness on PACS proves that FedPACE can be applied to the problem of domain generalization as well. The motivation behind its application would be that FedPACE would encourage the learning of domain invariant stable and causal features across domains. One can easily extrapolate that if all clients are agreeing over some gradient direction, it means that the spatial direction in vector space encodes some common feature across domains. By following this line of reasoning, one can apply FedPACE to a domain generalization as well.

Since FedPACE uses gradients and clients' agreement on gradients to determine the optimization trajectory, a possible flaw can be if majority of the clients agree on a spurious direction which has no causal relationship with the input data. In that case, we will be discarding the minority clients which might be pointing in the right direction.

Another major downside to FedPACE is the communication overhead, FedPACE requires the transmission of gradients in each training round for determining the gradient agreement score. A concept of pseudo gradients can be introduced, by taking the difference between the initial weights of the model when a clients starts local training and the new weights after clients finishes local training round after k local epochs, communication overhead can be reduced by increasing the number of local epochs and then sending those pseudo gradients to the server.

5. Conclusion

We introduced FedPACE in this project, a three stage federated learning pipeline aimed to improve performance in non-IID settings and achieve a better global model accuracy. The three steps, which included Gradient Guided Annealing, Agreement Weighted Dampening, and pruning based on sign similarity ensure that the model learns invariant features across clients minimizing the adverse effects of noisy gradients and client conflict.

The results we achieved with CIFAR-10 and PACS dataset validate that our approach was effective, as FedPACE outperformed FedGGA on CIFAR-10 while also demonstrating robust performance on PACS dataset. Our results proved that ensuring client consensus and agreement in gradient updates resulted in a better model with filtered out conflicting parameters making it more effective for real-world problems with largely unseen-data.

References

Ballas, A. and Diou, C. Gradient-guided annealing for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2502.20162.

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

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1181–1190, 2017.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics (AISTATS)*, pp. 1273–1282. PMLR, 2017.

Yi, L., Wang, G., Liu, X., Shi, Z., and Yu, H. FedGH: Heterogeneous federated learning with generalized global header. *arXiv preprint arXiv:2303.13137*, 2023. URL https://arxiv.org/abs/2303.13137.