FedBaF: Federated Learning Aggregation Biased by a Foundation Model

Jong-Ik Park **Srinivasa Pranav** **José M. F. Moura** **Carlee Joe-Wong**
Electrical and Computer Engineering, Carnegie Mellon University

Abstract

Foundation models are now a major focus of leading technology organizations due to their ability to generalize across diverse tasks. Existing approaches for adapting foundation models to new applications often rely on Federated Learning (FL) and disclose the foundation model weights to clients when using it to initialize the global model. While these methods ensure client data privacy, they compromise model and information security. In this paper, we introduce Federated Learning Aggregation Biased by a Foundation Model (FedBaF), a novel method for dynamically integrating pre-trained foundation model weights during the FL aggregation phase. Unlike conventional methods, FedBaF preserves the confidentiality of the foundation model while still leveraging its power to train more accurate models, especially in non-IID and adversarial scenarios. Our comprehensive experiments use Pre-ResNet and foundation models like Vision Transformer to demonstrate that FedBaF not only matches, but often surpasses the test accuracy of traditional weight initialization methods by up to 11.4% in IID and up to 15.8% in non-IID settings. Additionally, FedBaF applied to a Transformer-based language model significantly reduced perplexity by up to 39.2%.

1 INTRODUCTION

Developing foundation models (Zhuang et al., 2023) has become a major focus for leading technology companies like OpenAI, Microsoft, and Amazon AWS. These deep learning models are often trained with vast amounts of high-quality data (Bommasani et al., 2021) and their ability to generalize across different tasks and domains has made them essential assets for industry, government, and academia. Foundation models have been applied to natural language processing (e.g., text generation, translation, summarization), image generation and recognition, healthcare diagnostics, finance predictive analytics, and customer service and virtual assistant tasks (Chen et al., 2023; Han et al., 2021; Duan et al., 2021; Joshi et al., 2022; Yosinski et al., 2014). When a foundation model's training data distribution overlaps with a new application, it provides a robust starting point for fine-tuning and customization. Instead of training a model from scratch, with limited data and computes, we can leverage pre-trained foundation models to enable faster training.

For many applications, data that could be used to customize or fine-tune foundation models is often distributed across multiple clients, such as a network of clinics or small companies spread across different jurisdictions. For example, fine-tuning a recommendation model to fit a small company's product offering may require data from clients in various regions; and adapting a healthcare model for a network of clinics would involve confidential, distributed data sources. Therefore, effective generalization requires access to diverse data from multiple clients (Pranav and Moura, 2024).

Federated Learning (FL) is a promising solution for fine-tuning these models without sharing client data: FL clients train models on diverse local data, and a central FL server aggregates the client updates to build and refine a global model (Li et al., 2021; McMahan et al., 2017; Singh et al., 2019; Lyu et al., 2020; Nguyen et al., 2021; Wang et al., 2020; Lee et al., 2023; Siew et al., 2024). Using a foundation model to initialize the global FL model leads to effective customization that leverages diverse, distributed data without directly accessing the client data (Nguyen et al., 2022; Chen et al., 2023). However, *there are significant risks associated with sending a foundation model to clients, as required in traditional FL fine-tuning methods.*

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

First, disclosing a foundation model's weights to FL clients poses a significant security risk. For example, malicious actors could carry out *membership inference attacks*, identifying whether specific data was part of the foundation model's training dataset. Then, an attacker could disrupt the global model's training by introducing updates that degrade performance on identified data through backdoor attacks, deliberately leading to targeted misclassification (Dayal et al., 2023; Hu et al., 2022; Wang et al., 2023b). Similarly, for *model inversion attacks*, attackers use known model weights to reverse-engineer sensitive training data (see Figure 1) (Fredrikson et al., 2015; Li et al., 2022a; Zhang et al., 2020). Protecting foundation models, often trained on sensitive, proprietary data, is critical for safeguarding the training data and maintaining model integrity (Bagdasaryan et al., 2020; Kim et al., 2023).

Second, in competitive business contexts, disclosing foundation model weights to clients risks leaking strategic insights and proprietary information to adversaries (Han et al., 2021; Yu et al., 2023). This undermines a company's competitive advantage and substantial investments in data collection and training.

To address these challenges, we present Federated Learning Aggregation Biased by a Foundation Model (**FedBaF**). Rather than using a foundation model to initialize the global model, FedBaF is a novel method for server-side foundation model integration during the task-specific global model aggregation phase of each FL round (see Figure 2). Since the server uses the foundation model in the aggregation phase, FedBaF ensures that *the foundation model is not disclosed to clients*. FedBaF also gradually reduces the foundation model's influence as training progresses, thereby improving personalization for the client pool's data and matching or outperforming existing methods.

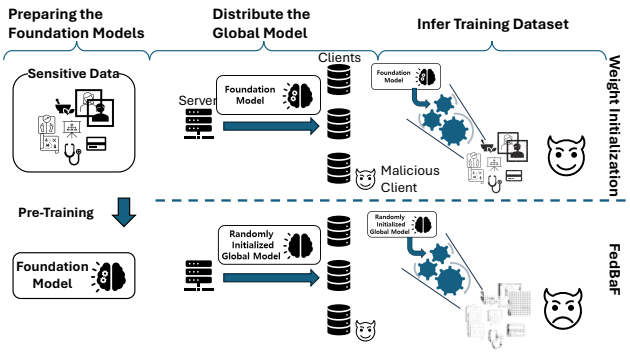

Figure 1: A visualization of a Model Inversion Attack. Since FedBaF does not initialize the global model with a pre-trained foundation model, it becomes difficult for malicious clients to reconstruct the pre-training data from the distributed global model.

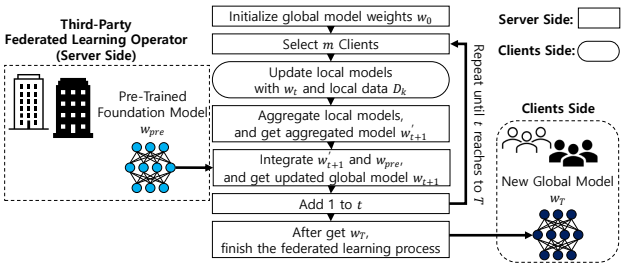

Figure 2: Visualization of FedBaF: in each FL round's aggregation phase (after client updates), the server integrates a foundation model into the global model.

FedBaF is particularly beneficial for a FL operator who owns the foundation model and needs to maintain security and integrity while fine-tuning it with a new set of clients. For instance, large technology companies such as Microsoft and Amazon, develop their own foundation models and often act as FL operators for domain-specific tasks across various industries.

When client data distributions are biased or non-IID, e.g., different ratios or a lack of certain labels, clients optimize correspondingly diverse local objective functions (Pranav and Moura, 2023) and may send conflicting updates to the server that skew the global model (Zhao et al., 2018). In FedBaF, the foundation model continuously serves as a form of regularization and stabilizes the global model by reducing the influence of these conflicting updates during the aggregation phase (Chen et al., 2023; Li et al., 2022b; Tan et al., 2022; Yosinski et al., 2014).

Furthermore, FedBaF uses a fixed foundation model as an anchor and continuously incorporates it throughout the FL training process. This adds a layer of protection from adversarial attacks beyond those introduced by disclosing foundation model weights – such as *misclassification attacks* or *backdoor attacks*, where compromised clients feed malicious updates to the server (Lyu et al., 2020; Bagdasaryan et al., 2020).

Our contributions:

1) To the best of our knowledge, we are the first to propose an **algorithm that integrates foundation models into FL without distributing the foundation model to clients**.

2) We provide **theoretical analysis of FedBaF's effectiveness** that reveals how foundation models can promote convergence in non-IID (not independent and identically distributed) and non-convex settings.

3) We conduct extensive empirical evaluation and show that **FedBaF matches or exceeds the training performance of traditional weight initialization methods** – with better test performance in 10 out of 14 cases. Our experiments use Pre-ResNet

and more complex architectures like Vision Transformer and Transformer-based language models frequently used as foundation models (Xu et al., 2023; Kenneweg et al., 2024). Compared to standard FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) with foundation models used for weight initialization, FedBaF achieves accuracy improvements of up to 10.8% in IID and up to 37.5% globally and 5.9% locally in non-IID settings. Simultaneously, FedBaF safeguards the foundation model. Similarly, applying FedBaF to a Transformer-based language model significantly reduced perplexity by up to 76.0%.

Under adversarial misclassification attacks, FedBaF demonstrates increased robustness by improving FedAvg and FedProx test performance by up to 19.4% in IID environments, up to 64.7% globally, and 7.2% locally in non-IID environments. Additionally, in 8 out of 12 cases, FedBaF was more robust than traditional weight initialization methods.

We outline related works in Sec. 2. We then detail our approach, FedBaF, in Sec. 3. In Sec. 4, we present theoretical analysis and, in Sec. 5, we provide extensive experimental evaluation. Finally, we conclude our research findings and discussion in Sec. 6.

2 RELATED WORK

Traditionally, pre-trained models are used in FL to initialize the weights of the global FL model. The server distributes this model to local clients, and the clients update it by using their local data. We refer to this approach as "weight initialization" throughout this paper. Such fine-tuning of a pre-trained model can significantly improve performance of the learned FL global model by integrating data from new clients (Nguyen et al., 2022). Several recent studies devised methods that leverage weight initialization to further improve performance: Federated Nearest Class Means (FedNCM) (Legate et al., 2023) for last-layer guidance, Federated Recursive Ridge Regression (Fed3R) (Fanì et al., 2024), Fractal Pair Similarity (FPS) (Chen et al., 2023), and FedPCL (Tan et al., 2022).

Several works also explore the use of foundation models in FL. These include cases where a subset of the weights of a large foundation model are chosen to initialize and fine-tune a smaller model (Xu et al., 2024) and when clients have diverse model architectures (Wang et al., 2023a; Park and Joe-Wong, 2024). Particularly in scenarios with limited pre-training data (Chen et al., 2023), approaches often rely on synthetic data (Nikolenko, 2021; Chen et al., 2023) for pre-training. Federated Prototype-wise Contrastive Learning (FedPCL) is a significant development that improves communication efficiency in FL by using class prototypes (Tan et al., 2022) and enhances personalized learning by having clients share class-specific information more effectively.

The related works discussed in this section so far achieve good performance, but they do not consider the significant **security vulnerabilities** that result from sharing a foundation model with local clients, which compromise data privacy and the integrity of the global model. Malicious clients with access to foundation models can exploit them through: Model Inversion Attacks, recovering original training data or sensitive attributes from the model's outputs (Fredrikson et al., 2015; Zhang et al., 2020; Li et al., 2022a); Membership Inference Attacks, analyzing model predictions to determine whether specific data records were used in training (Hu et al., 2022; Dayal et al., 2023; Wang et al., 2023b). These attacks compromise the security of the FL system, necessitating the development of more secure methods for leveraging pre-trained foundation models in FL settings.

FedBaF addresses these security challenges by not sharing the foundation model with clients during the weight initialization stage. Instead, it dynamically integrates the foundation model's pre-trained weights during the aggregation phase of each training round. We show that FedBaF improves privacy while matching or exceeding the performance achieved by weight initialization.

3 METHODOLOGY

In this section, we introduce FedBaF, whose approach is illustrated in Figure 2. FedBaF involves the server repeatedly leveraging pre-trained foundation model weights throughout the aggregation phases of the FL training process. For example, the pre-trained weights corresponding to feature extraction layers provide valuable representation mappings that guide the new model's feature extractor during training.

To mimic the performance gains of weight initialization, the server uses the foundation model as a strong anchor in the earlier FL rounds. To enable the FL global model to evolve and fit the clients' data as FL training continues, the server assigns rapidly decaying importance to the foundation model that is on the order of $1/\sqrt{t}$ and proportional to the change in model parameters caused by client updates. To further maintain foundation model confidentiality, the server randomly samples the aggregation weights (or importance) of the foundation model from a uniform distribution in each round. These aspects of FedBaF maintain foundation model privacy while enabling further application-specific tuning to achieve performance on par or exceeding that of weight initialization methods.

Algorithm 1 Federated Learning Aggregation Biased by a Foundation Model (FedBaF).

1: Initialize global model weights $\mathbf{w}_0$
2: **for** each round $t = 0, 1, 2, \ldots, T$ **do**
3: $m \leftarrow \max(C \cdot K, 1)$
4: $S_t \leftarrow$ (random set of m clients)
5: **for** each client $k \in S_t$ **in parallel do**
6: $\mathbf{w}_{t+1}^k \leftarrow \text{ClientUpdate}(\mathbf{w}_t, D_k)$
7: **end for**
8: $\mathbf{w}'_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{\sum_{k \in S_t} n_k} \mathbf{w}_{t+1}^k$
9: $\tau_t \leftarrow \frac{\left\| \frac{\mathbf{w}'_{t+1}}{\|\mathbf{w}'_{t+1}\|} - \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} \right\|}{\sqrt{t+1}}$
10: $\alpha_t \leftarrow \frac{\psi}{\tau_0} \mathrm{U}(1, 2)$
11: $\mathbf{w}_{t+1} \leftarrow \frac{1}{1+\alpha_t \tau_t} (\mathbf{w}'_{t+1} + \alpha_t \tau_t (\mathbf{w}_{pre} \setminus \mathbf{w}_t))$
12: **end for**
13: **ClientUpdate**$(\mathbf{w}, D)$
14: Initialize local model weights with $\mathbf{w}$
15: Update local model weights using local data D
16: **return** updated model weights

Algorithm 1 describes how FedBaF fits into the traditional FL framework by incorporating foundation model weights during aggregation, as illustrated in *Lines 9-11* (McMahan et al., 2017). FedBaF is versatile and can be embedded into many existing FL algorithms, e.g., SCAFFOLD (Karimireddy et al., 2020), FedProx (Li et al., 2020), FedAdam (Reddi et al., 2021), or other FL strategies, by modifying their aggregation methods (*Lines 8-11*) and using their existing **ClientUpdate**$(\mathbf{w}, D)$ logic for clients' local training in *Line 13*.

Modifying FL aggregation. FedBaF's aggregation process in each training round begins with the aggregation step of an existing FL algorithm, which, as mentioned above, includes FedAvg, SCAFFOLD, FedProx, and FedAdam. To illustrate an example with FedAvg, *Line 8* of Alg. 1 uses FedAvg's aggregation step and computes a weighted sum of the updated model parameters from each client. After this aggregation, *Line 11* incorporates the pre-trained model weights ($\mathbf{w}_{pre}$) into the global FL model, controlled by the factor τ_t defined in *Line 9*.

