BOBA: Byzantine-Robust Federated Learning with Label Skewness

Wenxuan Bao[1] **Jun Wu**[1] **Jingrui He**[1]
[1]University of Illinois Urbana-Champaign
{wbao4,junwu3,jingrui}@illinois.edu

Abstract

In federated learning, most existing robust aggregation rules (AGRs) combat Byzantine attacks in the IID setting, where client data is assumed to be independent and identically distributed. In this paper, we address label skewness, a more realistic and challenging non-IID setting, where each client only has access to a few classes of data. In this setting, state-of-the-art AGRs suffer from selection bias, leading to significant performance drop for particular classes; they are also more vulnerable to Byzantine attacks due to the increased variation among gradients of honest clients. To address these limitations, we propose an efficient two-stage method named *BOBA*. Theoretically, we prove the convergence of BOBA with an error of the optimal order. Our empirical evaluations demonstrate BOBA's superior unbiasedness and robustness across diverse models and datasets when compared to various baselines. Our code is available at `https://github.com/baowenxuan/BOBA`.

1 INTRODUCTION

Federated learning (FL) (McMahan et al., 2017) is a machine learning system where multiple clients collaboratively train a global model under the orchestration of a central server, without sharing their own private and sensitive data. It has wide applications in sales, finance, healthcare (Yang et al., 2019), etc. However, FL systems are vulnerable to attacks and failures (Kairouz et al., 2021; Lyu et al., 2020). Notably, *Byzantine attacks* can send arbitrary gradients to the server, causing sub-optimal convergence or even divergence (Blanchard et al., 2017). To defend against Byzantine attacks, a common approach is to replace gradient averaging with robust aggregation rules (AGRs) (Chen et al., 2017; Yin et al., 2018). These methods have demonstrated their effectiveness in achieving Byzantine-robustness when client data adheres to the independent and identically distributed (IID) assumption. However, in practical applications, client data often deviate from the IID pattern (McMahan et al., 2017; Wang et al., 2021; Kairouz et al., 2021). This non-IIDness introduces increased variation among honest clients' gradients, posing challenges in detecting and excluding Byzantine clients.

Our work mainly focuses on label skewness, a typical non-IID setting where each client only has access to a few classes of data (Li and Zhan, 2021; Shen et al., 2022). In this setting, while clients share the same conditional data distribution given labels, their label distributions can vary a lot. For instance, in animal image classification, users from various regions may capture images of distinct species prevalent in their areas, even though these species share similar visual characteristics. Label skewness introduces two key challenges for model performance. First, it introduces a selection bias of clients, causing robust AGRs to favor certain clients over others, thus biasing the model. Secondly, it amplifies the variation among gradients of honest clients, making AGRs more vulnerable to Byzantine attacks. Thus more advanced techniques are required to tackle these challenges.

Focusing on label skewness, we find that the gradients of honest clients distribute near a $(c-1)$-simplex, where c is the number of classes. Leveraging this insight, we introduce *BOBA* (Byzantine-rObust and unBiased Aggregator), a two-stage AGR to estimate this simplex effectively. In the first stage, we robustly estimate the low dimensional affine subspace containing the simplex and project all gradients onto the subspace. In the second stage, we use a few data samples on the server to estimate the $(c-1)$-simplex and further filter out potential Byzantine gradients. As a result, BOBA ensures that honest gradients re-

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

main largely unaffected, with only minor perturbations, while Byzantine gradients are either discarded or significantly weakened. Our contributions can be summarized as follows:

- We make a systematic analysis of FL robustness challenges in the presence of label skewness, including the identification of two key challenges: selection bias and increased vulnerability (Sec. 3).
- We introduce BOBA, which incorporates an objective addressing label skewness and robustness, along with an efficient optimization algorithm (Sec. 4).
- We provide theoretical analysis that derives gradient estimation error and convergence guarantee, demonstrating BOBA's unbiasedness and optimal order robustness (Sec. 5).
- We empirically evaluate the unbiasedness and robustness of BOBA across diverse models, datasets, and attack scenarios, outperforming various baseline AGRs and extending to more complex non-IID settings (Sec. 6).

2 RELATED WORKS

Robust AGRs with IID clients Extensive research has been conducted on robust AGRs tailored for IID clients. These AGRs modify the server's gradient averaging step, and can be categorized into two main groups: majority-based and reference-based methods. *Majority-based AGRs* operate under the assumption that the gradients of honest clients tend to cluster together. They employ robust mean estimators, including coordinate-wise median (Yin et al., 2018), geometric median (Chen et al., 2017; Pillutla et al., 2022), and Krum (Blanchard et al., 2017), to identify a vector close to the majority of gradients. While these methods have been theoretically proven to perform well in IID settings, our analysis reveals that they exhibit issues such as selection bias and increased vulnerability when confronted with label skewness scenarios. In many FL systems, the server possesses a limited amount of data (Zhao et al., 2018; Lin et al., 2020). Although this data may be insufficient to independently train a satisfactory model, reference-based AGRs leverage server data to assess each client's update and adjust their contributions to enhance robustness. Loss-based rejections (Fang et al., 2020) evaluate client updates with their loss on server data, and drop clients whose updates are the most harmful. Zeno (Xie et al., 2019b) extends this approach by considering both loss and gradient scales. FLTrust (Cao et al., 2021) computes a server gradient using server data and reweighs client gradients based on their similarity to the server gradient. ByGARS (Regatti et al., 2022) optimizes the aggregation weights for client gradients using server data in a meta-learning framework. However, it is worth noting that these methods are not specifically designed to address the non-IID challenges inherent in FL scenarios.

Robust AGRs with non-IID clients A few works have studied robustness with non-IID clients. Karimireddy et al. (2022) combine IID AGRs with bucketing to enhance homogeneity in AGR inputs, albeit with a trade-off in robustness. Similar to BOBA, RAGE (Data and Diggavi, 2021) also uses singular value decomposition (SVD) for robust aggregation. However, it uses SVD to remove Byzantine clients iteratively, whereas our work focuses on applying SVD to model the distribution of honest clients' gradients. Ghosh et al. (2019) group clients into IID clusters and train global models in each group. A topic related to selection bias is performance fairness, where each client should have similar accuracy. Hu et al. (2020) introduce a multi-task learning framework to learn a robust and fair global model. However, it is not robust to Byzantine attacks and can only guarantee Pareto optimal. Ditto (Li et al., 2021) learns personalized models to achieve fairness and robustness, but still requires training a robust global model.

For additional related works on FL with label skewness and non-IIDness, please refer to Appendix A.

3 FL WITH LABEL SKEWNESS

Setup We study the FedSGD (McMahan et al., 2017) system consisting of one central server and n clients. Each client is either *honest* (in honest set $\mathcal{H}$) or *Byzantine* (in Byzantine set $\mathcal{B}$), with $|\mathcal{H}|$ and $|\mathcal{B}|$ representing the *real* number of honest and Byzantine clients, respectively. In each communication round, the server broadcasts the parameter $\boldsymbol{w}_G \in \mathbb{R}^d$ to all clients. Each honest client $i \in \mathcal{H}$ computes the gradient with its own data $\{\boldsymbol{\xi}_{ij}\}_{j=1}^{m_i}$ sampled from P_i and sends back the *honest gradient* $\boldsymbol{g}_i = \nabla_{\boldsymbol{w}_G}\mathcal{L}_i(\boldsymbol{w}_G)$, where $\mathcal{L}_i(\boldsymbol{w}_G) = \frac{1}{m_i}\sum_{j=1}^{m_i}\ell(\boldsymbol{w}_G;\boldsymbol{\xi}_{ij})$ and ℓ is the loss function. Each Byzantine client can send arbitrary *Byzantine gradient* to the server. Finally, the server aggregates all n gradients $\hat{\boldsymbol{\mu}} = \mathrm{Agg}(\{\boldsymbol{g}_i\}_{i=1}^n)$ and updates the parameter $\boldsymbol{w}_G \leftarrow \boldsymbol{w}_G - \eta\hat{\boldsymbol{\mu}}$, where $\mathrm{Agg}(\cdot)$ is the aggregation rule (AGR), and η is the learning rate.

For each honest client $i \in \mathcal{H}$, let $\mathbb{E}\boldsymbol{g}_i$ be its *expected gradient*, where the expectation is taken on data sampling from P_i. During training, the system minimizes the empirical risk, $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathcal{L}_i(\boldsymbol{w}_G)$. FL aims to train a model with low population risk, $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}\mathcal{L}_i(\boldsymbol{w}_G)$. Let $\boldsymbol{\mu} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{g}_i$ denote the gradient of empirical risk and $\mathbb{E}\boldsymbol{\mu} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\mathbb{E}\boldsymbol{g}_i$ denote its expectation,

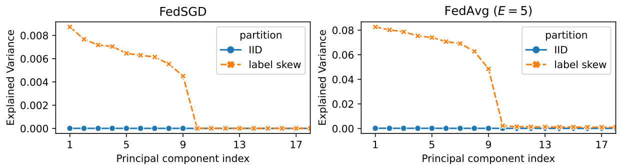

Figure 1: PCA of honest gradients on MNIST ($c = 10$). Over 99% of the variance concentrate on the first $(c-1)$ principal components, verifying that honest gradients distribute near the honest subspace.

which is also the gradient of population risk.

Byzantine attack In each round, Byzantine clients can send arbitrary vectors to the server, which may depend on current global model $\boldsymbol{w}_G$ and honest gradients $\{\boldsymbol{g}_i\}_{i\in\mathcal{H}}$. They can also collude to perform stronger attacks, e.g., by sending the same vector.

Robust AGRs aim to find a robust estimation of $\mathbb{E}\boldsymbol{\mu}$. Since the server has no prior knowledge about the exact number of Byzantines, we let f be the *Byzantine tolerance*, a hyperparameter such that AGRs guarantee to be robust when $|\mathcal{B}| \leq f$. Similar to previous works (Xie et al., 2019b; Cao et al., 2021; Lin et al., 2020), we assume the AGR has access to small amount of clean data to improve robustness. Notice that such data are collected and labeled by the server, rather than uploaded by clients (Cao et al., 2021). Finally, since Byzantines in FL can change their index over rounds, the AGR can only use the information from the current round, including the global model, all clients' uploaded gradients, and server data.

3.1 Distribution of Honest Gradients

This subsection analyzes the distribution of honest gradients with label skewness. We start with some definitions.

Definition 3.1 (Inner, outer and total variations)**.** For an honest client $i \in \mathcal{H}$, its inner variation is $\mathbb{E}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2$; its outer variation is $\|\mathbb{E}\boldsymbol{g}_i - \mathbb{E}\boldsymbol{\mu}\|_2^2$, and its total variation is $\mathbb{E}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{\mu}\|_2^2$.

Inner variation measures the randomness of sampling data from client i's local distribution P_i, while outer variation measures the difference between local distribution P_i and global distribution $\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} P_i$, without randomness.

