FedEve: On Bridging the Client Drift and Period Drift for Cross-device Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Federated learning (FL) is a machine learning paradigm that allows multiple clients to collaboratively train a shared model without exposing their private data. Data heterogeneity is a fundamental challenge in FL, which can result in poor convergence and performance degradation. *Client drift* has been recognized as one of the factors contributing to this issue resulting from the multiple local updates in FedAvg. However, in cross-device FL, a different form of drift arises due to the partial client participation, but it has not been studied well. This drift, we referred as *period drift*, occurs as participating clients at each communication round may exhibit distinct data distribution that deviates from that of all clients. It could be more harmful than client drift since the optimization objective shifts with every round. In this paper, we investigate the interaction between period drift and client drift, finding that period drift can have a particularly detrimental effect on cross-device FL as the degree of data heterogeneity increases. To tackle these issues, we propose a predict-observe framework and present an instantiated method, FedEve, where these two types of drift can compensate each other to mitigate their overall impact. We provide theoretical evidence that our approach can reduce the variance of model updates. Extensive experiments demonstrate that our method outperforms alternatives on non-iid data in cross-device settings.

1 Introduction

Federated learning is a decentralized machine learning approach that enables multiple clients to collaboratively train a shared model without exposing their private data [McMahan et al., 2017]. In this paradigm, each client independently trains a local model using its own data and subsequently sends the model updates to a central server. The server then periodically aggregates these updates to improve the global model until it reaches convergence. There are two primary settings in FL: cross-silo and cross-device [Kairouz et al., 2021]. Cross-silo FL typically involves large organizations (small number of clients), where most clients actively participate in every round of training [Chen and Chao, 2021, Lin et al., 2020]. In contrast, cross-device FL focuses on scenarios like smartphones (huge number of clients, e.g., millions), where only a limited number of clients participate in each round [Li et al., 2020b, Reddi et al., 2020], due to communication bandwidth, client availability, and other issues. This paper primarily focuses on the cross-device setting with partial client participation since we discover and then solve its unique challenge —"***period drift***".

Distinguished from traditional distributed optimization, the statistical heterogeneity of data has been acknowledged as a fundamental challenge in FL [Li et al., 2020a, Chen and Chao, 2021, Lin et al., 2020]. This data heterogeneity refers to the violation of the independent and identically distributed (non-iid) data assumption across clients, which can result in poor convergence and performance degradation when using FEDAVG . *Client drift* is recognized as one of the factors contributing to this issue and attracts numerous efforts to address it [Karimireddy et al., 2021, Li et al., 2020b, Reddi et al.,

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2020]. This phenomenon is characterized by clients who, after multiple local updates, progress too far towards minimizing their local objective, consequently diverging from the shared direction. However, in cross-device FL, a different form of drift exists and could be more detrimental to the training process than client drift, which has not been extensively studied. *This drift occurs periodically as different clients participate in each communication round, and these participating clients as a group may exhibit distinct data distribution that deviates from the overall distribution of all clients.* Period drift is fundamentally different from the noise in SGD, as further described in Appendix C.

This deviation could potentially lead to slow and unstable convergence, as the optimization objective shifts with every round. For simplicity, we refer to this phenomenon as ***period drift***. Despite both period drift and client drift being rooted in data heterogeneity, they stem from different causes (as illustrated in Figure 1). Client drift results from multiple local updates and the non-iid, while period drift arises due to partial client participation and the non-iid. The concept of period drift and client drift is shown in the following equation:

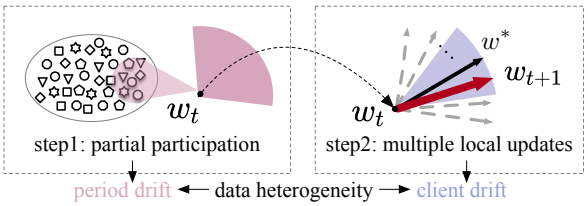


Figure 1: **The generation of period drift and client drift**

$$w^* \neq w^*_{\mathcal{N}} \neq w^*_{\mathcal{S}} \neq \bar{w} = \frac{1}{|\mathcal{S}|}\sum w^*_k \neq w^*_k, \tag{1}$$

Period Drift **Client Drift**

which is further described in detail in Appendix B.1. These two drifts, when occurring simultaneously, significantly increase the complexity of achieving stable and efficient convergence, particularly challenging the effectiveness of cross-device FL systems.

In this paper, we first investigate the impact of period drift and client drift, finding that period drift can have a particularly detrimental effect on cross-device FL as the degree of data heterogeneity increases (as demonstrated in detail in Section 3.2). *While the impacts of period drift and client drift are additive, we fortunately uncover a cooperative mechanism therby these two types of drift can compensate each other to mitigate their overall impact.* To achieve this, we propose a predict-observe framework, where we consider at each round 1) the server optimization (e.g., momentum) as a prediction of a update step of FL; 2) the clients' optimization (e.g., local SGD) as an observation of this update step. Note that the vanilla FEDAVG is a special case in which the server does not make any predictions and solely relies on the observation provided by clients. In this framework, period drift and client drift are viewed as the noise respectively associated with prediction and observation. We thereby incorporate a Bayesian filter to integrate prediction (with period drift) and observation (with client drift) to achieve a better estimation of update step and reduce uncertainties. Based on the predict-observe framework, we present an instantiated method, referred as FEDEVE , which combines the prediction and observation through linear interpolation. The coefficient of this linear combination indicate the relative confidence between prediction and observation, which is determined by the variance of the period drift and client drift, thus produces a more precise estimation of updates. FEDEVE does not increase the client storage or extra communication costs, and does not introduce additional hyperparameter tunning, making it ideal for cross-device FL.

Contributions We summarize the primary contributions of this paper as follows:

- We analyze the impact of period drift and client drift for cross-device FL, and observe that period drift has a particularly detrimental effect as the degree of data heterogeneity increases.
- We propose a predict-observe framework for cross-device FL that incorporates a Bayesian filter to integrate server optimization and clients' optimization so that period drift and client drift can compensate for each other.
- As an instantiation of the proposed framework, we present FEDEVE to combine prediction and observation through linear interpolation based on the variance of the period drift and client drift.
- We provide theoretical evidence within our framework that FEDEVE can reduce the variance of model updates. Extensive experiments demonstrate that our method outperforms alternatives on non-iid data in cross-device settings. [1]

[1]For a smoother reading experience, please feel free to check out our reading guide in Appendix A.

2 Methodology

In this section, we discuss the problem of some methods (e.g., FEDAVG) in cross-device FL, and then propose our predict-observe framework and a method FEDEVE to deal with it.

2.1 Typical Federated Learning Setup

Federated learning, as described by McMahan et al. [2017], involves utilizing multiple clients and a central server to optimize the overall learning objective. The goal is to minimize the following objective function:

$$\min_{w} \ f(w) = \sum_{k=1}^{N} p_k F_k(w) = \mathbb{E}_k[F_k(w)], \tag{2}$$

where N is the number of clients, $p_k \geq 0$, and $\sum_k p_k$=1. In general, the global objective is the expectation of the local objective over different data distributions $\mathcal{D}_k$, i.e., $F_k(w) = \mathbb{E}_{x_k \sim \mathcal{D}_k} f_k(w; x_k)$, with n_k samples on each client k and weighted by p_k. We set p_k=$\frac{n_k}{n}$, where n= $\sum_k n_k$ is the total number of data points. In deep learning setting, $F_k(w)$ is often non-convex. A common approach to solve the objective (2) in federated settings is FEDAVG [McMahan et al., 2017]. For example, in cross-device FL, a small subset $\mathcal{S}_t$ ($|\mathcal{S}_t| \ll N$) of the total clients are selected at each round (ideally randomly, but possibly biased in practice), and then the server broadcasts its global model to the selected client. In parallel, each of the selected clients runs SGD on their own loss function $F_k(\cdot)$ for E number of epochs, and sends the resulting model to the server. The server then updates its global model as the average of these local models and repeats this process until convergence. One problem of FL is the non-iid data across clients, which can bring about "client drift" in the updates of each client, resulting in slow and unstable convergence [Karimireddy et al., 2021]. Despite efforts to address the problem of client drift [Karimireddy et al., 2021, Li et al., 2020b, Reddi et al., 2020], there is a lack of research on the issue of period drift, i.e. the data distribution of selected clients at each round may differ from the overall data distribution of all clients. Period drift along with client drift can greatly impact the convergence of the learning process in FL, thus we propose a predict-observe framework to deal with them.

2.2 The Concept of Drift

In contrast to conventional distributed optimization, federated learning possesses distinct characteristics, such as client sampling, multiple local epochs, and non-iid data distribution. These attributes may lead to a drift in the updates of global model, resulting in suboptimal performance. This drift can be thought of as a noise term that is added to the true optimization states during the optimization process. Thus, we can make the definition of period drift and client drift as:

Definition 2.1 (Period Drift and Client Drift)**.** In federated learning, two types of drift arise from data heterogeneity:

- **Period Drift:** The deviation when the data distribution of selected clients differs from the overall distribution. Formally:

$$\text{Period Drift} := \mathbb{E}_{\mathcal{S}_t} \left[\left\| \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} \nabla F_k(w) - \nabla f(w) \right\|^2 \right], \tag{3}$$

  where $\mathbb{E}_{\mathcal{S}_t}$ is the expectation over client sampling, $\mathcal{S}_t$ is the subset of clients selected at round t, $\nabla F_k(w)$ is the gradient of client k's objective, and $\nabla f(w)$ is the gradient of the global objective.

- **Client Drift:** The deviation when the averaged optima of local objectives does not align with the optimum of the averaged objective. Formally:

$$\text{Client Drift} := \mathbb{E}_{k \in \mathcal{S}_t} \left[\left\| \nabla F_k(w_k^*) - \nabla F_k(w_{\mathcal{S}_t}^*) \right\|^2 \right], \tag{4}$$

  where w_k^* is the local optimum for client k and $w_{\mathcal{S}_t}^*$ is the optimum for the selected clients' averaged objective.

These drifts are conceptually independent but together affect model convergence and performance. See Appendix B.1 for detailed discussion.

Assumption 2.2. *The aggregated model parameters on the server w_{server}, can be represented as the sum of the optimal parameters w^* and a drift (noise) that follows a normal distribution $w_{drift} \sim \mathcal{N}(0, \sigma^2_{drift})$:*

$$w_{server} = w^* + w_{drift} \leftarrow noise, \tag{5}$$

where w^* represents the optimal parameters obtained through the use of stochastic gradient descent (SGD), w_{drift} represents the noise term caused by factors such as client sampling, multiple local epochs, and non-iid data distribution that we assume a normal distribution, and w_{server} represents the aggregated model parameters also follows a normal distribution $w_{server} \sim \mathcal{N}(w^*, \sigma^2_{drift})$, with the expectation of the aggregate model parameters being equal to the optimal parameters, i.e. $\mathbb{E}[w_{server}] = w^*$. Note that the assumption of Gaussian-like noise is natural, and its justification can be found in Appendix D.1.

2.3 The Predict-observe Framework

Initially, we establish the concept of period drift, represented by Q_t, and client drift, represented by R_t at the t-th communication round. We first make an assumption of independence concerning the two types of drift, which states that the two drifts are independent of one another. This assumption allows us to more accurately analyze the impact of each drift on the model's performance and devise methods to mitigate their effects.

Assumption 2.3. *The initialization model parameters are independent of all period drifts Q_t and client drifts R_t at each communication round, that is $w_0 \perp Q_0, Q_1, \cdots, Q_t$ and $w_0 \perp R_0, R_1, \cdots, R_t$.*

The justification and limitation of this assumption can be found in Appendix D.2. Since the clients participating in each round in cross-device FL is only a small fraction of all clients, period drift can be attributed to the discrepancy that the objective of selected clients at each round does not align with the overall objective. Thus, an effective prediction of updates can potentially help reduce the period drift. As formulated in Equation (5), we express the prediction of updates on the server as:

$$\hat{w}_{t+1} = g(w_t) + Q_t, \quad Q_t \sim \mathcal{N}(0, \sigma^2_{Q_t}), \tag{6}$$

where $\hat{w}_{t+1}$ is the prediction model of $(t+1)$-th round as the output of predcit function $g(\cdot)$ with the current model w_t as input. It is noteworthy that the period drift at the t-th round is represented by Q_t, and just like the drift in assumption 2.2, it is assumed to follow a normal distribution $\mathcal{N}(0, \sigma^2_{Q_t})$, characterized by a mean of zero and a variance of $\sigma^2_{Q_t}$. Client drift can be attributed to the phenomenon that the averaged optima of objectives does not align with the optima of averaged objectives. Thus, we consider the updates provided by these clients is a kind of observation of global updates. As formulated in Equation (5), we express it as:

$$\tilde{w}_{t+1} = h(\hat{w}_{t+1}) + R_t, \quad R_t \sim \mathcal{N}(0, \sigma^2_{R_t}), \tag{7}$$

where $\tilde{w}_{t+1}$ is the model of $(t+1)$-th round as the output of observe function $h(\cdot)$ with the predict model w_t as input. Also, the client drift at the t-th round is represented by R_t, and just like the drift in assumption 2.2, it is assumed to follow a normal distribution $\mathcal{N}(0, \sigma^2_{R_t})$, characterized by a mean of zero and a variance of $\sigma^2_{R_t}$. It is clear that standard FEDAVG is a special case since there is no prediction for server optimization, and it solely relies on the observations provided by clients. Furthermore, the period drift, Q_t, and the client drift, R_t, are represented as noise terms that are incorporated into the prediction and observation functions. According to assumption (2.3), these drifts are independent of the current model states, and the lemma of independence noise is posited:

Lemma 2.4. *(Independence of Noise). the noise present in the prediction and observation at each communication round is independent of the current model state, specifically, $w_t \perp Q_t$ and $w_t \perp R_t$.*

The complete proof of the independence of noise can be found in appendix D.3. The equations presented in equations 6 and 7 depict the prediction and observation of updates, respectively, taking into account both period drift and client drift. In order to reconcile the discrepancy between the prediction (including period drift) and observation (including client drift), a Bayesian filter is introduced to allow for compensation between the two sources of drift. The prior probability of w_{t+1} is represented by $P(\hat{w}_{t+1})$, and by combining the observation $P(\tilde{w}_{t+1})$ and the likelihood $P(\tilde{w}_{t+1} \mid \hat{w}_{t+1})$, the posterior probability $P(w_{t+1} \mid \tilde{w}_{t+1})$ of w_{t+1} can be calculated as the new model at the $(t+1)$-th round, as shown in Equation (8).

$$P(\, w_{t+1} \,) := P(\, \hat{w}_{t+1} \mid \tilde{w}_{t+1} \,) = \frac{P(\, \tilde{w}_{t+1} \mid \hat{w}_{t+1} \,) P(\, \hat{w}_{t+1} \,)}{P(\, \tilde{w}_{t+1} \,)}. \tag{8}$$

By utilizing the Bayesian filter in our predict-observe framework, an update mechanism is implemented that first performs prediction and then observes the predicted model state, as described in the following procedure:

$$\begin{aligned} f^+_{w_t}(w) \overset{\text{predict}}{\Longrightarrow} f^-_{\hat{w}_{t+1}}(w) &= \int_{-\infty}^{+\infty} f_{Q_t}[w - f(v)] f^+_{w_t}(v)\mathrm{d}v \\ \overset{\text{observe}}{\Longrightarrow} f^+_{w_{t+1}}(w) &= \eta_t \cdot f_{R_t}\left[w_{t+1} - h(w)\right] \cdot f^-_{\hat{w}_{t+1}}(w), \end{aligned} \tag{9}$$

where $f^+_{w_t}(w)$ is the posterior probability of w_t, $f^-_{\hat{w}_{t+1}}(w)$ is the prior probability of w_{t+1}, f_{Q_t} is the PDF of period drift, $f^+_{w_{t+1}}(w)$ is the posterior probability of w_{t+1}, f_{R_t} is the PDF of client drift, and $\eta_t = \left\{\int_{-\infty}^{+\infty} f_{R_t}\left[\tilde{w}_{t+1} - h(\hat{w}_{t+1})\right] f^-_{\hat{w}_{t+1}}(w)\mathrm{d}w\right\}^{-1}$. By combining prediction and observation, the fused model can be estimated by taking the expectation of the posterior probability as follow:

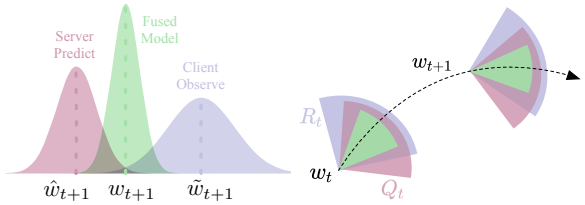
$$\hat{w}_{t+1} = E\left[f^+_{w_{t+1}}(w)\right] = \int_{-\infty}^{+\infty} w f^+_{w_{t+1}}(w)\mathrm{d}w. \tag{10}$$

Theorem 2.5. *Given assumption 2.2 and lemma 2.4, the composite model will exhibit a diminished degree of variance in comparison to the individual variances of both period drift and client drift, and the mean will be a linear combination that is weighted by the variances:*

$$\mu_{fused} = \frac{\mu_1\sigma^2_{R_t} + \mu_2\hat{\sigma}^2_{t+1}}{\sigma^2_{R_t}}, \quad \sigma^2_{fused} = \frac{\hat{\sigma}^2_{t+1}\sigma^2_{R_t}}{\hat{\sigma}^2_{t+1} + \sigma^2_{R_t}}, \tag{11}$$

where $\mu_1, \mu_2, \mu_{fused}$ is the mean of prediction, observation and fused model, and $\hat{\sigma}^2_{t+1}, \sigma^2_{R_t}, \sigma^2_{fused}$ is the variance of prediction, client drift and fused model.