Here, $(\mathbf{w}_{pre} \setminus \mathbf{w}_t)$ refers to *the subset of layers from the foundation model* ($\mathbf{w}_{pre}$) that have the same architecture as the corresponding layers in the global FL model ($\mathbf{w}_t$), ensuring that only compatible layers are used during aggregation. When the foundation and FL models have the same architecture except that the input and output layers (i.e., first and last layers) differ due to variations in input features or the number of classes, FedBaF excludes these input and output layers from the aggregation and integrates only the shared intermediate (hidden) layers. The differing layers are randomly initialized and then trained using data from the client pool, as in standard FL.

Foundation and Global Model Architecture Mismatch. In practice, the foundation model and the global model architectures may differ beyond the input and output layers. These differences can arise in terms of the number of layers or the number of parameters per layer. Since foundation models are typically larger than global FL models – consistent with their role as highly expressive networks pre-trained on extensive data (Meng et al., 2023; Awais et al., 2025) – we consider the following two cases:

• **Foundation model with more layers:** When the foundation model is deeper than the global FL model, only a subset of its layers is used during aggregation. We take advantage of the fact that most large models have layers grouped into *sections*. Here, a section is a contiguous subset of layers within a model that shares similar structural properties, such as the number of parameters, functional roles, or connectivity patterns. FedBaF selects and matches sections between the foundation and global models, prioritizing feature extraction sections near the input layer. Within each matched section, only layers that align with the global model's architecture are integrated into FL training. Methods for selecting compatible layers during aggregation are explored in Park and Joe-Wong (2024); Xu et al. (2024).

• **Foundation model with more parameters per layer:** If the foundation model has layers with more parameters than the global model, only a subset of parameters within each section and layer is selected to match the global FL model. Since different sections may have varying parameter distributions, the aggregation is performed iteratively per section and per layer. To consider these mismatches, *Line 11* of Alg. 1 is expanded:

for each section s in shared sections between $\mathbf{w}_{pre}$ and $\mathbf{w'}_t$ **do**
  for each layer l in section s **do**
    for each parameter subset p in layer l up to $P_t^{(s,l)}$ parameters **do**
      $\mathbf{w}_{t+1}^{(s,l,p)} \leftarrow \frac{1}{1+\alpha_t \tau_t} \left(\mathbf{w'}_{t+1}^{(s,l,p)} + \alpha_t \tau_t \mathbf{w}_{pre}^{(s,l,p)} \right)$
    end for
  end for
end for

Here, $P_t^{(s,l)}$ denotes the number of parameters selected per layer l within section s. Only the first $P_t^{(s,l)}$ parameters in each layer are incorporated into the global model.

FedBaF can be analogously modified to handle the case where the foundation model is smaller than the global FL model. By applying these modifications, FedBaF can seamlessly adapt to different network architectures, ensuring that foundation model knowledge is effectively transferred while maintaining structural compatibility with the FL model.

Designing τ_t. Our careful design of τ_t uses the $L2$ norm of the difference between consecutive normalized weights of $\mathbf{w'}_{t+1}$ and $\mathbf{w}_t$, divided by $\sqrt{t+1}$. This change in the model's weights between rounds reflects how much the global model adapts to new client updates. The normalization prevents τ_t from becoming too large. The factor $\sqrt{t+1}$ ensures that, as training progresses, the influence of $\mathbf{w}_{pre}$ gradually diminishes, but not too quickly, and $\mathbf{w}_{t+1}$ approaches the improving averaged weights $\mathbf{w'}_{t+1}$. This strategy is critical to keeping the global model flexible and effective, especially when client data differs from the data used to train the foundation model (Karimireddy et al., 2020).

Depending on the network architectures (e.g., the number of weights or scale of the initialized weights), the scale of τ_t can vary. In non-IID situations and during adversarial attacks, the factor τ_t becomes significant. In particular, a large τ can indicate the presence of non-IID data or an attack, as such scenarios often result in large differences in consecutive weight updates. To keep τ_t within a suitable range, we introduce the parameter α_t in *Line 10*, which depends on τ_0 and the hyper-parameter ψ. We empirically find that setting α_t such that $\alpha_t\tau_0$ is less than 2 in the initial round ($t = 0$) prevents excessively large values that could overly bias the global model towards the foundation model. A lower bound of 1 for $\alpha_t\tau_0$ also ensures that the minimum impact of the foundation model is significant for small t. We thus ensure the influence of the foundation model in the critical initial training stages, while still allowing the global model to adapt as training progresses.

Designing α_t. Sampling α_t from the uniform distribution $\frac{\psi}{\tau_0}\mathrm{U}(1,2)$ for every round makes it difficult for clients to reverse-engineer the foundation model, thus meeting FedBaF's model security guarantees. To see this, *Line 11* can be rearranged for $\mathbf{w}_{pre}$ as

$$\mathbf{w}_{pre} = \frac{(1+\alpha_t\tau_t)\mathbf{w}_{t+1} - \mathbf{w'}_{t+1}}{\alpha_t\tau_t}.$$

In the worst-case scenario, where all local clients are malicious and collaborating to extract the foundation model's weights, they can access τ_t, $\mathbf{w}_{t+1}$, and $\mathbf{w'}_{t+1}$. However, because α_t is randomly chosen in each round t and known only to the server, the foundation model's weights cannot be extracted. If α were static, even if the server did not disclose it, malicious clients could determine this constant value by solving the residual equations from two successive rounds,

$$\frac{(1+\alpha\tau_t)\mathbf{w}_{t+1} - \mathbf{w'}_{t+1}}{\alpha\tau_t} = \frac{(1+\alpha\tau_{t+1})\mathbf{w}_{t+2} - \mathbf{w'}_{t+2}}{\alpha\tau_{t+1}}.$$

This would eventually reveal the foundation model's weights. *More details, along with empirical analysis on the role of α_t and the security advantages of FedBaF are provided in Appendix G.*

We formally examine this idea in the next section.

4 THEORETICAL ANALYSIS

In this section, we focus on deriving performance guarantees for FedBaF, focusing on its convergence properties. Sec. 4.1 examines the general convergence behavior of FedBaF, while Sec. 4.2 shows specifically on how FedBaF manages convergence in the presence of diverse, non-IID local client data distributions.

The following notation, problem setup, and assumptions are used throughout our analysis. Given m clients, let the kth device's training data be drawn from $\mathcal{D}_k$. The FL problem can be formulated as the following global objective,

$$\min_{\mathbf{w}} \frac{1}{\sum_{k=1}^{m} n_k} \sum_{k=1}^{m} n_k f_{\mathcal{D}_k}(\mathbf{w}), \tag{1}$$

where $\mathbf{w}$ are model (usually, deep neural network) weights and the $f_{\mathcal{D}_k}$ are L-smooth local objective functions. The convergence analysis presented in this section also makes the following standard assumptions made by (Karimireddy et al., 2020) and detailed in Appendix E. Each client locally optimizes $\mathbf{w}$ using stochastic gradient descent, where the stochastic gradients are (i) unbiased and (ii) have bounded variance. We also assume (iii) bounded gradient dissimilarity: the norm of the difference between the gradient of the global objective and the gradients computed using different local objective functions is bounded. Lastly, we assume that (iv) the foundation model has the same architecture as the global model, ensuring compatibility during aggregation. See Appendix E.1.1 for mathematical details regarding the assumptions.

4.1 General Convergence Analysis

Proposition 1. *Let $\mathbf{w}^*$ be a (bounded) local minimum of the global objective function in* (1)*. Consider an FL algorithm that converges to $\mathbf{w}^*$ and let $\mathbf{w}'_t$ be its global model in each training round t. Suppose we run the same algorithm but using FedBaF for the aggregation, and let $\mathbf{w}_t$ be the FedBaF global model at round t. Let*

α_t satisfy

$$\alpha_t < \frac{2\|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2}{(\|\mathbf{w}_{pre} - \mathbf{w}^*\|^2 - \|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2)\tau_t} \quad (2)$$

for all t where $\|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2 < \|\mathbf{w}_{pre} - \mathbf{w}^*\|^2$. Then $\forall t\ \|\mathbf{w}_t - \mathbf{w}^*\| < \|\mathbf{w}'_t - \mathbf{w}^*\|$.

This means that, at any given round t, FedBaF's model weights are closer to $\mathbf{w}^*$.

Using the same restrictions on local and global learning rates placed by the FedAvg convergence analysis in (Karimireddy et al., 2020), the aforementioned bounded gradient variance, bounded gradient dissimilarity, and L-smoothness assumptions ensure that our method converges to $\mathbf{w}^*$ faster than FedAvg. Similar convergence rate arguments for other FL methods with appropriately modified aggregation can be shown, as discussed in Sec. 3.

4.2 Effectiveness of FedBaF with Diverse Client Data

This section shows the impact of integrating a foundation model close to the optimal weights on the learning process and convergence behavior in non-IID settings.

In round t, client k uses multiple SGD steps to update the global model $\mathbf{w}_t$ and obtains the local model $\mathbf{w}_t^k$. Letting S_t represent the randomly selected set of active clients at time t, we define δ_t as the maximum deviation of the client models from $\mathbf{w}^*$:

$$\delta_t := \max_{k \in S_t} \|\mathbf{w}_t^k - \mathbf{w}^*\|.$$

Aggregating the updated models from clients according to Alg. 1 *Lines 8-10* forms the global model $\mathbf{w}'_t$. By the triangle inequality, we get

$$\|\mathbf{w}'_t - \mathbf{w}^*\| \leq \delta_t.$$

Assumption: Foundation Model Proximity. The foundation model's pre-trained weights, $\mathbf{w}_{\text{pre}}$, are close to the optimal weights $\mathbf{w}^*$, i.e., $\|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\| \leq \gamma$ for a small $\gamma > 0$. Furthermore, we assume that $\gamma \leq \delta_t$ for earlier rounds (small t which makes $\tau_t \gg 0$), i.e., the foundation model is closer to the optimal model than clients' local weights.

These are reasonable assumptions in practice since selecting a foundation model with a large γ would correspond to selecting an unsuitable foundation model that hampers the training process.

Proposition 2. *Let $\mathbf{w}^*$ be a (bounded) local minimum of the global objective function in* (1)*. Consider an FL algorithm that converges to $\mathbf{w}^*$ and let $\mathbf{w}'_t$ be its global model. Consider FedBaF based on the same FL algorithm (with appropriately modified client updates and Lines 8-10 in Alg. 1) and let $\mathbf{w}_t$ be the FedBaF global model. FedBaF's global model error has an upper bound of $\|\mathbf{w}_t - \mathbf{w}^*\| \leq \frac{\delta_t + \alpha_t \tau_t \gamma}{1 + \alpha_t \tau_t} < \delta_t$.*

Similar to (29) in Sec. 4.1, we bounded the distance between the FedBaF global model $\mathbf{w}_t$ and $\mathbf{w}^*$ in terms of δ_t. Prop. 2 shows that the integration of the foundation model not only helps in stabilizing the learning process but also accelerates the convergence rate. The foundation model acts as a stabilizing factor and reduces the impact of this variance on the global model's convergence. This is particularly significant in the early stages of learning with non-IID data, when local models' weights are more prone to diverge from each other.

5 EXPERIMENTAL EVALUATIONS

In this section, we conduct a detailed evaluation of FedBaF's performance on both local and global test datasets, comparing its performance to the **no foundation model** and **weight initialization** baseline algorithms for training FL models.

• **No foundation model**: The global FL model is trained from scratch without weight initialization or FedBaF (i.e., no use of foundation models).

• **Weight initialization**: The global model's initial weights are set to equal the foundation model's weights, and the FL training then proceeds as usual.

We aim to **1)** illustrate that FedBaF offers security advantages over weight initialization while attaining equivalent performance. Furthermore, by verifying how τ_t in line 9 of Algorithm 1 converges to 0, we also **2)** establish FedBaF's ability to effectively adapt the influence of the foundation model.

More detailed testing results are provided in Appendix C, including the **1)** computational efficiency of FedBaF and **2)** additional evaluations using foundation models of varying quality trained on different amounts of data and real-world foundation model weights that are publicly available online.

5.1 Experimental Setup

Our experiments with popular image classification tasks encompass experiments using the CIFAR-10 and Rome Weather Image (Vaz, 2021) datasets. We use Pre-ResNet and Vision Transformer (Dosovitskiy et al., 2021) architectures as Vision Transformers are popular foundation model architectures known for achieving remarkable performance (Zhou et al., 2024) for ImageNet challenges (Dosovitskiy et al., 2021). We

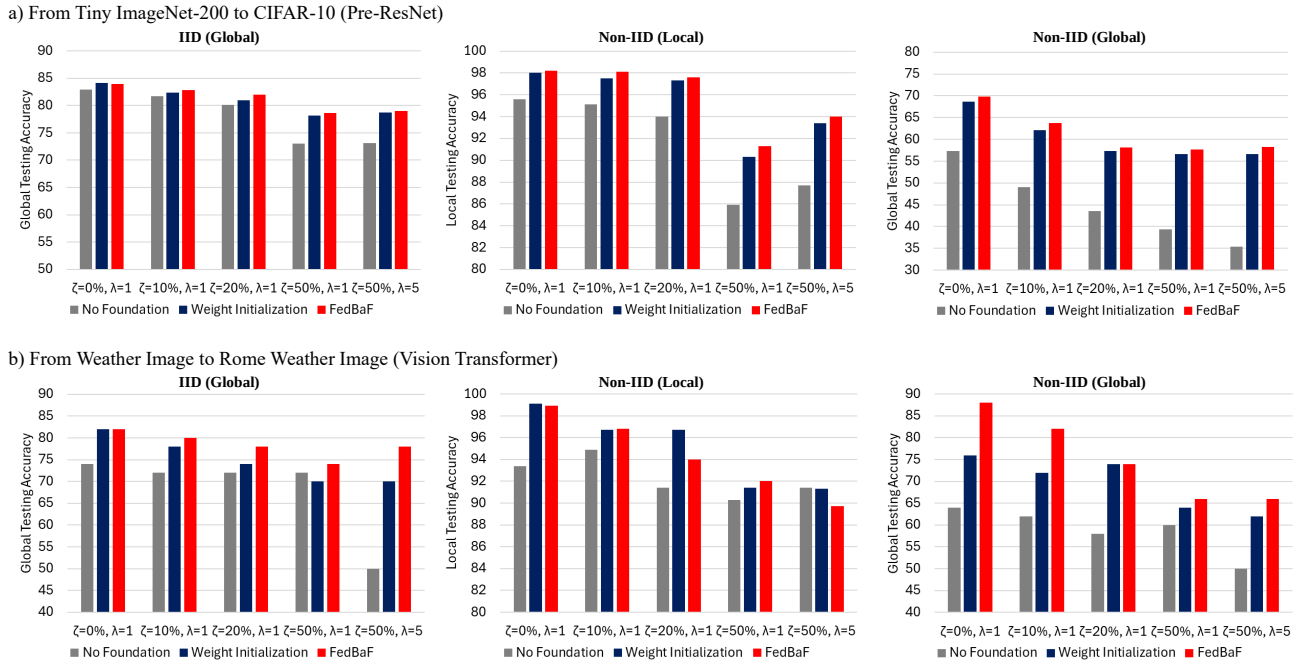

Figure 3: FedBaF maintains higher test accuracy when used with **FedAvg** with different proportions of malicious clients, ζ (0%, 10%, 20%, 50%), and attack intensity, λ (1, 5), executing misclassification attacks, under IID and non-IID settings. Three different foundation models, trained with different datasets, are used for three tasks. Red, blue, and gray bars respectively represent FedBaF, weight initialization, and no foundation model cases.

train the foundation models on the Tiny ImageNet-200 and Weather Image (Xiao, 2021) datasets. Our evaluations consider both IID and non-IID settings (see Appendix A). To demonstrate FedBaF's generalizability to other tasks, we also evaluate it on a next-word prediction task using a Transformer language model pre-trained on the WikiText-2 dataset and tested on the Penn Treebank dataset.