In the IID setting, the outer variation is zero, implying that *honest gradients* $\{\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ *are distributed around the same center* $\mathbb{E}\boldsymbol{\mu}$. However, this implication does not hold under label skewness, since the outer variations are non-zero. We formally define label skewness and analyze the distribution of honest gradients.

Definition 3.2 (c-label skew distribution)**.** The data distributions $\{P_i\}_{i\in\mathcal{H}}$ of honest clients are considered c-label skew distributions if they can be expressed as

$$P_i(\boldsymbol{\xi}) = \sum_{z=1}^{c} p_{iz} Q_z(\boldsymbol{\xi}), \quad \forall i \in \mathcal{H}$$

where $P_i(\boldsymbol{\xi})$ is the data distribution of client i, the label z can take c finite values, $p_{iz} \geq 0$ is the label distribution of client i subject to $\sum_{z=1}^{c} p_{iz} = 1$, and $Q_z(\boldsymbol{\xi}) = P_i(\boldsymbol{\xi}|z)$ represents the conditional distribution given label z. Different clients share the same $\{Q_z(\boldsymbol{\xi})\}_{z=1}^{c}$ while having distinct label distributions $\boldsymbol{p}_i = [p_{i1}, \cdots, p_{iz}]^\top$.

The c-label skew distribution assumes the heterogeneity among honest clients can be characterized by their divergence in label distribution. With this condition, we can analyze the distribution of honest gradients.

Proposition 3.3 (Expectation of honest gradients)**.** *With c-label skew distribution, $\forall i \in \mathcal{H}$, we have*

$$\begin{aligned}\mathbb{E}\boldsymbol{g}_i &= \sum_{\boldsymbol{\xi}} P_i(\boldsymbol{\xi})\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w};\boldsymbol{\xi}) = \sum_{\boldsymbol{\xi}}\sum_{z=1}^{c} p_{iz}Q_z(\boldsymbol{\xi})\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w};\boldsymbol{\xi}) \\ &= \sum_{z=1}^{c} p_{iz}\nabla_{\boldsymbol{w}}\sum_{\boldsymbol{\xi}} Q_z(\boldsymbol{\xi})\mathcal{L}(\boldsymbol{w};\boldsymbol{\xi}) = \sum_{z=1}^{c} p_{iz}\mathbb{E}\boldsymbol{\gamma}_z\end{aligned}$$

where $\mathbb{E}\boldsymbol{\gamma}_z = \nabla_{\boldsymbol{w}}\sum_{\boldsymbol{\xi}} Q_z(\boldsymbol{\xi})\mathcal{L}(\boldsymbol{w};\boldsymbol{\xi})$ is the expected gradient computed with data from class z.

Proposition 3.3 shows that each expected honest gradient is a convex combination of $\{\mathbb{E}\boldsymbol{\gamma}_z\}_{z=1}^{c}$, forming a $(c-1)$-simplex in its range. We define the *honest simplex* as $\{\sum_{z=1}^{c} p_z\mathbb{E}\boldsymbol{\gamma}_z : \sum_{z=1}^{c} p_z = 1, p_z \geq 0\}$, and the *honest subspace* as $\{\sum_{z=1}^{c} p_z\mathbb{E}\boldsymbol{\gamma}_z : \sum_{z=1}^{c} p_z = 1\}$.

As honest gradients are perturbations of their expectations, *they distribute near the honest simplex*, approximately forming a $(c-1)$-dimensional affine subspace. Thus, if we conduct principal component analysis (PCA) on honest gradients, the variance should concentrate on the first $(c-1)$ principal components. Figure 1 verifies our finding on MNIST (Lecun et al., 1998). Appendix C.7 gives details of this experiment.

3.2 Challenges of Label Skewness

In the IID scenario, each honest gradient serves as an unbiased estimator of $\mathbb{E}\boldsymbol{\mu}$, simplifying the design of robust AGRs which merely require the identification of one honest gradient (or a close Byzantine gradient). However, in label skewness settings, each honest gradient can exhibit substantial deviations from $\mathbb{E}\boldsymbol{\mu}$, giving rise to two key challenges: *selection bias* and *increased vulnerability*.

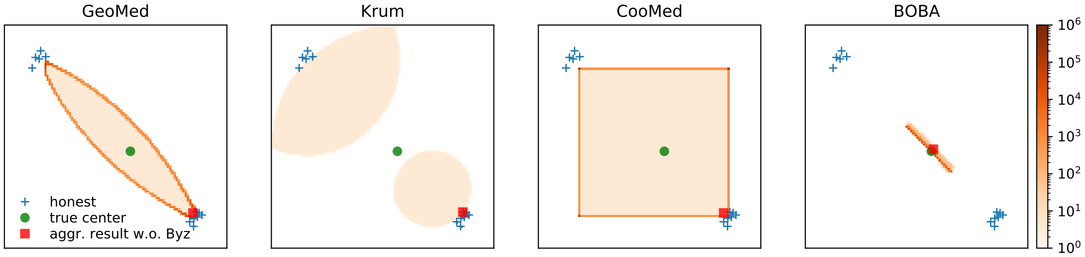

Figure 2: Comparison of aggregation results. (1) *Selection bias*: Without attacks, the aggregation results (■) for GeoMed, Krum and CooMed are biased toward the majority class in the lower-right corner and deviate from the honest gradient center (●), indicating their large biases. Meanwhile, BOBA is unbiased. (2) *Increased vulnerability*: With different attacks, the aggregation results will be different. The orange region represents the heatmap (2D histogram) of possible aggregation results given various attacks, where larger radius indicates worse robustness. BOBA has smallest radius, showing its stronger robustness than IID AGRs.

Selection bias Many robust AGRs, e.g., Krum (Blanchard et al., 2017), select a subset of gradients for aggregation. With label skewness, these AGRs tend to select some clients more frequently, often discarding clients with higher outer variations or deviates from the majority. This selection bias introduces bias into the aggregation results, *even in the absence of any attacks*. In Figure 2, where honest gradients form two clusters, each representing a different class of samples, baseline AGRs consistently choose the majority class. This results in the FL model exclusively training on one class of samples, converging to a trivial solution.

Increased vulnerability With label skewness, baseline AGRs are more vulnerable to attacks, resulting in larger variations in the aggregation results, primarily due to the increased total variation. In Figure 2, the aggregation results of baseline AGRs exhibit a considerable range, much larger than the inner variation (variation of each cluster). Interestingly, this vulnerability occurs not only on the direction of outer variation, but also its orthogonal direction.

In summary, IID AGRs are neither unbiased nor sufficiently robust in the more realistic label skewness setting. It is necessary to design a new robust AGR.

4 PROPOSED BOBA ALGORITHM

In this section, we propose BOBA and explain its two stages in detail. In stage 1, we robustly find the honest subspace, and project all gradients to this subspace. In stage 2, we estimate the vertices of the honest simplex, reconstruct the label distribution for each client, and drop clients with abnormal label distribution (i.e., with strongly negative entries). Intuitively, all honest gradients will be kept with small perturbation, while all Byzantine gradients will either be weakened (projected to the honest simplex in stage 1) or discarded (in stage 2). Therefore, the negative impact of Byzantine gradients can be largely mitigated.

Algorithm 1 BOBA Framework

```
Input: $\boldsymbol{G} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n]$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_c]$, $n, f, c, p_{\min}$
Output: Aggregation result $\hat{\boldsymbol{\mu}}$
Initialize subspace $\hat{\mathcal{P}}$: $\boldsymbol{m}, \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \mathrm{TrSVD}_{c-1}(\boldsymbol{\Gamma})$
while not converge do
    Update $\boldsymbol{r}$: $\boldsymbol{G}_{[n-f]} = \{n - f$ gradients in $\boldsymbol{G}$ with smallest $\|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i)\|_2\}$ where $\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) = \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m}) + \boldsymbol{m}$
    Update $\hat{\mathcal{P}}$: $\boldsymbol{m}, \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \mathrm{TrSVD}_{c-1}(\boldsymbol{G}_{[n-f]})$
Encode: $\tilde{\boldsymbol{g}}_i = \boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m}), \forall i$; $\tilde{\boldsymbol{\Gamma}} = \boldsymbol{U}^\top(\boldsymbol{\Gamma} - \boldsymbol{m}\boldsymbol{1}^\top)$
Estimate: $\hat{\boldsymbol{p}}_i = \begin{bmatrix} \tilde{\boldsymbol{\Gamma}} \\ \boldsymbol{1}^\top \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\boldsymbol{g}}_i \\ 1 \end{bmatrix}, \forall i$
Filter: $\boldsymbol{a} = \mathcal{A}(\{\hat{\boldsymbol{p}}_i\}_{i=1}^n)$
Aggregate: $\tilde{\boldsymbol{\mu}} = \sum_{i=1}^n a_i \tilde{\boldsymbol{g}}_i / \sum_{i=1}^n a_i$
Decode: $\hat{\boldsymbol{\mu}} = \boldsymbol{U}\tilde{\boldsymbol{g}}_G + \boldsymbol{m}$
```

4.1 Stage 1: Fitting the Honest Subspace

The goal of stage 1 is to find a $(c-1)$-dimensional affine subspace close to all honest gradients under the influence of Byzantine gradients. When there are no Byzantines, a standard way to find the subspace is TrSVD, i.e., truncated singular value decomposition on centralized gradients,

$$\boldsymbol{m}, \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \mathrm{TrSVD}_{c-1}(\boldsymbol{G}), \text{ s.t. } \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \approx \boldsymbol{G} - \boldsymbol{m}\boldsymbol{1}^\top$$

where $\boldsymbol{G} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n] \in \mathbb{R}^{d \times n}$ is the client gradient matrix, $\boldsymbol{m} = \frac{1}{n}\boldsymbol{G}\boldsymbol{1} \in \mathbb{R}^d$ is their average, $\boldsymbol{U} \in \mathbb{R}^{d \times (c-1)}, \boldsymbol{V} \in \mathbb{R}^{n \times (c-1)}$ are column-orthogonal and $\boldsymbol{\Sigma} \in \mathbb{R}^{(c-1) \times (c-1)}$ is diagonal. TrSVD fits a $(c-1)$-dimensional affine subspace $\mathcal{P} = \{\boldsymbol{U}\boldsymbol{\lambda} + \boldsymbol{m} : \boldsymbol{\lambda} \in \mathbb{R}^{c-1}\}$ minimizing the *reconstruction loss*

$$\ell(\mathcal{P}) = \sum_{i=1}^n \|\boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i)\|_2^2$$

where $\Pi_{\mathcal{P}}(\boldsymbol{g}_i) = \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m}) + \boldsymbol{m}$ is a projection function that projects vectors to $\mathcal{P}$.

However, vanilla TrSVD is not robust to Byzantine attacks. When there are Byzantine gradients deviating from the honest subspace, the fitted subspace will be dragged to these Byzantine gradients at the cost of underfitting honest ones. For example in $\mathbb{R}^2$, when n honest gradients are uniformly distributed on a segment of $\{(x, y) : x \in [-1, 1], y = 0\}$. TrSVD will fit a subspace of $\{y = 0\}$. However, one Byzantine gradient of $(100n, 100n)$ can alter the fitted subspace to about $\{y = x\}$. Therefore, we design a new objective.