The complete proof of the bayesian filter can be found in appendix D.5. The application of Bayesian filtering allows for the interaction of period drift and client drift to generate a new model, which is characterized by a reduced level of variance as compared to the individual variances of period drift and client drift, as depicted in Figure 2(a). However, the computation of the new model is challenging due to the presence of infinite integrals in Equation (10) and η_t, as it is a general framework for any prediction and observation function. In the following section, we will propose a specialized method to facilitate the convergence of FL.



(a) Bayesian filter (b) FEDEVE

Figure 2: **Illustrations of the framework and FEDEVE**

2.4 The FEDEVE Method

The predict-observe framework has been proposed as a strategy for mitigating the challenges of period drift and client drift. However, it also raises some questions regarding the effective method of prediction and the variance associated with both period and client drift. In this section, we demonstrate that the utilization of momentum as a server optimization [Hsu et al., 2019, Reddi et al., 2020], can serve as an effective prediction method. Furthermore, we present a method for estimating the variance of period drift and client drift. In the context of the predict-observe framework, we have adapted it to a specific setting where Nesterov momentum is employed as the prediction function $g(\cdot)$, and the observation function $h(\cdot)$ is the average of the models from the clients the same as FEDAVG . We have reformulated FEDAVG in an incremental form as the starting point of our approach.

$$\begin{aligned} w_{t+1} &= \sum_{k\in\mathcal{S}_t} p_k w^k_t = w_t - \sum_{k\in\mathcal{S}_t} p_k \left(w_t - w^k_t\right) \\ &= w_t - \sum_{k\in\mathcal{S}_t} p_k \Delta w^k_t = w_t - \Delta w_t. \end{aligned} \tag{12}$$

This formulation facilitates the accumulation of Δw_t as the momentum on the server, which serves as a prediction of updates, as the empirical value of the hyperparameter $\beta = 0.9$ suggests that the

direction of historical updates is likely to be maintained. By introducing the Nesterov momentum and specialize $g(w_t) = w_t - \eta_g M_t$ in Equation (6) as the prediction function. Additionally, we specialize $h(\hat{w}_{t+1}) = \hat{w}_{t+1} - \eta_g \Delta\tilde{w}_t$ in Equation (7) as the observation function. Thus, the predict-observe equation can be rewritten as follow:

$$\hat{w}_{t+1} = w_t - \eta_g M_t + Q_t, \tag{13}$$

$$\tilde{w}_{t+1} = \hat{w}_{t+1} - \eta_g \Delta\tilde{w}_t + R_t, \tag{14}$$

where M_t is the momentum (the accumulation of Δw_t) at t-th round, $\Delta\tilde{w}_t$ is the average of model update in Equation (12) from clients at the states of $\hat{w}_{t+1}$, and η_g is the global learning rate. By assuming a normal distribution for Q_t and R_t based on the equations (6), (7), and (8), the problem of infinite integral in Equation (10) and η_t can be solved in a closed-form, as detailed in reference D.5.2. Additionally, due to the normal distribution, the form of distribution like equations 15e, 10 is not necessary, and only the mean and variance are used to depict the model update process. Since these equations are linear in nature, the Bayesian filter can be specialized as the Kalman Filter (KF). The process of model update can thus be summarized as the use of KF, as represented by the following formulation:

$$\hat{w}t + 1 = w_t - \eta_g M_t, \tag{15a}$$

$$\hat{\sigma}^2 t + 1 = \sigma_t^2 + \sigma_{Q_t}^2, \tag{15b}$$

$$G_{kal} = \frac{\hat{\sigma}_{t+1}^2}{\hat{\sigma}_{t+1}^2 + \sigma_{R_t}^2}, \tag{15c}$$

$$M_{t+1} = M_t + G_{kal}(\Delta\tilde{w}t - M_t), \tag{15d}$$

$$w_{t+1} = w_t - \eta_g M_{t+1}, \tag{15e}$$

$$\sigma_{t+1}^2 = (1 - G_{kal})\hat{\sigma}_{t+1}^2. \tag{15f}$$

The six steps of model update for each communication round in our method are outlined in Equations (15a)-(15f). Equation (15a) predicts the model states w_t using the momentum M_t. Equation (15b) estimates the variance of the prediction model by summing the variance of w_t and the period drift Q_t. To provide a clear representation, the variance of the prediction model is represented by σ^- and the variance of the fused model is represented by σ^+. The core of our method is presented in Equation (15c), where the Kalman gain G_{kal} is calculated based on the ratio of the variance of the prediction $\hat{\sigma}_{t+1}^2$ and the observation (client drift) R_t. The value of G_kal determines the relative weight of the prediction and observation when they are combined. Equation (15d) fuses the prediction and observation in a linear fashion, weighted by the Kalman gain G_{kal} calculated in (15c). The fourth line updates the global model with the fused M_{t+1} calculated in (15d). Equation (15e) estimates the variance of the fused model w_{t+1} using G_{kal} in (15d) and $\hat{\sigma}_{t+1}^2$ in (15b), which will be used in the next communication round. It is worth noting that all these calculations are performed on the server, thus our method retains the same level of communication cost as FEDAVG while also being compatible with cross-device FL settings.

While Equations (15a)-(15f) provide an efficient and accurate method for model updates, the variance of the period drift $\sigma_{Q_t}^2$ in Equation (15b) and the client drift $\sigma_{R_t}^2$ in Equation (15c) remains unresolved. To address this issue, we propose an effective method for estimating the variance of the period drift and client drift.The period drift, which is a measure of the deviation from the consistency of the optimization objective at each communication round, can be quantified by analyzing the discrepancy between the prediction and the observation. Specifically, this can be done by computing the variance between the momentum M_t and the average of the model updates $\Delta\tilde{w}_t$. Similarly, the client drift, which represents the inconsistency of the updates made by different clients, can be estimated by computing the variance between the average of the model updates $\Delta\tilde{w}_t$ and the updates made by each individual client $\Delta\tilde{w}_t^k$. We formulate the estimation of the variance of period drift and client drift as follows:

$$\begin{aligned} \sigma_{Q_t}^2 &:= \frac{\sum_{i=1}^{d}(M_t^i - \Delta\tilde{w}_t^i)^2}{|\mathcal{S}_t|d}, \\ \sigma_{R_t}^2 &:= \frac{\sum_{k\in\mathcal{S}_t}\sum_{i=1}^{d}(\Delta\tilde{w}_t^{k,i} - \Delta\tilde{w}_t^i)^2}{|\mathcal{S}_t|^2 d}, \end{aligned} \tag{16}$$

where the index of model parameters is represented by the uppercase i and the dimension of the model is represented by d. With the estimation of the variance of period drift and client drift, the

Table 1: **Results on FEMNIST and CIFAR-100.** The best method is highlighted in **bold** fonts.

Dataset	FEMNIST				CIFAR-100		
Methods/NonIID	natural	$\alpha = 1$	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 1$	$\alpha = 0.1$	$\alpha = 0.01$
FEDAVG	82.37 ± 0.18	83.60 ± 0.11	82.02 ± 0.23	73.23 ± 1.36	47.04 ± 0.21	43.93 ± 0.36	30.11 ± 0.53
FEDAVGM	82.53 ± 0.43	83.67 ± 0.10	82.30 ± 0.49	74.96 ± 2.34	48.22 ± 0.19	44.74 ± 0.40	31.59 ± 0.98
FEDPROX	82.34 ± 0.17	83.58 ± 0.11	82.04 ± 0.27	74.16 ± 1.19	46.86 ± 0.38	43.74 ± 0.27	30.10 ± 0.55
SCAFFOLD	81.66 ± 0.28	83.06 ± 0.14	79.82 ± 0.42	5.13 ± 0.00	47.26 ± 1.49	36.36 ± 4.98	1.00 ± 0.00
FEDOPT	5.13 ± 0.00	81.86 ± 0.38	78.13 ± 0.39	5.13 ± 0.00	47.26 ± 1.49	45.43 ± 1.18	32.17 ± 1.38
FEDEVE	$\mathbf{82.68 \pm 0.19}$	$\mathbf{83.81 \pm 0.09}$	$\mathbf{82.69 \pm 0.31}$	$\mathbf{75.99 \pm 1.61}$	$\mathbf{48.38 \pm 0.24}$	$\mathbf{45.68 \pm 0.16}$	$\mathbf{32.68 \pm 0.62}$

Table 2: **Results on MovieLens-1M.** The best method is highlighted in **bold** fonts.

	AUC	HR@5	HR@10	NGCG@5	NGCG@10
FEDAVG	0.7633 ± 0.0065	0.2774 ± 0.0100	0.4294 ± 0.0120	0.1835 ± 0.0058	0.2324 ± 0.0064
FEDAVGM	0.7555 ± 0.0128	0.2705 ± 0.0384	0.4290 ± 0.0196	0.1771 ± 0.0319	0.2280 ± 0.0257
FEDPROX	0.7819 ± 0.0033	0.2700 ± 0.0129	0.4279 ± 0.0083	0.1803 ± 0.0078	0.2310 ± 0.0065
FEDOPT	0.7751 ± 0.0085	0.2868 ± 0.0055	0.4392 ± 0.0101	0.1886 ± 0.0044	0.2377 ± 0.0040
FEDEVE	$\mathbf{0.7967 \pm 0.0016}$	$\mathbf{0.2916 \pm 0.0077}$	$\mathbf{0.4460 \pm 0.0088}$	$\mathbf{0.1924 \pm 0.0039}$	$\mathbf{0.2407 \pm 0.0037}$

overall process of model update can be described in Algorithm 1. The fundamental principle of FEDEVE is to calculate the Kalman gain G_{kal} which is used to determine the relative weight of the prediction and observation when they are combined. The value of G_{kal} is calculated based on the ratio of the variance of the prediction $\sigma^2_{Q_t}$ and the variance of the observation $\sigma^2_{R_t}$. This coefficient is used to adjust the update direction of the model. A small G_{kal} means that the observation is close to the prediction, hence the update direction will also be close to the prediction. A large G_{kal} means that the observation deviates significantly from the prediction, hence the update direction will deviate from the prediction and be closer to the observation. This allows the algorithm to adapt to different scenarios in which the observations may deviate more or less from the predictions.

3 Experiments

3.1 Setup

Datasets and models. We evaluate FEDEVE on three computer vision (CV) and recommender system (RS) datasets under realistic cross-device FL settings. ***For CV dataset***, we use FEMNIST[2] Caldas et al. [2018], consisting of 671,585 training examples and 77,483 test samples of 62 different classes including 10 digits, 26 lowercase and 26 uppercase images with 28x28 pixels, handwritten by 3400 users. We also use CIFAR-10/100 [3] Caldas et al. [2018], consisting of 50,000 training examples and 10,000 test samples of 10/100 different classes with 32x32 pixels. For FEMNIST dataset, we use the lightweight model LeNet5 LeCun et al. [1998] and for CIFAR-10/100 dataset, we use ResNet-18 (replacing batch norm with group norm [Hsieh et al., 2020, Reddi et al., 2020]). ***For RS dataset***, we use MovieLens 1M [4]Harper and Konstan [2015], including 1,000,209 ratings by unidentifiable 6,040 users on 3,706 movies. It is a click-through rate (CTR) task, and we use the popular DIN Zhou et al. [2018] model. For performance evaluation, we follow a widely used leave-one-out protocol Muhammad et al. [2020]. For each user, we hold out their latest interaction as testset and use the remaining data as trainset, and binarize the user feedback where all ratings are converted to 1, and negative instances are sampled 4:1 for training and 99:1 for test times the number of positive ones.

Federated learning settings. It is important to note that the datasets FEMNIST and MovieLens 1M have a "natural" non-iid distribution, which means that the data is split by "user_id". For example, in FEMNIST, images are handwritten by different users, and in MovieLens 1M, movies are rated by different users. Furthermore, we use the Dirichlet distribution, to simulate the label distribution skew setting for FEMNIST, as described in Hsu et al. [2019]. This distribution allows us to control the degree of heterogeneity by adjusting the hyperparameter α (the smaller, the more non-iid). This allows us to test the robustness of the algorithm under different levels of heterogeneity, which is a common scenario in real-world FL settings. For the FL training, we set a total of $T = 1500$ communication rounds for the CV task and sample 10 clients per round with SGD optimizer. For the RS task, we set a total of $T = 1000$ communication rounds and sample 20 clients per round with

[2]https://github.com/TalwalkarLab/leaf/tree/master/data/femnist

[3]https://www.cs.toronto.edu/ kriz/cifar.html

[4]https://grouplens.org/datasets/movielens/

Table 3: **Results on FEMNIST with different α and E.** The best method is highlighted in **bold** fonts.

Method	Natural			$\alpha = 1$			$\alpha = 0.1$			$\alpha = 0.01$		
	$E = 1$	$E = 3$	$E = 5$	$E = 1$	$E = 3$	$E = 5$	$E = 1$	$E = 3$	$E = 5$	$E = 1$	$E = 3$	$E = 5$
FEDAVG	82.46 ± 0.18	81.95 ± 0.26	66.38 ± 30.63	83.64 ± 0.11	83.57 ± 0.12	83.38 ± 0.09	82.12 ± 0.23	81.91 ± 0.23	81.70 ± 0.26	74.51 ± 1.36	73.43 ± 1.73	72.67 ± 1.39
FEDAVGM	82.55 ± 0.43	81.99 ± 0.42	50.97 ± 37.43	83.67 ± 0.10	83.79 ± 0.11	83.65 ± 0.08	82.36 ± 0.49	82.23 ± 0.26	82.16 ± 0.34	75.18 ± 2.34	74.20 ± 2.81	73.79 ± 3.13
FEDPROX	82.43 ± 0.17	81.90 ± 0.27	51.07 ± 37.51	83.62 ± 0.11	83.52 ± 0.14	67.65 ± 31.26	82.17 ± 0.27	82.12 ± 0.19	81.94 ± 0.30	75.07 ± 1.19	74.38 ± 1.46	60.22 ± 27.56
SCAFFOLD	81.66 ± 0.28	81.08 ± 0.36	80.76 ± 0.30	83.18 ± 0.14	82.75 ± 0.16	82.46 ± 0.16	79.82 ± 0.42	79.00 ± 0.65	78.44 ± 0.70	5.13 ± 0.00	5.13 ± 0.00	5.13 ± 0.00
FEDOPT	5.13 ± 0.00	5.13 ± 0.00	5.13 ± 0.00	81.86 ± 0.38	35.90 ± 37.69	35.66 ± 37.39	78.13 ± 0.39	5.13 ± 0.00	5.13 ± 0.00	5.13 ± 0.00	5.13 ± 0.00	5.13 ± 0.00
FEDEVE	$\mathbf{82.66 \pm 0.19}$	$\mathbf{82.20 \pm 0.20}$	$\mathbf{81.93 \pm 0.16}$	$\mathbf{83.81 \pm 0.09}$	$\mathbf{83.88 \pm 0.08}$	$\mathbf{83.72 \pm 0.05}$	$\mathbf{82.69 \pm 0.31}$	$\mathbf{82.66 \pm 0.18}$	$\mathbf{82.52 \pm 0.19}$	$\mathbf{75.99 \pm 1.61}$	$\mathbf{75.00 \pm 2.24}$	$\mathbf{74.56 \pm 2.05}$

Adam optimizer Kingma and Ba [2014]. In all datasets, each client trains for $E = 1$ epoch at the local update with a learning rate of $\eta_l = 0.01$. In our proposed FEDEVE , we set the global learning rate $\eta_g = 1$ for all experiments.

