Minimax Theory and its Applications
Volume 11 (2026), No. 4s, 859 to 868

Robust Optimization for Federated Learning Under Client Drift

Dr. Phani Sridhar Addepalli[1], Dr. S. Swarnalatha[2], Shalini E[3], S. Monika[4], Ganesa Murthy A[5], Dr. M. Kavitha[6]

[1]Department of Computer Science and Engineering, Aditya University, Surampalem, Andhra Pradesh – 533437. Email: phani.addepalli@adityauniversity.in, ORCID ID: 0000-0002-6021-3433
[2]Associate Professor, Department of Hindi, Sri Ramakrishna College of Arts & Science, Email: swarnalathar023@gmail.com, ORCID ID: 0009-0005-3061-8623
[3]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Email: shalini@maher.ac.in, ORCID ID: 0009-0002-1391-0057
[4]Department of Artificial Intelligence and Data Science, New Prince Shri Bhavani College of Engineering, Email: smonicse@gmail.com, ORCID ID: 0009-0003-7448-7085
[5]Librarian, Library and Information Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India – 600117, Email: aganesamoorthy@gmail.com, ORCID ID: 0000-0002-6433-7134"
[6]Associate Professor, Government College of Engineering-Bargur, Email: m.kavitha@gcebargur.ac.in, orcid id: 0000-0002-4307-4929

Abstract
Federated learning (FL) enables collaborative model training without requiring clients to share their raw data with a central server. However, the statistical heterogeneity that characterizes many federated environments can cause local optimization trajectories to diverge from the direction of the global objective. Client drift has always been a major concern in federated learning, and it becomes more so when the client performs several local optimization iterations before communicating. The popular approaches to counter this, namely FedProx and SCAFFOLD, rely on proximal regularization and control-variate-driven drift correction, respectively. In the literature, it is believed that most of the existing methods implicitly do not make a distinction between persistent client disagreement and temporary deviations.Persistent Client-Drift Aware Proximal Federated Learning (PCD-Prox) is a computationally efficient, adaptive regularization technique that takes into account the time-invariant nature of client drift. Rather than a single static proximal coefficient for all clients, PCD-Prox retains an exponential average of each client's direction of deviation from the global update, and uses this to scale the strength of their local proximal regularization.The method preserves the typical fed-averaging scheme, with the addition of one scalar state variable per user. In order to verify the method, is to perform a small numerical validation with 50 heterogeneous users, partial participation, long-term and short-term user shifts, and 20 seeds.There are four methods evaluated in this test: Fed Avg, Fed Prox, a proximal version depending on the instantaneous drift and PCD-Prox. The results indicate that PCD-Prox exhibits the smallest final value for the objective for both persistent and transient drifting. It performs better than Fed Avg by almost 0.24% for the persistent drifting and by 0.08% in the case of transient drifting.The sensitivity analysis provides additional evidence of the importance of the parameter for temporal persistence on the accumulative drift estimate as well as the final optimization performance. These findings suggest that temporal information in the clients' update pattern can be used to guide adaptive local regularization without changing the existing aggregation framework. Further experiments with real non-IID data are necessary to establish the approach more generally.

*Keywords:*Federated Learning; Client Drift; Non-IID Data; Fedprox; Adaptive Regularization; Heterogeneous Optimization; Robust Federated Learning

1. Introduction

Federated learning is now considered a strong paradigm for training machine learning models with distributed data sets where the training data stays on the device. In the standard setup, the clients are trained with a common model provided by a coordinating server; then they train on their local data and update the model, which is collected and averaged.The original FedAvg framework showed that communication efficiency is a large benefit of this approach over classical training in the centralized setting, while at the same time allowing learning from unlabeled distributed data (McMahan et al., 2017).The value of federated learning in practice is directly related to one of the most difficult optimization issues it creates. Client data sets are infrequently distributed identically. Variations in

user behaviour, geographic region, institutional demographics, device capabilities, or acquisition situations can lead to substantial divergence in local data distributions.Thus, the local goal when an individual client works on it can be quite different from the ultimate goal of the full federation. Thus, a client can improve what goes forward for them, while also breaking away from the sequence that would be good for the overall model (Li et al., 2020; Wang et al., 2020; Ye et al., 2023).This effect is sometimes called client drift. Client drift due to statistical heterogeneity has been shown to be a major drawback of vanilla federated averaging in the analysis of federated optimization (Optimization Strategies for Client Drift in Federated Learning, 2022).