Objective We design *trimmed reconstruction loss* to robustify TrSVD:

$$\ell_t(\mathcal{P}) = \min_{\substack{\boldsymbol{r}\in\{0,1\}^n \\ \sum_{i=1}^n r_i = n-f}} \sum_{i=1}^n r_i \left\|\boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i)\right\|_2^2$$

BOBA stage 1 fits an affine subspace $\hat{\mathcal{P}}$ by minimizing the trimmed reconstruction loss above, which selects the $n-f$ nearest neighbors ($r_i = 1$) and ignores f gradients furthest from $\hat{\mathcal{P}}$ ($r_i = 0$). Intuitively, if Byzantines are far from the $\hat{\mathcal{P}}$, they will be ignored so $\hat{\mathcal{P}}$ is not affected; if Byzantines are close to $\hat{\mathcal{P}}$, the $n - f$ nearest neighbors of $\hat{\mathcal{P}}$ still includes at least $n - 2f$ honest gradients, which are enough to reconstruct the honest subspace (by Assumption 5.3 in Section 5). We show in Appendix B.2.5 that stage 1 is theoretically guaranteed to estimate the honest subspace robustly.

The strongest colluding Byzantines may focus on another dimension different from the $c - 1$ honest dimensions. But BOBA stage 1 will not identify the Byzantine dimension as honest. If it makes such a mistake, the $n - f$ nearest neighbors will form a c-dimensional affine subspace, including one Byzantine dimension and $c-1$ honest dimensions (since there are at least $n - 2f$ honest gradients in the $n - f$ nearest neighbors). Conducting TrSVD on these $n - f$ nearest neighbors results in large loss proportional to the outer variation, which is clearly sub-optimal. Meanwhile, correctly identifying all honest dimensions results in a loss unrelated to outer variations, which is much smaller. In our experiments, we also show that BOBA can resist such colluding Byzantines, e.g. IPM (Xie et al., 2019a) and LIE (Baruch et al., 2019).

Optimization To minimize trimmed reconstruction loss, we solve a joint minimization problem

$$\hat{\mathcal{P}}, \hat{\boldsymbol{r}} = \mathop{\mathrm{argmin}}_{\substack{\mathcal{P}, \boldsymbol{r}\in\{0,1\}^n \\ \sum_{i=1}^n r_i = n-f}} \ell_t(\mathcal{P}, \boldsymbol{r}) = \sum_{i=1}^n r_i \left\|\boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i)\right\|_2^2$$

Fixing $\mathcal{P}$, the optimal $\boldsymbol{r}$ selects the $n-f$ nearest neighbors of $\mathcal{P}$; while fixing $\boldsymbol{r}$, the optimal $\mathcal{P}$ can be fitted by conducting TrSVD on the selected $n - f$ gradients. A naive way to minimize trimmed reconstruction loss is *exhaustive searching* (BOBA-ES), which iterates every possible value of $\boldsymbol{r}$, conducts TrSVD to fit $\mathcal{P}$, and chooses the $\mathcal{P}$ with the smallest trimmed reconstruction loss. It can guarantee the global minimum but have exponentially high computational complexity.

Instead, we use *alternating optimization*, with details in lines 2 - 4 in Algorithm 1. It alternatively updates $\mathcal{P}$ and $\boldsymbol{r}$ until convergence. Although the global minimum may not be guaranteed, alternating optimization can converge to a high-quality local minimum with just a few steps. Thus, it is more efficient and practical for large-scale FL.

After minimization, we project every gradient to the fitted subspace $\hat{\mathcal{P}}$. The projection can weaken Byzantine gradients by eliminating its components orthogonal to $\hat{\mathcal{P}}$; meanwhile, it only introduces small bounded perturbation to honest gradients. However, only applying stage 1 does not fully guarantee robustness: a Byzantine may still have large components along $\hat{\mathcal{P}}$ that bias the aggregation. We design stage 2 to further rule out such Byzantine gradients.

4.2 Stage 2: Finding the Honest Simplex

In stage 2, BOBA uses a small amount of server data to estimate c vertices of the honest simplex, and estimates the label distribution of each client. Gradients with negative entries in the label distribution lie outside the honest simplex, and will be discarded.

Proposition 3.3 shows that each vertex of the honest simplex is the expected gradient computed with one class of data. Thus, we initialize c virtual clients on the server, each with one class of data, and compute *server gradients* $\{\boldsymbol{\gamma}_z\}_{z=1}^c$ with the same process of honest clients. To estimate the label distribution of a client i, we solve for $\{\hat{p}_{iz}\}_{z=1}^n$, s.t.

$$\sum_{z=1}^c \hat{p}_{iz}\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i), \quad \sum_{z=1}^c \hat{p}_{iz} = 1$$

Solving this linear system in the gradient space $\mathbb{R}^d$ is inefficient. Instead, we split the projection into two steps: encoding ($\tilde{\boldsymbol{g}}_i = \boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m})$) and decoding ($\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) = \boldsymbol{U}\tilde{\boldsymbol{g}}_i + \boldsymbol{m}$), and solve the linear system in the latent space $\mathbb{R}^{c-1}$, which has an explicit solution (see line 6 in Algorithm 1).

If our estimation is perfect (e.g., when $\boldsymbol{g}_i = \mathbb{E}\boldsymbol{g}_i, \boldsymbol{\gamma}_z = \mathbb{E}\boldsymbol{\gamma}_z$), $\hat{\boldsymbol{p}}_i$ will lie in the probability simplex, i.e. $\{\boldsymbol{p} : \boldsymbol{1}^\top\boldsymbol{p} = 1, \boldsymbol{p} \geq \boldsymbol{0}\}$ if client i is honest, while it can be arbitrary if client i is Byzantine. So we can discard clients with negative entries in $\hat{\boldsymbol{p}}_i$, since they must be Byzantines. However in practice, our estimation has a bounded error (Appendix B.2.6). Thus, if an honest client does not have data from a class, which is very common, it can also have a slightly negative entry.

Therefore, we design an acceptance criterion

$$\boldsymbol{a} = \mathcal{A}(\{\hat{\boldsymbol{p}}_i\}_{i=1}^n), \quad \text{where } a_i = \mathbb{I}\{\min_z p_{iz} \geq p_{\min}\}$$

where $\mathbb{I}$ is the indicator function and $p_{\min} \leq 0$ is a hyper-parameter deciding the threshold of rejecting Byzantines. In our implementation, we will accept $n-f$ clients with largest $\min_{z=1}^c p_{iz}$ if $\sum_{i=1}^n a_i \leq n-f$ (i.e., our acceptance criterion drops too many clients), since there should be at least $n-f$ honest clients. After dropping Byzantines ($a_i = 0$), we average the remaining projected gradients as the aggregation result of BOBA.

4.3 Computational complexity

The computational complexity of BOBA is $\mathcal{O}(kcnd)$, if it conducts TrSVD for k times. The complexity of TrSVD is $\mathcal{O}(cnd)$ (Halko et al., 2011), where c is the number of classes, n is the number of clients and d is the dimension of gradients. When k, c are small constants, BOBA has the same complexity as vanilla averaging, which is very efficient. Practically we also observe that k is very small. In our experiments with MNIST, CIFAR-10 and AG-News, $k = 3.29, 3.20, 4.77$ on average, respectively. A detailed analysis of the complexity for each step is provided in Appendix B.4.

5 THEORETICAL ANALYSIS

This section presents the convergence analysis of BOBA. We first establish a connection between convergence and gradient estimation error in Proposition 5.1. Subsequently, we demonstrate in Theorem 5.5 that BOBA has bounded gradient estimation error, ensuring guaranteed convergence. Through our analysis, we confirm that the order of BOBA's gradient estimation error aligns with the lower bound of the gradient estimation error in non-IID setting, surpassing IID AGRs. This illustrates the unbiasedness and optimal order robustness of BOBA. *Detailed proofs are deferred to Appendix B due to space limit.*

Proposition 5.1 (Convergence)**.** *With non-negative L-smooth population risk $\mathcal{L}(\boldsymbol{w})$, we conduct SGD with noisy gradient $\hat{\boldsymbol{\mu}} = \hat{g}(\boldsymbol{w})$ and step size $\eta = \frac{1}{L}$. If the gradient estimation error $\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathbb{E}\|\hat{g}(\boldsymbol{w}) - \nabla\mathcal{L}(\boldsymbol{w})\|_2^2 \leq \Delta^2$ for all $\boldsymbol{w}$, then for any weight initialization $\boldsymbol{w}^{(0)}$, after T steps,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \leq 2\frac{L}{T}\mathcal{L}(\boldsymbol{w}^{(0)}) + \Delta^2$$

Proposition 5.1 shows that with a robust AGR featuring bounded gradient estimation error, FedSGD converges to a flat region with small gradient in expectation, with convergence rate $\frac{1}{T}$ and error rate Δ^2. Essentially, a smaller gradient estimation error contributes to improved model convergence. Subsequently, we proceed to derive the gradient estimation error of BOBA. To facilitate this analysis, we introduce the following assumptions:

Assumption 5.2 (Bounded variations)**.** For all $\boldsymbol{w}$,

1. Bounded honest client inner variations: $\exists\epsilon^2$ s.t., $\mathbb{E}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 \leq \epsilon^2, \forall i \in \mathcal{H}$.
2. Bounded honest client outer variations: $\exists\delta^2$ s.t., $\|\mathbb{E}\boldsymbol{g}_i - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq \delta^2, \forall i \in \mathcal{H}$.
3. Bounded server inner variations: $\exists\epsilon_s^2$ s.t., $\mathbb{E}\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 \leq \epsilon_s^2, \forall z = 1, \cdots, c$.
4. Bounded server outer variations: $\exists\delta_s^2$ s.t., $\|\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq \delta_s^2, \forall z = 1, \cdots, c$.

Assumption 5.2 is standard in FL (Wu et al., 2020; Wang et al., 2021), and is applied to both honest clients and server since they both have clean data.

Assumption 5.3 (Bounded client singular value)**.** There exists $\sigma > 0$ such that for all $\boldsymbol{w}$, conducting centralized SVD on any $n-2f$ expectations of honest gradients, the $(c-1)$-th singular value $\sigma_{c-1} \geq \sigma$.

Assumption 5.3 is a natural extension of the standard "$n-2f > 0$" assumption prevalent in IID AGRs (Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2017). This extension entails that, with c-label skewness, it is imperative for *all honest components to simultaneously outweigh the Byzantine component*. To fulfill this requirement, removing any arbitrary subset of f clients from the set of $n-f$ honest clients should still ensure that the remaining $n-2f$ honest clients affinely span the honest subspace, indicated by $\sigma_{c-1} \geq \sigma, \exists\sigma > 0$. Failure to meet this condition could empower Byzantines to form a cluster to replace an honest component.