Attack setup. To assess security robustness of FedBaF, we randomly shuffle local data labels for a subset of clients, treating them as backdoor attackers aiming to induce misclassification. We also increase the attack intensity by varying the number of local epochs for malicious clients. For image classification tasks (Pre-ResNet and Vision Transformer), we increase the local epochs by a factor $\lambda > 1$, which introduces more bias from the initial global model and strengthens the attack's impact. For the language task (Transformer), we decrease the local epochs by a factor $1/\lambda < 1$ to prevent convergence to a small loss, ensuring the calculated perplexity remains high regardless of misclassification, thereby intensifying the attack. We vary the proportion of attacking clients, ζ, and evaluate algorithm resiliency when 0%, 10%, 20%, and 50% of the client base are attackers.

Evaluation metrics. To evaluate testing performance, we calculate the *global testing accuracy* using a global test dataset after the aggregation phase of an FL round. In non-IID settings, we also use local test datasets that are extracted from the global test dataset and reflect the class distribution of the local clients. After local training and prior to aggregation, we test the local models to determine an *average local testing accuracy*. For the Transformer model, we use *global perplexity* to assess the performance of the global language model. Perplexity is **inversely** related to how well a probability model predicts a sample.

Details of other deep neural network architectures employed in our experiments and additional training specifics are provided in Tables 1 and 2 in Appendix A.

5.2 Experimental Results

Figures 3, 4, and 5 display the extensive test accuracy/perplexity evaluation results for FedBaF and our two baselines (*no foundation model* and *weight initialization*). We evaluate the three methods using both FedAvg and FedProx (Li et al., 2020) as the base FL training algorithms. We incorporate FedBaF into FedProx by modifying *Line 8* and the ClientUpdate routine in Algorithm 1, including both IID and non-IID settings, with one non-adversarial scenario and four adversarial scenarios. We use FedProx's aggregation step: $\mathbf{w}'_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{\sum_{k \in S_t} n_k} \mathbf{w}^k_{t+1} - \mu(\mathbf{w}^k_{t+1} - \mathbf{w}_t)$. Here, μ represents the regularization term that con-

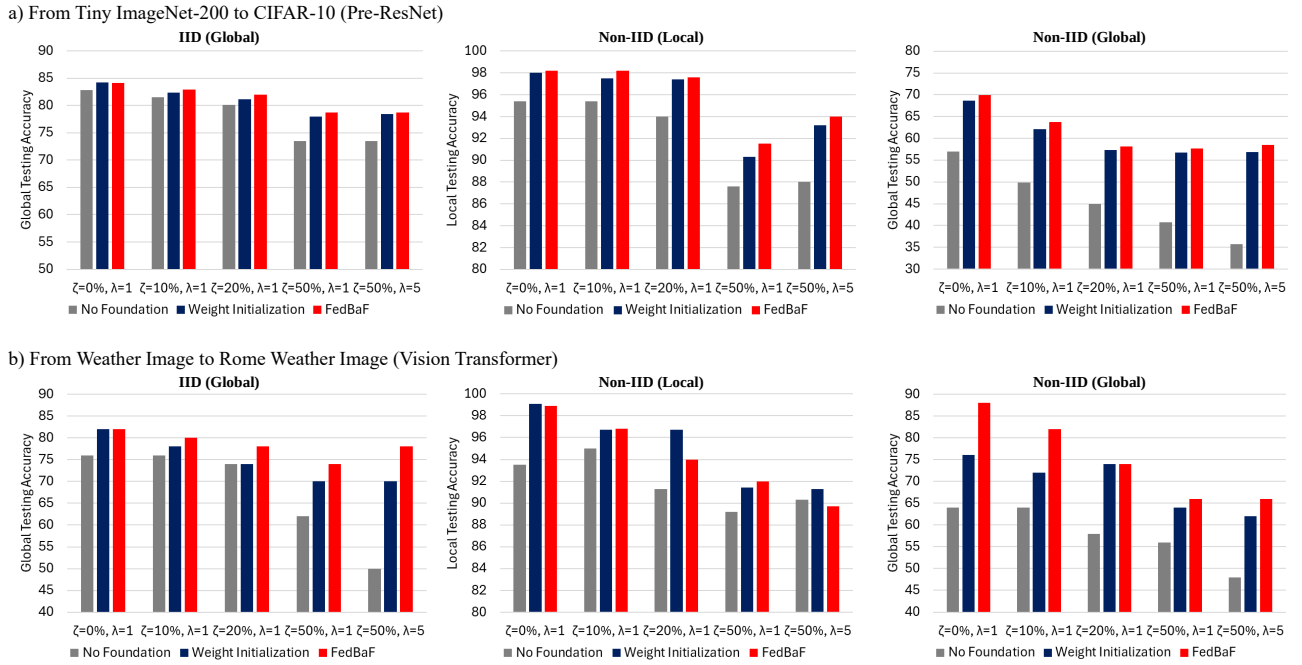

Figure 4: FedBaF maintains higher test accuracy than both baselines when used with **FedProx** with different proportions of malicious clients, attack intensity, and IID as well as non-IID settings. All other settings are identical to those in Figure 3.

trols the trade-off between the local and global objectives. FedBaF then incorporates the foundation model weights as in *Lines 9-11* of Algorithm 1.

5.2.1 Testing FedBaF Performance

In non-adversarial scenarios, FedBaF showcased superior testing performance compared to the no foundation model and weight initialization methods across both IID and non-IID configurations.

Figures 3 and 4 respectively show test accuracies of all three methods using FedAvg and FedProx for both Pre-ResNet and the Vision Transformer. In comparison to Pre-ResNet trained with no foundation model, FedBaF improved global model accuracy by 1.3% for FedAvg and 1.6% for FedProx in IID scenarios, and by 21.8% for FedAvg and 22.6% for FedProx in non-IID scenarios. For the Vision Transformer, FedBaF improved global performance relative to no foundation model by 10.8% for FedAvg and 0.0% for FedProx in IID settings and by 37.5% for FedAvg and 15.8% for FedProx in non-IID settings. We observe that **both FedAvg and FedProx benefit from FedBaF's inclusion of the foundation model**, with particular benefits in more challenging scenarios with non-IID client data. Incorporating the foundation model mitigates slower FL convergence caused by non-IID data.

The weight initialization method, which also incorporates a foundation model but does not keep it private, exhibits similar performance gains as FedBaF compared to training without a foundation model. For example, on Pre-ResNet weight initialization exhibited global performance gains of 19.7% for FedAvg and 20.4% for FedProx in non-IID scenarios, while on Vision Transformers it achieves gains of 18.8% for both FedAvg and FedProx in non-IID scenarios.

In the next-word prediction task using a Transformer, FedBaF significantly outperformed training with no foundation model, reducing perplexity by 76.0% with FedAvg, whereas weight initialization yielded a 67.8% decrease relative to training without a foundation model with FedAvg.

Collectively, these findings indicate **negligible differences between the test accuracies attained by FedBaF and those achieved with weight initialization**. Simultaneously, **FedBaF achieves privacy advantages** as, unlike weight initialization, it does not reveal the foundation model weights to FL clients. Moreover, FedBaF showed better testing performance than the weight initialization or no foundation model methods in 10 out of the 14 experiment settings.

In Appendix C.3, we provide additional examples of FedBaF achieving privacy and test performance improvements. They include comparisons with FedAdam (Reddi et al., 2021), a state-of-the-art FL method leveraging pre-trained models.

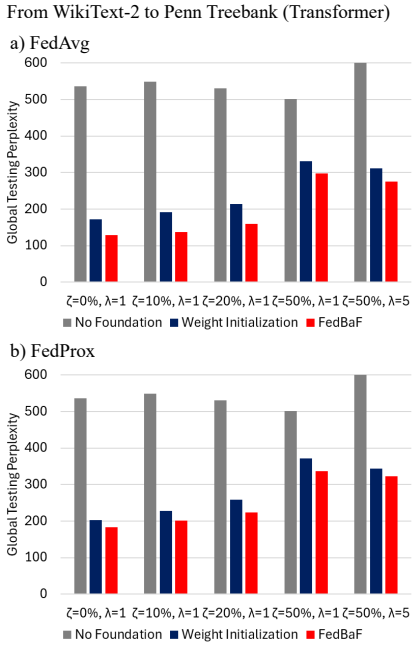

Figure 5: FedBaF maintains lower test perplexity when used with **FedAvg** and **FedProx** with different proportions of malicious clients and attack intensities. **Note: lower perplexity is better**

5.2.2 FedBaF's Robustness to Attacks

FedBaF remains effective in maintaining robustness when faced with misclassification attacks and shows a more modest performance degradation in comparison to both baseline methods.

We evaluate the robustness of FedBaF and our two baselines in the presence of adversarial clients. As the proportion of attacking clients (ζ in Figures 3, 4, and 5) increases, test accuracy declines in all cases. This matches our intuition since neither FedBaF nor the two baselines are designed to perfectly defend against these attacks, which become more effective as more clients act as attackers.

The test accuracy for the no foundation model baseline method drops significantly when 50% of the clients are attackers and the attack intensity $\lambda = 5$. For Pre-ResNet under such attacks, the global performance drop, when compared to the case with no attackers, is 11.8% for FedAvg and 11.2% for FedProx in IID scenarios and 38.2% for FedAvg and 37.4% for FedProx in non-IID scenarios. With the Vision Transformer, these accuracy drops are 32.4% for FedAvg and 34.2% for FedProx in IID and 21.9% for FedAvg and 25.0% for FedProx in non-IID scenarios. Thus, **training with no foundation model results in vulnerability to attacks in both IID and non-IID settings**. This matches our expectations since it has no built-in defenses.

In contrast, **FedBaF experiences a much more modest performance decline under attack**, demonstrating its robustness. For Pre-ResNet, FedBaF's global performance decreases by 6.0% for FedAvg and 6.4% for FedProx in IID settings and by 16.5% for FedAvg and 16.3% for FedProx in non-IID settings, where the decrease is again measured for the most intense attack ($\zeta = 50\%, \lambda = 5$) relative to no attack. These accuracy drops are *less than half* of those experienced by the no foundation model method. With the Vision Transformer, FedBaF's global performance decreases by only 4.9% for FedAvg and FedProx in IID settings and by 25% for both FedAvg and FedProx in non-IID settings.

Finally, we comment that the results also show that the weight initialization method, which naïvely incorporates the foundation model and reveals the foundation model weights to clients, does confer some robustness to attacks. Weight initialization shows a performance decrease of 6.4% for FedAvg and 6.9% for FedProx in IID settings and 17.5% for FedAvg and 17.2% for FedProx in non-IID settings globally for Pre-ResNet. With the Vision Transformer, the decreases are 14.6% for both FedAvg and FedProx in IID settings and 18.4% for both FedAvg and FedProx in non-IID settings globally.

These findings demonstrate that, similar to weight initialization, **FedBaF offers considerable attack robustness** compared to training with no foundation model. In 8 out of 12 cases, FedBaF suffers the minimal loss in performance when compared with the other two baselines, indicating its superior effectiveness in maintaining robustness under adversarial conditions.

6 CONCLUSION

This paper introduced Federated Learning Aggregation Biased by a Foundation Model (FedBaF). FedBaF enhances adaptability and security in dynamic FL scenarios without sharing the foundation model with clients. This is crucial in environments with ever-changing data and non-IID scenarios, where foundation models are used across several domains as seed models. Our findings show that FedBaF increases resilience against adversarial attacks while matching or outperforming traditional weight initialization performance in both IID and non-IID settings.

Acknowledgments

The authors are partially supported by NSF Grants CCF-2327905 and CNS-2106891. Srinivasa Pranav is partially supported by an NSF Graduate Research Fellowship (DGE-1745016, DGE-2140739) and an ARCS Fellowship.

References

Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H., and Khan, F. S. (2025). Foundation Models Defining a New Era in Vision: a Survey and Outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T. F., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S. P., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J. F., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y. H., Ruiz, C., Ryan, J., R'e, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K. P., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M. A., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Chen, H.-Y., Tu, C.-H., Li, Z., Shen, H. W., and Chao, W.-L. (2023). On the Importance and Applicability of Pre-Training for Federated Learning. In *International Conference on Learning Representations*.

Dayal, S., Alhadidi, D., Abbasi Tadi, A., and Mohammed, N. (2023). Comparative Analysis of Membership Inference Attacks in Federated Learning. In *Proceedings of the 27th International Database Engineered Applications Symposium*, pages 185–192.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Duan, M., Liu, D., Ji, X., Wu, Y., Liang, L., Chen, X., Tan, Y., and Ren, A. (2021). Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674.

Fanì, E., Camoriano, R., Caputo, B., and Ciccone, M. (2024). Accelerating Heterogeneous Federated Learning with Closed-form Classifiers. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13029–13048. PMLR.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al. (2021). Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., and Zhang, X. (2022). Membership inference attacks on machine learning: A survey. *ACM Computing Surveys*, 54(11s):1–37.

Joshi, M., Pal, A., and Sankarasubbu, M. (2022). Federated Learning for Healthcare Domain - Pipeline, Applications and Challenges. *ACM Trans. Comput. Healthcare*, 3(4).

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR.

Kenneweg, T., Kenneweg, P., and Hammer, B. (2024). Foundation Model Vision Transformers are Great Tracking Backbones. In *International Conference on Artificial Intelligence, Computer, Data Sciences and Applications*, pages 1–6. IEEE.