Baselines. To evaluate the performance of FEDEVE , we compare it with several state-of-the-art FL methods: 1) The vanilla FL method FEDAVG McMahan et al. [2017], which is a widely used method for FL; 2) A client-side FL method FEDPROX Li et al. [2020b], which improves the model aggregation by adding a proximal term to the local update; 3) A server-side FL method FEDAVGM Hsu et al. [2019], which adapts the momentum in FL optimization; 4) A server-side FL method FEDOPT Reddi et al. [2020], which introduces adaptive optimization methods in FL. See more experimental details in the Appendix E.

3.2 Analysis

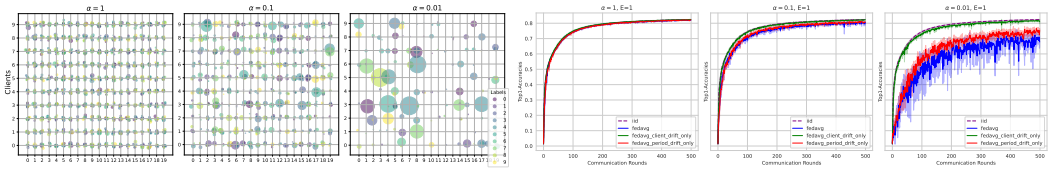


Figure 3: **Visualization of period drift and its impact on performance (a) Visualization of period drift** The color of the scatter points represents different classes, and the size denotes the number of samples of a given class on a particular client. **(b) Visualization of its impact on performance** It is revealed in cross-device FL when data is rather non-iid, period drift has a greater effect than client drift. Appendix E.2 for setting details.

Visualizing the period drift and its impact. Figure 3 (a) visualizes the data distributions of these sampled clients. Client drift arises due to the shift in label distribution among sampled clients ***within a single round***, while period drift results from the shift in the data distribution of participating clients ***across different rounds***. The scatter points' size and distribution grow more diverse both within and across communication rounds as the value of α decreases (indicating increasing non-iid). The implications of these drifts on the global model's convergence are presented in Figure 3 (b). Utilizing the vanilla FEDAVG algorithm for illustration, we experimented with four settings: 1) FEDAVG with iid data; 2) FEDAVG experiencing only period drift; 3) FEDAVG subject to only client drift; and 4) FEDAVG impacted by both drifts (See appendix E.2 for detailed settings). As heterogeneity intensifies, the effects of both drifts become evident. Specifically, in a highly non-iid environment ($\alpha = 0.01$), FEDAVG affected only by client drift yields results akin to the iid setting. In contrast, FEDAVG influenced solely by period drift significantly disrupts the stability and convergence of the FL process. The combination of both drifts results in the poorest performance, underlining that in cross-device FL, period drift poses a more considerable challenge to model convergence than client drift.

The performance of FEDEVE . We evaluate our algorithm on real-world datasets and compare it with the relevant state-of-the-art methods in Tables 1 and 2. We conducted simulations on three datasets: FEMNIST, CIFAR-100, and MovieLens. The FEMNIST and Movielens datasets have a naturally-arising client partitioning setting in real-world FL scenarios, making them highly representative. For FEMNIST and CIFAR-100 datasets, each of the datasets includes three non-iid settings, established through the Dirichlet distribution partition method [Hsu et al., 2019]. *Generally, the results show that our proposed algorithm,* FEDEVE *, consistently outperforms the baselines, and the performance gains are more dominant in more non-iid settings ($\alpha = 0.01$).* We also conduct experiments with different local epochs (E), please refer to the Figure 3.2 for the setting details. Also, our FEDEVE has more leading advantages in RS experiments, indicating its large potential in real-world industrial applications. Our method can better utilize the server-side adaptation through the Bayesian filter's predict-observer framework. Besides, it is important to note that our method does not introduce other

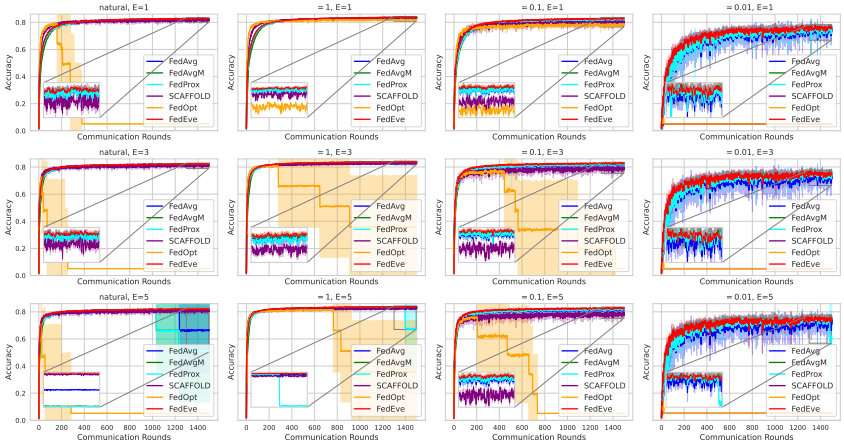


Figure 4: **Accuracies with different α and E**

hyperparameters while these baselines have multiple hyperparameters to tune, which means that our FEDEVE is more flexible and advantageous in real-world practices.

Performace of FEDEVE on FEMNIST with different local epochs. Table 3 showcases the results on the FEMNIST dataset across various methods, specifically: FEDAVG , FEDAVGM , FEDPROX , SCAFFOLD , FEDOPT , and FEDEVE , with different settings of parameters α and E. FEDEVE method stands out consistently as the superior approach across all configurations. This consistent performance signifies that FEDEVE is a potent and reliable method for the FEMNIST dataset across the tested configurations. The performance drop of SCAFFOLD in specific experiments may be attributed to two primary reasons: Staleness of control variate: SCAFFOLD mandates that each client maintain a control variant. However, given the large number of clients and the fact that only a limited subset is chosen for training during each communication round, most control variants remain outdated. As a result, they fail to effectively correct the bias in local updates. This point was also reported in FedOpt [Reddi et al., 2020]. Excessiveness of correction: Upon detailed inspection of our experiments, we discerned that the training of SCAFFOLD tends to fail when there exists a client with more substantial data than others. This stems from the fact that the fixed batch size and training epoch will result in more local updates in the clients with more data, but it will be corrected by the same control variant in SCAFFOLD. Excessive corrections drive the model further from the optimal point, resulting in the divergence of the model. We reckon that the poor performances of FedOpt in some settings primarily result from period drift. Period drift impedes FedOpt's adaptivity across rounds. FedOpt tailors the learning rates of individual weights by accumulating past gradients' squares. However, with the ever-shifting optimization objectives in each communication round (period drift), these rate adjustments become misaligned for subsequent updates, thereby skewing model training. It is validated in the experiments that FedOpt fails on FEMNIST with natural and heterogeneity, where period drift is more dominant (more client number, more non-i.i.d. data).

4 Conclusion

In this work, we conducted a comprehensive exploration of the impact of client drift and period drift on the performance of cross-device FL, discovering that period drift can be particularly harmful as data heterogeneity increases. To solve this challenge, we introduced a novel predict-observe framework and a method, FedEve, that views these drifts as noise associated with prediction and observation. By integrating these two sources in a principled way through a Kalman filter-based approach, we provided a better estimation of model update steps, reducing variance and improving the stability and convergence speed of FL. Our theoretical analysis showed that FedEve can effectively bound the variance of model updates, while extensive empirical evaluations demonstrated that it significantly outperforms alternative methods across different tasks and datasets. We believe our framework opens up new possibilities for developing more robust and efficient FL algorithms.

References

Barak Battash and Ofir Lindenbaum. Revisiting the noise model of stochastic gradient descent. *arXiv preprint arXiv:2303.02749*, 2023.

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

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR, 2022.

Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. A general theory for client sampling in federated learning. In *Trustworthy Federated Learning: First International Workshop, FL 2022, Held in Conjunction with IJCAI 2022, Vienna, Austria, July 23, 2022, Revised Selected Papers*, pages 46–58. Springer, 2023.

Yongxin Guo, Tao Lin, and Xiaoying Tang. Towards federated learning on time-evolving heterogeneous data. *arXiv preprint arXiv:2112.13246*, 2021.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. *arXiv:1910.06378 [cs, math, stat]*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. *arXiv:1812.06127 [cs, stat]*, 2020b.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Jarl Waldemar Lindeberg. Eine neue herleitung des exponentialgesetzes in der wahrscheinlichkeit-srechnung. *Mathematische Zeitschrift*, 15(1):211–225, 1922.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18:1–35, 2017.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv:1602.05629 [cs]*, 2017.

Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. Fedfast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1234–1242, 2020.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

Zhenyu Sun, Xiaochun Niu, and Ermin Wei. Understanding generalization of federated learning via stability: Heterogeneity matters. *arXiv preprint arXiv:2306.03824*, 2023.

Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, pages 10367–10376. PMLR, 2020.

Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. Understanding short-horizon bias in stochastic meta-optimization. *arXiv preprint arXiv:1803.02021*, 2018.

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=wXgk_iCiYGo`.

Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. Federated Learning with Unbiased Gradient Aggregation and Controllable Meta Updating. *CoRR*, abs/1910.08234, 2019.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. *Advances in neural information processing systems*, 32, 2019.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated Learning with Non-IID Data. *arXiv:1806.00582 [cs, stat]*, 2018.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018.

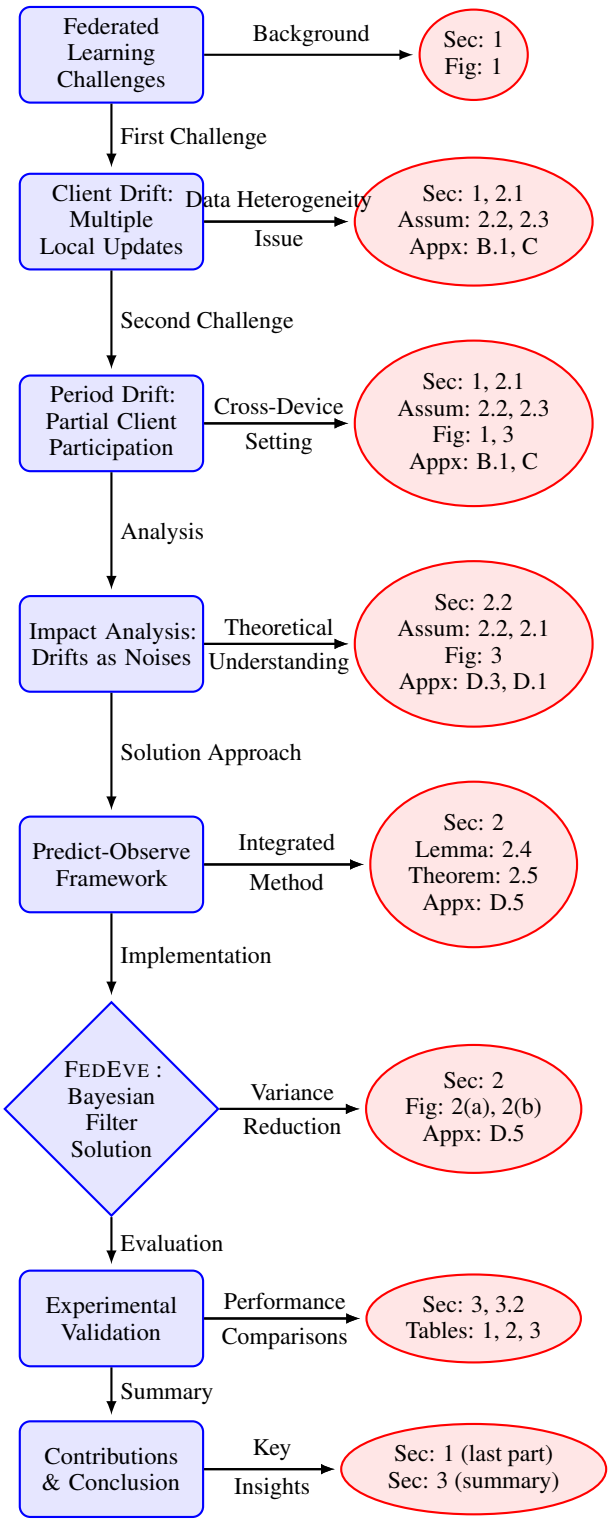
Chen Zhu, Zheng Xu, Mingqing Chen, Jakub Konečnỳ, Andrew Hard, and Tom Goldstein. Diurnal or nocturnal? federated learning of multi-branch networks from periodically shifting distributions. In *International Conference on Learning Representations*, 2022.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning*, pages 7654–7663. PMLR, 2019.

Liu Ziyin, Kangqiao Liu, Takashi Mori, and Masahito Ueda. Strength of minibatch noise in sgd. *arXiv preprint arXiv:2102.05375*, 2021.

A Appendix

Reading Guide



This reading guide provides a comprehensive roadmap to navigate through our paper on addressing the challenges in Federated Learning (FL).

We begin by introducing the fundamental challenges of FL in Section 1, illustrated in Figure 1. Here, we identify two critical issues: Client Drift and Period Drift.

Client Drift occurs due to multiple local updates and data heterogeneity, detailed in Sections 1 and 2.1. This concept is formalized in Assumptions 2.2 and 2.3, with further explanations in Appendices B.1 and C.

Period Drift, a less studied but equally important challenge, arises from partial client participation in cross-device settings. This phenomenon is thoroughly discussed in Sections 1 and 2.1, visually represented in Figures 1 and 3, and conceptually explained in Appendices B.1 and C.

The impact of these drifts on FL performance is analyzed in Section 2.2, where we formally define them in Definition 2.1 and model them as noise under Assumption 2.2. The detrimental effects are visualized in Figure 3, with additional theoretical analysis in Appendix D.3 and justification of the noise assumption in Appendix D.1.

To address these challenges, we propose a Predict-Observe Framework in Section 2. This framework leverages Lemma 2.4 and Theorem 2.5 to establish a principled approach for integrating predictions and observations, with detailed Bayesian derivations in Appendix D.5.

Our specific solution, FEDEVE , implements this framework using a Bayesian filter. The methodology is presented in Section 2 with illustrations in Figures 2(a) and 2(b), demonstrating how it achieves variance reduction through the fusion of predictions and observations.

We validate our approach through extensive experiments in Sections 3 and 3.2. The results, presented in Tables 1, 2, and 3, demonstrate that FEDEVE consistently outperforms state-of-the-art methods, particularly in highly heterogeneous settings.

The paper concludes by summarizing our key contributions in the final part of Section 1 and providing a comprehensive overview of our findings in Section 3.

This guide is designed to facilitate a clear understanding of our research, highlighting the logical progression from identifying challenges to proposing and validating solutions in the context of Federated Learning.

B Related works

There are many works that have attempted to address the non-iid problem in federated learning. FEDAVG , first presented by McMahan et al. [2017], has been demonstrated to have issues with convergence when working with non-iid data. Zhao et al. [2018] depict the non-iid trap as weight divergence, and it can be reduced by sharing a small set of data. However, in traditional federal setting, data sharing violates the principle of data privacy. Karimireddy et al. [2021] highlight the phenomenon of "client drift" that occurs when data is heterogeneous (non-iid), and uses control variates to address this problem. However, using Scaffold in cross-device FL may not be effective, as it requires clients to maintain the control variates, which may become outdated and negatively impact performance. Li et al. [2020b] propose FedProx that utilizes a proximal term to deal with heterogeneity.

In addition to these works, some research has noticed the presence of period drift, but have not specifically addressed it in their analysis. For example, Cho et al. [2022], Fraboni et al. [2023] investigate the problem of biased client sampling and proposes an sampling strategy that selects clients with large loss. However, active client sampling can potentially alter the overall data distribution by having unrandom clients participation, which can raise concerns about fairness. Similarly, Yao et al. [2019] propose a meta-learning based method for unbiased aggregation, but it requires training the global model on a proxy dataset, which may not be feasible in certain scenarios where such a dataset is not available. Zhu et al. [2022] observe that the data on clients have periodically shifting distributions that changed with the time of day, and model it using a mixture of distributions that gradually shifted between daytime and nighttime modes. Guo et al. [2021] study the impact of time-evolving heterogeneous data in real-world scenarios, and solve it in a framework of continual learning. Although these two papers define similar terms, they focus on the case of client data changing over time. However, in this paper, we find that even if the distribution of client data remains unchanged, period drift can seriously affect the convergence of FL.