Assumption 5.3 also reveals that the robustness of an FL system with label skewness depends not only on n, f and c, but also on the label distribution for each honest client. Considering $c = 2$, when $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ distributes uniformly on the honest simplex (a line segment), Assumption 5.3 holds as long as $n - 2f > 1$ (i.e., $f < \frac{n-1}{2}$), closely resembling the IID setting. However, when half of the honest clients have only positive samples, while the other half have only negative samples, $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ will only be distributed at the two vertices of the honest simplex. In this case, Assumption 5.3 only holds when $n - 2f > \frac{n-f}{2}$ (i.e., $f < \frac{n}{3}$).

With label skewness, a gradient distributed around the honest simplex can either be an honest gradient or a Byzantine gradient mimicking honest gradients to bias the aggregation without being detected. However, it is impossible for *any* AGR to distinguish between the two scenarios. Consequently, this introduces an inevitable component to the gradient estimation error as demonstrated in Proposition 5.4.

Proposition 5.4 (Lower bound of gradient estimation error for any AGR)**.** *Given any AGR, we can find $|\mathcal{H}|$ honest gradients and $|\mathcal{B}|$ Byzantine gradients, such that $\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \geq \Omega(\beta^2\delta^2)$, where $\beta = \frac{|\mathcal{B}|}{n} = \frac{|\mathcal{B}|}{|\mathcal{H}|+|\mathcal{B}|}$ is the fraction of Byzantine clients.*

We finally derive BOBA's gradient estimation error and show that it matches with the error bound above.

Theorem 5.5 (Upper bound of gradient estimation error for BOBA)**.** *With Assumptions 5.2 and 5.3, BOBA has*

$$\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq C_1\epsilon^2 + C_2\epsilon_s^2 + C_3\beta^2\delta_s^2$$

where $\beta = \frac{|\mathcal{B}|}{n}$ is the fraction of Byzantine clients, $C_1 = 4+8(\frac{1}{n-2f}+\frac{\delta^2}{\sigma^2})(2(n-f)+|\mathcal{H}|)$, $C_2 = 16(\frac{1}{n-2f}+\frac{\delta^2}{\sigma^2})(n-f)+16c(1+c|p_{\min}|)^2\beta^2$, $C_3 = 16(1+c|p_{\min}|)^2$.

When the outer variation increases t times, both δ and σ increase t times. When all clients are duplicated, δ^2 does not change but σ^2 is doubled. Thus generally we have $\frac{\delta^2}{\sigma^2} \propto \frac{1}{n}$. When $\epsilon_s = \mathcal{O}(\epsilon), \delta_s = \mathcal{O}(\delta)$, $c = \mathcal{O}(1)$, $\frac{1}{n-2f} = \mathcal{O}(\frac{1}{n})$, $|\mathcal{H}| = \mathcal{O}(n)$, and $|p_{\min}| = \mathcal{O}(1)$, we have $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathcal{O}(\epsilon^2 + \beta^2\delta^2)$. We conclude that

- *BOBA is unbiased.* Without attacks, BOBA preserves all honest gradients, resulting in a gradient estimation error unaffected by outer variation δ.
- *BOBA has optimal order robustness.* With attacks, BOBA' gradient estimation error matches the optimal order in Proposition 5.4 in terms of the outer variation δ, while IID AGRs only guarantee $\mathcal{O}(\epsilon^2 + \delta^2)$ even when $\beta = 0$ (see Appendix B.3.2).

A detailed comparison and analysis of the gradient estimation error is presented in Appendix B.2 and B.3.

6 EXPERIMENTS

In this section, we conduct experiments to answer the following research questions.

- **RQ1**: Is BOBA unbiased and more robust to attacks than baseline AGRs?
- **RQ2**: Is BOBA efficient?
- **RQ3**: How is BOBA affected by the quality and quantity of server data, hyper-parameters, and different label skewness settings?
- **RQ4**: Can BOBA be extended to more complex non-IID settings and other FL frameworks?

Setup We conduct the experiments on a wide range of models and datasets: a 3-layer MLP for MNIST (Lecun et al., 1998), a 5-layer CNN for CIFAR-10 (Krizhevsky and Hinton, 2009), and a GRU network for AG-News (Zhang et al., 2015). We partition training sets to $|\mathcal{H}| = 100/100/160$ honest clients respectively with pathological partition (McMahan et al., 2017), where each client has data from at most two classes. To evaluate unbiasedness, we use $|\mathcal{B}| = 0$. To evaluate robustness, we add $|\mathcal{B}| = 15/15/54$ Byzantine clients *as supplements, not replacements*, resulting in totally $n = 115/115/214$ clients. This design simulates real-world FL systems where adversaries use additional devices to participate in FL training, instead of replacing existing users' devices. Meanwhile, since no data is removed from training, we can directly compare the accuracy with/without Byzantine clients. Appendix C.1 gives the detailed experimental settings.

Attacks We consider six representative attacks: Gauss (Blanchard et al., 2017), IPM (Xie et al., 2019a), LIE (Baruch et al., 2019), Mimic (Karimireddy et al., 2022), MinMax, and MinSum (Shejwalkar and Houmansadr, 2021).

Baseline AGRs We consider 15 baseline AGRs:

- *Average* (McMahan et al., 2017) simply averages all gradients. It is unbiased but vulnerable to attacks.
- *Server* only uses server data to fit a model. We use it to verify that one cannot train a good model with server data only.
- *Majority-based IID AGRs*: coordinate median (CooMed), trimmed mean (TrMean) (Yin et al., 2018), Krum, Multi-Krum (MKrum) (Blanchard et al., 2017), and geometric median (GeoMed) (Chen et al., 2017).
- *Reference-based IID AGRs.* SelfRej, AvgRej (Fang et al., 2020), Zeno (Xie et al., 2019b), FLTrust (Cao et al., 2021) and ByGARS (Regatti et al., 2022).
- *Non-IID AGRs.* Bucketing (Karimireddy et al., 2022) with Krum (B-Krum) or Multi-Krum (B-MKrum), and RAGE (Data and Diggavi, 2021).

All AGRs are set to be robust to $f = 16$ Byzantines on MNIST/CIFAR-10 and $f = 60$ on AG-News. BOBA uses $p_{\min} = -0.5$. We assume limited server data: 20 per class for MNIST/CIFAR-10 and 30 per class for AG-News, much fewer than the samples on each client.

Evaluation of unbiasedness (RQ1) We evaluate the unbiasedness with $|\mathcal{B}| = 0$. Besides accuracy, we introduce max-recall-drop (MRD) as a complement. It computes how the recall scores of each class differ from the model trained with Average (with $|\mathcal{B}| = 0$) and picks the largest absolute drop. Smaller MRD in-

Table 1: Evaluation of unbiasedness (mean (s.d.) % over five random seeds, $|\mathcal{H}| = 100, 100, 160, |\mathcal{B}| = 0$)

Method	MNIST		CIFAR-10		AG-News	
	Acc ↑	MRD ↓	Acc ↑	MRD ↓	Acc ↑	MRD ↓
Average	**92.5** (0.1)	-	**71.7** (0.8)	-	88.3 (0.1)	-
Server	82.0 (0.5)	18.8 (1.9)	24.4 (2.0)	61.7 (1.9)	82.7 (1.4)	8.8 (3.5)
CooMed	73.4 (5.8)	62.9 (24.3)	18.0 (2.8)	79.8 (3.3)	80.4 (4.5)	18.6 (12.0)
TrMean	82.3 (2.7)	59.4 (20.9)	22.3 (11.3)	81.4 (2.2)	86.9 (0.5)	5.8 (3.6)
Krum	39.6 (4.3)	98.1 (0.2)	35.0 (3.0)	81.5 (1.9)	66.8 (2.9)	89.2 (7.0)
MKrum	91.7 (0.1)	10.0 (2.3)	70.5 (0.7)	11.1 (3.7)	88.0 (0.1)	4.6 (2.1)
GeoMed	91.9 (0.1)	3.1 (0.3)	71.6 (0.8)	5.1 (1.1)	**88.4** (0.1)	0.4 (0.2)
SelfRej	91.7 (0.1)	9.6 (0.8)	70.1 (1.2)	13.5 (6.1)	86.6 (1.8)	13.5 (9.4)
AvgRej	91.1 (0.5)	18.1 (8.0)	71.0 (0.5)	11.2 (6.8)	85.8 (0.9)	15.6 (6.2)
Zeno	91.7 (0.1)	10.3 (2.0)	70.2 (0.8)	11.5 (4.1)	86.4 (1.5)	14.1 (8.6)
FLTrust	85.6 (0.6)	18.9 (3.5)	53.1 (0.9)	32.2 (2.7)	86.3 (0.4)	5.8 (1.0)
ByGARS	76.7 (1.4)	59.9 (10.2)	32.0 (1.7)	60.7 (6.4)	44.9 (6.5)	82.0 (4.3)
B-Krum	73.8 (4.8)	93.8 (3.1)	59.0 (1.0)	81.4 (2.2)	87.3 (0.6)	5.0 (2.8)
B-MKrum	92.0 (0.1)	2.9 (0.5)	70.9 (0.8)	6.2 (0.9)	87.8 (0.3)	3.3 (1.5)
RAGE	59.8 (0.5)	90.1 (0.5)	58.3 (1.5)	56.4 (10.0)	63.9 (6.1)	80.2 (5.2)
BOBA	**92.5** (0.1)	**1.3** (1.7)	70.9 (0.9)	**4.0** (1.7)	88.3 (0.1)	**0.2** (0.1)

dicates a less biased AGR. As selection bias may dramatically decrease some classes' recalls while increasing others, MRD can reflect selection bias better than accuracy. As shown in Table 1, most baseline AGRs suffer from significant selection bias, resulting in large MRD. Among baselines, GeoMed and B-MKrum aim to retain as many gradients as possible, consequently achieving smaller MRDs. *We observe that BOBA has accuracy very close to Average, and the smallest MRD among all robust AGRs. It verifies the superior unbiasedness of BOBA.*

Evaluation of robustness (RQ1) We evaluate the robustness with $|\mathcal{B}| = 15, 15, 54$ on three datasets respectively with results shown in Table 2. Considering that Byzantines would select the attack strategy that most effectively degrades model accuracy, we summarize the worst-case accuracy for each defense in the "Wst" column for a clear comparison. *BOBA significantly improves the worst-case accuracy by **6.1%, 18.3%, 1.6%** on three datasets, respectively, showing that BOBA has better robustness than baselines.* Interestingly, we observed that some AGRs (e.g., Mkrum and SelfRej) achieve higher accuracy under certain attacks (e.g., Gauss) compared to no attack conditions. This phenomenon arises from these AGRs relying on accurate estimates of the number of attackers. Without attacks, these AGRs overestimate the number of attackers ($f \gg |\mathcal{B}|$), leading to dropping honest clients. However, with attacks ($f \approx |\mathcal{B}|$), these AGRs drop fewer honest clients, resulting in higher accuracy. Considering that majority-based AGRs do not use server data, we also study whether server data can further improve their robustness in Appendix C.2. We show that server data cannot enhance the most competitive of these AGRs.