If the data distributions are heterogeneous, repeated local optimization may always push each model towards individual optima. As a result, the aggregate updates may be away from the optimization path of the global objective (Karimireddy et al., 2020; Li et al., 2020). This problem may be exacerbated by clients performing several steps of local optimization before doing the communication. This causes local updates to accumulate, which can further exacerbate the dissonance between client optimization direction and the global model (Wang et al., 2020; Ye et al., 2023).Different approaches have been designed to compensate for the impact of heterogeneous local optimization. Fed Prox employs a proximal term in the local objective to constrain the deviation from the current global model (Li et al., 2020). SCAFFOLD employs the control variate to counteract the client drift problem by estimating the differences between the local and the global update directions (Karimireddy et al., 2020).FedNova tackles the problem of client discrepancy that can occur due to different amounts of local computation and normalize client updates (Wang et al., 2020). These two schemes are two different ways of handling the tradeoff between local optimization and learning of the global model. They form a solid foundation for further work on heterogeneous federated optimization (Optimization Strategies for Client Drift in Federated Learning, 2022; Ye et al., 2023).However, it is possible that a client's disagreement has a time-related aspect that cannot be described within one communication round. The client that disagrees with the global update in one round is not guaranteed to be the same as the one that always produces conflicting updates for many rounds.

The disagreement that is observed on at least one occasion could be due to either a stochastic effect or a temporary change in the distribution of data on the client. On the other hand, if disagreement occurs repeatedly, it could be a sign of a longer-lasting source of heterogeneity on the client. Considering these two situations in the same way might thus give us limited insight into whether it should hold stronger local regularizers in subsequent rounds of communication.Temporal changes in distributed data and optimization behaviors have been similarly studied under the scope of federated concept drift. Jothimurugesan et al. (2023) proposed studying federated learning under the scenario of distributed concept drift, where clients' data-generating processes evolve over time. They showed that these temporal changes in distribution are associated with the diversity of client behaviours but represent a different problem than the directional disagreement at the optimization level. This is an interesting distinction, as a client may always send an update whose direction is different without necessarily requiring the explicit detection of a change in the distribution that generates its data. Several existing studies about client-drift mitigation also cite the importance of differentiating between different sources and types of heterogeneity for the design of optimization strategies (Optimization Strategies for Client Drift in Federated Learning, 2022).The authors highlight the temporal aspect as the underlying idea of what they coin the 'Persistent Client-Drift Aware Proximal Federated Learning' (PCD-Prox): a form of proximal federated optimization with a regularizer on the local training limited by the temporal nature of the persistent directional client drift.This basic idea remains consistent with classic federated optimization. A client that consistently moves toward the global optimum will be allowed to use diminishing amounts of further regularization, whereas a client that continually moves in the opposite direction could be required to use greater amounts of proximal regularization.

As a result, the local optimization problem can now be augmented by time-series data in addition to the known weighting scheme of the local optimizer used in federated averaging, as proposed in McMahan et al. (2017). The history maintained for the client is written by a scalar drift state. In addition, the clients no longer need to store all past model updates.The above expression yields a more mechanistic way of assessing whether the action-length of memory, as shown by the temporal decay of update behaviour, may be acting as an adaptive proximal regularizer.In particular, persistent disagreement plus instantaneous disagreement allows us to separate the effects of temporal memory from those of instantaneous client drift. This distinction turns out to be particularly important in the setting of heterogeneous federated optimization, where the local constraint factor is affected by both the magnitude of the disagreement as well as how long it persisted across multiple rounds of communication.