B.1 The concepts of *Drifts*

To distinguish between the concepts of period drift and client drift, we conduct a thorough analysis of the entire FL process, as shown in the upper part of Fig 5. We utilize a chain of sampling (from left to right) to identify the errors introduced at each stage of FL. We aim to optimize an ideal objective $f(x)$ to obtain the ideal optima x^* under the assumption of infinite data.. However, in the real world, we can only optimize $f_{\mathcal{N}}(x)$ over a finite training set $\mathcal{D}_{\mathcal{N}}$ to obtain the optima $x^*_{\mathcal{N}}$. Sampling finite data from infinite data introduces the first variance $\sigma^2_{\mathcal{N}}$, also known as the generalization error, which is small because we usually assume that the training data and the overall data are IID in the context of machine learning.

In FL, directly optimize $f_{\mathcal{N}}(x)$ is not feasible due to the non-IID distribution of data across different clients, requiring distributed training instead. In cross-device FL, only a small subset of clients is sampled each round for training. However, due to data heterogeneity, the data distribution $\mathcal{D}_{\mathcal{S}}$ of the sampled clients differs from the full client set $\mathcal{D}_{\mathcal{N}}$, leading to different optimization objectives $f_{\mathcal{S}}(x)$ and $f_{\mathcal{N}}(x)$, as well as a significant deviation between the optima $x^*_{\mathcal{S}}$ and $x^*_{\mathcal{N}}$. This difference introduces the second variance $\sigma^2_{\mathcal{S}}$ referred to as "period drift", which can be substantial in scenarios with strong data heterogeneity and significantly slow down model convergence, as depicted in Fig 3 in the main paper. Additionally, these sampled clients independently optimize their local objectives $f_k(x)$ on their respective datasets $\mathcal{D}_k$. Due to the presence of data heterogeneity, the optima x^*_k and $x^*_{\mathcal{S}}$ obtained by different clients also differ, introducing the third variance σ^2_k, which is "client drift". On this chain of sampling, each sampling introduces an error, each affecting the model's convergence and performance.

To clarify the concepts of "period drift" and "client drift" in FL, we refine the commonly cited inequality $\frac{1}{S}\sum_{i\in S} x^*_i \neq x^*$. We propose a more nuanced formulation: $x^* \neq x^*_{\mathcal{N}} \neq x^*_{\mathcal{S}} \neq \bar{x} = \frac{1}{|\mathcal{S}|}\sum x^*_k \neq x^*_k$. In FL, the goal is to estimate the global optimum $x^*_{\mathcal{N}}$—or ideally, x^*—through the weighted average $\bar{x} = \frac{1}{|\mathcal{S}|}\sum x^*_k$. However, this average $\bar{x}$ often only approximates $x^*_{\mathcal{S}}$ and falls short of estimating the true global optimum $x^*_{\mathcal{N}}$ due to significant distributional discrepancies between $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{N}}$. This variance is highlighted in the lower part of Figure 5 and illustrates the challenges in drawing reliable inferences about the global data distribution from locally sampled subsets.

Given that we only have data from current samples and lack comprehensive information about the broader sampling framework, it's clear that attributing the inequality $\frac{1}{S}\sum_{i\in S} x^*_i \neq x^*$ solely to client drift is misleading. This inequality encompasses effects beyond local updates, including

Figure 1: A Chain of Sampling

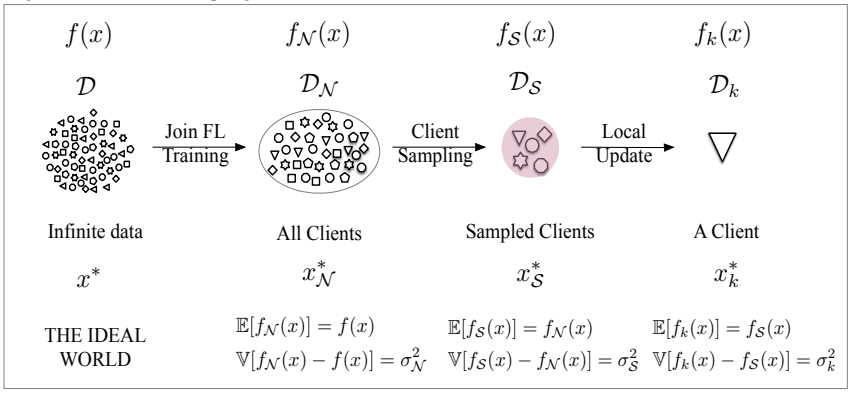


Figure 2: A Gap from $x^*_{\mathcal{S}}$ to $x^*_{\mathcal{N}}$

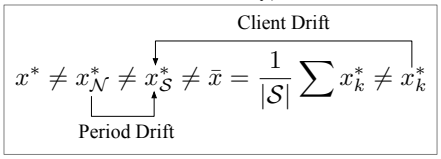


Figure 5: **Illustration of period drift and client drift in the FL process.**

generalization error $\sigma^2_{\mathcal{N}}$ and client sampling $\sigma^2_{\mathcal{S}}$. A precise representation of client drift would therefore be $\bar{x} = \frac{1}{|\mathcal{S}|}\sum x^*_k \neq x^*_{\mathcal{S}}$. An illustrative example of this is when each client performs a single gradient update step; there is effectively no client drift, as the average of these updates reflects the gradient of the current batch. Yet, the disparity $x^* \neq x^*_{\mathcal{N}} \neq x^*_{\mathcal{S}} = \bar{x} = \frac{1}{|\mathcal{S}|}\sum x^*_k$ still persists, and the influence of period drift remains. Hence, period drift is fundamentally distinct from client drift and operates independently.

C The difference between *Drift* in FL and the *Noise* of centralized SGD

Federated learning possesses unique characteristics compared to traditional centralized optimization, such as client sampling, multiple local epochs, and non-iid data distribution. In this context, drifts in federated learning can be viewed as noises to the training dynamics. More specifically, period drift, originating from non-iid data and partial participation (only a subset of clients participate in each round), can be likened to the implementation of a mini-batch technique in the full-batch gradient descent of centralized training [Ziyin et al., 2021]. Here, the distinction is that in centralized optimization issues, each batch is iid, and each batch's data distribution closely mirrors the overall distribution, albeit with a noise component. This noise becomes remarkably pronounced in federated learning, given that client data is non-iid. Client drift, arising from non-iid data and multiple local updates (where each client runs local SGD with multiple steps), is a well-structured noise [Lin et al., 2018]. Due to the combined impact of client drift and period drift, the situation can be perceived as adding a noise term to the original model (or gradient). The research outlined in this paper based on the difference between the drifts in federated learning and the noises in centralized SGD.

C.1 Visualization of difference between Period Drift and noise in SGD

In this section, we provide a visualization to illustrate the fundamental difference between period drift in cross-device federated learning and the noise introduced by stochastic gradient descent (SGD). While both phenomena introduce stochasticity into the optimization process, their nature and impact on model convergence differ significantly.

Figure 6 depicts this distinction by comparing the gradient directions across different communication rounds in federated learning. Period drift arises from the systematic shifts in client data distributions between communication rounds, creating consistent biases in gradient direction for extended periods. In contrast, SGD noise exhibits more random fluctuations that tend to average out over time.

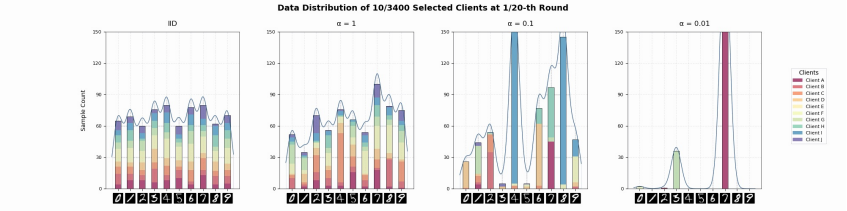


Figure 6: **Visualization of the difference between period drift and SGD noise**

This visualization highlights why period drift poses a more significant challenge to federated learning than standard SGD noise. While SGD noise is often assumed to have zero mean and can be mitigated through increased batch sizes or adaptive optimization methods, period drift introduces differet kind of noise that depends on data heterogeneity that require specialized techniques like our proposed FEDEVE algorithm to effectively address. The systematic nature of period drift makes it particularly problematic in highly heterogeneous cross-device settings, as demonstrated in our experimental results in Section 3.

C.2 A simple analysis of data heterogeneity and period drift

Let us analyze how data heterogeneity impacts the data distribution difference between sampled clients and the overall population. We begin by examining the expectation of the sampled distribution P_S. Since clients are randomly selected, the expectation equals the overall data distribution:

$$\mathbb{E}[P_S] = \mathbb{E}\left[\frac{1}{|S|}\sum_{i\in S} D_i\right] = \frac{1}{|S|}\sum_{i=1}^{N} D_i = P \tag{17}$$

Next, we calculate the variance of P_S. Using the linearity of variance and independence:

$$\begin{aligned}\text{Var}(P_S) &= \text{Var}\left(\frac{1}{|S|}\sum_{i\in S} D_i\right) \\ &= \frac{1}{|S|^2}\sum_{i\in S}\text{Var}(D_i) \\ &= \frac{1}{|S|^2}\sum_{i\in S}\frac{1}{M}\sum_{j=1}^{M}(p_{ij}-p_j)^2\end{aligned} \tag{18}$$

Since each client in S is independently and identically distributed, taking the expectation yields:

$$\mathbb{E}[\text{Var}(P_S)] = \frac{1}{|S|}\cdot\frac{1}{N}\sum_{i=1}^{N}\text{Var}(D_i) = \frac{H}{|S|} \tag{19}$$

Finally, we derive the expectation of the data distribution difference between selected clients:

$$\begin{aligned}\mathbb{E}[D_S] &= \mathbb{E}[\text{Var}(P_S, P)] \\ &= \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M}\left(\frac{1}{|S|}\sum_{i\in S} p_{ij} - p_j\right)^2\right] \\ &= \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\left[\left(\frac{1}{|S|}\sum_{i\in S} p_{ij} - p_j\right)^2\right] \\ &= \text{Var}(P_S) \\ &= \frac{H}{|S|}\end{aligned} \tag{20}$$

This result demonstrates that the expected data distribution difference among selected clients is directly proportional to the system's data heterogeneity H and inversely proportional to the number of selected clients $|S|$. This finding has two important implications:

1. As data heterogeneity H increases, the expected data distribution difference among selected clients also increases. In the IID case where $H = 0$, we have $\mathbb{E}[D_S] = 0$, indicating that any client selection will match the overall distribution.
2. As the number of selected clients $|S|$ increases, the expected data distribution difference decreases, suggesting that increasing participation can help mitigate the impact of data heterogeneity.

D The relationship between period drift and client drift

The relationship between period drift and client drift can be understood through their distinct yet interrelated effects on the FL optimization process. First, there is a clear sequential nature to these drifts - period drift occurs first in the optimization process, as it is determined by client selection at the beginning of each round. Client drift then follows as a consequence of local updates performed by the selected clients. This sequential relationship means that period drift can influence the magnitude and direction of subsequent client drift.

When data heterogeneity is high, period drift leads to biased optimization objectives for each round. This bias can be amplified by client drift during local updates, as clients further optimize towards their local objectives. The combination of these drifts results in increased variance in model updates, slower convergence rates, and potential divergence from the global optimum. Despite their potentially harmful effects, these drifts can sometimes compensate for each other. Period drift may help prevent over-fitting to the data distribution of currently selected clients, while client drift can help explore the local optimization landscape more thoroughly. Additionally, the interaction between these drifts can create a natural regularization effect. The magnitude of these drifts operates at different scales, which can be expressed mathematically as:

$$\begin{aligned} \text{Period Drift} &\propto \frac{H}{|S|} \\ \text{Client Drift} &\propto K \cdot H \end{aligned} \tag{21}$$

where H is the heterogeneity measure, $|S|$ is the number of selected clients, and K is the number of local steps. Understanding this relationship is crucial for designing effective FL algorithms that can handle both types of drift simultaneously, as addressing one type of drift in isolation may inadvertently exacerbate the other.

D.1 The justification of Gaussian-like noise assumption 2.2

Assumption D.1 (2.2)**.** *The aggregated model parameters on the server w_{server}, can be represented as the sum of the optimal parameters w^* and a drift (noise) that follows a normal distribution $w_{drift} \sim \mathcal{N}(0, \sigma^2_{drift})$:*

$$w_{server} = w^* + w_{drift} \leftarrow noise, \tag{22}$$

where w^* represents the optimal parameters obtained through the use of stochastic gradient descent (SGD), w_{drift} represents the noise term caused by factors such as client sampling, multiple local epochs, and non-iid data distribution that we assume a normal distribution, and w_{server} represents the aggregated model parameters also follows a normal distribution $w_{server} \sim \mathcal{N}(w^*, \sigma^2_{drift})$, with the expectation of the aggregate model parameters being equal to the optimal parameters, i.e. $\mathbb{E}[w_{server}] = w^*$.

In this paper, we conceptualize the aggregated model parameters on the server as the summation of optimal parameters and a certain drift (or noise), represented as: $w_{server} = w^* + w_{drift}$. We also assume that w_{drift} is subject to a normal (Gaussian-like) distribution, and justify this assumption by demonstrating its prevalence, and explaining it in FL.

From a historical standpoint, modeling noise in dynamic systems as a Gaussian-like distribution is a widely accepted practice. This dates back to [Kramers, 1940], and many studies analyzing Stochastic Gradient Descent (SGD) optimization have emphasized the Gaussian nature of noise on gradients or parameters [Mandt et al., 2017, Zhu et al., 2019, Simsekli et al., 2019, Ziyin et al., 2021]. This assumption of Gaussianity for SGD noise is justified by Wu et al. [2020], which guarantees the

SGD noise's convergence to a specific infinite divisible distribution. This falls under the Gaussian class provided the noise's second moment is finite (as per Lindeberg's condition). While it has been proposed that the noise in SGD might be better represented by $S\alpha S$ noise [Simsekli et al., 2019], this idea has been challenged and redirected back to the earlier proposed Gaussian noise model [Xie et al., 2021, Battash and Lindenbaum, 2023].

We further elucidate the occurrence of Gaussian noise in the context of FL. The Lindeberg-Feller Central Limit Theorem (CLT) [Lindeberg, 1922] plays a key role in explaining the prevalence of the normal distribution. It posits that the sum (or average) of random variables gravitates towards a normal distribution (no need to assume iid of these random variables themselves), irrespective of the individual distributions of these variables. In FL, the emergence of noise can be attributed to partial participation (referred to as period drift) and multiple local updates (referred to as client drift). The drifted model that we observe, w_{server}, is typically the result of the combination of these factors, making the normal distribution an apt model for characterizing the noise.

The impact of noise on generalization In order to investigate the effect of the deviation on performance in FL, we utilize a regression optimization objective as in previous studies, such as [Zhang et al., 2019] and [Wu et al., 2018]:

$$\hat{\mathcal{L}}(w) = \frac{1}{2}(w + w_{drift})^T A(w + w_{drift}),$$

where $w_{drift} \sim \mathcal{N}(0, \sigma^2)$ is the drift caused by the characteristics of FL. Therefore, the generalization error can be formulated as:

$$\begin{aligned}\mathcal{L}\left(w^t\right) &= \mathbb{E}\left[\hat{\mathcal{L}}\left(w^t\right)\right] = \frac{1}{2}\mathbb{E}\left[\sum_i a_i\left(w_i^{t^2} + \sigma_i^2\right)\right] \\ &= \frac{1}{2}\sum_i a_i\left(\mathbb{E}\left[w_i^t\right]^2 + \mathbb{V}\left[w_i^t\right] + \sigma_i^2\right),\end{aligned}$$

where A is the matrix of quadratic form of the MSE loss function, and a_i is the elements of A. As results, the generalization error can also be decomposed into three components: bias, variance, and noise. The noise component in FL context is further influenced by factors such as client sampling, multiple local epochs, and non-iid data distribution, leading to a much larger overall generalization error compared to centralized SGD. This formulation reveals the reason of why FL usually performs worse than centralized training. Thus, our goal is to reduce the variance of drift σ^2 in order to improve both the convergence and performance of the model.