Kim, T., Li, J., Madaan, N., Singh, S., and Joe-Wong, C. (2023). Adversarial Robustness Unhard-

ening via Backdoor Attacks in Federated Learning. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*.

Lee, S., Zhang, T., and Avestimehr, A. S. (2023). Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8491–8499.

Legate, G., Bernier, N., Page-Caccia, L., Oyallon, E., and Belilovsky, E. (2023). Guiding The Last Layer in Federated Learning with Pre-Trained Models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69832–69848. Curran Associates, Inc.

Li, J., Rakin, A. S., Chen, X., He, Z., Fan, D., and Chakrabarti, C. (2022a). Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10194–10202.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*.

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

Li, Z., Ren, K., Jiang, X., Li, B., Zhang, H., and Li, D. (2022b). Domain generalization using pretrained models without fine-tuning. *arXiv preprint arXiv:2203.04600*.

Lyu, L., Yu, H., Zhao, J., and Yang, Q. (2020). *"Threats to Federated Learning"*, pages 3–16. Springer International Publishing, Cham.

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

Meng, F., Shao, W., Jiang, C., Zhang, K., Qiao, Y., Luo, P., et al. (2023). Foundation model is efficient multimodal multitask model selector. *Advances in Neural Information Processing Systems*, 36:33065–33094.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., and Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658.

Nguyen, J., Malik, K., Sanjabi, M., and Rabbat, M. (2022). Where to begin? Exploring the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*.

Nikolenko, S. I. (2021). *Synthetic data for deep learning*, volume 174. Springer.

Park, J.-I. and Joe-Wong, C. (2024). Federated Learning with Flexible Architectures. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 143–161. Springer.

Pranav, S. and Moura, J. M. F. (2023). Peer-to-Peer Learning+Consensus with Non-IID Data. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pages 709–713. IEEE.

Pranav, S. and Moura, J. M. F. (2024). Peer-to-Peer Deep Learning for Beyond-5G IoT. In *2024 58th Asilomar Conference on Signals, Systems, and Computers*. IEEE.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. (2021). Adaptive Federated Optimization. In *International Conference on Learning Representations*.

Siew, M., Zhang, H., Park, J.-I., Liu, Y., Ruan, Y., Su, L., Ioannidis, S., Yeh, E., and Joe-Wong, C. (2024). Fair Concurrent Training of Multiple Models in Federated Learning. *CoRR*, abs/2404.13841.

Singh, A., Vepakomma, P., Gupta, O., and Raskar, R. (2019). Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145*.

Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T., and Jiang, J. (2022). Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems*, 35:19332–19344.

Vaz, R. (2021). Rome Weather Classification. `https://www.kaggle.com/datasets/rogeriovaz/rome-weather-classification`.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. (2020). Federated Learning with Matched Averaging. In *International Conference on Learning Representations*.

Wang, K., He, Q., Chen, F., Chen, C., Huang, F., Jin, H., and Yang, Y. (2023a). FlexiFed: Personalized federated learning for edge clients with heterogeneous model architectures. In *Proceedings of the ACM Web Conference 2023*, pages 2979–2990.

Wang, X., Wang, N., Wu, L., Guan, Z., Du, X., and Guizani, M. (2023b). GBMIA: Gradient-based

Membership Inference Attack in Federated Learning. In *ICC 2023-IEEE International Conference on Communications*, pages 5066–5071. IEEE.

Xiao, H. (2021). Weather phenomenon database (WEAPD). `https://doi.org/10.7910/DVN/M8JQCR`.

Xu, P., Zhu, X., and Clifton, D. A. (2023). Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12113–12132.

Xu, Z., Chen, Y., Vishniakov, K., Yin, Y., Shen, Z., Darrell, T., Liu, L., and Liu, Z. (2024). Initializing Models with Larger Ones. In *The Twelfth International Conference on Learning Representations*.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Yu, J., Wang, Y., Zhao, C., Ghanem, B., and Zhang, J. (2023). FreeDoM: Training-Free Energy-Guided Conditional Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184.

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-IID data. *arXiv preprint arXiv:1806.00582*.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. (2024). A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT. *International Journal of Machine Learning and Cybernetics*, pages 1–65.

Zhuang, W., Chen, C., and Lyu, L. (2023). When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, see Section 4.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, see Appendix C.4.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, this will be included in the supplementary.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes, this will be included in the supplementary.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, this will be included in the supplementary.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, this will be included in the supplementary.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix Overview

This appendix provides additional details and results to complement the main text, offering further insight into the experimental setup, theoretical analysis, and security considerations for FedBaF. The appendix is organized as follows:

Experimental Setup in Section A

This section details the experimental configurations, including data distributions, network architectures for vision and language tasks, and hyper-parameters under IID and non-IID settings.

Formulas for Evaluating Computational Complexity in Section B

We present the mathematical formulas used to compute the computational complexity of the FedBaF algorithm, specifically focusing on multiply-accumulate (MAC) operations across clients and training rounds.

Additional Experimental Evaluations in Section C

This section includes supplementary experimental results, analyzing the effect of varying foundation model quality on FedBaF's performance and comparisons using the official pre-trained foundation model. Computational complexity is also compared to scenarios without foundation models and weight initialization. Additionally, we include further experiments employing FedAdam (Reddi et al., 2021).

Training Curves in Section D

We provide training curves that display the progression of model accuracy over training epochs for the Pre-ResNet and Vision Transformer models, illustrating comparisons between FedBaF, weight initialization, and without foundation model cases under IID and non-IID conditions.

Convergence Analysis in Section E

This section provides a theoretical analysis of FedBaF's convergence properties, detailing how the algorithm performs under different client data distributions and demonstrating the theoretical guarantees for its performance.

Proofs of Propositions in Section F

In this section, we include the formal proofs of Propositions 1 and 2, outlined in the theoretical analysis in Section 4, which support the claims made regarding FedBaF's performance and convergence behavior.

Security Analysis in the Presence of Adversarial Attacks in Section G

This section provides an in-depth security analysis of FedBaF, focusing on its robustness against adversarial attacks such as misclassification and backdoor attacks. We compare FedBaF's resilience to malicious clients with traditional methods, showing how FedBaF mitigates the negative impact of attacks and preserves global model integrity.

A Experimental Setup

Table 1: The specific conditions under which our experiments were conducted, including data distribution and model training settings.

Evaluation	**Pretrained Model Training**			**Performance Comparison**						
				From Tiny ImageNet-200		**From Weather Image**		**From WikiText-2**	**From ImageNet (PyTorch)**	
				IID	**Non-IID**	**IID**	**Non-IID**	-	**IID**	**Non-IID**
Number of clients	1			100		10		100	100	
Fraction of active clients C	1			0.1						
Number of classes for each client	100	11	-	10	2	5	2	-	10	2
Number of samples for each client	50,000 ~ 100,000	2,500 ~ 5,000	640,000 ~ 1,280,000	125 ~ 250		10 ~ 20		7,680 ~ 8960	125 ~ 250	
Data	Tiny ImageNet-200	Weather Image	WikiText-2	CIFAR-10		Rome Weather Image		Penn Treebank	CIFAR-10	
Model	Pre-ResNet	Vision Transformer	Transformer	Pre-ResNet		Vision Transformer		Transformer	Vision Transformer	
Local epochs E	300	300	300	5						
Local mini-batch size B	128			50		50		128	125	
Communication rounds	1			200	250	200	250	200	200	250
Optimizer	SGD									
Momentum	0.9									
Weight decay	1e-4									
Learning rate η	0.1			0.01		0.01		0.1	0.01	
Learning rate 0.1x decay schedule	[150, 225]	[150, 225]	Not applied	Not applied						
Batch normalization layer	Non-static									
μ **for proximal term in FedProx**	Not applied			0.01				Not applied	0.01	

In this section, we provide details about our experimental setup and network architectures. For vision tasks using Pre-ResNets and Vision Transformers, we conduct evaluations in both IID and non-IID environments. In **IID settings**, each client's data distribution is uniform across all classes, with an equal number of samples from each class. In **non-IID settings**, clients receive samples from only 20% of the dataset's classes for CIFAR-10 and 50% of the dataset's classes for Rome Weather Image, but maintain an equal number of samples for each class they have. *During local training in these settings, clients zero out logits for classes not present in their data.* For language tasks using Transformers, each client has specific numbers of tokenized words grouped sequentially. Details of the experimental settings can be found in Table 1.

For the network architecture configuration, details for Pre-ResNets and Transformers can be found in Table 2. For Vision Transformer, we used the standard ViT_B_16 model with no modifications except changing the last output layers according to the new data. This model was obtained from the PyTorch library. Additionally, we also tested FedBaF with official pre-trained foundation model weights that are available online (not developed by us). We specifically used the ImageNet pre-trained weights for a standard Vision Transformer from PyTorch's official model repository: (ViT_B_16_Weights.IMAGENET1K_SWAG_E2E_V1).

Table 2: This table presents the detailed structures of the neural networks (Pre-ResNet and Transformer) utilized in our Federated Learning experiments and making foundation models.

Model	**Section 1**	**Section 2**	**Section 3**	**Section 4**	**Section 5**	**Section 6**	**Section 7**	**Section 8**
Pre-ResNet	$[3\times3, 64]\times1$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$N_{classes}$-d fc		

Model	**Encoder**					**Decoder**				**Classifier**
	Attention	**FeedForward**	**Attention**	**FeedForward**			**Attention**	**FeedForward**	**Attention**	**FeedForward**	
Transformer	192-d fc 64-d fc	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	192-d fc 64-d fc	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	N_{token}-d fc	N_{token}-d fc	192-d fc 64-d fc	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times1$	192-d fc 64-d fc	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times1$	N_{token}-d fc

B Formulas for Evaluating Computational Complexity

In this section, we provide metrics for evaluating the computational complexity, as discussed in Section C.4.

To evaluate *computational complexity*, we track the number of multiply-accumulate (MAC) operations, denoted as MACS. MACS_k represents the number of MAC operations required to process all the local data samples held by client k. The average MAC per client, denoted as $\overline{\text{MACS}}$, is calculated by averaging the MACS values for all clients participating in a single round of FL:

$$\overline{\text{MACS}} = \frac{1}{m} \sum_{k=1}^{m} \text{MACS}_k$$

Here, m is the number of participating clients in the given round. For each local training epoch, the computational complexity, referred to as MACE (Multiply-Accumulate Complexity per Epoch), is calculated as:

$$\text{MACE} = m \times \tilde{n} \times \overline{\text{MACS}}.$$

Here, $\tilde{n}$ is the median number of data samples per client. This metric reflects the computational load incurred by m clients during local training in each epoch.

To compute the total computational load for the entire FL system, denoted as TMAC (Total Multiply-Accumulate Complexity), we multiply the number of local epochs E, the number of aggregation rounds T, and the previously calculated MACE:

$$\text{TMAC} = T \times E \times \text{MACE}.$$

This provides the total number of MAC operations required for the entire training process across all clients and rounds, accounting for both the number of local epochs and the aggregation rounds in the FL system.

C Additional Experimental Evaluations

In this section, we present additional experimental results that provide further insight into the performance of FedBaF across various tasks and scenarios. These evaluations focus on the impact of different qualities of foundation models, the application of the real pre-trained foundation model, and computational complexities. We compare the use of foundation models in both IID and non-IID settings with weight initialization and cases where no foundation model is used.

C.1 Differentiating the Quality of Pre-Trained Foundation Models

We conducted experiments to evaluate the generalized performance of FedBaF by varying the quality of foundation models. Tables 3, 4, and 5 present the results for image classification tasks using Pre-ResNet and Vision Transformer models, as well as a next-word prediction task using a Transformer model.

Table 3: Image classification test accuracy results for Pre-ResNet using the no foundation, weight initialization, and FedBaF methods (best of 3 trials).

FedAvg										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
50,000	ζ=0%, λ=1	82.9	**84.1**	84.0	95.6	98.0	**98.2**	57.3	68.6	**69.8**
	ζ=10%, λ=1	81.7	82.4	**82.8**	95.1	97.5	**98.1**	49.0	62.1	**63.8**
	ζ=20%, λ=1	80.1	81.0	**82.0**	94.0	97.3	**97.6**	43.6	57.3	**58.2**
	ζ=50%, λ=1	73.0	78.2	**78.6**	85.9	90.3	**91.3**	39.4	56.6	**57.7**
	ζ=50%, λ=5	73.1	78.7	**79.0**	87.7	93.4	**94.0**	35.4	56.6	**58.3**
100,000	ζ=0%, λ=1	82.9	85.9	**85.9**	95.6	98.4	**98.6**	57.3	71.9	**73.4**
	ζ=10%, λ=1	81.7	84.3	**84.8**	95.1	98.0	**98.3**	49.0	64.2	**67.0**
	ζ=20%, λ=1	80.1	83.5	**83.7**	94.0	97.5	**97.9**	43.6	59.8	**61.9**
	ζ=50%, λ=1	73.0	80.6	**80.8**	85.9	91.1	**92.0**	39.4	60.1	**60.8**
	ζ=50%, λ=5	73.1	80.6	**80.8**	87.7	94.0	**94.8**	35.4	56.4	**61.0**
FedProx										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
50,000	ζ=0%, λ=1	82.8	**84.2**	84.1	95.4	98.0	**98.2**	57.0	68.6	**69.9**
	ζ=10%, λ=1	81.5	82.4	**82.9**	95.4	97.5	**98.2**	49.8	62.1	**63.7**
	ζ=20%, λ=1	80.1	81.1	**82.0**	94.0	97.4	**97.6**	45.0	57.3	**58.1**
	ζ=50%, λ=1	73.5	78.0	**78.7**	87.6	90.3	**91.5**	40.8	56.7	**57.7**
	ζ=50%, λ=5	73.5	78.4	**78.7**	88.0	93.2	**94.0**	35.7	56.8	**58.5**
100,000	ζ=0%, λ=1	82.8	**85.9**	85.8	95.4	98.4	**98.6**	57.0	71.9	**73.4**
	ζ=10%, λ=1	81.5	84.4	**84.8**	95.4	98.0	**98.3**	49.8	64.4	**67.3**
	ζ=20%, λ=1	80.1	83.8	**84.0**	94.0	97.6	**97.9**	45.0	59.8	**61.9**
	ζ=50%, λ=1	73.5	80.5	**80.6**	87.6	90.9	**92.3**	40.8	59.9	**60.7**
	ζ=50%, λ=5	73.5	80.5	**80.9**	88.0	94.6	**94.8**	35.7	55.1	**60.5**

Table 4: Image classification test accuracy results for Vision Transformer using no foundation, weight initialization, and FedBaF methods (best of 3 trials).