2. Related Work

2.1 Federated optimization under heterogeneous data

FedAvg is still the core benchmark algorithm of federated learning, where clients locally optimize and the server averages models (weighted by the number of clients' data points) (McMahan et al., 2017). On the other hand,

multiple rounds of local optimization may intensify the divergence in case of non-IID clients' data. Such issues have received much attention in the research on heterogeneous federated optimization and client-drift mitigation.Specifically, Non-IID data can be caused by feature distribution shift, label distribution shift, sample size discrepancy and client distribution discrepancy, which results in unaligned local gradients (Li et al., 2022; Ma et al., 2022). Ye et al. (2023) also review statistical and system heterogeneity as two main challenges of federated learning. Adaptive optimization and personalization have thus become hot research topics.This is important to differentiate this work. PCD-Prox can be classified as a local-optimization method. Instead of changing the client aggregation process, PCD-Prox alters the influence of the constraints when training locally.

2.2 FedProx and proximal regularization

FedProx adds a proximal penalty term to FedAvg to control excessive local updates from the current global model (Li et al., 2020). FedProx allows more control over local model divergence, though most frameworks use the same fixed proximal coefficient.A form of dynamic regularization has been investigated to automatically adapt optimization to a changing client state (Acar et al., 2021). In PCD-Prox, the proximal form is used as before, but with an adaptive coefficient, depending on the current drift measurement and its history.

2.3 SCAFFOLD and drift correction

SCAFFOLD deals with client drift through control variates that are used to estimate local and global update directions (Karimireddy et al., 2020). Therefore, SCAFFOLD seeks to correct the local update direction, while PCD-Prox seeks to control the magnitude of local update directions when their directional disagreement persists.

Other works on non-IID federated learning suggest that heterogeneity in the data distribution may cause systematic optimization problems (Li et al., 2022; Ma et al., 2022). Jothimurugesan et al. (2023) apply this to the setting of distributed concept drift, where client distributions change over time.

2.4 Normalization and other optimization strategies

FedNova handles heterogeneous local computation by averaging client updates with respect to the local optimization steps (Wang et al., 2020). Additional modifications to the optimization procedure or to the local representations include adaptive federated optimization (Reddi et al., 2021), local normalization for non-IID features (Li et al., 2021), and personalized federated learning (Li et al., 2021; Long et al., 2023).

These techniques give rise to heterogeneity through data, computation, participation, or temporal behaviour. The method proposed directly targets and manipulates the data distribution so as to generate persistent directional disagreement in client updates without any modifications to the aggregation rule or server optimizer.

2.5 Temporal and concept drift

In the distributed concept-drift case, the distribution on each client might drift at varying times and levels. Jothimurugesan et al. (2023) discuss the difficulties of federated studying within the distributed concept-drift scenario. Furthermore, additional recent works proceed to check adaptive and robust methods for heterogeneous federated learning (Zheng et al., 2024; Liang et al., 2024).These experiments inspire a desire to do persistence-aware optimization. Don't treat client disagreement that is just fleeting differently from persistent divergence. PCD-Prox does not directly detect concept drift, but rather uses the path of optimization as a proxy for persistent disagreement.

3. Proposed Method

3.1 Motivation

PCD-Prox is based on the idea that the passage of time between the two rounds of disagreement yields information that a single round of drift measurement cannot provide. In one direction, a deviation may be due to a stochastic local optimization, sampled data, or a momentary distribution shift, and thus not necessarily indicative of ongoing disagreement.On the other hand, if disagreement remains over multiple rounds of communication, it is a stronger signal that a client's optimization trajectory is fundamentally inconsistent with the global trajectory. Persistent statistical heterogeneity, or mismatched local and global objectives, can drive this type of client behaviour. PCD-Prox therefore records directional disagreement per client and uses it to adaptively modify proximal regularization.This method is especially suitable for proximal optimization since the proximal coefficient controls the constraint placed on local deviation from the current global model. Therefore, a moving drift can affect the local regularization term without changing the server aggregation.

3.2 Drift-Aware Proximal Regularization

During a communication round $t$the server delivers the global model w^t to client k. After local optimization, they evaluate the direction of the client update against the aggregate update. Cosine similarity is utilized to obtain the degree of alignment, from which they derive the drift score d_k^t.

Instead of using the instantaneous drift directly, PCD-Prox maintains an exponentially smoothed drift state:

$$s_k^t = \rho s_k^{t-1} + (1-\rho) d_k^t$$

where s_k^trepresents the accumulated drift state, and ρ governs the influence exerted by previous observations. A greater degree of persistence implies that the system retains past values for a longer period; conversely, a lower degree of persistence places greater emphasis on recent observations.