D.2 The justification of independence assumption 2.3 of client drift and period drift

Assumption D.2 (2.3)**.** *The initialization model parameters are independent of all period drifts Q_t and client drifts R_t at each communication round, that is $w_0 \perp Q_0, Q_1, \cdots, Q_t$ and $w_0 \perp R_0, R_1, \cdots, R_t$.*

This assumption can be justified from various perspectives, demonstrating its reasonableness:

- **Independence of the initial model parameters from other noise variables:** This assumption suggests that there is no direct relationship between the initial model parameters (w_0) and period drifts or client drifts. This is a reasonable assumption since initial model parameters are typically determined prior to training and hence are not influenced by any noise processes.
- **Independence of client drifts between each communication round:** According to this assumption, client drifts ($R_0, R_1, \cdots, R_t$) in different communication rounds are independent of each other. The client drift is influenced by data heterogeneity and multiple local updates. A higher degree of data heterogeneity and an increased number of local updates can result in greater client drift. Each client has its own fixed client drift[Guo et al., 2021], and the client drift in each communication round doesn't impact other rounds, the assumption of client drift independence is reasonable.
- **Independence of period drifts between each communication round:** This assumption contends that the period drifts ($Q_0, Q_1, \cdots, Q_t$) across different communication rounds are independent. Period drift is influenced by data heterogeneity and client sampling. Though period drift may be caused by biased client sampling due to factors like time and geographic locations, leading to a dependence between period drifts in different communication rounds, this paper considers the case where random client sampling occurs in each communication round. Here, one round of client sampling doesn't affect others, thus rendering the independence of period drift reasonable.

- **Independence between period drifts and client drifts at each communication round:** This assumption argues that the client drifts $(R_0, R_1, \cdots, R_t)$ and the period drifts $(Q_0, Q_1, \cdots, Q_t)$ at each communication round are independent. While both client drift and period drift are influenced by data heterogeneity, they are conditionally independent given the heterogeneous data. We offer a causal graph to depict their relationships:
multiply local updates $\longrightarrow$ client drift $\longleftarrow$ data heterogeneity $\longrightarrow$ period drift $\longleftarrow$ client sampling
This graph indicates that data heterogeneity is a common cause of client drift and period drift, and varying levels of data heterogeneity result in different magnitudes of client drift and period drift. However, given that the heterogeneous data is constant across clients (i.e., given D), we can express P(client drift, period drift|D) = P(client drift|D) * P(period drift|D). Indeed, conditioning on a given heterogeneous data set is a fundamental assumption in Federated Learning.

Empirical Analysis To verify that both period drift and client drift adhere to the Gaussian assumption, we design this experiment by means of randomly selecting a model parameter and tracing its period drift and client drift across the subsequent 1500 rounds of training. During this process, we collected the global optima $x^*_{\mathcal{N}}$ of global objective on the whole training dataset for each communication round, as well as the optima $x^*_{\mathcal{S}}$ on the dataset of sampled clients for each communication round, and the local optima x^*_k of the dataset of a single client, respectively. The period drift is represented by $x^*_{\mathcal{S}} - x^*_{\mathcal{N}}$, while the client drift is depicted by $x^*_k - x^*_{\mathcal{S}}$. The sampling results are visualized using histograms. Experiments were conducted with alpha values in the scope $\alpha = [1, 0.1, 0.01]$, as illustrated in Fig 7. We also employ the 'normaltest' function from 'scipy.stats', utilizing the *D'Agostino-Pearson Test* to determine if a sample deviates from a normal distribution. Here, a p-value > 0.05 indicates conformity to a normal distribution. The experimental results showed that both period drift and client drift follow a normal distribution, confirming the reliability of the Gaussian distribution assumption.

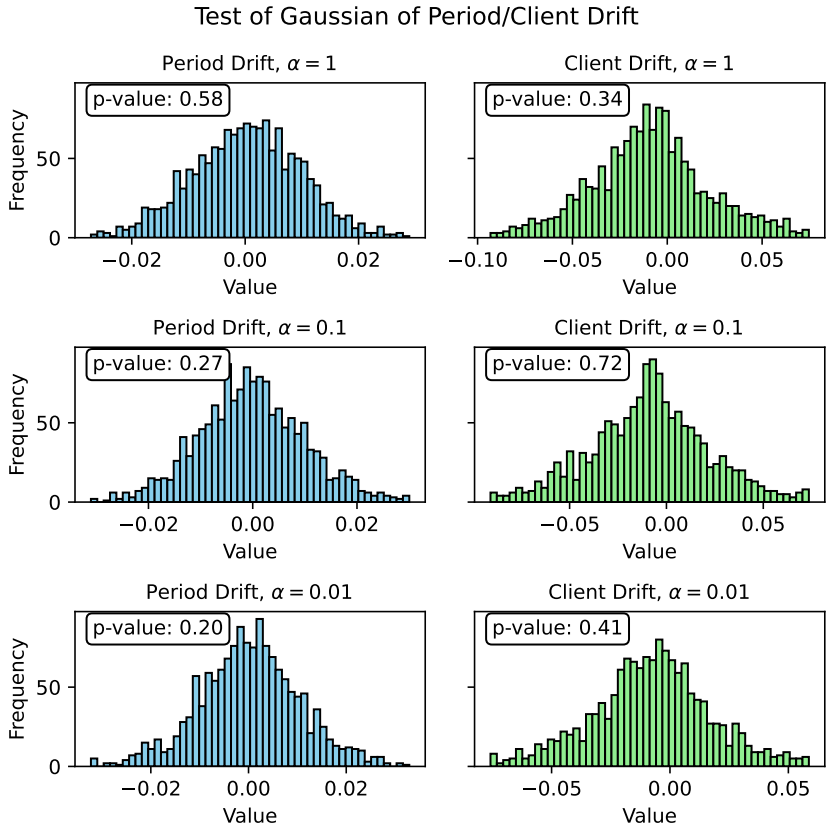


Figure 7: **Empirical justification of Gaussian distribution assumption.**

D.3 The Independence of Noise

Lemma D.3 (2.4). *(Independence of Noise). the noise present in the prediction and observation at each communication round is independent of the current state of the model, specifically,* $w_t \perp Q_t$ *and* $w_t \perp R_t$.

To prove this, we will first need to understand the relationship between the variables w_{t+1}, Q_t, and R_t. In the given context, w_{t+1} represents the state of the model at a particular communication round (say, round $t+1$). It is affected by the values of the period drift (Q_t) and client drift (R_t) at the previous round. This is represented by the state transfer function $w_{t+1} = T_t(w_t, Q_t, R_t)$. However, this relationship does not imply that w_{t+1} is dependent on Q_t or R_t. To see why, let's look at how w_t is formed. Using the state transfer function, we can express w_t as:

$$\begin{aligned} w_t &= T_{t-1}(w_{t-1}, Q_{t-1}, R_{t-1}), \\ w_{t-1} &= T_{t-2}(w_{t-2}, Q_{t-2}, R_{t-2}), \\ &\vdots \\ w_2 &= T_1(w_1, Q_1, R_1), \\ w_1 &= T_0(w_0, Q_0, R_0), \end{aligned} \tag{23}$$

From this chain of equations, it is evident that w_t is not only a function of the current round's period drift Q_t and client drift R_t, but also of their past values and the past values of w_t itself. In a more generalized form, we can write this as: $w_t = T(w_0, Q_0, Q_1, \cdots, Q_{t-1}, R_0, R_1, \cdots, R_{t-1})$. Assumption 2.3 states that the period drift and client drift are independent of each other at each communication round and also independent of the initial model parameters. That is, $w_0 \perp Q_0 \perp Q_1 \perp \cdots Q_t \perp R_0 \perp R_1 \perp \cdots R_t$. This assumption implies that the previous states of w_t (w_{t-1}, w_{t-2}, and so on) do not have any influence on the current values of Q_t and R_t. In other words, the noise present at each round is independent of the model's current state. Thus, w_t is independent of Q_t and R_t, denoted as $w_t \perp Q_t \perp R_t$. Therefore, it can be concluded that the noise present in the prediction and observation at each communication round is indeed independent of the current state of the model, thereby confirming the independence of noise. This statement about the independence of noise is significant because it confirms that the noise encountered during each communication round does not affect the model's state. This means that the model is robust and not affected by random perturbations, which is a desirable property in any machine learning model.

D.4 Analysis of Kalman Gain in FEDEVE

We conducted an in-depth analysis of the Kalman Gain K of FedEve under various experimental settings, incorporating four levels of data heterogeneity and various local epochs, as shown in Figure 4. We observed that as data heterogeneity increases (i.e., as the value of α decreases), the Kalman Gain K progressively enlarges. With the rise of data heterogeneity, the period drift starts to play a more dominant role. In this context, the primary role of Kalman Gain K is to adjust the weights between global and local updates, as depicted by Equation (15b) and (15c). Further, according to Equation (15d), the model update tends to trust local updates more, stabilizing the optimization process. For varied counts of local updates, the relative change in Kalman Gain K is marginal. This is primarily because, in cross-device FL, the client drift is not a pivotal or dominant factor, which aligns with our prior analysis in Figure 3.

D.5 Model Update with Bayesian filter

D.5.1 Bayesian filter

This section begins by describing the main idea behind the approach: the combination of prediction and observation models using a Bayesian filter, which is a statistical tool for estimating an unknown probability density function (PDF) based on observations. The prediction process is explained using the concept of cumulative distribution functions (CDFs), which are functions giving the probability that a random variable is less than or equal to a certain value. The prediction process is characterized

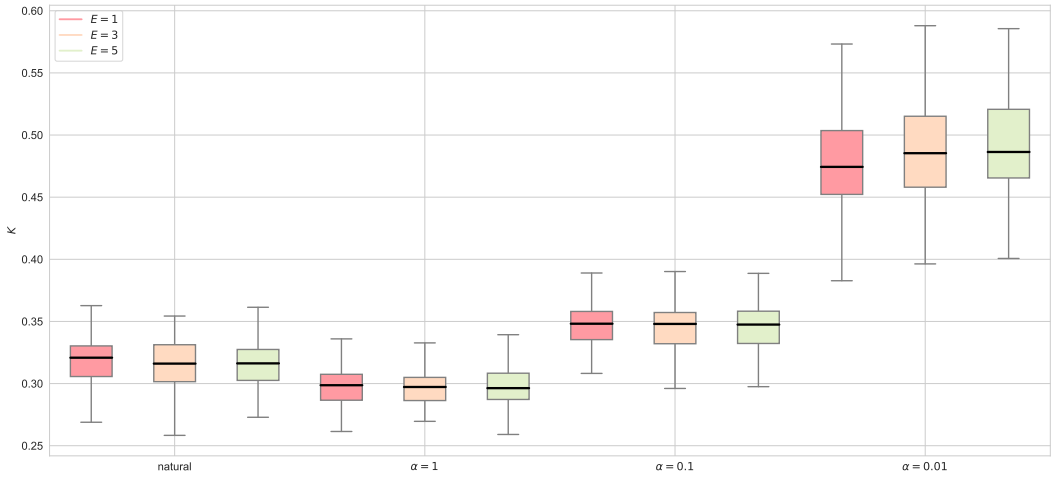


Figure 8: **Boxplots for Kalman Factors**

by the distribution:

$$
\begin{aligned}
F^{-}_{\hat{w}_{t+1}}(w) &= P\left(\hat{w}_{t+1} \leq w\right) \\
&= \sum_{u=-\infty}^{w} P\left(\hat{w}_{t+1} = u\right) \\
&= \sum_{u=-\infty}^{w} \sum_{v=-\infty}^{+\infty} P\left(\hat{w}_{t+1} = u \mid w_t = v\right) P\left(w_t = v\right) \\
&= \sum_{u=-\infty}^{w} \sum_{v=-\infty}^{+\infty} P\left[\hat{w}_{t+1} - f\left(w_t\right) = u - f(v) \mid w_t = v\right] P\left(w_t = v\right) \\
&= \sum_{u=-\infty}^{w} \sum_{v=-\infty}^{+\infty} P\left[Q_t = u - f(v) \mid w_t = v\right] P\left(w_t = v\right) \quad \Rightarrow \text{Prediction Equation} \\
&= \sum_{u=-\infty}^{w} \sum_{v=-\infty}^{+\infty} P\left[Q_t = u - f(v)\right] P\left(w_t = v\right) \quad \Rightarrow \text{Lemma (2.4)} \\
&= \sum_{u=-\infty}^{w} \left\{ \lim_{\epsilon \to 0} \sum_{v=-\infty}^{+\infty} f_{Q_t}[u - f(v)] \cdot \epsilon \cdot f^{+}_{w_t}(v) \cdot \epsilon \right\} \\
&= \sum_{u=-\infty}^{w} \left\{ \lim_{\epsilon \to 0} \int_{-\infty}^{+\infty} f_{Q_t}[u - f(v)] f^{+}_{w_t}(v) \mathrm{d}v \cdot \epsilon \right\} \\
&= \int_{-\infty}^{w} \int_{-\infty}^{+\infty} f_{Q_t}[u - f(v)] f^{+}_{w_t}(v) \mathrm{d}v \, \mathrm{d}u \\
&= \int_{-\infty}^{w} \int_{-\infty}^{+\infty} f_{Q_t}[w - f(v)] f^{+}_{w_t}(v) \mathrm{d}v \, \mathrm{d}w
\end{aligned}
\tag{24}
$$

The first three steps apply the definition of the cumulative distribution function (CDF), which is the sum of probabilities up to a certain point. In the fourth step, the change of variables is used to switch from u to v. The fifth and sixth steps apply Lemma 2.4, which states that the drift is independent of the weights w_t. The last three steps show how to convert the sum to an integral, which is a common method in probability theory for dealing with continuous random variables. Finally, the PDF of the

prediction is obtained by taking the derivative of the CDF, which can be expressed as:

$$f^{-}_{\hat{w}_{t+1}}(w) = \frac{\mathrm{d}F^{-}_{\hat{w}_{t+1}}(w)}{\mathrm{d}w} = \int_{-\infty}^{+\infty} f_{Q_t}[w - f(v)] f^{+}_{w_t}(v)\mathrm{d}v \tag{25}$$

The observation process is also characterized by a PDF. Similar steps are used to manipulate and simplify the expression for the PDF of the observed value w_{t+1}, given the predicted value $\hat{w}_{t+1}$. Specifically:

$$\begin{aligned}
&f_{\tilde{w}_{t+1}|\hat{w}_{t+1}}\left(w_{t+1} \mid w\right) = \lim_{\epsilon \to 0} \frac{F_{\tilde{w}_{t+1}|\hat{w}_{t+1}}\left(w_{t+1}+\epsilon \mid w\right) - F_{\tilde{w}_{t+1}|\hat{w}_{t+1}}\left(w_{t+1} \mid w\right)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{P\left(w_{t+1} \leq \tilde{w}_{t+1} \leq w_{t+1}+\epsilon \mid \hat{w}_{t+1} = w\right)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{P\left[w_{t+1} - h(w) \leq \tilde{w}_{t+1} - h\left(\hat{w}_{t+1}\right) \leq w_{t+1} - h(w) + \epsilon \mid \hat{w}_{t+1} = w\right]}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{P\left[w_{t+1} - h(w) \leq R_t \leq w_{t+1} - h(w) + \epsilon \mid \hat{w}_{t+1} = w\right]}{\epsilon} \Rightarrow \text{Lemma (2.4)} \\
&= \lim_{\epsilon \to 0} \frac{F_{R_t}\left[w_{t+1} - h(w) + \epsilon\right] - F_{R_t}\left[w_{t+1} - h(w)\right]}{\epsilon} \\
&= f_{R_t}\left[w_{t+1} - h(w)\right].
\end{aligned} \tag{26}$$

The first two steps apply the definition of the probability density function (PDF), which is the derivative of the cumulative distribution function (CDF). The third and fourth steps use a change of variables to express the PDF in terms of the difference between the observed and predicted values. The fifth step applies the independence Lemma (2.4) to simplify the conditional probability. The sixth step calculates the limit to arrive at the PDF of the observation process. As a consequence of combining the prediction and observation distributions, a fused model can be obtained, which can be described by its own probability density function (PDF) as:

$$f^{+}_{w_{t+1}}(w) = \eta_t \cdot f_{\tilde{w}_{t+1}|\hat{w}_{t+1}}\left(\tilde{w}_{t+1} \mid \hat{w}_{t+1}\right) \cdot f^{-}_{\hat{w}_{t+1}}(w) = \eta_t \cdot f_{R_t}\left[\tilde{w}_{t+1} - h(\hat{w}_{t+1})\right] \cdot f^{-}_{\hat{w}_{t+1}}(w), \tag{27}$$

where

$$\eta_t = \left[\int_{-\infty}^{+\infty} f_{\tilde{w}_{t+1}|\hat{w}_{t+1}}\left(\tilde{w}_{t+1} \mid \hat{w}_{t+1}\right) f^{-}_{\hat{w}_{t+1}}(w)\mathrm{d}w\right]^{-1} = \left\{\int_{-\infty}^{+\infty} f_{R_t}\left[\tilde{w}_{t+1} - h(\hat{w}_{t+1})\right] f^{-}_{\hat{w}_{t+1}}(w)\mathrm{d}w\right\}^{-1}. \tag{28}$$

The PDF of the fused model is obtained by multiplying the PDFs of the prediction and observation processes, normalized by a factor η_t. The process of updating the fused model, also known as the Bayesian filter, can be summarized in the following steps:

$$\begin{aligned}
f^{+}_{w_t}(w) &\overset{\text{predict}}{\Longrightarrow} f^{-}_{\hat{w}_{t+1}}(w) = \int_{-\infty}^{+\infty} f_{Q_t}[w - f(v)] f^{+}_{w_t}(v)\mathrm{d}v \\
&\overset{\text{observe}}{\Longrightarrow} f^{+}_{w_{t+1}}(w) = \eta_t \cdot f_{R_t}\left[w_{t+1} - h(w)\right] \cdot f^{-}_{\hat{w}_{t+1}}(w),
\end{aligned} \tag{29}$$

where $\eta_t = \left\{\int_{-\infty}^{+\infty} f_{R_t}\left[\tilde{w}_{t+1} - h(\hat{w}_{t+1})\right] f^{-}_{\hat{w}_{t+1}}(w)\mathrm{d}w\right\}^{-1}$. The fused model combines the prediction and observation distributions, and it describes the sequential steps of the Bayesian filter: starting with the PDF at time t, a prediction is made for the PDF at time $t+1$, and then this prediction is updated based on the observation to obtain the PDF at time $t+1$. The final estimation of the parameter can be obtained as a result of these update steps and can be represented as:

$$\hat{w}_{t+1} = E\left[f^{+}_{w_{k+1}}(w)\right] = \int_{-\infty}^{+\infty} w f^{+}_{w_{k+1}}(w)\mathrm{d}w, \tag{30}$$

which is done by calculating the expected value of the PDF at time $t+1$. This is performed by multiplying the parameter w by its probability density and integrating over all possible values of w. The integral provides a single, average value for w weighted by its probability density, which serves as the final estimate of the parameter.