Byzantines within the honest simplex Byzantine clients can upload vectors on the boundary of the honest simplex, thereby maximizing the bias in the aggregation results without being detected. The Mimic attack is an example of this type of attack. Although this attack cannot be detected by any AGRs, including BOBA, we found that this attack has a limited impact on model accuracy.

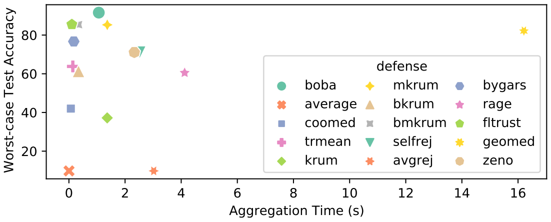

Figure 3: Running time of AGRs on MNIST

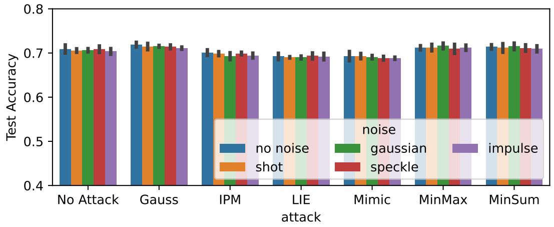

Figure 4: BOBA is robust to corrupted server data

Efficiency (RQ2) We compare the aggregation time of BOBA with baselines on MNIST. *Figure 3 shows that BOBA is faster than half of the baseline AGRs.*

Effect of server data (RQ3) We investigate how the performance of BOBA is influenced by both the quality and quantity of server data. To simulate low-quality data, we introduce four types of random noises to the server data, following the approach proposed by (Hendrycks and Dietterich, 2019). As illustrated in Figure 4, BOBA exhibits remarkable consistency across various noise types, highlighting its robustness to variations in server data quality. Additionally, as demonstrated in Appendix C.3, BOBA exhibits greater resilience to label skewness in server data compared to baseline reference-based AGRs. Moreover, BOBA proves to be effective even with a minimal amount of server data, surpassing all baseline AGRs with just 5 samples per class on CIFAR-10.

Effect of hyper-parameters (RQ3) We show in Appendix C.4 that BOBA is robust to a wide range of f and $p_{\min}$ under multiple fractions of Byzantines $\beta = |\mathcal{B}|/n$.

More label skewness settings (RQ3) In Appendix C.5, we evaluate BOBA under two more label skewness settings: step partition (Chen and Chao, 2021) and Dirichlet partition (Yurochkin et al., 2019). We also test BOBA under different levels of non-IIDness, and different participation rates. We observe

Table 2: Evaluation of robustness (Accuracy, mean (s.d.) % over five random seeds)

Method	MNIST ($\lvert\mathcal{H}\rvert = 100, \lvert\mathcal{B}\rvert = 15$)							CIFAR-10 ($\lvert\mathcal{H}\rvert = 100, \lvert\mathcal{B}\rvert = 15$)							AG-News ($\lvert\mathcal{H}\rvert = 160, \lvert\mathcal{B}\rvert = 54$)						
	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst
Average	9.8 (0.0)	9.8 (0.0)	92.4 (0.1)	**92.1** (0.1)	90.0 (0.2)	90.8 (0.1)	9.8	10.0 (0.0)	10.0 (0.0)	68.2 (0.8)	**70.3** (0.8)	33.2 (5.9)	33.1 (5.3)	10.0	25.4 (2.6)	25.0 (0.0)	87.5 (0.2)	87.2 (0.3)	35.9 (3.6)	30.5 (3.0)	25.0
CooMed	68.0 (6.9)	42.0 (3.7)	89.6 (0.3)	65.0 (6.2)	77.2 (3.1)	77.2 (3.1)	42.0	18.2 (0.8)	7.0 (1.3)	22.0 (0.8)	14.9 (1.9)	18.0 (2.3)	18.0 (2.3)	7.0	86.0 (0.3)	58.6 (9.9)	81.7 (0.3)	82.2 (1.7)	61.2 (17.6)	60.9 (17.4)	58.6
TrMean	91.7 (0.1)	63.8 (10.0)	88.9 (0.6)	83.2 (2.0)	88.8 (0.2)	88.8 (0.2)	63.8	57.3 (1.5)	14.4 (2.6)	30.6 (1.5)	30.1 (5.1)	22.4 (2.4)	23.2 (4.1)	14.1	88.1 (0.3)	57.5 (7.7)	85.2 (0.2)	82.4 (3.8)	67.5 (16.3)	74.4 (5.5)	57.5
Krum	42.6 (3.8)	42.6 (3.8)	91.3 (0.1)	37.2 (6.4)	44.0 (5.1)	42.9 (4.4)	37.2	38.4 (1.7)	35.9 (3.7)	40.1 (2.3)	31.8 (3.7)	34.0 (2.5)	39.1 (2.6)	31.8	66.3 (1.9)	66.8 (1.7)	80.3 (1.0)	46.6 (0.4)	66.2 (2.1)	65.7 (3.3)	46.6
MKrum	92.4 (0.2)	85.3 (5.3)	92.0 (0.2)	91.4 (0.2)	**92.4** (0.1)	**92.3** (0.1)	85.3	71.7 (0.8)	50.9 (11.2)	66.0 (1.1)	69.6 (0.5)	70.1 (0.3)	60.5 (3.0)	50.9	88.3 (0.2)	80.7 (6.0)	86.6 (0.2)	83.4 (0.6)	**88.3** (0.1)	85.9 (0.3)	80.7
GeoMed	91.9 (0.1)	82.2 (0.5)	91.6 (0.1)	89.5 (0.3)	91.2 (0.1)	91.3 (0.1)	82.2	71.5 (0.6)	52.6 (2.5)	43.9 (2.3)	62.1 (0.6)	43.5 (3.0)	43.4 (2.3)	43.4	88.3 (0.1)	77.5 (2.9)	83.5 (0.2)	84.1 (0.2)	83.5 (0.3)	83.6 (0.3)	77.5
SelfRej	92.4 (0.2)	71.1 (2.5)	92.0 (0.1)	91.4 (0.1)	87.6 (1.1)	88.6 (0.7)	71.5	71.7 (0.9)	14.2 (3.3)	66.0 (1.2)	69.3 (0.9)	32.1 (2.3)	32.4 (1.9)	14.2	**88.4** (0.1)	25.0 (0.0)	86.4 (0.3)	84.4 (0.8)	38.2 (10.8)	32.6 (2.3)	25.0
AvgRej	9.8 (0.0)	91.0 (0.4)	91.8 (0.2)	90.7 (0.4)	92.3 (0.1)	92.2 (0.1)	9.8	10.0 (0.0)	**70.5** (0.7)	67.0 (1.2)	71.6 (0.5)	61.7 (5.2)	58.6 (4.6)	10.0	41.1 (7.7)	**88.0** (0.3)	84.6 (0.4)	**88.3** (0.1)	40.7 (7.3)	41.8 (12.1)	40.7
Zeno	92.4 (0.2)	71.1 (2.4)	92.0 (0.1)	91.4 (0.1)	87.6 (1.1)	88.6 (0.7)	71.1	71.5 (0.5)	14.1 (3.3)	65.8 (1.0)	69.4 (0.5)	32.3 (1.1)	31.3 (3.8)	14.1	88.3 (0.1)	25.0 (0.0)	86.5 (0.2)	85.9 (2.1)	53.9 (5.4)	61.6 (13.3)	25.0
FLTrust	85.6 (0.6)	85.6 (0.6)	88.4 (0.7)	85.5 (0.6)	85.8 (0.6)	85.6 (0.6)	85.5	53.0 (0.7)	52.6 (1.1)	48.9 (2.0)	53.3 (1.0)	52.0 (1.7)	51.9 (1.5)	48.9	86.2 (0.5)	86.2 (0.4)	86.2 (0.4)	85.7 (0.8)	85.8 (0.9)	85.8 (0.5)	85.7
ByGARS	76.7 (1.4)	87.5 (0.7)	85.0 (0.7)	77.1 (1.3)	76.6 (1.3)	76.6 (1.3)	76.6	31.9 (1.7)	53.6 (0.8)	30.8 (2.6)	32.2 (1.3)	26.9 (1.9)	26.9 (1.6)	26.9	45.4 (11.2)	48.0 (8.1)	44.5 (11.3)	77.2 (20.1)	59.0 (22.6)	40.7 (2.4)	40.7
B-Krum	78.8 (2.8)	80.0 (1.0)	90.9 (0.4)	61.3 (2.2)	79.3 (2.9)	77.6 (2.5)	61.3	58.1 (2.3)	58.1 (1.1)	42.4 (2.4)	46.0 (2.6)	58.8 (0.8)	57.8 (1.1)	42.4	88.3 (0.1)	51.1 (30.0)	87.0 (1.2)	81.6 (3.8)	86.9 (0.4)	86.2 (0.6)	51.1
B-MKrum	92.4 (0.1)	85.4 (1.8)	92.2 (0.1)	91.4 (0.0)	91.8 (0.2)	91.1 (0.1)	85.4	71.8 (0.6)	32.0 (2.3)	66.0 (0.7)	69.7 (0.8)	45.8 (4.9)	42.9 (2.7)	32.0	88.3 (0.2)	24.9 (12.6)	85.9 (0.2)	84.9 (0.2)	63.7 (14.2)	60.4 (28.3)	24.9
RAGE	82.6 (1.0)	60.5 (0.9)	80.6 (14.0)	63.9 (2.3)	60.4 (0.9)	59.8 (0.5)	59.8	71.7 (0.5)	63.7 (1.3)	48.3 (2.2)	60.2 (1.1)	59.6 (3.0)	56.8 (1.1)	48.3	28.5 (5.6)	69.5 (2.6)	61.2 (9.4)	48.8 (21.7)	70.6 (1.0)	65.5 (7.3)	28.5
BOBA	**92.5** (0.1)	**91.6** (0.2)	**92.5** (0.2)	91.7 (0.4)	92.0 (0.3)	92.0 (0.6)	**91.6**	**71.9** (0.5)	70.1 (0.6)	**69.2** (0.7)	69.3 (1.1)	**71.2** (0.5)	**71.4** (0.5)	**69.2**	88.3 (0.1)	87.7 (0.7)	**88.4** (0.1)	87.3 (0.3)	88.1 (0.1)	**88.3** (0.2)	**87.3**

Table 3: Performance (mean (s.d.) % over five random seeds) on CIFAR-10 with label skewness and image corruptions (see full table in Appendix C.6)