The adaptive proximal coefficient is given by the following expression:

$$\mu_k^t = \mu_0 (1 + \lambda s_k^t)$$

where $\mu_0$represents the base proximal coefficient, and λ controls the sensitivity to drift. The more accumulated drift occurs, the more proximal regularization is imposed; on the contrary, it has the least impact on the proximal coefficient if there is minimal disagreement. The local objective is then expressed as

$$\mathscr{L}_k(w) = F_k(w) + \frac{\mu_k^t}{2} \parallel w - w^t \parallel^2$$

where $F_k(w)$ denotes the objective specific to the client. The proximal term serves to prevent overly large deviations from the current global model.

The formulation retains FedProx's essential structure but replaces its constant proximal coefficient with a time-varying coefficient that depends on the client and the time-step, thereby requiring, for each communication round, downloading the global model, performing local optimization, measuring the directional disagreement, updating the drift state, setting the proximal coefficient, and uploading the client update for standard aggregation.This merely storesa scalar state for each customer as opposed to the history of earlier updates.

3.3 Interpretation of the Adaptive Mechanism

PCD-Prox establishes a feedback loop by translating the current directional disagreement into an estimate of persistence with exponential smoothing, which in turn influences the proximal strength by local regularization. A client that is balanced in accordance with the global trend will keep a low drift state and will be regularized more tightly to the base level. Clients that have repeatedly diverged will build up a larger state and will be regularized with more force.Thus, the mechanism emphasizes the long term disagreement more than momentary deviation. The persistence parameter determines how much weight to give the historical data as opposed to the client behaviour. The client-specific state enables different clients to have different proximal strengths.

3.4 Server Aggregation and Communication Structure

PCD-Prox uses the standard weighted aggregation rule. The received local models (or updates) are aggregated by employing the standard federated averaging rule, which suggests that the aggregation rule on the server does not need to be different.The supplemental state is also only the continuous drift estimate for the specific client in question. The entire historical update trajectory does not need to be transmitted. For occasional clients the stored state can be kept during several rounds of participation.For comparison, the proposed adaptation generally bounds to local optimization and adds temporal information without changing the existing federated communication scheme.

4. Experimental Design

4.1 Controlled Optimization Setting

Experimental setting based on a controlled convex optimization is utilized to assess the technique, in which each consumer has a local quadratic goal, tailored to the individual consumer. This enables explicit control over

heterogeneity of sources and levels of heterogeneity.Federation of 50 clients with a 10-dimensional model parameter. 120 communication rounds, with 50% of the clients participating per round, and 5 local optimization steps per participating client. Main setting used a learning rate of 0.12, a base proximal coefficient of 0.8, a drift sensitivity of 2.0, and a persistence parameter of 0.9. The same setting is used for the main comparisons of methods. Same configurations are tested with twenty random seeds each, thus enabling us to gather objective values from different executions.The controlled convex setup is intended to serve as validation at the mechanism level, not as a stand-in for a current deep-learning benchmark. It aims to set controlled conditions that enable discernment between nonfading client disagreement and transitory deviations.

4.2 Baselines

Four algorithms, Fed Avg, Fed Prox, Instant-Prox and PCD-Prox, are tested with the same experimental setting.In Fed Prox, the proximal regularization term is a constant with respect to the current global model. The proximal coefficient does not change with client behavior.Instant-Prox scales the proximal coefficient according to the current estimate of client drift, but it doesn't remember earlier instances of drift. Instant-Prox thus separates transient adaptation from persistent drift.PCD-Prox leverages directional drift adaptation with an exponentially smoothed persistence state, where disagreement can be subsequently used for client-specific regularization in multiple rounds.The difference between them is especially crucial since they both rely on drift-aware regularization, yet only PCD-Prox makes use of memory on past drift. Despite this, they have the same number of clients, model dimension, communication rounds, percentage of participating clients, and number of local optimization steps.