D.5.2 FEDEVE

In this section, we delve into the derivation of the FEDEVE algorithm using the Bayesian filter. The model update process within the FEDEVE algorithm is outlined as follows: Firstly, we calculate the predictive value of the model's parameters ($\hat{w}_{t+1}$) at the next time step using the current parameters (w_t) and the step-size scaled momentum ($\eta_g M_t$):

$$\hat{w}_{t+1} = w_t - \eta_g M_t, \tag{31}$$

The inverse of the variance at time $t+1$ ($\hat{\sigma}^2_{t+1}$) is determined as the sum of the predicted variance at time t (σ_t) and the squared variance associated with the process noise ($\sigma^2_{Q_t}$):

$$\hat{\sigma}^2_{t+1} = \sigma_t + \sigma^2_{Q_t}, \tag{32}$$

The Kalman Gain (K) is computed as the ratio of the inverse of the variance at time $t+1$ to the sum of the inverse of the variance at time $t+1$ and the variance of the observation noise ($\sigma^2_{R_t}$):

$$K = \frac{\hat{\sigma}^2_{t+1}}{\hat{\sigma}^2_{t+1} + \sigma^2_{R_t}}, \tag{33}$$

The momentum for the next time step (M_{t+1}) is obtained by adjusting the current momentum (M_t) based on the difference between the observed value ($\Delta\tilde{w}_t$) and the current momentum:

$$M_{t+1} = M_t + K(\Delta\tilde{w}_t - M_t), \tag{34}$$

The parameters (w_{t+1}) for the next time step are calculated by subtracting the step-size scaled updated momentum from the current parameters:

$$w_{t+1} = w_t - \eta_g M_{t+1}, \tag{35}$$

Finally, the predicted variance for the next time step (σ^2_t) is computed by scaling the inverse of the variance at time $t+1$ by $(1-K)$:

$$\sigma^2_{t+1} = (1-K)\hat{\sigma}^2_{t+1}. \tag{36}$$

A key assumption for this derivation is that the two random variables, A and B, follow a normal distribution. Specifically, A is assumed to be distributed as $\mathcal{N}\left(\mu_A, \sigma^2_A\right)$, and B is assumed to be distributed as $\mathcal{N}\left(\mu_B, \sigma^2_B\right)$. Given these assumptions, it can be mathematically proven that the sum and the product of A and B also follow a normal distribution. In particular, we have:

$$A + B \sim \mathcal{N}\left(\mu_A + \mu_B, \sigma^2_A + \sigma^2_B\right), \tag{37}$$

$$A \times B \sim \mathcal{N}\left(\frac{\mu_A\sigma^2_B + \mu_B\sigma^2_A}{\sigma^2_A + \sigma^2_B}, \frac{\sigma^2_A\sigma^2_B}{\sigma^2_A + \sigma^2_B}\right), \tag{38}$$

In the context of the predict-observe framework and the Bayesian filter, we make a few key assumptions: firstly, w_t is distributed as $\mathcal{N}\left(w_t, \sigma^2_t\right)$; secondly, Q_t is distributed as $\mathcal{N}\left(0, \sigma^2_{Q_t}\right)$; and finally, the momentum M_t is considered a non-random variable. Specializing the prediction function as a linear function, as shown in Equation (13), leads us to the following result:

$$\hat{w}_{t+1} \sim \mathcal{N}\left(w_t - \eta_g M_t, \sigma^2_t + \sigma^2_{Q_t}\right), \tag{39}$$

This is essentially equivalent to the equations (15a) and (15b) in the model update process. Moreover, we set $\sigma^{-2}_{t+1} = \sigma^2_t + \sigma^2_{Q_t}$. The distribution of w_{t+1} is considered as the posterior, which is calculated by applying the Bayesian formula and combining the product of the likelihood and the prior. Here, the likelihood corresponds to the observation, and the prior corresponds to the prediction. The observed $\tilde{w}_{t+1}$ is distributed as follows:

$$\tilde{w}_{t+1} \sim \mathcal{N}\left(\hat{w}_{t+1} - \eta_g\Delta\tilde{w}_t, \sigma^2_{R_t}\right). \tag{40}$$

To calculate the product of the prior and the likelihood, we evaluate the proportionality coefficient $K = \frac{\hat{\sigma}^2_{t+1}}{\hat{\sigma}^2_{t+1}+\sigma^2_{R_t}}$. We can then assert that w_{t+1} also adheres to a normal distribution:

$$\tilde{w}_{t+1} \sim \mathcal{N}\left(w_t - \eta_g((1-K)M_t + K\Delta\tilde{w}_t), (1-K)\hat{\sigma}^2_{t+1}\right), \tag{41}$$

This result serves as a stepping stone for the subsequent round of computation. Consequently, the variance of w_{t+1} is minimized as:

$$\sigma^2_{\text{fused}} = \frac{\hat{\sigma}^2_{t+1}\sigma^2_{R_t}}{\hat{\sigma}^2_{t+1} + \sigma^2_{R_t}}. \tag{42}$$

In summary, the above proof demonstrates how the Bayesian filter can be used to derive the model update process of the FEDEVE algorithm. The predict-observe framework and the feature of normal distribution are key elements in this derivation.

D.6 Pseudo-code

The pseudo-code of FEDEVE is depicted in Algorithm 1.

Algorithm 1 FEDEVE The selected clients are indexed by k; E is the number of local epochs, and η_l is the local learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots, T$ **do**
  $\hat{w}_{t+1} \leftarrow w_t - \eta_g M_t$ as in Equation (12) in the main paper // predict
  $\mathcal{S}_t \leftarrow$ randomly select $|\mathcal{S}_t|$ clients
  for each client $k \in \mathcal{S}_t$ **in parallel do**
    $w_t^k \leftarrow \text{ClientUpdate}(k, \hat{w}_{t+1})$
  end for
  $\Delta\tilde{w}_t = \sum_{k \in \mathcal{S}_t} p_k \left(\hat{w}_{t+1} - w_t^k\right)$ // observe
  Model update: executes Equations (13)-(17) in the main paper
end for

ClientUpdate(k, w)**:** // run on client k
$\mathcal{B} \leftarrow$ (split local data into batches of size)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta_l F_k(w; b)$
  end for
end for
return w to server

E Experimental Details

Implementation. All the experiments are implemented using PyTorch, a popular machine learning library. We simulate the federated learning environment, including clients, and run all experiments on a deep learning server equipped with an NVIDIA GTX 2080 ti GPU.

E.1 Evaluation metric.

For the RS task, the model performance is evaluated using the following metrics: area under curve (AUC), Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). For the CV task, the model performance is measured by the widely used Top-1 accuracy metric. In the experiments, for the CTR (Click-Through Rate) task, the model performance is evaluated using the following metrics: area under curve (AUC), Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG).

$$\text{AUC} = \frac{\sum_{x_0 \in D_T} \sum_{x_1 \in D_F} \mathbf{1}\left[f\left(x_1\right) < f\left(x_0\right)\right]}{\left|D_T\right|\left|D_F\right|},$$

$$\text{HitRate@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbf{1}\left(R_{u,g_u} \leq K\right),$$

$$\text{NDCG@K} = \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{U}|} \frac{2^{\mathbf{1}\left(R_{u,g_u} \leq K\right)} - 1}{\log_2\left(\mathbf{1}\left(R_{u,g_u} \leq K\right) + 1\right)},$$

where $\mathcal{U}$ is the set of users, $\mathbf{1}$ is the indicator function, R_{u,g_u} is the rank generated by the model for the ground truth item g_u, f is the model being evaluated, and D_T and D_F are the positive and negative sample sets in the testing data, respectively. For the image classification task, the model performance is measured by the widely used Top-1 accuracy metric.

E.2 Detailed Settings of Figure 3 in the Main Paper.

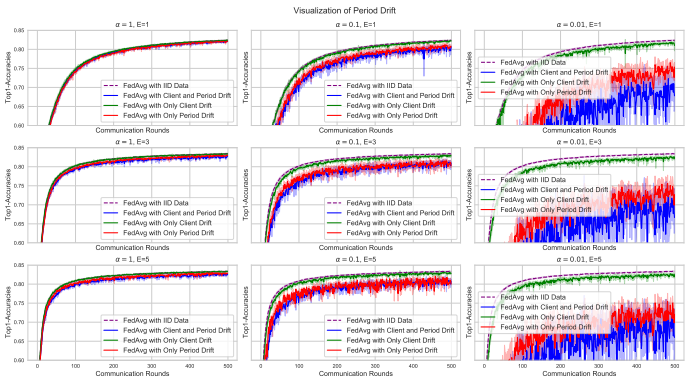


Figure 9: **Visualization of period drift with different α and E**

- **Figure 3 (a).** To provide a clearer illustration, we displayed the label distribution of the 10-digit classes, rather than the complete 62 classes in the original FEMNIST dataset, for 20 communication rounds.
- **Figure 3 (b).** 1) FEDAVG *with iid data with no period or client drift:* FEDAVG with iid data, where training data is randomly partitioned among all clients, resulting in no period or client drift; 2) FEDAVG *with only period drift:* non-iid data is initially partitioned, but the training data of the sampled clients is randomly reshuffled and iid-distributed evenly among clients in each round, resulting in period drift but no client drift; 3) FEDAVG *with only client drift:* iid data is initially partitioned, but the training data of the sampled clients in each round is re-partitioned in non-iid setting, resulting in client drift but no period drift; 4) FEDAVG *with both period and client drift:* FEDAVG with non-iid data, where data is partitioned in non-iid setting, resulting in both period and client drift.

To make the results more convincing, we conducted more experiments on FEMNIST. Specifically, we add experiments with various local epochs and data heterogeneity. Each experiment is repeated 5 times, and the results are shown as follow:

F Convergence Analysis

This section provides a comprehensive theoretical analysis of the FedEve algorithm. We establish formal guarantees on the convergence of our proposed method by first stating necessary assumptions, then presenting the main convergence theorem, and finally providing supporting lemmas and detailed proofs.

F.1 Assumptions

For our theoretical analysis, we rely on the following standard assumptions in federated optimization literature:

Assumption F.1 (Lipschitz Continuity)**.** The loss function $l(\cdot, z)$ is L-Lipschitz continuous, that is, $|l(w; z) - l(w'; z)| \leq L\|w - w'\|$, and is L-smooth for any z, that is, $\|\nabla l(w; z) - \nabla l(w'; z)\| \leq L\|w - w'\|$ for any z, w, w'.

The Lipschitz continuity assumption ensures that the loss function doesn't change too rapidly as the model parameters change, which is crucial for establishing the stability of our optimization procedure. The smoothness property ensures that the gradients of the loss function are well-behaved, which is necessary for proving convergence rates.

Assumption F.2 (Bounded Variance)**.** The function f_i has σ-bounded variance, i.e., $\mathbb{E}\|f_i(w) - \nabla f_i(w)\| \leq \sigma$ for all $w \in \mathbb{R}^d$ and $i \in [N]$.

This assumption limits the variance of the stochastic gradients at each client, which is essential in federated learning where we only have access to gradient estimates from a subset of clients. This bounded variance allows us to control the error introduced by client sampling and stochastic optimization.

F.2 Theorems

Our main theoretical result establishes the convergence rate of the FedEve algorithm:

Theorem F.3 (Convergence of FedEve)**.** *Suppose Assumptions F.1,F.2 hold and $\eta_g \leq \frac{1}{L}$. Then, for the FedEve algorithm, we have:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(w_t)\|^2] \leq \mathcal{O}\left(\frac{L(f(w_0) - f^*)}{T} + \frac{G_{kal}^2\sigma^2}{S}\left(1 - \frac{S}{N}\right)\right).$$

This theorem provides a bound on the average squared norm of gradients, which is a standard measure of convergence in non-convex optimization. The bound consists of two terms:

- The first term $\mathcal{O}\left(\frac{L(f(w_0)-f^*)}{T}\right)$ decreases with the number of iterations T, indicating that the algorithm converges as T increases.
- The second term $\mathcal{O}\left(\frac{G_{kal}^2\sigma^2}{S}\left(1 - \frac{S}{N}\right)\right)$ represents the irreducible error due to client sampling and stochastic gradients, which decreases as the number of sampled clients S increases and vanishes when all clients participate ($S = N$).

This convergence bound demonstrates that FedEve achieves the optimal convergence rate for non-convex optimization while effectively handling the challenges of federated learning settings.

F.3 Lemmas

To prove our main theorem, we establish the following key lemmas that characterize the behavior of client sampling and the Kalman filter-based momentum updates:

Lemma F.4 (Variance of Local Gradients)**.** *Given Assumptions F.1,F.2, the variance of the local gradients can be bounded as:*

$$\mathbb{E}[\|\nabla f_{\mathcal{S}}(w) - \nabla f(w)\|^2] \leq \frac{\sigma^2}{S}\left(1 - \frac{S}{N}\right).$$

This lemma quantifies the error introduced by client sampling in federated learning. It shows that the variance of the gradients from a subset of clients $\mathcal{S}$ compared to the full gradient decreases as we sample more clients (larger S), and vanishes when all clients participate ($S = N$).

Lemma F.5 (Kalman Filter Update Variance)**.** *Given Assumptions F.1,F.2, the variance of the momentum update using the Kalman filter can be bounded as:*

$$\mathbb{E}[\|M_{t+1} - M_t\|^2] \leq G_{kal}^2 \left(\frac{\sigma^2}{S} \left(1 - \frac{S}{N} \right) + \mathbb{E}[\|\nabla f(w_t) - M_t\|^2] \right).$$

This lemma characterizes the stability of the momentum updates in our FedEve algorithm. It shows that the variance of the momentum changes is controlled by both the client sampling variance and the current error in the momentum estimate. The Kalman gain parameter G_{kal} allows us to balance between adaptivity and stability in the momentum updates.

F.4 Proofs

We now provide detailed proofs for our lemmas and main theorem. Each proof is structured to clearly show the logical progression from assumptions to conclusions, with detailed explanations of intermediate steps.