FedAvg										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
2,500	ζ=0%, λ=1	74.0	82.0	**82.0**	93.4	**99.1**	98.9	64.0	76.0	**88.0**
	ζ=10%, λ=1	72.0	78.0	**80.0**	94.9	96.7	**96.8**	62.0	72.0	**82.0**
	ζ=20%, λ=1	72.0	74.0	**78.0**	91.4	**96.7**	94.0	58.0	74.0	**74.0**
	ζ=50%, λ=1	72.0	70.0	**74.0**	90.3	91.4	**92.0**	60.0	64.0	**66.0**
	ζ=50%, λ=5	50.0	70.0	**78.0**	**91.4**	91.3	89.7	50.0	62.0	**66.0**
5,000	ζ=0%, λ=1	**74.0**	72.0	72.0	93.4	98.0	**98.0**	64.0	**78.0**	76.0
	ζ=10%, λ=1	72.0	72.0	**74.0**	**94.9**	94.3	93.9	62.0	74.0	**76.0**
	ζ=20%, λ=1	72.0	**78.0**	72.0	91.4	90.6	**92.0**	58.0	**72.0**	68.0
	ζ=50%, λ=1	**72.0**	68.0	68.0	90.3	**92.3**	90.0	60.0	64.0	**68.0**
	ζ=50%, λ=5	50.0	**72.0**	70.0	91.4	**92.3**	90.5	50.0	66.0	**66.0**
FedProx										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
2,500	ζ=0%, λ=1	76.0	82.0	**82.0**	93.5	**99.1**	98.9	64.0	76.0	**88.0**
	ζ=10%, λ=1	76.0	78.0	**80.0**	95.0	96.7	**96.8**	64.0	72.0	**82.0**
	ζ=20%, λ=1	74.0	74.0	**78.0**	91.3	**96.7**	94.0	58.0	74.0	**74.0**
	ζ=50%, λ=1	62.0	70.0	**74.0**	89.2	91.4	**92.0**	56.0	64.0	**66.0**
	ζ=50%, λ=5	50.0	70.0	**78.0**	90.3	**91.3**	89.7	48.0	62.0	**66.0**
5,000	ζ=0%, λ=1	**76.0**	72.0	72.0	93.5	98.0	**98.0**	64.0	**78.0**	76.0
	ζ=10%, λ=1	**76.0**	72.0	74.0	**95.0**	94.3	93.9	64.0	74.0	**76.0**
	ζ=20%, λ=1	74.0	**78.0**	72.0	91.3	90.6	**92.0**	58.0	**72.0**	68.0
	ζ=50%, λ=1	62.0	68.0	**68.0**	89.2	**92.3**	90.0	56.0	64.0	**68.0**
	ζ=50%, λ=5	50.0	**72.0**	70.0	90.3	**92.3**	90.5	48.0	66.0	**66.0**

To assess the impact of foundation model quality, we varied the number of pre-trained samples used for each

model and assessed each method's performance under a varying number of malicious clients and attack intensity. Interestingly, larger sample sizes do not always lead to better results, as seen in Table 3, where excessive pre-training can negatively impact performance. This trend is further evidenced in Tables 4, 5. The reason behind this is likely due to overfitting or reduced adaptability to new tasks. Despite these variations, FedBaF consistently outperforms models without foundation models and delivers similar testing performance to weight initialization.

Training curves for selected cases can be found in Section D.

Table 5: Next-word prediction perplexity results for Transformer models using no foundation, weight initialization, and FedBaF methods (best of 3 trials). **Lower perplexity is better.**

FedAvg				
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**Global Testing Perplexity**		
		No Foundation	**Weight Initialization**	**FedBaF**
640,000	ζ=0%, λ=1	536.5	172.8	**128.6**
	ζ=10%, λ=1	549.4	191.8	**137.8**
	ζ=20%, λ=1	531.3	213.7	**159.4**
	ζ=50%, λ=1	501.6	330.6	**298.4**
	ζ=50%, λ=5	680.4	311.7	**275.1**
1,280,000	ζ=0%, λ=1	536.5	202.6	**183.3**
	ζ=10%, λ=1	549.4	227.3	**201.4**
	ζ=20%, λ=1	531.3	258.3	**224.2**
	ζ=50%, λ=1	501.6	371.9	**337.3**
	ζ=50%, λ=5	680.4	343.9	**322.8**

C.2 Using the Official Pre-Trained Foundation Model

Table 6: Image classification test accuracy results for Vision Transformer using official pre-trained foundation model weights from PyTorch. Comparisons are made between no foundation model, weight initialization, and FedBaF methods (best of 3 trials).

FedAvg										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
1,281,167	ζ=0%, λ=1	47.9	**81.5**	81.3	84.2	**96.9**	96.0	41.3	70.8	**71.5**
	ζ=10%, λ=1	47.6	**81.2**	80.5	85.5	96.3	**96.6**	38.6	67.0	**68.7**
	ζ=20%, λ=1	45.6	**80.3**	80.1	85.0	**94.9**	94.1	36.3	63.7	**65.4**
	ζ=50%, λ=1	42.3	**77.2**	76.4	73.7	**83.3**	80.4	33.7	49.8	**51.4**
	ζ=50%, λ=5	37.8	**74.3**	72.7	74.7	**85.6**	82.4	28.7	40.4	**50.4**
FedProx										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
1,281,167	ζ=0%, λ=1	47.4	**80.9**	80.5	78.3	**89.0**	87.3	41.2	68.7	**71.2**
	ζ=10%, λ=1	46.7	**80.4**	80.3	75.9	**85.5**	83.1	37.7	61.0	**62.2**
	ζ=20%, λ=1	45.1	79.4	**79.7**	73.1	**81.1**	79.2	35.4	59.0	**59.7**
	ζ=50%, λ=1	41.7	**76.7**	75.9	62.5	64.9	**64.9**	32.8	48.8	**49.2**
	ζ=50%, λ=5	36.8	**73.5**	69.8	63.9	66.6	**67.4**	28.5	37.3	**46.5**

We also evaluated the performance of FedBaF using the official pre-trained foundation model that was not developed by us. Specifically, we used ImageNet pre-trained weights for the Vision Transformer model, obtained from PyTorch's official model repository. As shown in Table 6, FedBaF outperforms scenarios without foundation models and delivers similar performance to weight initialization using pre-trained weights. These results demonstrate that FedBaF can effectively integrate widely adopted pre-trained models.

C.3 Generalized Applicability of FedBaF with FedAdam

To further assess the generalized applicability of FedBaF, we conducted additional experiments using the FedAdam Reddi et al. (2021) algorithm. Specifically, we evaluated its performance on image classification tasks with CIFAR-10 and Rome Weather Image datasets using Pre-ResNet and Vision Transformer, as well as on a language modeling task with a Transformer model pre-trained on the WikiText-2 dataset. We followed the same experimental setup, attack scenarios, and hyperparameters described in Section 5 and Appendix A.

For FedAdam, we set the global aggregation update learning rate to 0.01 for CIFAR-10 and Rome Weather Image datasets and 0.1 for the WikiText-2 dataset. Additionally, we configured the momentum parameters as $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for all experiments.

Table 7: Image classification test accuracy results for Pre-ResNet using the no foundation, weight initialization, and FedBaF methods employing **FedAdam** (best of 3 trials).

FedAdam										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
50,000	ζ=0%, λ=1	78.7	**84.6**	83.2	95.9	98.1	**98.3**	57.5	69.1	**69.8**
	ζ=10%, λ=1	79.7	**83.3**	83.2	95.1	97.9	**98.1**	47.7	63.8	**66.4**
	ζ=20%, λ=1	78.5	82.6	**82.6**	94.3	97.3	**98.1**	42.5	60.0	**62.2**
	ζ=50%, λ=1	70.6	**80.0**	79.8	83.7	90.4	**93.5**	34.5	52.6	**54.2**
	ζ=50%, λ=5	67.3	78.6	**79.4**	86.7	**94.4**	91.9	27.3	50.5	**53.9**
100,000	ζ=0%, λ=1	78.7	**86.0**	85.0	95.9	98.4	**98.6**	57.5	70.0	**71.4**
	ζ=10%, λ=1	79.7	**85.1**	84.7	95.1	98.2	**98.2**	47.7	65.9	**70.5**
	ζ=20%, λ=1	78.5	83.9	**84.2**	94.3	**98.0**	97.9	42.5	63.7	**65.7**
	ζ=50%, λ=1	70.6	**82.0**	81.9	83.7	89.7	**92.2**	34.5	56.7	**57.7**
	ζ=50%, λ=5	67.3	80.8	**81.6**	86.7	**94.1**	93.8	27.3	54.3	**57.4**

Table 8: Image classification test accuracy results for Vision Transformer using no foundation, weight initialization, and FedBaF methods employing **FedAdam** (best of 3 trials).

FedAdam										
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**IID - Global Testing Accuracy**			**Non-IID - Local Testing Accuracy**			**Non-IID – Global Testing Accuracy**		
		No Foundation	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**	**No Foundation**	**Weight Initialization**	**FedBaF**
2,500	ζ=0%, λ=1	58.0	**64.0**	62.0	92.2	93.6	**96.9**	52.0	62.0	**62.0**
	ζ=10%, λ=1	58.0	66.0	**66.0**	90.2	**93.3**	93.1	54.0	**62.0**	60.0
	ζ=20%, λ=1	56.0	64.0	**66.0**	87.3	**93.3**	92.3	48.0	56.0	**56.0**
	ζ=50%, λ=1	56.0	**64.0**	58.0	84.7	85.9	**86.9**	54.0	56.0	**60.0**
	ζ=50%, λ=5	48.0	58.0	**64.0**	**86.7**	85.5	84.4	44.0	48.0	**56.0**
5,000	ζ=0%, λ=1	58.0	68.0	**70.0**	92.2	95.0	**95.0**	52.0	**64.0**	62.0
	ζ=10%, λ=1	58.0	**66.0**	62.0	90.2	92.2	**95.7**	54.0	60.0	**60.0**
	ζ=20%, λ=1	56.0	64.0	**64.0**	87.3	90.5	**90.6**	48.0	56.0	**58.0**
	ζ=50%, λ=1	56.0	**60.0**	56.0	84.7	**90.2**	85.9	54.0	52.0	**58.0**
	ζ=50%, λ=5	48.0	**64.0**	54.0	**86.7**	86.5	**88.5**	44.0	**52.0**	50.0

Table 9: Next-word prediction perplexity results for Transformer models using no foundation, weight initialization, and FedBaF methods employing **FedAdam** (best of 3 trials). **Lower perplexity is better.**

FedAdam				
Pre-trained Samples	**Malicious Clients (ζ) Attack Intensity (λ)**	**Global Testing Perplexity**		
		No Foundation	**Weight Initialization**	**FedBaF**
640,000	ζ=0%, λ=1	157.9	45.9	**43.4**
	ζ=10%, λ=1	167.8	**45.6**	45.7
	ζ=20%, λ=1	199.0	50.9	**48.7**
	ζ=50%, λ=1	218.9	91.2	**85.4**
	ζ=50%, λ=5	262.7	94.7	**91.0**
1,280,000	ζ=0%, λ=1	157.9	38.4	**34.0**
	ζ=10%, λ=1	167.8	38.4	**37.3**
	ζ=20%, λ=1	199.0	44.8	**42.8**
	ζ=50%, λ=1	218.9	**93.0**	95.8
	ζ=50%, λ=5	262.7	**82.0**	83.3

Tables 7, 8, and 9 present the experimental results obtained using FedAdam across different datasets, model architectures, and attack scenarios. The results demonstrate that FedBaF remains effective across various federated optimization frameworks, reinforcing its adaptability in federated learning.

Among the 70 tested cases, FedBaF achieved similar or superior performance in 68 cases when compared to weight initialization approaches, outperforming cases without foundation models. This trend aligns with the findings in Section 5, further highlighting FedBaF's robustness across different optimization strategies and experimental settings.

These findings confirm the reliability of FedBaF in federated learning, ensuring its applicability to both vision and language tasks under diverse conditions.

C.4 Computational Complexity

Table 10: Pre-ResNet and Vision Transformer computational complexities using no foundation model, weight initialization, and FedBaF methods. **Note: T represents trillion.**

a) Pre-ResNet

	Pre-trained Samples	IID			Non-IID		
		No Foundation	Weight Initialization	FedBaF	No Foundation	Weight Initialization	FedBaF
FedAvg	50,000	1.51 T	0.20 T	0.18 T	2.63 T	0.26 T	0.27 T
	100,000	1.51 T	0.11 T	0.13 T	2.63 T	0.21 T	0.20 T
FedProx	50,000	1.52 T	0.21 T	0.18 T	2.64 T	0.26 T	0.27 T
	100,000	1.52 T	0.11 T	0.13 T	2.64 T	0.21 T	0.20 T

b) Vision Tranformer

	Pre-trained Samples	IID			Non-IID		
		No Foundation	Weight Initialization	FedBaF	No Foundation	Weight Initialization	FedBaF
FedAvg	2,500	7.34 T	0.20 T	0.23 T	8.08 T	0.20 T	0.49 T
	5,000	7.34 T	0.26 T	0.20 T	8.08 T	0.14 T	0.26 T
FedProx	2,500	6.66 T	0.20 T	0.23 T	8.91 T	0.20 T	0.49 T
	5,000	6.66 T	0.26 T	0.20 T	8.91 T	0.14 T	0.26 T

FedBaF demonstrates remarkable efficiency in terms of computational complexity, requiring significantly fewer computations than scenarios without foundation models and performing similarly to weight initialization methods.

To demonstrate that FedBaF's computational demands are minimal, even when integrating the foundation model in every training round, we assess its computational complexity. Table 10 presents the computational complexities, measured in TMAC (Total Multiply-Accumulate Operations) from Section B, for six non-adversarial scenarios in both IID and non-IID settings. To calculate these complexities, we consider the number of training rounds required to achieve specific global testing accuracies: 75% for IID and 50% for non-IID scenarios in Pre-ResNet cases. For Vision Transformer cases, we set the thresholds to 60% IID accuracy and 60% non-IID accuracy.

Compared to the no foundation model cases, **FedBaF requires significantly fewer computations** across both IID and non-IID scenarios. Specifically, for Pre-ResNet IID and non-IID cases, computations are reduced by 88.1-91.4% and 89.7-92.4% for FedAvg, and by 88.2-91.4% and 89.8-92.4% for FedProx, respectively. Similarly, for Vision Transformer, IID and non-IID scenarios see a reduction of 96.9-97.3% and 93.9-96.8% for FedAvg, and 96.5-97.0% and 94.5-97.1% for FedProx in computations. Therefore, these findings indicate that FedBaF's computational demands are relatively minimal despite the foundation model being integrated into every training round.