4.3 Persistent and Transient Drift

Two experimental conditions are studied: persistent-drift and transient-drift. In persistent-drift, the target shifted for the chosen clients persists for the entire duration of the experiment. This leads to the clients affected by the shift to continue optimizing in the wrong direction away from the global optimum, resulting in persistent disagreements about direction.In the transient-drift case, the client target changes for a brief period and then returns to normal. Therefore, the difference will be temporary.These two conditions provide a controlled comparison between sustained and short-lived client heterogeneity. Persistent drift is expected to reinforce the smoothed drift state, whereas the influence of transient disagreement can diminish after the shift subsides. Final objective values and optimization trajectories are evaluated under both conditions to examine the effect of temporal drift adaptation.

5. Results and Discussion

5.1 Performance under persistent drift

Consequently, these results can be viewed as indicative of a stable optimization effect rather than a large performance enhancement. In particular, the experiment shows that a persistent addition to the drift signal neither destabilizes optimization nor introduces a significant gain in performance.The transition from Fed Avg to Fed Prox, Instant-Prox and PCD-Prox follows naturally from the design. Fixed proximal regularization constrains local deviation, while adaptive regularization switches the constraint on based on client actions.

Table 1 reports the final objective values under persistent client drift. PCD-Prox has the lowest mean final objective compared to the other four methods. The difference from FedAvg is about 0.24%, whereas the difference to the proximal variants is quite insignificant.

Table 1. Final objective under persistent client drift

Method	Mean final objective	Standard deviation
FedAvg	13.0915	1.2347
FedProx	13.0751	1.2309
Instant-Prox	13.0663	1.2273
PCD-Prox	13.0604	1.2273

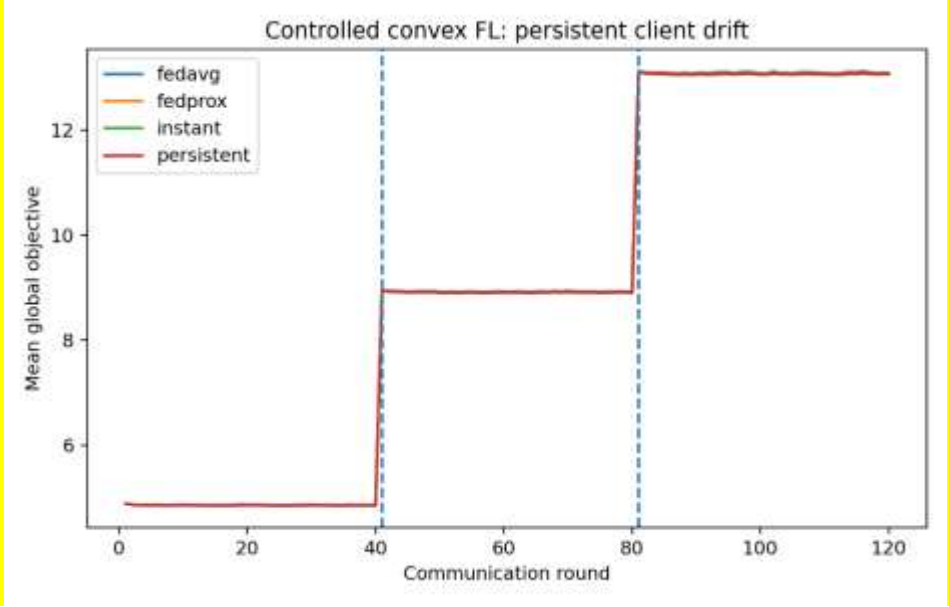


Figure 1. Convergence behaviour under persistent client drift.

Figure 1 shows similar convergence across methods, with clear objective shifts at rounds 40 and 80. The curves exhibit quite similar optimization behaviour among methods, with the adaptive methods providing slight variations in their final value of the objective. This is significant, as PCD-Prox obtains this similar behaviour without having to use an alternate aggregation procedure.

5.2 Adaptive response to client drift

Unlike fixed proximal regularization, this mechanism varies the strength of the constraint based on the client. Clients aligned to the global path experience less regularization than clients that always diverge.