F.4.1 Proof of Lemma F.4

Proof. Our goal is to bound the variance between the gradient estimated from a subset of clients and the full gradient across all clients. We start by analyzing the definition of the sampled gradient:

$$\nabla f_{\mathcal{S}}(w) = \frac{1}{S} \sum_{i \in \mathcal{S}} \nabla f_i(w).$$

Given that $\mathcal{S}$ is a randomly selected subset of S clients from the total N clients without replacement, the variance we want to bound can be expressed as:

$$\mathbb{E}[\|\nabla f_{\mathcal{S}}(w) - \nabla f(w)\|^2] = \mathbb{E}\left[\left\| \frac{1}{S} \sum_{i \in \mathcal{S}} \nabla f_i(w) - \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(w) \right\|^2 \right].$$

To analyze this expression, we will decompose the difference by separating the selected clients $\mathcal{S}$ and the non-selected clients $\mathcal{S}^c$. Note that $\mathcal{S} \cup \mathcal{S}^c = \{1, 2, \ldots, N\}$ and $\mathcal{S} \cap \mathcal{S}^c = \emptyset$. We can rewrite:

$$\begin{aligned} \frac{1}{S} \sum_{i \in \mathcal{S}} \nabla f_i(w) - \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(w) &= \frac{1}{S} \sum_{i \in \mathcal{S}} \nabla f_i(w) - \frac{1}{N} \left(\sum_{i \in \mathcal{S}} \nabla f_i(w) + \sum_{i \in \mathcal{S}^c} \nabla f_i(w) \right) \\ &= \left(\frac{1}{S} - \frac{1}{N} \right) \sum_{i \in \mathcal{S}} \nabla f_i(w) - \frac{1}{N} \sum_{i \in \mathcal{S}^c} \nabla f_i(w). \end{aligned}$$

To compute the expected squared norm of this expression, we note that the two terms are independent due to the random sampling process. Therefore:

$$\begin{aligned} &\mathbb{E}\left[\left\| \left(\frac{1}{S} - \frac{1}{N} \right) \sum_{i \in \mathcal{S}} \nabla f_i(w) - \frac{1}{N} \sum_{i \in \mathcal{S}^c} \nabla f_i(w) \right\|^2 \right] \\ &= \left(\frac{1}{S} - \frac{1}{N} \right)^2 \mathbb{E}\left[\left\| \sum_{i \in \mathcal{S}} \nabla f_i(w) \right\|^2 \right] + \frac{1}{N^2} \mathbb{E}\left[\left\| \sum_{i \in \mathcal{S}^c} \nabla f_i(w) \right\|^2 \right]. \end{aligned}$$

Now, we need to evaluate the expectations. Let's denote $\mu = \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(w)$ as the mean gradient across all clients. Under Assumption F.2, the client gradients have bounded variance σ^2 around this mean. For a set of S randomly sampled clients, the variance of their sum is:

$$\mathbb{E}\left[\left\| \sum_{i \in \mathcal{S}} \nabla f_i(w) - S\mu \right\|^2 \right] = S\sigma^2.$$

Since $\mathbb{E}[\nabla f_i(w)] = \mu$ for all clients, we have:

$$\mathbb{E}\left[\left\| \sum_{i \in \mathcal{S}} \nabla f_i(w) \right\|^2 \right] = S\sigma^2 + S^2 \|\mu\|^2.$$

Similarly, for the non-selected clients:

$$\mathbb{E}\left[\left\|\sum_{i\in\mathcal{S}^c}\nabla f_i(w)\right\|^2\right] = (N-S)\sigma^2 + (N-S)^2\|\mu\|^2.$$

Substituting these expressions into our variance formula:

$$\begin{aligned}&\left(\frac{1}{S}-\frac{1}{N}\right)^2(S\sigma^2+S^2\|\mu\|^2) + \frac{1}{N^2}((N-S)\sigma^2+(N-S)^2\|\mu\|^2)\\ &= \left(\frac{1}{S}-\frac{1}{N}\right)^2 S\sigma^2 + \left(\frac{1}{S}-\frac{1}{N}\right)^2 S^2\|\mu\|^2 + \frac{(N-S)\sigma^2}{N^2} + \frac{(N-S)^2\|\mu\|^2}{N^2}.\end{aligned}$$

After algebraic manipulation, the terms involving $\|\mu\|^2$ cancel out (which can be verified by expanding the expressions), leaving us with:

$$\begin{aligned}\mathbb{E}[\|\nabla f_{\mathcal{S}}(w)-\nabla f(w)\|^2] &= \left(\frac{1}{S}-\frac{1}{N}\right)^2 S\sigma^2 + \frac{(N-S)\sigma^2}{N^2}\\ &= \frac{\sigma^2}{S}\left(1-\frac{S}{N}\right).\end{aligned}$$

This final result elegantly quantifies how the variance scales with both the number of sampled clients S and the total number of clients N. We observe that:

- As S increases, the variance decreases, showing the benefit of sampling more clients.
- When $S = N$ (i.e., we use all clients), the variance becomes zero, as expected.
- The term $1 - \frac{S}{N}$ represents the finite population correction factor from sampling theory, accounting for sampling without replacement.

□

F.4.2 Proof of Lemma F.5

Proof. This lemma characterizes the stability of momentum updates using the Kalman filter. We begin with the momentum update equation in the FedEve algorithm:

$$M_{t+1} = M_t + G_{kal}(\Delta\tilde{w}_t - M_t),$$

where $\Delta\tilde{w}_t$ represents the average update from the selected clients at time t, and G_{kal} is the Kalman gain parameter that controls how much new information is incorporated into the momentum.

Rearranging the update equation, we have:

$$M_{t+1} - M_t = G_{kal}(\Delta\tilde{w}_t - M_t).$$

Our goal is to bound the expected squared norm of this difference, which measures how much the momentum changes between iterations:

$$\mathbb{E}[\|M_{t+1}-M_t\|^2] = \mathbb{E}[\|G_{kal}(\Delta\tilde{w}_t - M_t)\|^2].$$

Since G_{kal} is a scalar constant, we can factor it out:

$$\mathbb{E}[\|M_{t+1}-M_t\|^2] = G_{kal}^2\mathbb{E}[\|\Delta\tilde{w}_t - M_t\|^2].$$

To analyze $\mathbb{E}[\|\Delta\tilde{w}_t - M_t\|^2]$, we introduce the true full gradient $\nabla f(w_t)$ as an intermediate term:

$$\begin{aligned}\mathbb{E}[\|\Delta\tilde{w}_t - M_t\|^2] &= \mathbb{E}[\|\Delta\tilde{w}_t - \nabla f(w_t) + \nabla f(w_t) - M_t\|^2]\\ &= \mathbb{E}[\|(\Delta\tilde{w}_t - \nabla f(w_t)) + (\nabla f(w_t) - M_t)\|^2].\end{aligned}$$

Now, we expand the squared norm of the sum. If we can show that the cross-term $\mathbb{E}[\langle\Delta\tilde{w}_t - \nabla f(w_t), \nabla f(w_t) - M_t\rangle] = 0$, then we can separate the expression. This is indeed the case because:

- M_t depends only on information up to time $t-1$

- $\nabla f(w_t)$ is a deterministic function of w_t
- $\Delta\tilde{w}_t - \nabla f(w_t)$ is the sampling error at time t, which is independent of past information

Therefore, we can write:

$$\begin{aligned}\mathbb{E}[\|\Delta\tilde{w}_t - M_t\|^2] &= \mathbb{E}[\|\Delta\tilde{w}_t - \nabla f(w_t)\|^2] + \mathbb{E}[\|\nabla f(w_t) - M_t\|^2] \\ &\quad + 2\mathbb{E}[\langle \Delta\tilde{w}_t - \nabla f(w_t), \nabla f(w_t) - M_t\rangle] \\ &= \mathbb{E}[\|\Delta\tilde{w}_t - \nabla f(w_t)\|^2] + \mathbb{E}[\|\nabla f(w_t) - M_t\|^2].\end{aligned}$$

The first term, $\mathbb{E}[\|\Delta\tilde{w}_t - \nabla f(w_t)\|^2]$, represents the variance of the client updates. From Lemma F.4, we know that:

$$\mathbb{E}[\|\Delta\tilde{w}_t - \nabla f(w_t)\|^2] = \frac{\sigma^2}{S}\left(1 - \frac{S}{N}\right).$$

The second term, $\mathbb{E}[\|\nabla f(w_t) - M_t\|^2]$, represents how far the current momentum estimate is from the true gradient. This is the estimation error from previous iterations.

Substituting these results back into our original expression:

$$\mathbb{E}[\|M_{t+1} - M_t\|^2] = G_{kal}^2\left(\frac{\sigma^2}{S}\left(1 - \frac{S}{N}\right) + \mathbb{E}[\|\nabla f(w_t) - M_t\|^2]\right).$$

This result provides valuable insights into the momentum dynamics:

- The variance of momentum changes is controlled by two factors: client sampling variance and current momentum estimation error.
- The Kalman gain G_{kal} directly impacts the magnitude of momentum changes. A smaller G_{kal} leads to more stable but slower-adapting momentum, while a larger G_{kal} allows faster adaptation but with potentially higher variance.
- When $S = N$ (using all clients), the client sampling variance term disappears, but the estimation error term remains, showing that momentum still provides a beneficial smoothing effect even with full client participation.

□

F.4.3 Proof of Theorem F.3

Proof. Our convergence analysis follows the standard approach for analyzing first-order optimization methods for non-convex functions. We start by using the L-smoothness property from Assumption F.1 to bound the progress in the objective function between consecutive iterations.

For an L-smooth function f, we have the following inequality for any two points w and w':

$$f(w') \leq f(w) + \langle \nabla f(w), w' - w\rangle + \frac{L}{2}\|w' - w\|^2.$$

Applying this to consecutive iterates w_t and w_{t+1} in our algorithm:

$$f(w_{t+1}) \leq f(w_t) + \langle \nabla f(w_t), w_{t+1} - w_t\rangle + \frac{L}{2}\|w_{t+1} - w_t\|^2.$$

In the FedEve algorithm, the update rule is $w_{t+1} = w_t - \eta_g M_{t+1}$, where M_{t+1} is the momentum term updated using the Kalman filter and η_g is the global learning rate. Substituting this update rule:

$$\begin{aligned}f(w_{t+1}) &\leq f(w_t) + \langle \nabla f(w_t), -\eta_g M_{t+1}\rangle + \frac{L}{2}\| - \eta_g M_{t+1}\|^2 \\ &= f(w_t) - \eta_g\langle \nabla f(w_t), M_{t+1}\rangle + \frac{\eta_g^2 L}{2}\|M_{t+1}\|^2.\end{aligned}$$

Taking the expectation, we have:

$$\mathbb{E}[f(w_{t+1})] \leq f(w_t) - \eta_g\mathbb{E}[\langle \nabla f(w_t), M_{t+1}\rangle] + \frac{\eta_g^2 L}{2}\mathbb{E}[\|M_{t+1}\|^2].$$

Now we need to analyze the terms $\mathbb{E}[\langle \nabla f(w_t), M_{t+1} \rangle]$ and $\mathbb{E}[\|M_{t+1}\|^2]$. Let's start with the inner product term.

The momentum update in FedEve is given by:

$$M_{t+1} = M_t + G_{kal}(\Delta \tilde{w}_t - M_t),$$

where $\Delta \tilde{w}_t$ is the average gradient from the sampled clients. An important property of this update is that, under expectation, it is unbiased:

$$\mathbb{E}[\Delta \tilde{w}_t] = \nabla f(w_t).$$

Therefore:

$$\begin{aligned}
\mathbb{E}[M_{t+1}] &= \mathbb{E}[M_t + G_{kal}(\Delta \tilde{w}_t - M_t)] \\
&= \mathbb{E}[M_t] + G_{kal}(\mathbb{E}[\Delta \tilde{w}_t] - \mathbb{E}[M_t]) \\
&= \mathbb{E}[M_t] + G_{kal}(\nabla f(w_t) - \mathbb{E}[M_t]).
\end{aligned}$$

In the long run, this recursive relation converges to $\mathbb{E}[M_t] = \nabla f(w_t)$. For simplicity, we assume this has approximately been achieved, which gives us:

$$\mathbb{E}[\langle \nabla f(w_t), M_{t+1} \rangle] = \langle \nabla f(w_t), \mathbb{E}[M_{t+1}] \rangle = \langle \nabla f(w_t), \nabla f(w_t) \rangle = \|\nabla f(w_t)\|^2.$$

Next, we need to bound $\mathbb{E}[\|M_{t+1}\|^2]$. Using the update rule and the variance from Lemma F.5:

$$\begin{aligned}
\mathbb{E}[\|M_{t+1}\|^2] &= \mathbb{E}[\|\mathbb{E}[M_{t+1}] + (M_{t+1} - \mathbb{E}[M_{t+1}])\|^2] \\
&= \|\mathbb{E}[M_{t+1}]\|^2 + \mathbb{E}[\|M_{t+1} - \mathbb{E}[M_{t+1}]\|^2] \\
&= \|\nabla f(w_t)\|^2 + \mathbb{E}[\|M_{t+1} - \nabla f(w_t)\|^2] \\
&\leq \|\nabla f(w_t)\|^2 + G_{kal}^2 \frac{\sigma^2}{S} \left(1 - \frac{S}{N}\right).
\end{aligned}$$

The last inequality uses the bound on the variance of M_{t+1} derived from Lemma F.5.

Substituting these results back into our progress bound:

$$\begin{aligned}
\mathbb{E}[f(w_{t+1})] &\leq f(w_t) - \eta_g \|\nabla f(w_t)\|^2 + \frac{\eta_g^2 L}{2} \left(\|\nabla f(w_t)\|^2 + G_{kal}^2 \frac{\sigma^2}{S} \left(1 - \frac{S}{N}\right) \right) \\
&= f(w_t) - \eta_g \|\nabla f(w_t)\|^2 + \frac{\eta_g^2 L}{2} \|\nabla f(w_t)\|^2 + \frac{\eta_g^2 L G_{kal}^2 \sigma^2}{2S} \left(1 - \frac{S}{N}\right) \\
&= f(w_t) - \eta_g \left(1 - \frac{\eta_g L}{2}\right) \|\nabla f(w_t)\|^2 + \frac{\eta_g^2 L G_{kal}^2 \sigma^2}{2S} \left(1 - \frac{S}{N}\right).
\end{aligned}$$

The coefficient of $\|\nabla f(w_t)\|^2$ is maximized when $\eta_g = \frac{1}{L}$. Setting this optimal learning rate, we get:

$$\begin{aligned}
\mathbb{E}[f(w_{t+1})] &\leq f(w_t) - \frac{1}{L} \left(1 - \frac{1}{2}\right) \|\nabla f(w_t)\|^2 + \frac{1}{L^2} \frac{L G_{kal}^2 \sigma^2}{2S} \left(1 - \frac{S}{N}\right) \\
&= f(w_t) - \frac{1}{2L} \|\nabla f(w_t)\|^2 + \frac{G_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right).
\end{aligned}$$

This inequality shows that in each iteration, we make progress in decreasing the objective function, but there's a constant error term due to the variance in client sampling.

To derive the final convergence rate, we sum this inequality over all iterations from $t = 0$ to $t = T - 1$:

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}[f(w_{t+1})] &\leq \sum_{t=0}^{T-1} \left(f(w_t) - \frac{1}{2L} \|\nabla f(w_t)\|^2 + \frac{G_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right) \right) \\
&= \sum_{t=0}^{T-1} f(w_t) - \frac{1}{2L} \sum_{t=0}^{T-1} \|\nabla f(w_t)\|^2 + \frac{T G_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right).
\end{aligned}$$

Note that $\sum_{t=0}^{T-1} \mathbb{E}[f(w_{t+1})] = \sum_{t=1}^{T} \mathbb{E}[f(w_t)]$. Rearranging the terms:

$$\sum_{t=1}^{T} \mathbb{E}[f(w_t)] \leq \sum_{t=0}^{T-1} f(w_t) - \frac{1}{2L} \sum_{t=0}^{T-1} \|\nabla f(w_t)\|^2 + \frac{TG_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right)$$

$$\Rightarrow \mathbb{E}[f(w_T)] \leq f(w_0) - \frac{1}{2L} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] + \frac{TG_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right).$$

This is because the sum telescopes: $f(w_0) + (f(w_1) - f(w_1)) + \ldots + (f(w_{T-1}) - f(w_{T-1})) - \mathbb{E}[f(w_T)] = f(w_0) - \mathbb{E}[f(w_T)]$.