Method	$\lvert\mathcal{B}\rvert = 0$		$\lvert\mathcal{B}\rvert = 15$ (Acc ↑)						
	Acc ↑	MRD ↓	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst
Average	68.7 (0.4)	-	10.0 (0.0)	10.0 (0.0)	64.6 (0.7)	67.5 (0.5)	27.9 (4.9)	21.6 (7.5)	10.0
MKrum	66.8 (1.1)	16.7 (11.7)	68.2 (0.7)	52.9 (10.2)	63.1 (1.1)	54.9 (25.1)	67.2 (0.4)	62.3 (2.1)	52.9
FLTrust	50.1 (0.9)	29.1 (2.1)	50.0 (1.1)	47.8 (1.7)	47.3 (2.5)	49.8 (0.9)	49.0 (1.8)	49.1 (1.9)	47.3
BOBA	66.5 (1.0)	6.7 (2.6)	68.5 (0.3)	66.0 (0.7)	62.8 (1.6)	66.2 (0.7)	67.7 (0.5)	67.5 (0.6)	62.8

Table 4: Ablation study (Accuracy, mean (s.d.) % over five random seeds, AG-News with $\lvert\mathcal{H}\rvert = 16, f = 2$)

Method	$\lvert\mathcal{B}\rvert = 0$	$\lvert\mathcal{B}\rvert = 2$					
		Gauss	IPM	LIE	Mimic	MinMax	MinSum
Average	88.3 (0.1)	25.8 (4.1)	25.0 (0.0)	88.3 (0.1)	88.1 (0.2)	82.7 (0.3)	84.7 (0.1)
BOBA-ES	88.3 (0.1)	88.3 (0.1)	86.3 (0.5)	88.3 (0.1)	88.1 (0.1)	88.3 (0.1)	88.2 (0.1)
BOBA	88.3 (0.1)	88.3 (0.1)	88.1 (0.3)	88.3 (0.1)	88.0 (0.1)	88.4 (0.2)	88.3 (0.2)
BOBA w.o. stage 1	83.0 (1.1)	82.8 (0.8)	82.2 (1.3)	82.8 (1.0)	82.6 (1.1)	82.7 (1.4)	82.7 (1.0)
BOBA w.o. stage 2	88.3 (0.1)	24.8 (0.4)	25.0 (0.0)	88.3 (0.1)	88.0 (0.1)	88.4 (0.2)	88.3 (0.2)

that BOBA has consistent performance across all setting.

Beyond label skewness (RQ4) In a real FL system, label skewness may not be the sole kind of distribution shifts. We consider a setting with both label skewness and feature skewness on CIFAR-10, where we additionally add different types of image corruption to each client (Hendrycks and Dietterich, 2019). Results in Table 3 shows that BOBA still achieves significantly higher worst-case accuracy than baseline AGRs.

More FL frameworks (RQ4) We extend BOBA to more FL frameworks, including FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020b) in Appendix C.7. BOBA still remains effective for these frameworks.

Ablation Study We study how each component of BOBA contributes to the aggregation in Table 4. *BOBA w.o. stage 1* skips the subspace optimization and uses the subspace initialized with server gradients. Though being robust to attacks, it fails to fully utilize clients' data, and thus has a worse performance. *BOBA w.o. stage 2* averages all projected gradients without discarding Byzantine gradients. It is unbiased, but not robust to attacks. *BOBA-ES* uses exhaustive searching instead of alternating optimization to fit the honest subspace, globally minimizing the trimmed reconstruction loss. We observe that BOBA has performance comparable to BOBA-ES while calling TrSVD for much fewer times (≈ 3 v.s. $\binom{n}{f}$), which reduces the computation time from 5.69s to only 13.6ms. We can conclude that (1) both stages in BOBA are necessary to guarantee performance and robustness, and (2) alternating optimization significantly improves the efficiency while maintaining the performance.

7 CONCLUSION

This paper focuses on Byzantine-robustness in FL with label skewness. We show that existing AGRs suffer from selection bias and increased vulnerability, and propose BOBA to alleviate these problems. We verify the unbiasedness and robustness of BOBA theoretically and empirically.

Acknowledgments

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, and the U.S. Department of Homeland Security under Grant Award Number, 17STQAC00001-06-00. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

G. Baruch, M. Baruch, and Y. Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, pages 8632–8642, 2019.

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

X. Cao, M. Fang, J. Liu, and N. Z. Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

H. Chen and W. Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Y. Chen, L. Su, and J. Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2), dec 2017.

D. Data and S. Diggavi. Byzantine-resilient high-dimensional sgd with local iterations on heterogeneous data. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2478–2488. PMLR, 18–24 Jul 2021.

M. Fang, X. Cao, J. Jia, and N. Z. Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium, USENIX Security 2020*, pages 1605–1622. USENIX Association, 2020.

A. Ghosh, J. Hong, D. Yin, and K. Ramchandran. Robust federated learning in a heterogeneous environment. *CoRR*, abs/1906.06629, 2019.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

D. Hendrycks and T. G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu. Fedmgda+: Federated learning meets multi-objective optimization. *CoRR*, abs/2006.11489, 2020.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14 (1–2):1–210, 2021. ISSN 1935-8237.

S. P. Karimireddy, L. He, and M. Jaggi. Learning from history for byzantine robust optimization. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5311–5319. PMLR, 2021.

S. P. Karimireddy, L. He, and M. Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2022.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 1998.

J. Li, W. Abbas, and X. D. Koutsoukos. Byzantine resilient distributed multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020a.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020b.

T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 2021.

X. Li and D. Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 995–1005. ACM, 2021.

T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, 2020.

L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *CoRR*, abs/2012.06337, 2020.

O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal. Federated multi-task learning under a mixture of distributions. In *Advances in Neural Information Processing Systems*, pages 15434–15447, 2021.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

E. M. E. Mhamdi, R. Guerraoui, and S. Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

J. Regatti, H. Chen, and A. Gupta. Byzantine resilience with reputation scores. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2022.

V. Shejwalkar and A. Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

Z. Shen, J. Cervino, H. Hassani, and A. Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022.

G. W. Stewart. Perturbation theory for the singular value decomposition. Technical report, USA, 1990.

J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, B. A. y Arcas, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, S. N. Diggavi, H. Eichner, A. Gadhikar, Z. Garrett, A. M. Girgis, F. Hanzely, A. Hard, C. He, S. Horváth, Z. Huo, A. Ingerman, M. Jaggi, T. Javidi, P. Kairouz, S. Kale, S. P. Karimireddy, J. Konečný, S. Koyejo, T. Li, L. Liu, M. Mohri, H. Qi, S. J. Reddi, P. Richtárik, K. Singhal, V. Smith, M. Soltanolkotabi, W. Song, A. T. Suresh, S. U. Stich, A. Talwalkar, H. Wang, B. E. Woodworth, S. Wu, F. X. Yu, H. Yuan, M. Zaheer, M. Zhang, T. Zhang, C. Zheng, C. Zhu, and W. Zhu. A field guide to federated optimization. *CoRR*, abs/2107.06917, 2021.

Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

C. Xie, O. Koyejo, and I. Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. AUAI Press, 2019a.

C. Xie, S. Koyejo, and I. Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6893–6901. PMLR, 2019b.

Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), jan 2019. ISSN 2157-6904.

D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659. PMLR, 10–15 Jul 2018.

T. Yoon, S. Shin, S. J. Hwang, and E. Yang. Fedmix: Approximation of mixup under mean augmented federated learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

F. X. Yu, A. S. Rawat, A. K. Menon, and S. Kumar. Federated learning with only positive labels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10946–10956. PMLR, 2020.

M. Yurochkin, M. Agarwal, S. Ghosh, K. H. Greenewald, T. N. Hoang, and Y. Khazaeni. Bayesian nonparametric federated learning of neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 2019.

L. Zhang, B. Shen, A. Barnawi, S. Xi, N. Kumar, and Y. Wu. Feddpgan: Federated differentially private generative adversarial networks framework for the detection of COVID-19 pneumonia. *Inf. Syst. Frontiers*, 23(6):1403–1415, 2021.

X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018.

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]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, in Subsection 4.3]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

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, in Appendix B]
   (c) Clear explanations of any assumptions. [Yes, in Section ? and Appendix B.2.2]

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, the code is available at `https://github.com/baowenxuan/BOBA`]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, in Appendix C.1]
   (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, in Appendix C.1]

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. [Not Applicable]
   (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]

Contents

A DETAILED RELATED WORKS

Majority-based robust aggregator with IID clients Majority-based aggregators assume the gradients of honest clients should cluster together, and find a vector near to the majority of the gradients with robust mean estimators, including coordinate-wise median (CooMed) / trimmed mean (TrMean) (Yin et al., 2018), geometric median (GeoMed) (Chen et al., 2017; Pillutla et al., 2022), and Krum (Blanchard et al., 2017). Specifically, CooMed / TrMean use median and trimmed mean on each dimension of the gradient separately. GeoMed minimizes the sum of L2 distances between each gradient and the aggregation result. Krum computes each gradient's sum of square L2 distances to its k nearest neighbors, and picks the gradients with the lowest score. All these methods are robust mean estimators with bounded gradient estimation error, and are theoretically proven not to fail arbitrarily in IID settings. However, as shown in our analysis, they suffer from selection bias and increased vulnerability in label skewness settings. The bounds for gradient estimation error still hold, but become vacuous under severe non-IIDness.

Reference-based robust aggregator with IID clients Another line of works utilize server data to evaluate each client update, and reweigh these clients to achieve better robustness. Loss-based rejections (Fang et al., 2020) evaluate client updates with their loss on server data, and drop clients whose updates are the most harmful. Specifically, we consider SelfRej, which selects $n - f$ clients whose local models $\boldsymbol{w}_i = \boldsymbol{w}_G - \eta \boldsymbol{g}_i$ have the smallest loss, and AvgRej, which selects $n - f$ clients that can lower the loss of averaged model the most. Zeno (Xie et al., 2019b) generalizes this idea by considering both loss and gradient scales, believing that honest gradients should lower the model loss with small gradient norm. FLTrust (Cao et al., 2021) uses server data to compute a server gradient, and reweighs the client gradients with their cosine similarity to the server gradient. ByGARS (Regatti et al., 2022) optimizes the aggregation weights of client gradients with server data in a meta-learning fashion. However in each update step, it still relies on the inner product of (normalized) client gradients and server gradient. Different from the original ByGARS which saves the aggregation weights as the initialization for the next round, we use zero initialization for every round to avoid using historical information. These methods use server data to improve aggregation, however, they are not specifically designed to tackle the non-IID challenge in FL.

Robust aggregator with non-IID clients Recently, a few works have studied robustness with non-IID clients. Karimireddy et al. (2022) combine IID aggregation with bucketing, using averages of random subset of client gradients as inputs of an IID aggregator, e.g., Krum. It makes the inputs of the aggregator more homogeneous. However, bucketing also increases the ratio of Byzantine gradients, which sacrifices some robustness. For example, if there are $|\mathcal{B}|$ Byzantine gradients among totally n gradients, after bucketing with subset size s, there can be as much as $|\mathcal{B}|$ corrupted gradients among totally n/s gradients fed to the aggregator, which increases the ratio of Byzantines from $\frac{|\mathcal{B}|}{n}$ to $s \cdot \frac{|\mathcal{B}|}{n}$. Similar to BOBA, RAGE (Data and Diggavi, 2021) also uses singular value decomposition (SVD) for robust aggregation. However, it uses SVD to remove Byzantine clients iteratively, whereas our work focuses on applying SVD to model the distribution of honest clients' gradients.