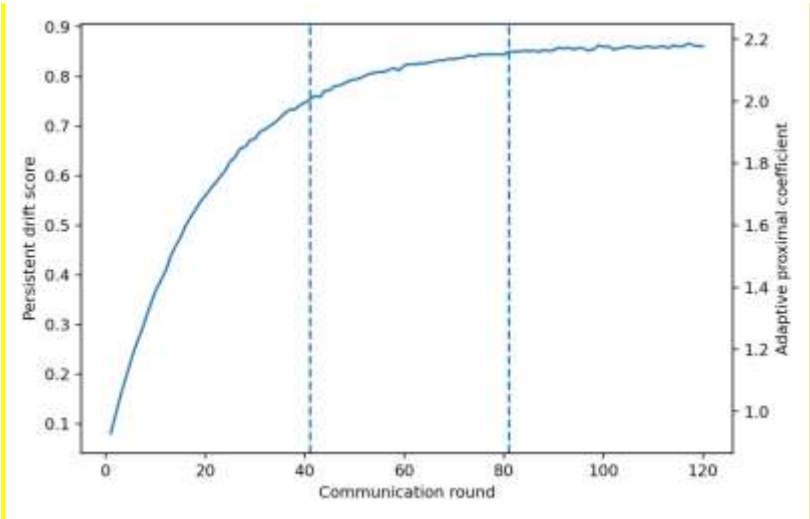


Figure 2. Relationship between persistent client drift and adaptive proximal regularization.

Figure 2 illustrates the relationship between persistent drift and the adaptive proximal coefficient. It shows that persistent drift increases progressively, causing the adaptive proximal coefficient to rise accordingly. As anticipated, the adaptive coefficient grows with the total learned drift estimate. This enables a simple and direct interpretation of the novel algorithm: the existing disagreement causes the algorithm to lessen the extent of unrestricted local optimization.

This kind of approach is most useful in non-IID federated learning, where heterogeneity is different for different clients. It is not about trying to take all clients to a common behaviour.

5.3 Transient drift.

The proposed method leverages the fact that adaptive regularization depends both on the size and duration of the disagreement.This finding has implications for dynamic federated scenarios, where the local data distributions change asynchronously over time. In fact, there is evidence that distributed concept drift occurs at different clients and different times, which in turn poses an optimization problem that cannot be expressed fully in a static non-IID partition (Jothimurugesan et al., 2023).

In **Table 2,** the lowest mean final objective is again reached by PCD-Prox but is only slightly superior to Fed Avg, by about 0.08%. This is consistent with the desired behaviour of the persistence mechanism, as the transient disagreement need not be persisted for any length of time and so its drift estimate doesn't accumulate as much.

Table 2. Final objective under transient client drift

Method	Mean final objective	Standard deviation
FedAvg	4.8591	0.3281
FedProx	4.8536	0.3270
Instant-Prox	4.8504	0.3254
PCD-Prox	4.8481	0.3258

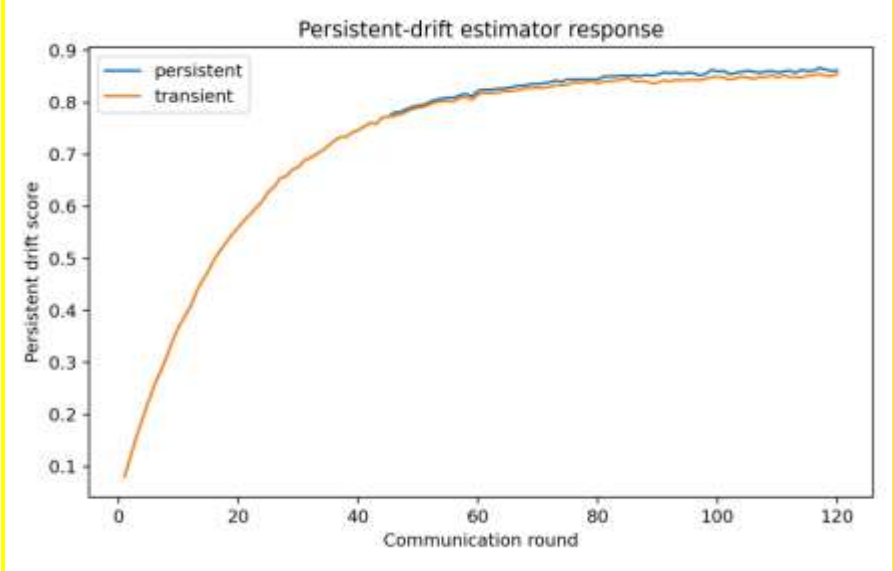


Figure 3. Persistent and transient client-drift responses.