Rearranging to isolate the gradient norm terms:

$$\frac{1}{2L} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] \leq f(w_0) - \mathbb{E}[f(w_T)] + \frac{TG_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right).$$

Since f is bounded below by some value f^* (which could be the global minimum), we have $\mathbb{E}[f(w_T)] \geq f^*$. Thus:

$$\frac{1}{2L} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] \leq f(w_0) - f^* + \frac{TG_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right).$$

Dividing both sides by T:

$$\frac{1}{2LT} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] \leq \frac{f(w_0) - f^*}{T} + \frac{G_{kal}^2 \sigma^2}{2LS} \left(1 - \frac{S}{N}\right)$$

$$\Rightarrow \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] \leq \frac{2L(f(w_0) - f^*)}{T} + \frac{G_{kal}^2 \sigma^2}{S} \left(1 - \frac{S}{N}\right).$$

This final bound can be expressed in big-O notation as:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_t)\|^2] \leq \mathcal{O}\left(\frac{L(f(w_0) - f^*)}{T} + \frac{G_{kal}^2 \sigma^2}{S} \left(1 - \frac{S}{N}\right)\right).$$

This result demonstrates several important properties of the FedEve algorithm:

- The first term $\mathcal{O}\left(\frac{L(f(w_0)-f^*)}{T}\right)$ shows that the convergence rate is $\mathcal{O}(1/T)$, which is optimal for first-order methods on non-convex functions.
- The second term $\mathcal{O}\left(\frac{G_{kal}^2 \sigma^2}{S}\left(1 - \frac{S}{N}\right)\right)$ represents the irreducible error due to client sampling and stochastic gradients.
- This error decreases as we increase the number of sampled clients S, and vanishes completely when $S = N$ (i.e., when we use all clients).
- The Kalman gain parameter G_{kal} appears quadratically in the error term, showing that smaller values of G_{kal} can reduce the impact of stochasticity, but at the cost of potentially slower adaptation to changes in the gradient.

In practical implementations, the parameters η_g and G_{kal} can be tuned to balance the trade-off between convergence speed and stability based on the specific characteristics of the federated learning task. □

G Generalization bound

This Generalization bound is inspired by Sun et al. [2023].

G.1 Theorems

Theorem G.1 (On-average Algorithmic Stability)**.** *Suppose a federated learning algorithm $\mathcal{A}$ is ϵ-on-averagely stable. Then,*

$$\epsilon_{gen} \leq \mathbb{E}_{\mathcal{A},\mathcal{S}}\left[\left|f(\mathcal{A}(\mathcal{S})) - \hat{f}_{\mathcal{S}}(\mathcal{A}(\mathcal{S}))\right|\right] \leq \epsilon.$$

Theorem G.2 (Generalization Bound)**.** *Suppose Assumptions G.3-G.5 hold and $\eta_l \leq \frac{1}{L_m K(t+1)}$. Then,*

$$\epsilon_{gen} \leq \mathcal{O}\left(\frac{L_p}{nL_m}\right)\left[T(\sigma_l + \sigma_g) + \sqrt{TL_m\Delta_0} + T\sqrt{\frac{G_{kal}^2\sigma^2}{S}}\right].$$

G.2 Assumptions

Assumption G.3 (Lipschitz Continuity)**.** The loss function $l(\cdot, z)$ is L_p-Lipschitz continous, that is, $|l(w;z) - l(w';z)| \leq L_p\|w - w'\|$, and is L_m-smooth for any z, that is, $\|\nabla l(w;z) - \nabla l(w';z)\| \leq L_m\|w - w'\|$ for any z, w, w'.

Assumption G.4 (Bounded Variance)**.** The function F_i have σ_l-bounded (local) variance i.e., $\mathbb{E}\|f_i(w) - \nabla f_i(w)\| \leq \sigma_l$ for all $w \in \mathcal{U}^d$, $j \in [d]$ and $i \in [m]$. Furthermore, we assume the (global) variance is bounded, $\mathbb{E}\|f_i(w) - \nabla f(w)\| \leq \sigma_g$ for all $x \in \mathcal{U}^d$ and $j \in [d]$.

Assumption G.5 (Bounded Gradients)**.** The function $f_i(x, z)$ have G-bounded gradients i.e., for any $i \in [m]$, $x \in \mathcal{U}^d$ and $z \in \mathcal{Z}$ we have $|[\nabla f_i(x, z)]_j| \leq G$ for all $j \in [d]$.

G.3 Lemmas

Lemma G.6 (Bounded Local Updates)**.** *Suppose Assumptions G.3-G.5 hold. For any step-size, we can bound the local updates as*

$$\mathbb{E}\|w_{i,k} - w_t\| \leq \frac{(1 + \eta_l L_m)^k - 1}{L_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right).$$

where $w_{i,k}$ is the model parameters of client i at k-th local updates.

Lemma G.7 (Bounded Local Gradients)**.** *Given Assumptions G.3-G.5. For any step-size, we can bound the local gradients as $\mathbb{E}\|f_i(w_{i,k})\| \leq (1 + \eta_l L_m)^k\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right)$, where $f_i(\cdot)$ is the sampled gradient of client i.*

Lemma G.8 (Bounded Global Model with Sample Perturbation)**.** *Given Assumptions G.3-G.5. For any step-size, we can bound the local gradients as $\mathbb{E}\|w_T - w_T'\| \leq \sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right)$, where w_T' is the model parameters with sample perturbation at T-th communication rounds.*

G.4 Proofs

G.4.1 Proof of Lemma G.6

Proof. Bounding Local Updates:

$$\begin{aligned}
&\mathbb{E} \quad \|w_{i,k+1} - w_t\| \\
&= \quad \mathbb{E}\|w_{i,k} - \eta_l f_i(w_{i,k}) - w_t\| \\
&\leq \quad \mathbb{E}\|w_{i,k} - w_t - \eta_l(f_i(w_{i,k}) - f_i(w_t))\| + \eta_l\mathbb{E}\|f_i(w_t)\| \\
&\leq \quad (1 + \eta_l L_m)\mathbb{E}\|w_{i,k} - w_t\| + \eta_l\mathbb{E}\|f_i(w_t)\| \\
&\leq \quad (1 + \eta_l L_m)\mathbb{E}\|w_{i,k} - w_t\| + \eta_l(\mathbb{E}\|f_i(w_t) - \nabla f_i(w_t)\| + \mathbb{E}\|\nabla f_i(w_t) - \nabla f(w_t)\| + \mathbb{E}\|\nabla f(w_t)\|) \\
&\leq \quad (1 + \eta_l L_m)\mathbb{E}\|w_{i,k} - w_t\| + \eta_l(\sigma_l + \sigma_g + \mathbb{E}\|\nabla f(w_t)\|),
\end{aligned}$$

unrolling the above and noting $w_{i,0} = w_t$ yields

$$\mathbb{E}\|w_{i,k} - w_t\| \quad \leq \quad \frac{(1 + \eta_l L_m)^k - 1}{L_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right).$$

□

G.4.2 Proof of Lemma G.7

Proof. Bounding Local Gradients:

$$\begin{aligned}\mathbb{E}\|f_i(w_{i,k})\| &= \mathbb{E}\|f_i(w_{i,k}) - \nabla f_i(w_{i,k}) + \nabla f_i(w_{i,k}) - \nabla f(w_t) + \nabla f(w_t)\| \\ &\leq \mathbb{E}\|f_i(w_{i,k}) - \nabla f_i(w_{i,k})\| + \mathbb{E}\|\nabla f_i(w_{i,k}) - \nabla f(w_t)\| + \mathbb{E}\|\nabla f(w_t)\| \\ &\leq \sigma_l + L_m\mathbb{E}\|w_{i,k} - w_t\| + \mathbb{E}\|\nabla f(w_t),\end{aligned}$$

based on Lemma G.6, we obtain:

$$\begin{aligned}&\leq \sigma_l + \mathbb{E}\|\nabla f(w_t) \\ &+ \left((1+\eta_l L_m)^k - 1\right)\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right) \\ &\leq \left(1+\eta_l L_m\right)^k\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right).\end{aligned}$$

□

G.4.3 Proof of Lemma G.8

Proof. Given time index t and for client j with $j \neq i$, we have

$$\begin{aligned}\mathbb{E}\|w_{j,k+1} - w'_{j,k+1}\| &= \mathbb{E}\|w_{j,k} - w'_{j,k} - \eta_l(g_j(w_{j,k}) - g_j(w'_{j,k}))\| \\ &\leq (1+\eta_l L_m)\mathbb{E}\|w_{j,k} - w'_{j,k}\|.\end{aligned}$$

And unrolling it gives

$$\mathbb{E}\|w_{j,K} - w'_{j,K}\| \leq e^{\eta_l K L_m}\mathbb{E}\|w_t - w'_t\|, \ \forall j \neq i,$$

since $1 + x < e^x$. For client i, there are two cases to consider. In the first case, SGD selects non-perturbed samples in $\mathcal{S}$ and $\mathcal{S}^{(i)}$, which happens with probability $1 - 1/n_i$. Then, we have $\|w_{i,k+1} - w'_{i,k+1}\| \leq (1+\eta_l L_m)\|w_{i,k} - w'_{i,k}\|$. In the second case, SGD encounters the perturbed sample at time step k, which happens with probability $1/n_i$. Then, we have

$$\begin{aligned}\|w_{i,k+1} - w'_{i,k+1}\| &= \|w_{i,k} - w'_{i,k} - \eta_l(f_i(w_{i,k}) - g'_i(w'_{i,k}))\| \\ &\leq \|w_{i,k} - w'_{i,k} - \eta_l(f_i(w_{i,k}) - f_i(w'_{i,k}))\| + \eta_l\|f_i(w'_{i,k}) - g'_i(w'_{i,k})\| \\ &\leq (1+\eta_l L_m)\|w_{i,k} - w'_{i,k}\| + \eta_l\|f_i(w'_{i,k}) - g'_i(w'_{i,k})\|.\end{aligned}$$

Combining these two cases for client i we have

$$\begin{aligned}\mathbb{E}\|w_{i,k+1} - w'_{i,k+1}\| &\leq (1+\eta_l L_m)\mathbb{E}\|w_{i,k} - w'_{i,k}\| + \frac{\eta_l}{n_i}\mathbb{E}\|f_i(w'_{i,k}) - g'_i(w'_{i,k})\| \\ &\leq (1+\eta_l L_m)\mathbb{E}\|w_{i,k} - w'_{i,k}\| + \frac{2\eta_l}{n_i}\mathbb{E}\|f_i(w_{i,k})\|,\end{aligned}$$

based on Lemma G.7, we obtain:

$$\begin{aligned}&\leq (1+\eta_l L_m)\mathbb{E}\|w_{i,k} - w'_{i,k}\| \\ &+ \frac{2\eta_l}{n_i}e^{\eta_l k L_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right),\end{aligned}$$

then unrolling it gives

$$\begin{aligned}&\mathbb{E} \ \|w_{i,K} - w'_{i,K}\| \\ &\leq e^{\eta_l K L_m}\mathbb{E}\|w_t - w'_t\| \\ &+ \frac{2e^{\eta_l K L_m}}{n_i L_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right) \ \forall j = i. \end{aligned} \tag{43}$$

Combines 43 and 43 we have

$$\begin{aligned}\mathbb{E}\|w_{t+1} - w'_{t+1}\| &\leq \sum_{i=1}^{m} p_i\mathbb{E}\|w_{i,K} - w'_{i,K}\| \\ &\leq e^{\eta_l K L_m}\mathbb{E}\|w_t - w'_t\| + \frac{2e^{\eta_l K L_m}}{nL_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right)\end{aligned}$$

where we also use $p_i = n_i/n$ in the last step. Further, unrolling the above over t and noting $w_0 = w'_0$, we obtain

$$\mathbb{E}\|w_T - w'_T\| \leq \sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right).$$

□

G.4.4 Proof of Theorem G.2

Proof. According to the fact that:

$$\left(\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(w_t)\|\right)^2 \leq T\sum_{t=0}^{T-1}\left(\mathbb{E}\|\nabla f(w_t)\|\right)^2 \leq T\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(w_t)\|^2,$$

where the second inequality follows Jensen's inequality, and the convergence analysis of FedEve:

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla f(w_t)\|^2 \leq \mathcal{O}\left(\frac{L_m\Delta_0}{T} + \frac{G_{kal}^2\sigma^2}{S}\left(1-\frac{S}{N}\right)\right),$$

where $\Delta_0 := \mathbb{E}[f(w_0) - f(w^*)]$. The generalization bound is

$$\begin{aligned}
\epsilon_{gen} &\leq L_p\mathbb{E}\|w_T - w_T'\| \\
&\leq L_p\sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\mathbb{E}\|\nabla f(w_t)\| + \sigma_l + \sigma_g\right),
\end{aligned}$$

when $\eta_l < \frac{1}{K(t+1)L_m}$, we obtain:

$$\begin{aligned}
&\leq L_p\sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\sqrt{T\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(w_t)\|^2} + T(\sigma_l+\sigma_g)\right), \\
&\leq L_p\sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\sqrt{T\mathcal{O}\left(\frac{L_m\Delta_0}{T} + \frac{G_{kal}^2\sigma^2}{S}\left(1-\frac{S}{N}\right)\right)} + T(\sigma_l+\sigma_g)\right), \\
&\leq L_p\sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\sqrt{L_m\Delta_0 + T\frac{G_{kal}^2\sigma^2}{S}\left(1-\frac{S}{N}\right)} + T(\sigma_l+\sigma_g)\right), \\
&\leq L_p\sum_{t=0}^{T-1}\frac{2e^{\eta_l K(t+1)L_m}}{nL_m}\left(\sqrt{L_m\Delta_0} + \sqrt{T\frac{G_{kal}^2\sigma^2}{S}\left(1-\frac{S}{N}\right)} + T(\sigma_l+\sigma_g)\right), \\
&\leq \mathcal{O}\left(\frac{L_p}{nL_m}\right)\left[T(\sigma_l+\sigma_g) + \sqrt{TL_m\Delta_0} + T\sqrt{\frac{G_{kal}^2\sigma^2}{S}\left(1-\frac{S}{N}\right)}\right].
\end{aligned}$$

□

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's contributions, including: identifying and defining period drift, proposing a predict-observe framework, developing FedEve as an implementation, and providing theoretical and empirical evidence of its effectiveness.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: The paper discusses limitations in Appendix, particularly with respect to assumptions of independence between model parameters and drift, and provides justifications for these assumptions.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper provides clear assumptions (Assumption 1 and 2 in Section 3.2 and 3.3) for the theoretical results, with complete proofs in the Appendix (e.g., Appendix sections on noise independence and Bayesian filter proofs).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

Justification: The paper provides comprehensive details on datasets (FEMNIST, CIFAR-10/100, MovieLens), models (LeNet5, ResNet-18, DIN), training settings (rounds, clients per round, epochs), and algorithm parameters in Section 4.1 and Appendix.

Guidelines:

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

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

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

Answer: [No]

Justification: While the paper uses publicly available datasets (FEMNIST, CIFAR-10/100, MovieLens) with appropriate citations and links, there is no explicit mention of code availability for the proposed FedEve algorithm.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper details data splits (e.g., Dirichlet distribution for non-iid settings), hyperparameters (learning rates, epochs, communication rounds), optimizers (SGD for CV, Adam for RS), and evaluation protocols in Section 4.1 and Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports standard deviations for all experimental results as shown in Tables 1, 2, and 3, indicating statistical significance across multiple runs of each experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

Answer: [No]

Justification: The paper does not explicitly mention the hardware specifications, memory requirements, or running time for the experiments.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research focuses on improving federated learning algorithms which inherently promote privacy protection. It uses standard benchmark datasets and does not present any apparent ethical concerns.

Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: While the paper focuses on improving federated learning, which has positive privacy implications, it does not explicitly discuss broader societal impacts, either positive or negative.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper does not release any high-risk models or datasets. It uses standard benchmark datasets and proposes an optimization algorithm for federated learning which does not pose risks of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper properly cites and links to all datasets used (FEMNIST, CIFAR-10/100, MovieLens) and acknowledges the original creators through standard citations.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification: The paper does not introduce new datasets or models, but rather proposes a new algorithm (FedEve) based on existing FL paradigms and evaluates it on standard datasets.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing or research with human subjects. It uses pre-existing datasets and simulation-based evaluations.

Guidelines:

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

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

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

Answer: [NA]

Justification: The research does not involve human subjects or study participants, so IRB approval was not required for this work.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The paper does not use large language models as part of its core methodology or experiments. The proposed FedEve method is based on mathematical formulations and traditional ML techniques.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.