Robust aggregator using historical information Some works (Mhamdi et al., 2021; Karimireddy et al., 2021) assume stateful clients or use historical information to improve robustness. They mainly focus on distributed learning, where the index for both honest and Byzantine clients remains the same across communication rounds. However, such assumptions do not hold in FL, especially cross-device FL, where the training clients are different across communication rounds. Therefore, we only focus on algorithms that do not use any historical information.

Robust aggregator for personalized FL While our paper focuses on global FL, where all clients share the same global model, robustness is also studied in personalized FL. Ghosh et al. (2019) divide clients into IID groups and train global models in each group. Ditto (Li et al., 2021) learns personalized models to achieve fairness and robustness, but still requires training a robust global model. Li et al. (2020a) propose a Byzantine-robust multi-task learning system.

Non-IIDness in FL Besides robustness, non-IIDness also raises optimization challenges in FL. When clients take multiple local steps, non-IIDness makes local updates diverge and thus degrades the model. A common method to handle non-IIDness is to share a limited amount of data as augmentation (Zhao et al., 2018), which can be collected in many real applications. To further protect privacy, some works replace the raw samples with aggregated samples (Yoon et al., 2021), or synthetic samples (Zhang et al., 2021). Compared to them, our work assumes very limited server data.

Label Skewness and Mixture Distribution Plenty of works focus on label skewness, a particular sub-class of non-IIDness. FedAwS (Yu et al., 2020) studies an extreme case where each client has only access to one class, while FedRS (Li and Zhan, 2021) focuses on a general label skewness setting. A related non-IID setting is a mixture distribution (Marfoq et al., 2021), where each client's data distribution is a mixture of several shared distributions with its own mixture weights. BOBA mainly focuses on label skewness and can be easily extended to mixture distribution.

B MISSING PROOFS

B.1 Convergence Analysis

In this subsection we provides classical convergence analysis which connects convergence to the gradient estimation error. We consider two cases:

- Smooth and non-negative loss (Proposition 5.1)
- Smooth and strongly convex loss (Proposition B.4)

We start with formal definitions.

Definition B.1 (L-smoothness)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2$$

equivalently, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

Definition B.2 (μ-strong convexity)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

B.1.1 Convergence with Smooth and Non-Negative Loss

In Proposition 5.1, we provides convergence analysis with L-smooth and non-negative loss.

Proposition 5.1 (Convergence with smooth non-negative loss)**.** *With non-negative L-smooth population risk $\mathcal{L}(\boldsymbol{w})$, conducting SGD with noisy gradient $\hat{\boldsymbol{\mu}} = \hat{g}(\boldsymbol{w})$ and step size $\eta = \frac{1}{L}$. If the gradient estimation error $\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathbb{E}\|\hat{g}(\boldsymbol{w}) - \nabla\mathcal{L}(\boldsymbol{w})\|_2^2 \leq \Delta^2$ for all $\boldsymbol{w}$, then for any weight initialization $\boldsymbol{w}^{(0)}$, after T steps,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \leq 2\frac{L}{T}\mathcal{L}(\boldsymbol{w}^{(0)}) + \Delta^2$$

Proof. For any $\boldsymbol{w}^{(t)}$,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}^{(t+1)}) &\leq \mathcal{L}(\boldsymbol{w}^{(t)}) + \nabla\mathcal{L}(\boldsymbol{w}^{(t)})^\top(\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}) + \frac{L}{2}\left\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}\right\|_2^2 && (L\text{-smoothness}) \\
&= \mathcal{L}(\boldsymbol{w}^{(t)}) + \nabla\mathcal{L}(\boldsymbol{w}^{(t)})^\top\left[-\eta\left(\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)}) + \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right)\right] \\
&\quad + \frac{L}{2}\left\|-\eta\left(\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)}) + \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right)\right\|_2^2 \\
&= \mathcal{L}(\boldsymbol{w}^{(t)}) + \left(\frac{L\eta^2}{2} - \eta\right)\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 + \left(L\eta^2 - \eta\right)\nabla\mathcal{L}(\boldsymbol{w}^{(t)})^\top\left(\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right) \\
&\quad + \frac{L\eta^2}{2}\left\|\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \\
&= \mathcal{L}(\boldsymbol{w}^{(t)}) - \frac{1}{2L}\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 + \frac{1}{2L}\left\|\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 && (\eta = \tfrac{1}{L})
\end{aligned}$$

Equivalently,

$$\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \leq 2L\left(\mathcal{L}(\boldsymbol{w}^{(t)}) - \mathcal{L}(\boldsymbol{w}^{(t+1)})\right) + \left\|\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2$$

Average over $t = 0, \cdots, T-1$, we get

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 &\le 2\frac{L}{T}\left(\mathcal{L}(\boldsymbol{w}^{(0)}) - \mathcal{L}(\boldsymbol{w}^{(T)})\right) + \frac{1}{T}\sum_{t=0}^{T-1}\left\|\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \\ &\le 2\frac{L}{T}\mathcal{L}(\boldsymbol{w}^{(0)}) + \frac{1}{T}\sum_{t=0}^{T-1}\left\|\hat{g}(\boldsymbol{w}^{(t)}) - \nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2\end{aligned}$$

Finally, take expectations at both sides

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\mathcal{L}(\boldsymbol{w}^{(t)})\right\|_2^2 \le 2\frac{L}{T}\mathcal{L}(\boldsymbol{w}^{(0)}) + \Delta^2$$

□

B.1.2 Convergence with Smooth and Strongly Convex Loss

Lemma B.3. *Let $f(\boldsymbol{w})$ be L-smooth and μ-strongly convex, conducting GD with exact gradient $\nabla f(\boldsymbol{w})$ and step size $\eta = \frac{2}{L+\mu}$. For all t,*

$$\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^*\|_2 \le \left(\frac{L-\mu}{L+\mu}\right)\|\boldsymbol{w}^{(t)} - \boldsymbol{w}^*\|_2$$

Proof. See Theorem 3.12 in Bubeck (2015). □

Proposition B.4 (Convergence with smooth and strongly convex loss)**.** *With L-smooth and μ-strongly convex population risk $\mathcal{L}(\boldsymbol{w})$, conducting SGD with noisy gradient $\hat{\boldsymbol{\mu}} = \hat{g}(\boldsymbol{w})$ and step size $\eta = \frac{2}{L+\mu}$. If the gradient estimation error $\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathbb{E}\|\hat{g}(\boldsymbol{w}) - \nabla\mathcal{L}(\boldsymbol{w})\|_2^2 \le \Delta^2$ for all $\boldsymbol{w}$, then for any weight initialization $\boldsymbol{w}^{(0)}$, after T steps,*

$$\left\|\boldsymbol{w}^{(T)} - \boldsymbol{w}^*\right\|_2 \le \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^*\right\|_2 + \frac{1}{\mu}\Delta$$

Proof. For any $\boldsymbol{w}^{(t)}$,

$$\begin{aligned}\left\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^*\right\|_2 &= \left\|\boldsymbol{w}^{(t)} - \eta\hat{g}\left(\boldsymbol{w}^{(t)}\right) - \boldsymbol{w}^*\right\|_2 \\ &= \left\|\boldsymbol{w}^{(t)} - \eta\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \boldsymbol{w}^* + \eta\left(\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \hat{g}\left(\boldsymbol{w}^{(t)}\right)\right)\right\|_2 \\ &\le \left\|\boldsymbol{w}^{(t)} - \eta\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \boldsymbol{w}^*\right\|_2 + \eta\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \hat{g}\left(\boldsymbol{w}^{(t)}\right)\right\|_2 \\ &\le \left(\frac{L-\mu}{L+\mu}\right)\left\|\boldsymbol{w}^{(t)} - \boldsymbol{w}^*\right\|_2 + \frac{2}{L+\mu}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \hat{g}\left(\boldsymbol{w}^{(t)}\right)\right\|_2 \qquad \text{(Lemma B.3 and } \eta = \tfrac{2}{L+\mu}\text{)}\end{aligned}$$

By induction,

$$\left\|\boldsymbol{w}^{(T)} - \boldsymbol{w}^*\right\|_2 \le \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)} - \boldsymbol{w}^*\right\|_2 + \sum_{t=0}^{T-1}\left(\frac{L-\mu}{L+\mu}\right)^t\frac{2}{L+\mu}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right) - \hat{g}\left(\boldsymbol{w}^{(t)}\right)\right\|_2$$

Notice that for any $\boldsymbol{w}$,

$$\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2 = \sqrt{\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 - \mathrm{Var}\left(\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2\right)} \le \sqrt{\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2} \le \Delta$$

Finally, take expectations at both sides.

$$\begin{aligned}
\mathbb{E}\left\|\boldsymbol{w}^{(T)}-\boldsymbol{w}^*\right\|_2 &\leq \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\sum_{t=0}^{T-1}\left(\frac{L-\mu}{L+\mu}\right)^t\frac{2}{L+\mu}\mathbb{E}\left\|\nabla\mathcal{L}\left(\boldsymbol{w}^{(t)}\right)-\hat{g}\left(\boldsymbol{w}^{(t)}\right)\right\|_2 \\
&\leq \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\sum_{t=0}^{T-1}\left(\frac{L-\mu}{L+\mu}\right)^t\frac{2}{L+\mu}\Delta \\
&\leq \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\sum_{t=0}^{\infty}\left(\frac{L-\mu}{L+\mu}\right)^t\frac{2}{L+\mu}\Delta \\
&= \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\frac{1}{1-\frac{L-\mu}{L+\mu}}\frac{2}{L+\mu}\Delta \\
&= \left(\frac{L-\mu}{L+\mu}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\frac{1}{\mu}\Delta
\end{aligned}$$

□

Remark. Some previous literature, including Yin et al. (2018), use $\frac{1}{L}$ as the step size, which results in the same parameter estimation error but sub-optimal convergence rate

$$\mathbb{E}\left\|\boldsymbol{w}^{(T)}-\boldsymbol{w}^*\right\|_2 \leq \left(\frac{L-\mu}{L}\right)^T\left\|\boldsymbol{w}^{(0)}-\boldsymbol{w}^*\right\|_2+\frac{1}{\mu}\Delta$$

where $1 > \frac{L-\mu}{L} > \frac{L-\mu}{L+\mu}$. The proof can be found in Theorem 3.10 in Bubeck (2015). Instead, we choose step size $\eta = \frac{2}{L+\mu}$ which improves the convergence rate.