Figure 3 shows that both persistent and transient drift start off approximately at the same level, but persistent drift consistently remains higher over time. The long-lasting signal represents the long-lasting drift signal generated by the persistent-drift condition, while the transient condition only results in a short-lived signal.

5.4 Persistence parameter sensitivity

The minimum mean final loss over the tested values happens at image. Because the differences are so small, the results do not suggest considering 0.90 an inherently superior final setting.

Rather, the experiment indicates a compromise between us and stability. Smaller would enable the method to have a quicker response to the clients' behaviour. Higher would give more priority to the past behaviour.

Table 3 indicates that the persistence parameter influences the drift state. For

$\gamma = 0$, the mechanism has no temporal memory and therefore acts as an instantaneous drift estimator, and as γ increases, the resulting accumulated drift estimate is progressively smoother with less sensitivity to individual rounds.

Table 3. Sensitivity to the persistence parameter

γ	**Mean final loss**	**Standard deviation**	**Mean persistent drift**
0.00	12.8806	0.9741	0.8592
0.50	12.8780	0.9742	0.8442
0.90	12.8732	0.9731	0.7292
0.99	12.8767	0.9732	0.2168

This, in turn, is relevant to deployment because applications can have different time dependencies. Systems with fast-changing distributions of clients may prefer a smaller memory, whereas with more persistent client-specific distributions, a higher degree of persistence can be enjoyed.

6. Limitations and Future Work

Several limitations should be considered when interpreting the current results. The main limitation is the controlled experimental setting. The study uses convex quadratic objectives to validate the proposed mechanism, providing a proof of concept, but its behaviour on deep neural networks, stochastic mini-batch training, and realistic image or text datasets remains unknown.A second limitation is the relatively small magnitude of the observed improvements. PCD-Prox reduces the final objective by approximately **0.24%** compared with FedAvg under persistent drift and **0.08%** under transient drift. These results support the proposed mechanism but are insufficient to establish practical superiority.The current evaluation also does not include several established optimization methods, including SCAFFOLD, FedNova, FedDyn, and adaptive server optimizers. Their inclusion would provide a broader comparison with alternative approaches to heterogeneous federated optimization.Another limitation is the use of directional cosine disagreement as the drift indicator. This captures whether local and global updates move in similar directions but does not represent update magnitude or gradient variance. Future versions could therefore combine directional information with update magnitude or loss-based signals. Further experiments can intensively compare PCD-Prox with standard local algorithms (independent, non-identity partitioned data like CIFAR-10, partial participation, multiple random seeds) and federated optimization algorithms (we compare with one more algorithm). It would also be interesting to perform ablation studies to study parameters related to persistence and drift-sensitivity. Finally, since distributed concept drift is a major direction for future work, the state persistence could be designed to decay in time as user behaviour converges to follow the global optimization path, allowing us to distinguish between transient and long-term heterogeneity without explicit knowledge of user data distribution.

7. Conclusion

The experiments validate the motivation for persistence-based local regularization, although the observed gains are modest.The first observation is that proximal regularization provides additional control over local optimization under heterogeneous objectives. FedProx consistently achieves a marginally lower final objective than FedAvg in both experimental settings, indicating that restricting excessive local divergence can improve optimization stability.The second finding is that the drift signal can be used to adapt the proximal coefficient. Instant-Prox performs better than fixed FedProx in both experiments, suggesting that adjusting regularization according to observed client behaviour can provide an additional optimization benefit. The third observation concerns temporal persistence. PCD-Prox achieves the lowest mean objective in both persistent and transient settings, with a larger improvement in the persistent case. This behaviour is consistent with the temporal-memory mechanism, where repeated disagreement produces a stronger accumulated signal than an isolated deviation.The set of controlled experiments provides preliminary results to this proposition. In the set of methods tested, PCD-Prox obtains the lowest average final objective in the presence of both persistent and transient drift, with an especially large enhancement in the case of persistent drift. Additionally, analysis on the persistence parameter indicates that temporal smoothing affects the accumulated drift estimate and the optimization performance.This approach adheres to the conventional federated communication framework, with the exception of a single scalar persistence state. Consequently, the change is straightforward to apply as no extra communication rounds are needed. This shows results in a setting where the clients are optimized under controlled conditions, and the improvements are

moderate. It is necessary to test this approach on real non-IID data and compare it to more federated optimization algorithms. Additional work might involve more sophisticated drift indicators and adaptive persistence strategies to deal with varying client distributions.