These findings clearly indicate that **FedBaF's computational demands are minimal**, even when integrating the foundation model into every training round, making it a highly efficient solution in both IID and non-IID environments.

D Training curves

In this section, we present the evolution of model accuracy over epochs for the experiments described in Section C, as shown in Figure 6. The experiments involve both Pre-ResNet and Vision Transformer models, focusing on two setups: **1) Pre-ResNet**, with foundation models pre-trained using TinyImageNet-200 (50,000 pre-trained samples), and **2) Vision Transformer**, with foundation models pre-trained using the Weather Image dataset (2,500 pre-trained samples). We present results for both IID and non-IID settings, using FedAvg and FedProx as the aggregation methods.

These training curves illustrate the progression of model accuracy throughout the training process, comparing the behaviors of models using weight initialization and FedBaF. The curves demonstrate that FedBaF performs similarly to traditional weight initialization methods and achieves higher accuracies than when no foundation model is used.

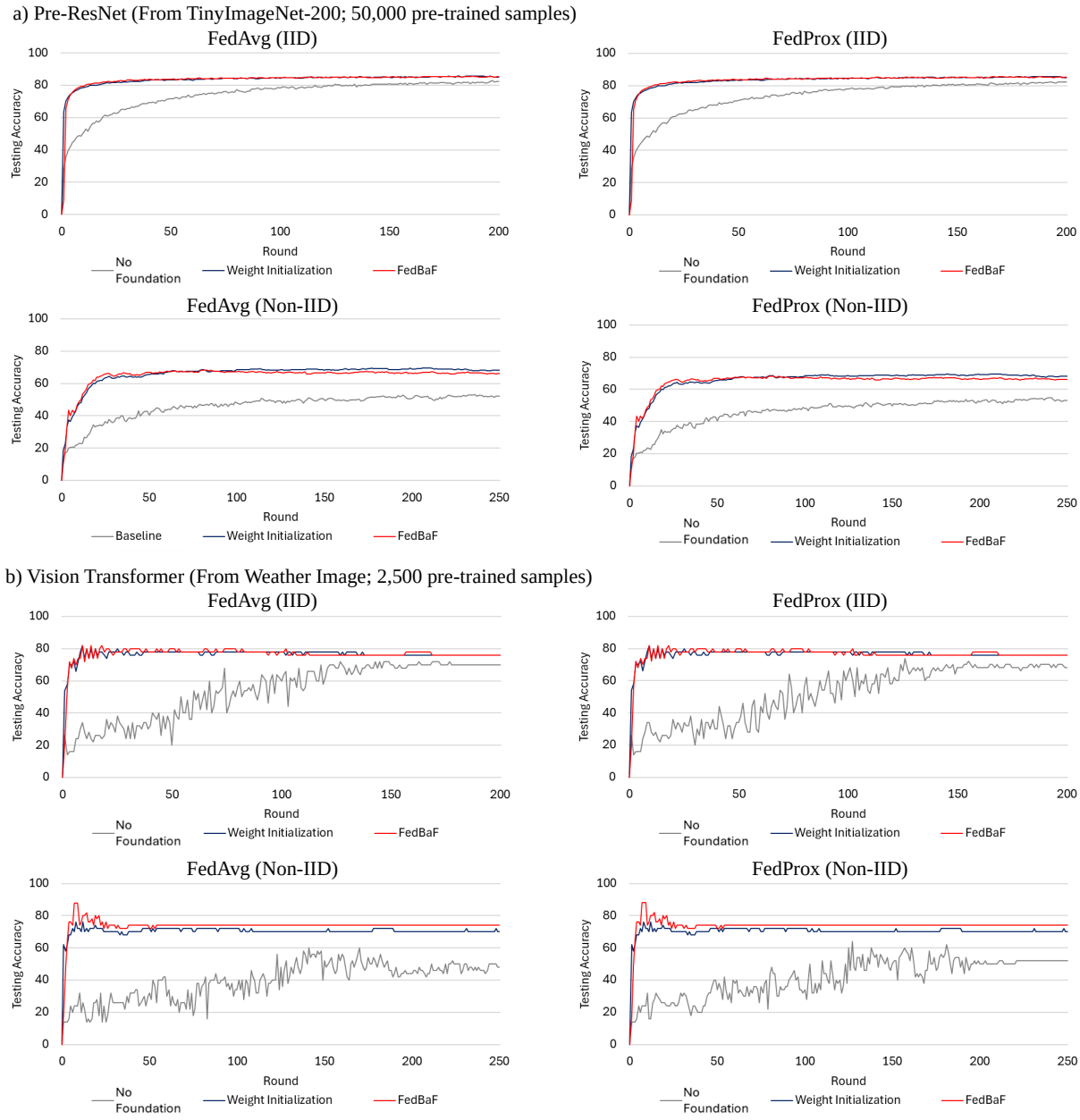

Figure 6: This figure shows the evolution of model accuracy over training epochs for Pre-ResNets and Vision Transformers under IID and Non-IID scenarios. It compares the performance using no foundation model, weight initialization, and FedBaF with FedAvg and FedProx.

E Convergence Analysis

This section provides a convergence analysis for the FedBaF algorithm under non-convex settings using Stochastic Gradient Descent (SGD). The study demonstrates how integrating a foundation model during aggregation can improve convergence rates, even in non-IID scenarios.

E.1 Problem Setup

Consider the global objective function in federated learning, which is defined as:

$$F(\mathbf{w}) = \frac{1}{\sum_{k=1}^{m} n_k} \sum_{k=1}^{m} n_k f_{D_k}(\mathbf{w}), \tag{3}$$

where m is the number of clients, $f_{D_k}(\mathbf{w})$ represents the local objective function on client k, n_k is the number of data samples at client k, $\mathbf{w} \in \mathbb{R}^d$ are the model weights, $F(\mathbf{w})$ is the global objective function. *To simplify our analysis, we assume all clients participate in every global round.*

We define $\mathbf{w}_k^{(t,\ell)}$ as the model weights of client k at global round t and local update step ℓ. Therefore, before the first local update step in each global round, each client's model weights $\mathbf{w}_k^t$ are equal to the global model weights $\mathbf{w}_t$ and

$$\mathbf{w}_k^{(t,0)} = \mathbf{w}_k^t = \mathbf{w}_t. \tag{4}$$

E.1.1 Assumptions

We make the following standard assumptions to facilitate the convergence analysis:

L-Smoothness: Each local objective function $f_{D_k}(\mathbf{w})$ is L-smooth with respect to $\mathbf{w}$, meaning:

$$f_{D_k}(\mathbf{v}) \leq f_{D_k}(\mathbf{w}) + \nabla f_{D_k}(\mathbf{w})^\top (\mathbf{v} - \mathbf{w}) + \frac{L}{2}\|\mathbf{v} - \mathbf{w}\|^2, \quad \forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^d. \tag{5}$$

and

$$\|\nabla f_{D_k}(\mathbf{v}) - \nabla f_{D_k}(\mathbf{w})\| \leq L\|\mathbf{v} - \mathbf{w}\|, \quad \forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^d. \tag{6}$$

Unbiased Mini-Batch Gradients: The stochastic gradients computed over mini-batches during local updates are unbiased estimates of the true gradients.

$$\mathbb{E}\left[g_{D_k}(\mathbf{w}; b)\right] = \nabla f_{D_k}(\mathbf{w}) \quad \forall \mathbf{w} \in \mathbb{R}^d. \tag{7}$$

Bounded Gradient Norm: The gradients of the local objective functions are bounded by a constant β. Specifically,

$$\|\nabla f_{D_k}(\mathbf{w})\| \leq \beta \quad \forall \mathbf{w} \in \mathbb{R}^d. \tag{8}$$

Bounded Gradient Noise:

$$\mathbb{E}\left[\|g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2\right] \leq G^2 + B^2\|\nabla F(\mathbf{w}_t)\|^2 \tag{9}$$

where G^2 and B^2 are constants that bound the dissimilarity, $b_{k,\ell}$ denotes the j-th mini-batch on client k, and ℓ indexes the local update steps during global round t.

Foundation Model Alignment: The foundation model shares the same architecture as the global model, guaranteeing seamless integration during aggregation.

E.2 FedBaF Aggregation Step with Multiple Local Updates

The FedBaF algorithm operates in global rounds, where each global round t includes multiple local iterations denoted by ℓ. Each client performs multiple local updates in each global round.

The global model is updated at round t using the rule:

$$\mathbf{w}_{t+1} = \frac{1}{1+\alpha_t\tau_t}\left(\mathbf{w}'_t + \alpha_t\tau_t\mathbf{w}_{\text{pre}}\right), \tag{10}$$

where $\mathbf{w}'_t$ is the aggregated model from the client updates, $\mathbf{w}_{\text{pre}}$ is the foundation model's weights, α_t is a scaling factor, and τ_t represents the correction factor based on the foundation model.

Each client k performs multiple local SGD updates over local iterations $\ell = 0, 1, \ldots, \Lambda_k - 1$, where Λ_k is the number of local updates for each client k. For each local update, the local model is updated as:

$$\mathbf{w}_k^{(t,\ell+1)} = \mathbf{w}_k^{(t,\ell)} - \eta g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}), \tag{11}$$

where $g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})$ is the stochastic gradient computed on the mini-batch $b_{k,\ell}$ at local iteration ℓ.

At the end of the local updates, each client sends the model $\mathbf{w}_k^{(t,\Lambda_k)}$ to the server, where the global model $\mathbf{w}'_t$ is computed as the weighted average of the client updates:

$$\mathbf{w}'_t = \mathbf{w}_t - \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}), \tag{12}$$

where $p_k = \frac{n_k}{\sum_{k=1}^m n_k}$ is the weight of client k.

Substituting this into the FedBaF update rule:

$$\mathbf{w}_{t+1} = \frac{1}{1+\alpha_t\tau_t}\left(\mathbf{w}_t + \alpha_t\tau_t\mathbf{w}_{\text{pre}} - \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\right) \tag{13}$$

For small values of $\alpha_t\tau_t \ll 1$, we use the fact that:

$$\begin{aligned}\frac{1}{1+\alpha_t\tau_t} &= \sum_{i=0}^{\infty}(-\alpha_t\tau_t)^i \\ &= 1 - \alpha_t\tau_t + \sum_{i=2}^{\infty}(-\alpha_t\tau_t)^i \\ &\approx 1 - \alpha_t\tau_t\end{aligned} \tag{14}$$

which simplifies the update rule to:

$$\begin{aligned}\mathbf{w}_{t+1} \approx \mathbf{w}_t &- \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) \\ &- \alpha_t\tau_t\left(\left[\mathbf{w}_t - \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\right] - (1-\alpha_t\tau_t)\,\mathbf{w}_{\text{pre}}\right).\end{aligned} \tag{15}$$

The update rule can now be interpreted as multiple local gradient descent steps combined with a bias correction. Here, we define the correction term χ_t as follows:

$$\chi_t = \alpha_t\tau_t\left(\left[\mathbf{w}_t - \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\right] - (1-\alpha_t\tau_t)\,\mathbf{w}_{\text{pre}}\right). \tag{16}$$

χ_t acts as a correction that adjusts the direction of the gradient descent to leverage the foundation model's knowledge. The update rule becomes:

$$\mathbf{w}_{t+1} \approx \mathbf{w}_t - \eta\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \chi_t. \tag{17}$$

E.3 Decrease in Objective Function

Using the smoothness property of $F(\mathbf{w})$:

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t)^\top (\mathbf{w}_{t+1} - \mathbf{w}_t) + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2. \tag{18}$$

We substitute the update rule:

$$\mathbf{w}_{t+1} - \mathbf{w}_t \approx -\eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \chi_t. \tag{19}$$

Thus:

$$\begin{aligned} F(\mathbf{w}_{t+1}) \lessapprox & F(\mathbf{w}_t) - \eta \nabla F(\mathbf{w}_t)^\top \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \nabla F(\mathbf{w}_t)^\top \chi_t \\ & + \frac{L}{2}\|\eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \chi_t\|^2. \end{aligned} \tag{20}$$

Given the update rule, the change in the objective function can be bounded using the triangle inequality:

$$\begin{aligned} F(\mathbf{w}_{t+1}) - F(\mathbf{w}_t) \lessapprox & -\eta \nabla F(\mathbf{w}_t)^\top \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \nabla F(\mathbf{w}_t)^\top \chi_t \\ & + L\eta^2 \| \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2 + L\|\chi_t\|^2 \end{aligned} \tag{21}$$

E.3.1 Taking Expectations

We now take expectations of both sides of (21), based on the unbiasedness of mini-batch gradients and the assumptions about bounded gradient norms and smoothness.