References

1. Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P. N., & Saligrama, V. (2021). Federated learning based on dynamic regularization. *International Conference on Learning Representations*.

2. Chung, W.-C., & Peng, C.-H. (2025). FedCOLA: Federated learning with heterogeneous feature concatenation and local acceleration for non-IID data. *Future Generation Computer Systems, 166*, 107674. https://doi.org/10.1016/j.future.2024.107674

3. Jothimurugesan, E., Hsieh, K., Wang, J., Joshi, G., & Gibbons, P. B. (2023). Federated learning under distributed concept drift. *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, 206*, 5834–5853.

4. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. *Proceedings of the 37th International Conference on Machine Learning, 119*, 5132–5141.

5. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, M. J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., ... Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning, 14*(1–2), 1–210.

6. Li, M., & Wang, D. (2020). FedMA: Federated learning with model averaging. *International Conference on Learning Representations*.

7. Li, Q., Diao, Y., Chen, Q., & He, B. (2022). Federated learning on non-IID data: A survey. *Neurocomputing, 465*, 371–390. https://doi.org/10.1016/j.neucom.2021.07.098

8. Li, Q., He, B., & Song, D. (2021). Model-contrastive federated learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10713–10722.

9. Li, T., Hu, S., Beirami, A., & Smith, V. (2021). Ditto: Fair and robust federated learning through personalization. *Proceedings of the 38th International Conference on Machine Learning, 139*, 6357–6368.

10. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems, 2*, 429–450.

11. Li, X., Jiang, M., Zhang, X., Kamp, M., & Dou, Q. (2021). FedBN: Federated learning on non-IID features via local batch normalization. *International Conference on Learning Representations*.

12. Liang, X.-Y., Zhang, H.-R., Tang, W., & Min, F. (2024). Robust federated learning with voting and scaling. *Future Generation Computer Systems, 153*, 113–124. https://doi.org/10.1016/j.future.2023.11.015

13. Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., & Jiang, J. (2023). Multi-center federated learning: Clients clustering for better personalization. *World Wide Web, 26*, 481–500. https://doi.org/10.1007/s11280-022-01046-x

14. Ma, X., Zhu, J., Lin, Z., Chen, S., & Qin, Y. (2022). A state-of-the-art survey on solving non-IID data in federated learning. *Future Generation Computer Systems, 135*, 244–258. https://doi.org/10.1016/j.future.2022.05.003

15. McMahan, B., Moore, E., Ramage, D., Hampson, S., &Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54*, 1273–1282.

16. Mármol Campos, E., González-Vidal, A., Hernández-Ramos, J. L., &Skarmeta, A. (2025). FedRDF: A robust and dynamic aggregation function against poisoning attacks in federated learning. *IEEE Transactions on Emerging Topics in Computing, 13*(1), 48–67.

17. Pfeiffer, K., Rapp, M., Khalili, R., & Henkel, J. (2023). Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Computing Surveys, 55*(14s), 1–27. https://doi.org/10.1145/3596907

18. Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2021). Adaptive federated optimization. *International Conference on Learning Representations*.

19. Wang, J., Liu, Q., Liang, H., Joshi, G., & Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems, 33*.

20. Ye, M., Fang, X., Du, B., Yuen, P. C., & Tao, D. (2023). Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys, 56*(3), 1–44.

21. Zheng, S., Yuan, W., Wang, X., & Duan, L. (2024). Adaptive federated learning via new entropy approach. *IEEE Transactions on Mobile Computing, 23*(12), 11920–11936. https://doi.org/10.1109/TMC.2024.3402080