We now compute the expectation of the change in the objective function:

$$\begin{aligned} & \mathbb{E}\left[\nabla F(\mathbf{w}_t)^\top \left(-\eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\right)\right] \\ & = \nabla F(\mathbf{w}_t)^\top \left(-\eta (\sum_{k=1}^{m} p_k \Lambda_k) \nabla F(\mathbf{w}_t) + \eta (\sum_{k=1}^{m} p_k \Lambda_k) \nabla F(\mathbf{w}_t) - \eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} \mathbb{E}\left[g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\right]\right) \\ & = -\eta (\sum_{k=1}^{m} p_k \Lambda_k) \|\nabla F(\mathbf{w}_t)\|^2 + \nabla F(\mathbf{w}_t)^\top \left(\eta (\sum_{k=1}^{m} p_k \Lambda_k) \sum_{k=1}^{m} p_k \nabla f_{D_k}(\mathbf{w}_t) - \eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} \nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)})\right). \end{aligned} \tag{22}$$

Next, we simplify the remaining term:

$$\begin{aligned} & = -\eta (\sum_{k=1}^{m} p_k \Lambda_k) \|\nabla F(\mathbf{w}_t)\|^2 + \eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} \nabla F(\mathbf{w}_t)^\top \left(\frac{\sum_{k=1}^{m} p_k \Lambda_k}{\Lambda_k} \nabla f_{D_k}(\mathbf{w}_t) - \nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)})\right) \\ & \leq -\eta (\sum_{k=1}^{m} p_k \Lambda_k) \|\nabla F(\mathbf{w}_t)\|^2 + \frac{\eta}{2} \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} \left[\|\nabla F(\mathbf{w}_t)\|^2 + \|\frac{\sum_{k=1}^{m} p_k \Lambda_k}{\Lambda_k} \nabla f_{D_k}(\mathbf{w}_t) - \nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)})\|^2\right]. \end{aligned} \tag{23}$$

Finally, using the fact that the gradient is bounded, we get:

$$\begin{aligned}
&\leq -\left(\frac{\eta}{2}\sum_{k=1}^{m} p_k\Lambda_k\right)\|\nabla F(\mathbf{w}_t)\|^2 + \frac{\eta}{2}\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} \|\frac{\sum_{k=1}^{m} p_k\Lambda_k}{\Lambda_k}\nabla f_{D_k}(\mathbf{w}_t) - \nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)})\|^2 \\
&\leq -\left(\frac{\eta}{2}\sum_{k=1}^{m} p_k\Lambda_k\right)\|\nabla F(\mathbf{w}_t)\|^2 + \frac{\eta}{2}\sum_{k=1}^{m} 2p_k \sum_{\ell=0}^{\Lambda_k-1} \left[\|\frac{\sum_{k=1}^{m} p_k\Lambda_k}{\Lambda_k}\nabla f_{D_k}(\mathbf{w}_t)\|^2 + \|\nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)})\|^2\right] \\
&\leq -\left(\frac{\eta}{2}\sum_{k=1}^{m} p_k\Lambda_k\right)\|\nabla F(\mathbf{w}_t)\|^2 + \eta\beta^2\sum_{k=1}^{m} p_k\left(\frac{(\sum_{k=1}^{m} p_k\Lambda_k)^2}{\Lambda_k} + \Lambda_k\right).
\end{aligned} \tag{24}$$

This provides a bound on the expected decrease in the global objective function.
Similarly,

$$\begin{aligned}
&\mathbb{E}\left[L\eta^2\|\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2\right] \\
&\leq Lm\eta^2\sum_{k=1}^{m}\mathbb{E}\left[\|p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2\right] \\
&\leq Lm\eta^2\sum_{k=1}^{m} p_k^2\Lambda_k \sum_{\ell=0}^{\Lambda_k-1}\mathbb{E}\left[\|g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2\right] \\
&\leq Lm\eta^2\sum_{k=1}^{m} p_k^2\Lambda_k^2(G^2 + B^2\|\nabla F(\mathbf{w}_t)\|^2)
\end{aligned} \tag{25}$$

Plugging these bounds into (21) gives

$$\begin{aligned}
\mathbb{E}\left[F(\mathbf{w}_{t+1}) - F(\mathbf{w}_t)\right] &\lessapprox -\eta\nabla F(\mathbf{w}_t)^\top \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) - \nabla F(\mathbf{w}_t)^\top\mathbb{E}\left[\chi_t\right] \\
&\quad + L\eta^2\|\sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k-1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell})\|^2 + L\mathbb{E}\left[\|\chi_t\|^2\right] \\
&\leq -\left(\frac{\eta}{2}\sum_{k=1}^{m} p_k\Lambda_k\right)\|\nabla F(\mathbf{w}_t)\|^2 + \eta\beta^2\sum_{k=1}^{m} p_k\left(\frac{(\sum_{k=1}^{m} p_k\Lambda_k)^2}{\Lambda_k} + \Lambda_k\right) - \nabla F(\mathbf{w}_t)^\top\mathbb{E}\left[\chi_t\right] \\
&\quad + Lm\eta^2\sum_{k=1}^{m} p_k^2\Lambda_k^2(G^2 + B^2\|\nabla F(\mathbf{w}_t)\|^2) + L\mathbb{E}\left[\|\chi_t\|^2\right] \\
&\leq \left(-\frac{\eta}{2}\sum_{k=1}^{m} p_k\Lambda_k + B^2Lm\eta^2\sum_{k=1}^{m} p_k^2\Lambda_k^2\right)\|\nabla F(\mathbf{w}_t)\|^2 + \eta\beta^2\sum_{k=1}^{m} p_k\left(\frac{(\sum_{k=1}^{m} p_k\Lambda_k)^2}{\Lambda_k} + \Lambda_k\right) \\
&\quad - \nabla F(\mathbf{w}_t)^\top\mathbb{E}\left[\chi_t\right] + Lm\eta^2\sum_{k=1}^{m} p_k^2\Lambda_k^2G^2 + L\mathbb{E}\left[\|\chi_t\|^2\right]
\end{aligned} \tag{26}$$

E.4 Summing Over Iterations

To establish a convergence result, we sum this inequality over T iterations:

$$\begin{aligned}\sum_{t=1}^{T} F(\mathbf{w}_{t+1}) \leq & \sum_{t=1}^{T} F(\mathbf{w}_t) + \left(-\frac{\eta}{2} \sum_{k=1}^{m} p_k \Lambda_k + B^2 L m \eta^2 \sum_{k=1}^{m} p_k^2 \Lambda_k^2 \right) \sum_{t=1}^{T} \|\nabla F(\mathbf{w}_t)\|^2 \\ & + \eta \beta^2 T \sum_{k=1}^{m} p_k \left(\frac{(\sum_{k=1}^{m} p_k \Lambda_k)^2}{\Lambda_k} + \Lambda_k \right) - \sum_{t=1}^{T} \nabla F(\mathbf{w}_t)^\top \mathbb{E}\left[\chi_t\right] \\ & + L m \eta^2 T \sum_{k=1}^{m} p_k^2 \Lambda_k^2 G^2 + L \sum_{t=1}^{T} \mathbb{E}\left[\|\chi_t\|^2\right]\end{aligned} \tag{27}$$

As long as $\eta < \frac{\sum_{k=1}^{m} p_k \Lambda_k}{2B^2 L m \sum_{k=1}^{m} p_k^2 \Lambda_k^2}$, the term $\frac{1}{2}\sum_{k=1}^{m} p_k \Lambda_k - B^2 L m \eta \sum_{k=1}^{m} p_k^2 \Lambda_k^2$ is positive. Therefore, rearranging and dividing by T gives:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|\nabla F(\mathbf{w}_t)\|^2] \leq & \frac{F(\mathbf{w}_1) - F(\mathbf{w}_{T+1})}{T(\frac{1}{2}\sum_{k=1}^{m} p_k \Lambda_k - B^2 L m \eta \sum_{k=1}^{m} p_k^2 \Lambda_k^2)} \\ & + \frac{\eta \beta^2 \sum_{k=1}^{m} p_k \left(\frac{(\sum_{k=1}^{m} p_k \Lambda_k)^2}{\Lambda_k} + \Lambda_k \right) + L m \eta^2 \sum_{k=1}^{m} p_k^2 \Lambda_k^2 G^2}{\frac{1}{2}\sum_{k=1}^{m} p_k \Lambda_k - B^2 L m \eta \sum_{k=1}^{m} p_k^2 \Lambda_k^2} \\ & + \frac{L \sum_{t=1}^{T} \mathbb{E}\left[\|\chi_t\|^2\right] - \sum_{t=1}^{T} \nabla F(\mathbf{w}_t)^\top \mathbb{E}\left[\chi_t\right]}{T(\frac{1}{2}\sum_{k=1}^{m} p_k \Lambda_k - B^2 L m \eta \sum_{k=1}^{m} p_k^2 \Lambda_k^2)}\end{aligned} \tag{28}$$

If the sign of

$$L\mathbb{E}[\|\chi_t\|^2] - \nabla F(\mathbf{w}_t)^\top \mathbb{E}[\chi_t]$$

is negative, we obtain a tighter bound, which implies that the correction terms positively influence convergence.

We know that:

$$\chi_t = \alpha_t \tau_t \left(\left[\mathbf{w}_t - \eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} g_{D_k}(\mathbf{w}_k^{(t,\ell)}; b_{k,\ell}) \right] - (1 - \alpha_t \tau_t)\, \mathbf{w}_{\text{pre}} \right).$$

This means:

$$\mathbb{E}\left[\|\chi_t\|^2\right] = \alpha_t^2 \tau_t^2 \left\| \left[\mathbf{w}_t - \eta \sum_{k=1}^{m} p_k \sum_{\ell=0}^{\Lambda_k - 1} \nabla f_{D_k}(\mathbf{w}_k^{(t,\ell)}) \right] - (1 - \alpha_t \tau_t)\, \mathbf{w}_{\text{pre}} \right\|^2.$$

Given that $\alpha_t \tau_t$ is small enough at a sufficiently large global round t, the higher-order terms with $\alpha_t^2 \tau_t^2$ become negligible. Therefore, for large t, $[\|\chi_t\|^2] \approx 0$. Next, the direction of the correction χ_t and the direction of the gradient of the global objective $\nabla F(\mathbf{w}_t)$ agreeing implies that the inner product $\nabla F(\mathbf{w}_t)^\top \mathbb{E}\left[\chi_t\right]$ is positive.

We conclude that:

$$L\mathbb{E}\left[\|\chi_t\|^2\right] < \nabla F(\mathbf{w}_t)^\top \mathbb{E}[\chi_t].$$

This shows that the variance of the correction term χ_t is significantly smaller than its impact on the inner product, leading to a tighter convergence bound, especially when $\alpha_t \tau_t$ is small but positive.

E.5 Influence of the Correction Term

The correction term χ_t, derived from the foundation model, plays a significant role in the convergence behavior. The influence of χ_t ensures that the gradient descent step is adjusted based on the foundation model's knowledge. By controlling the size of $\alpha_t \tau_t$, the foundation model can guide the global model towards better solutions, especially in non-IID scenarios. The correction term provides additional stability and enhances convergence, particularly when the local models exhibit significant heterogeneity.

F Proofs of Propositions

F.1 Proof of Proposition 1

Proposition 1. *Let $\mathbf{w}^*$ be a (bounded) local minimum of the global objective function in (1). Consider an FL algorithm that converges to $\mathbf{w}^*$ and let $\mathbf{w}'_t$ be its global model in each training round t. Suppose we run the same algorithm but using FedBaF for the aggregation, and let $\mathbf{w}_t$ be the FedBaF global model at round t. Let α_t satisfy*

$$\alpha_t < \frac{2\|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2}{(\|\mathbf{w}_{pre} - \mathbf{w}^*\|^2 - \|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2)\tau_t} \tag{2}$$

for all t where $\|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2 < \|\mathbf{w}_{pre} - \mathbf{w}^*\|^2$. Then $\forall t\ \|\mathbf{w}_t - \mathbf{w}^*\| < \|\mathbf{w}'_t - \mathbf{w}^*\|$.

This means that, at any given round t, FedBaF's model weights are closer to $\mathbf{w}^*$.

Proof. We present a convergence analysis of our FL framework that incorporates foundation models in the aggregation phase according to Alg. 1 *Lines 8-10*. By comparing the square distance between $\mathbf{w}_{t+1}$ and $\mathbf{w}^*$ to the square distance between $\mathbf{w}'_{t+1}$ and $\mathbf{w}^*$, we derive conditions under which our method converges to $\mathbf{w}^*$ faster than FedAvg. Noting that $\forall t\ \alpha_t, \tau_t \geq 0$,

$$\begin{aligned}
\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 &= \|\frac{1}{1+\alpha_t\tau_t}(\mathbf{w}'_{t+1} + \alpha_t\tau_t\mathbf{w}_{\text{pre}}) - \mathbf{w}^*\|^2 \\
&= \frac{1}{(1+\alpha_t\tau_t)^2}\|(\mathbf{w}'_{t+1} - \mathbf{w}^*) + \alpha_t\tau_t(\mathbf{w}_{\text{pre}} - \mathbf{w}^*)\|^2 \\
&\leq \frac{\|\mathbf{w}'_{t+1} - \mathbf{w}^*\|^2 + \alpha_t^2\tau_t^2\|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|^2}{1 + 2\alpha_t\tau_t + \alpha_t^2\tau_t^2}
\end{aligned} \tag{29}$$

For notational convenience, we define $\beta_t := \|\mathbf{w}'_{t+1} - \mathbf{w}^*\|$ and $\gamma := \|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|$. FedBaF is better than FedAvg when the right side is less than β_t^2. So, we upper bound the right side by β_t^2 and find values of α_t that satisfy the bound.

$$\begin{aligned}
\frac{\beta_t^2 + \alpha_t^2\tau_t^2\gamma^2}{1 + 2\alpha_t\tau_t + \alpha_t^2\tau_t^2} &< \beta_t^2 \\
\beta_t^2 + \alpha_t^2\tau_t^2\gamma^2 &< \beta_t^2 + 2\alpha_t\tau_t\beta_t^2 + \alpha_t^2\tau_t^2\beta_t^2 \\
\alpha_t^2\tau_t^2(\gamma^2 - \beta_t^2) - 2\alpha_t\tau_t\beta_t^2 &< 0
\end{aligned}$$

Note that α_t is sampled from the uniform distribution $\frac{\psi}{\tau_0}\text{U}(1,2)$. For the above inequality to be satisfied for a given t, there are three cases:

1. $\beta_t > \gamma$: This case occurs when t is small and $\mathbf{w}^*$ is closer to $\mathbf{w}_{\text{pre}}$ than $\mathbf{w}^*$ is to the FedBaF global model. In this case, we require
$$\alpha_t > \frac{2\beta_t^2}{(\gamma^2 - \beta_t^2)\tau_t}$$
α_t always satisfies this inequality since the RHS is negative and $\alpha_t > 0$ by definition.
2. $\beta_t = \gamma$: This means that we require $\alpha_t > 0$, which is always true by definition.
3. $\beta_t < \gamma$: This case may occur when t is large and $\mathbf{w}^*$ is closer to the FedBaF global model than $\mathbf{w}^*$ is to $\mathbf{w}_{\text{pre}}$. In this case, we get a meaningful bound for α_t:
$$\alpha_t < \frac{2\beta_t^2}{(\gamma^2 - \beta_t^2)\tau_t}$$

When $\forall t\ \alpha_t$ satisfies the above conditions, the proposition holds. □

F.2 Proof of Proposition 2

Proposition 2. *Let $\mathbf{w}^*$ be a (bounded) local minimum of the global objective function in* (1)*. Consider an FL algorithm that converges to $\mathbf{w}^*$ and let $\mathbf{w}'_t$ be its global model. Consider FedBaF based on the same FL algorithm (with appropriately modified client updates and Lines 8-10 in Alg. 1) and let $\mathbf{w}_t$ be the FedBaF global model. FedBaF's global model error has an upper bound of $\|\mathbf{w}_t - \mathbf{w}^*\| \leq \frac{\delta_t + \alpha_t \tau_t \gamma}{1+\alpha_t \tau_t} < \delta_t$.*

Proof.

$$\begin{aligned}
\|\mathbf{w}_t - \mathbf{w}^*\| &= \left\| \frac{1}{1+\alpha_t\tau_t}(\mathbf{w}'_t + \alpha_t\tau_t \mathbf{w}_{\text{pre}}) - \mathbf{w}^* \right\| \\
&= \frac{\|(\mathbf{w}'_t - \mathbf{w}^*) + \alpha_t\tau_t(\mathbf{w}_{\text{pre}} - \mathbf{w}^*)\|}{1+\alpha_t\tau_t} \\
&\leq \frac{\|\mathbf{w}'_t - \mathbf{w}^*\| + \alpha_t\tau_t\|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|}{1+\alpha_t\tau_t} \\
&= \frac{\left\|\sum_{k\in S_t} \frac{n_k}{n}(\mathbf{w}_t^k - \mathbf{w}^*)\right\| + \alpha_t\tau_t\|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|}{1+\alpha_t\tau_t} \\
&\leq \frac{\sum_{k\in S_t} \frac{n_k}{n}\|\mathbf{w}_t^k - \mathbf{w}^*\| + \alpha_t\tau_t\|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|}{1+\alpha_t\tau_t} \\
&\leq \frac{\delta_t + \alpha_t\tau_t\gamma}{1+\alpha_t\tau_t}
\end{aligned}$$

where we set $\gamma = \|\mathbf{w}_{\text{pre}} - \mathbf{w}^*\|$. Since non-IID data can cause significant variance in local updates, we compare the derived bound to FedAvg, where the bound on the distance between $\mathbf{w}_t$ and $\mathbf{w}^*$ is δ_t. By assumption, $\gamma \leq \delta_t$ for earlier rounds (small t). We get $\frac{\delta_t + \alpha_t\tau_t\gamma}{1+\alpha_t\tau_t} \leq \delta_t$, which is equivalent to

$$\delta_t + \alpha_t\tau_t\gamma \leq \delta_t + \alpha_t\tau_t\delta_t = \delta_t(1+\alpha_t\tau_t)$$

Therefore,

$$\|\mathbf{w}_t - \mathbf{w}^*\| \leq \frac{\delta_t + \alpha_t\tau_t\gamma}{1+\alpha_t\tau_t}$$

Since $\frac{\delta_t + \alpha_t\tau_t\gamma}{1+\alpha_t\tau_t} \leq \delta_t$, FedBaF has a tighter upper bound on $\|\mathbf{w}_t - \mathbf{w}^*\|$ than FedAvg. This demonstrates the advantage of using a foundation model in non-IID settings.

□

G Security Analysis in the Presence of Adversarial Attacks

In this section, we discuss the potential for extracting a foundation model in FedBaF and demonstrate FedBaF's robustness against backdoor attacks. These attacks pose unique security challenges to FL systems, involving malicious alterations within model updates to degrade system performance or embed hidden vulnerabilities. We will analyze how FedBaF mitigates these threats and ensures integrity and security.

G.1 Possibility of Extracting a Foundation Model

As discussed in Section 3, using a randomized α_t prevents the extraction of the foundation model's weights. However, the aggregated global models might still exhibit components of the foundation model by following a similar weight distribution. To investigate this, we analyze the distance between the global model and the foundation model over the first 200 aggregation rounds.

Let $\mathbf{w}_{t+1}$ and $\mathbf{w}_{pre}$ represent the weights of the global model and the foundation model, respectively. For each weight tensor $\mathbf{w}^i_{t+1}$ and $\mathbf{w}^i_{pre}$ with matching shapes, we calculate the normalized distance for each element and then average these distances. For each element j in the weight tensor $\mathbf{w}^i_{t+1}$ and $\mathbf{w}^i_{pre}$:

$$\text{dist}^i_j = \frac{|w^i_{t+1,j} - w^i_{pre,j}|}{|w^i_{t+1,j}|}$$

where $w^i_{t+1,j}$ is the j-th element of the i-th weight tensor of the global model; $w^i_{pre,j}$ is the j-th element of the i-th weight tensor of the foundation model; and dist^i_j is the normalized distance for the j-th element of the i-th weight tensor.

We concatenate all element-wise distances dist^i_j across all weight tensors and then compute the mean of these distances:

$$\text{Dist} = \frac{1}{N_{param}} \sum_{i=1} \sum_{j=1} \text{dist}^i_j$$

where N_{param} is the total number of elements across all matching weight tensors and Dist is the overall average normalized distance.

In Figure 7, the curves show the distances, Dist, for each aggregation round. The minimum Dist across all cases was 1.27, indicating that the distance has a 127% scale of the magnitude of the weights of the aggregated global model. This means the foundation model's weights differ in scale from the aggregated weights. To analyze the effect of distance intensity, we added Gaussian random noise based on the magnitude of each foundation model's weights to the foundation model's weights.

Figure 8 shows the testing accuracy as a function of the added noise. The x-axis represents the error rate, calculated by dividing the magnitude of the added Gaussian noise by the magnitude of the foundation model's weights. We used the best-performing foundation models from those with varying pre-trained sample sizes, as described in Section C. When a 127% error rate is applied, the Pre-ResNet model shows almost 0% testing accuracy, the Vision Transformer shows 30% testing accuracy, and the Transformer model exhibits excessively high testing perplexity. This empirical evidence indicates that extracting the foundation model's knowledge is impossible after training begins from the global model. The diverse updates during training in FedBaF significantly disrupt the alignment between the foundation model's weights and the global model, preventing any meaningful extraction of the foundation model's information.

To this end, we examine the proximity of the global model, $\mathbf{w}_t$, to the foundation model and to the averaged local models, $\mathbf{w}'_t$, throughout the training process. We first determine the distances between $\mathbf{w}'_t$ and $\mathbf{w}_t$ and

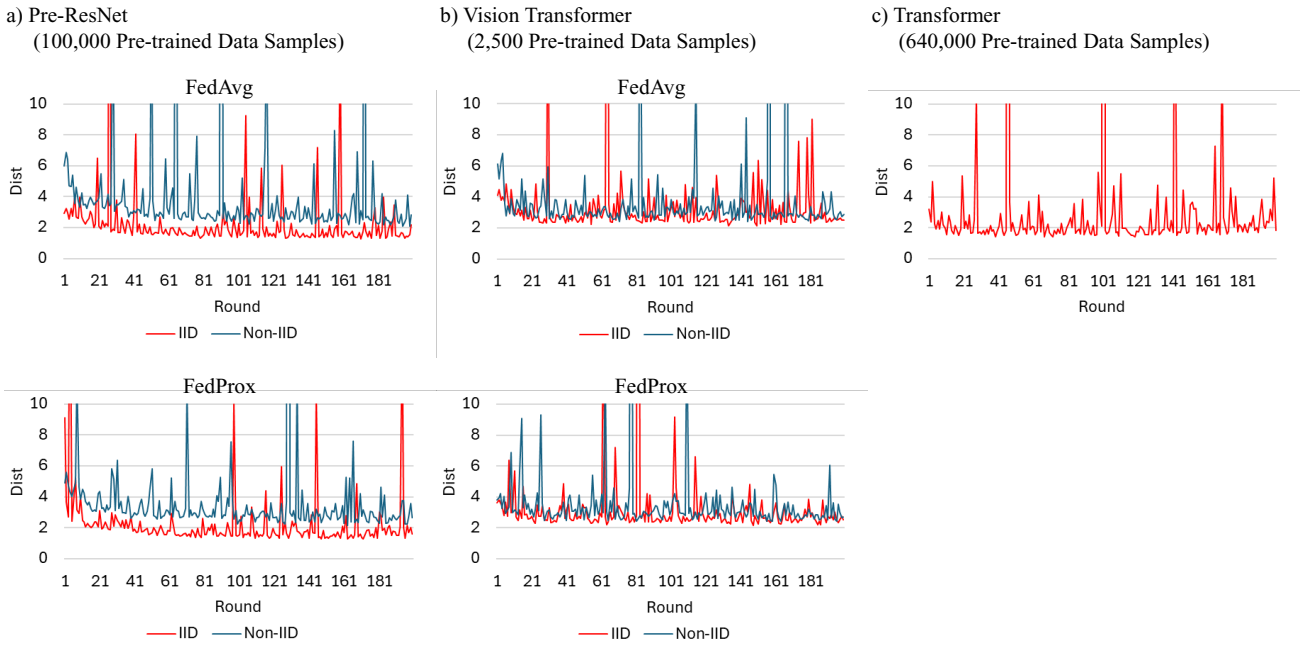

Figure 7: Distances (Dist) between the global model and the foundation model across aggregation rounds. The minimum Dist observed was 1.27, indicating significant differences in scale between the foundation model's weights and the aggregated global model's weights.

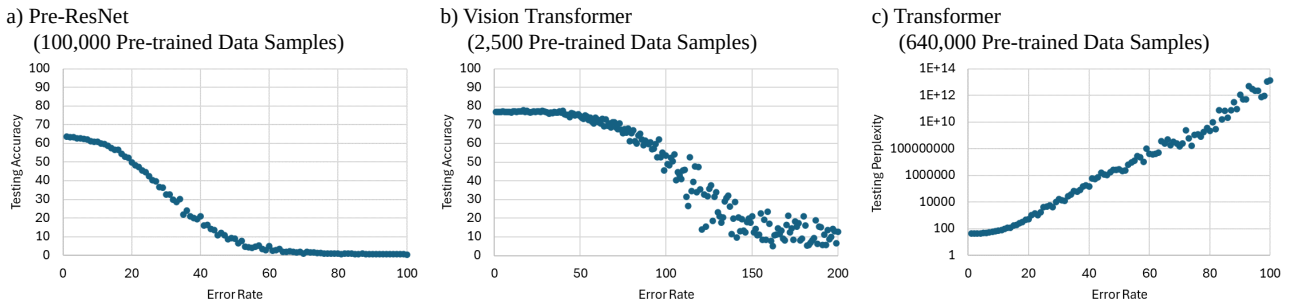

Figure 8: Testing accuracy according to the added noise. The x-axis error rate is calculated by the magnitude of the added Gaussian noise divided by the magnitude of the foundation model's weights.

between $\mathbf{w}_t$ and $\mathbf{w}_{pre}$:

$$\begin{aligned}\|\mathbf{w}_t - \mathbf{w}'_t\| &= \left\| \frac{1}{1+\alpha\tau_t}(\mathbf{w}'_t + \alpha\tau_t \mathbf{w}_{pre}) - \mathbf{w}'_t \right\| \\ &= \frac{\alpha\tau_t}{1+\alpha\tau_t}\|\mathbf{w}'_t - \mathbf{w}_{pre}\| \\ \|\mathbf{w}_t - \mathbf{w}_{pre}\| &= \left\| \frac{1}{1+\alpha\tau_t}\mathbf{w}'_t + \alpha\tau_t \mathbf{w}_{pre} - \mathbf{w}_{pre} \right\| \\ &= \frac{1}{1+\alpha\tau_t}\|\mathbf{w}'_t - \mathbf{w}_{pre}\|\end{aligned}$$

At the onset of training, both distances are equivalent since we make a strategic choice for the weight $\alpha_t\tau_0$ to be approximately 2. This simplifies the initial update rule for the global model such that the initial global model weights, $\mathbf{w}_0$, are an unweighted average of the client's updated model weights $\mathbf{w}'_0$ and the foundation model

weights $\mathbf{w}_{pre}$. As the training progresses, $\alpha\tau_t$ typically decays to less than 1. We deduce for $t > 0$:

$$\frac{\alpha\tau_t}{1+\alpha\tau_t} < \frac{1}{1+\alpha\tau_t} \implies \|\mathbf{w}_t - \mathbf{w}'_t\| < \|\mathbf{w}_t - \mathbf{w}_{pre}\|$$

As $t \to \infty$, $\mathbf{w}_t$ will drift away from $\mathbf{w}_{pre}$ and towards $\mathbf{w}'_t$. Due to the intricate dissemination of learned insights across all model weights, and the complexities of high-dimensional weight spaces, it is difficult to reverse-engineer $\mathbf{w}_{pre}$ from $\mathbf{w}_t$. Even a subset of weights does not provide enough information to predict the rest deterministically. The inherent complexity of the model weight (parameter) space is a natural defense mechanism in FedBaF.

G.2 Mitigating Backdoor Attacks

Backdoor attacks in FL involve embedding a dormant malicious function in a local model. Integrating foundation models mitigates such attacks by diluting the impact of individual client updates. Specifically, we have the updates

$$\Delta\mathbf{w}^t_{\text{client}} = \text{ClientUpdate}(\mathbf{w}_t)$$
$$\mathbf{w}_{t+1} = \frac{1}{1+\alpha_t\tau_t}(\Delta\mathbf{w}^t_{\text{client}} + \alpha_t\tau_t\mathbf{w}_{\text{pre}})$$

Here, $\Delta\mathbf{w}^t_{\text{client}}$ is the update from client c at iteration t, and $\mathbf{w}_{\text{pre}}$ is the foundation model weight. The factor τ_t controls the influence of the foundation model. This mathematical formulation showcases the security benefits of our method. By incorporating the foundation model, the aggregation counterbalances the (malicious) client update $\Delta\mathbf{w}^t_{\text{client}}$. This approach thus *enhances the system's resilience to adversarial attacks* by maintaining a consistent learning direction and reducing the impact of compromised updates.

G.3 Experiments on Variations of τ_t

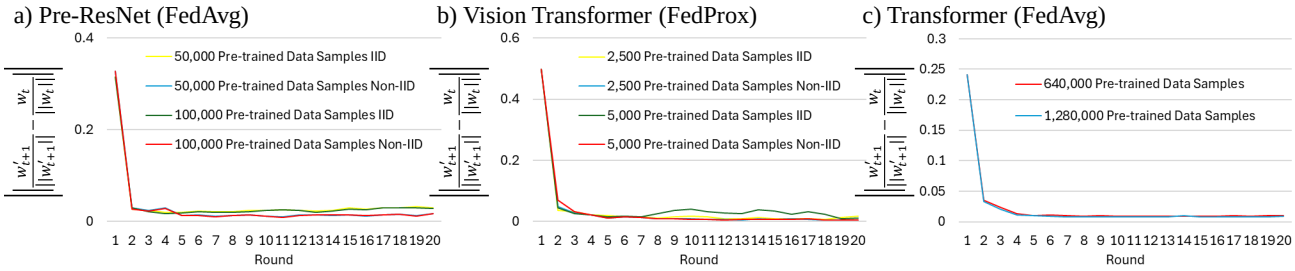

Figure 9: Variations of $\|\frac{\mathbf{w}'_{t+1}}{\|\mathbf{w}'_{t+1}\|} - \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|}\|$ $(= \tau_t\sqrt{t+1})$ across training rounds.

Figure 9 illustrates the non-adversarial IID and non-IID scenarios from Tables 3, 4, and 5. We observed that the numerator of τ_t (as referenced in Alg. 1 *Line 9*) consistently decreases towards 0, independent of the denominator $(\sqrt{t+1})$, after several rounds. This indicates that FedBaF benefits from the foundation model's guidance but retains the ability to effectively and quickly adapt to new data.