B.2 Upper Bound of Gradient Estimation Error of BOBA

In this subsection we prove bounded gradient estimation error for BOBA. Since the full proof is long, we split it to parts for clarity:

- Subsubsection B.2.1 summarizes the notation used in the proof.
- Subsubsection B.2.2 gives formal assumptions.
- Subsubsection B.2.3 provides useful lemmas used in the proof.
- Subsubsection B.2.4 proves that BOBA stage 1 can converge to an affine subspace with upper bounded trimmed reconstruction loss in expectation.
- Subsubsection B.2.5 proves the robustness of BOBA stage 1, i.e., the fitted subspace is closed enough to the honest subspace.
- Subsubsection B.2.6 proves the robustness of BOBA stage 2, i.e., all honest gradients will not be discarded.
- Subsubsection B.2.7 wraps up the previous subsubsections, and proves the robustness of BOBA.

B.2.1 Notation

We summarize all notations we use in our proof in Table 5.

Table 5: Notation

Notation	Description
d	dimensionality of model parameters and gradient
c	number of classes
n	number of clients
$\mathcal{H}$	set of honest clients
$\lvert\mathcal{H}\rvert$	number of honest clients
$\mathcal{B}$	set of Byzantine clients
$\lvert\mathcal{B}\rvert$	real number of Byzantine clients
f	declared number of Byzantine clients. The aggregator is robust when $f \geq \lvert\mathcal{B}\rvert$
$\boldsymbol{g}_i$	gradient uploaded by client i, $i = 1, \cdots, n$
$\boldsymbol{\mu}$	average of all honest gradients, $\boldsymbol{\mu} = \frac{1}{\lvert\mathcal{H}\rvert}\sum_{i\in\mathcal{H}} \boldsymbol{g}_i$. This is the gradient of empirical loss.
$\mathbb{E}\boldsymbol{g}_i$	expectation of honest gradient $\boldsymbol{g}_i, i \in \mathcal{H}$. Note that Byzantine gradient does not have expectation.
$\mathbb{E}\boldsymbol{\mu}$	expectation of $\boldsymbol{\mu}$. This is the gradient of population loss.
$\hat{\boldsymbol{\mu}}$	aggregation result, $\hat{\boldsymbol{\mu}} = \text{Agg}(\{\boldsymbol{g}_i\}_{i=1}^n)$
ϵ	upper bound of client inner variation, formally defined in Assumption 5.2
δ	upper bound of client outer variation, formally defined in Assumption 5.2
ϵ_s	upper bound of server inner variation, formally defined in Assumption 5.2
δ_s	upper bound of server outer variation, formally defined in Assumption 5.2
σ	lower bound of client singular value, formally defined in Assumption 5.3
σ_s	lower bound of server singular value, formally defined in Assumption 5.3
$\mathcal{P}$	an affine subspace
$\Pi_{\mathcal{P}}$	an affine projection function
$\mathcal{F}(\mathcal{P})$	$n - f$ gradients used to fit $\mathcal{P}$ (among $\{\boldsymbol{g}_i\}_{i=1}^n$)
$\mathcal{N}(\mathcal{P})$	$n - f$ nearest neighbors of $\mathcal{P}$ (among $\{\boldsymbol{g}_i\}_{i=1}^n$)
$\ell_t(\mathcal{P})$	trimmed reconstruction loss of $\mathcal{P}$, $\ell_t(\mathcal{P}) = \sum_{i\in\mathcal{N}(\mathcal{P})} \lVert\boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i)\rVert_2^2$
$\mathcal{P}^*$	the honest subspace. It goes through the expectation of honest gradients $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$
$\hat{\mathcal{P}}$	the projection function fitted by BOBA
$\mathcal{S}$	$n - 2f$ clients that is both honest and in $n - f$ nearest neighbors of $\hat{\mathcal{P}}$, $\mathcal{S} = \{s_1, \cdots, s_{n-2f}\} \subset (\mathcal{H} \cap \mathcal{N}(\hat{\mathcal{P}}))$
$\partial\boldsymbol{S}$	matrix of differences between projections to fitted and ideal affine subspaces of expected gradients in $\mathcal{S}$, $\partial\boldsymbol{S} = [\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_{s_1}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_{s_1}), \cdots, \Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_{s_{n-2f}}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_{s_{n-2f}})] \in \mathbb{R}^{d\times(n-2f)}$
$\Delta\boldsymbol{g}_i$	difference between (fitted) projection and expectation of honest gradient $\boldsymbol{g}_i, i \in \mathcal{H}$, $\Delta\boldsymbol{g}_i = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \mathbb{E}\boldsymbol{g}_i$
$\boldsymbol{\gamma}_z$	server gradient of class z, $z = 1, \cdots, c$
$\mathbb{E}\boldsymbol{\gamma}_z$	expectation of server gradient $\boldsymbol{\gamma}_z, z = 1, \cdots, c$
$\boldsymbol{\Gamma}$	matrix of server gradients, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_c]$
$\mathbb{E}\boldsymbol{\Gamma}$	matrix of expectations of server gradients, $\mathbb{E}\boldsymbol{\Gamma} = [\mathbb{E}\boldsymbol{\gamma}_1, \cdots, \mathbb{E}\boldsymbol{\gamma}_c]$
$\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})$	matrix of projections of server gradients, $\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) = [\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_1), \cdots, \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_c)]$
$\Delta\boldsymbol{\Gamma}$	matrix of differences between (fitted) projection and expectation of server gradients, $\Delta\boldsymbol{\Gamma} = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma} = [\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_1) - \mathbb{E}\boldsymbol{\gamma}_1, \cdots, \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_c) - \mathbb{E}\boldsymbol{\gamma}_c]$
$\boldsymbol{p}_i$	true label distribution of honest client $i \in \mathcal{H}$
$\hat{\boldsymbol{p}}_i$	estimated label distribution of client i
$p_{\min}$	hyperparameter of BOBA, $p_{\min} \leq 0$ in our case
$\hat{\boldsymbol{p}}_{\mathcal{H}}$	average of all estimated label distributions of honest clients, $\hat{\boldsymbol{p}}_{\mathcal{H}} = \frac{1}{\lvert\mathcal{H}\rvert}\sum_{i\in\mathcal{H}} \hat{\boldsymbol{p}}_i$
$\hat{\boldsymbol{p}}_{\mathcal{B}}$	average of all estimated label distributions of Byzantine clients that evading stage 2, $\hat{\boldsymbol{p}}_{\mathcal{B}} = \frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}} \hat{\boldsymbol{p}}_b$ when all Byzantine gradients evade stage 2

B.2.2 Assumptions

In this part, we re-introduce the assumptions mentioned in the main text and provide more explanations in remarks.

Assumption 5.2 (Bounded variations)**.**

1. Honest client inner variation: $\mathbb{E}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 \leq \epsilon^2, \forall i \in \mathcal{H}$.
2. Honest client outer variation: $\|\mathbb{E}\boldsymbol{g}_i - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq \delta^2, \forall i \in \mathcal{H}$.
3. Server inner variation: $\mathbb{E}\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 \leq \epsilon_s^2, \forall z = 1, \cdots, c$.
4. Server outer variation: $\|\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq \delta_s^2, \forall z = 1, \cdots, c$.

Remark.

- Assumption 5.2(1) and 5.2(2) are standard assumptions in both FL and Byzantine-robust FL, e.g., Assumption 6.1.1 (vi) and (vii) in Wang et al. (2021).
- Since server gradients are also 'honest', Assumption 5.2(3) and 5.2(4) simply rewrites Assumption 5.2(1) and 5.2(2) with updated notation.

Assumption 5.3 (Bounded singular values)**.**

1. Honest client singular value: conducting centralized SVD on any $n-2f$ expectations of honest gradients, the $(c-1)$-th singular value $\sigma_{c-1} \geq \sigma > 0$.
2. Server singular value: conducting centralized SVD on all c expectations of server gradients, the $(c-1)$-th singular value $\sigma_{c-1} \geq \sigma_s > 0$.

Remark.

- Assumption 5.3(1) is a natural extension of the standard "$n-2f > 0$" assumption prevalent in IID AGRs (Blanchard et al., 2017; Yin et al., 2018; Chen et al., 2017). This extension entails that, with c-label skewness, it is imperative for *all honest components to simultaneously outweigh the Byzantine component.* To fulfill this requirement, removing any arbitrary subset of f clients from the set of $n-f$ honest clients should still ensure that the remaining $n-2f$ honest clients affinely span the honest subspace, indicated by $\sigma_{c-1} \geq \sigma, \exists \sigma > 0$. Failure to meet this condition could empower Byzantines to form a cluster to replace an honest component.

  Assumption 5.3(1) also reveals that the robustness of an FL system with label skewness depends not only on n, f and c, but also on the label distribution for each honest client. Considering $c = 2$, when $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ distributes uniformly on the honest simplex (a line segment), Assumption 5.3(1) holds as long as $n-2f > 1$ (i.e., $f < \frac{n-1}{2}$), closely resembling the IID setting. However, when half of the honest clients have only positive samples, while the other half have only negative samples, $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ will only be distributed at the two vertices of the honest simplex. In this case, Assumption 5.3(1) only holds when $n-2f > \frac{n-f}{2}$ (i.e., $f < \frac{n}{3}$).
- Assumption 5.3(2) assumes that c server gradients form the vertices of a $(c-1)$-honest simplex, while they do not degrade, i.e., they are not on any $(c-2)$-simplex. We omit Assumption 5.3(2) in the main text for clarity, since Assumption 5.3(1) is a sufficient condition for Assumption 5.3(2).

B.2.3 Useful Lemmas of Affine Projection

In our proof, we frequently use lemmas related to affine subspace and affine projection. For clarity, we formally define these notions and summarize these lemmas.

Definition B.5 (Affine Subspace and Affine Projection)**.** $\mathcal{P}$ is a c-dimensional affine subspace in $\mathbb{R}^d$ if there exists a column-orthogonal $\boldsymbol{U} \in \mathbb{R}^{d\times c}$ and a bias vector $\boldsymbol{m} \in \mathbb{R}^d$, s.t.

$$\mathcal{P} = \{\boldsymbol{U}\boldsymbol{\lambda} + \boldsymbol{m} : \boldsymbol{\lambda} \in \mathbb{R}^c\}$$

The corresponding affine projection function $\Pi_{\mathcal{P}}$ is an affine projection function orthogonally projecting vectors to $\mathcal{P}$.

$$\Pi_{\mathcal{P}}(\boldsymbol{w}) = \boldsymbol{P}(\boldsymbol{w} - \boldsymbol{m}) + \boldsymbol{m}, \quad \forall \boldsymbol{w} \in \mathbb{R}^d$$

where $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top \in \mathbb{R}^{d\times d}$ is a projection matrix whose eigenvalues have c ones and $d - c$ zeros.

Then, we present useful lemmas of affine projection.

Lemma B.6 (Nearest neighbor projection)**.** *For any affine projection function $\Pi_{\mathcal{P}} : \mathbb{R}^d \to \mathbb{R}^d$ and two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$,*

$$\|\Pi_{\mathcal{P}}(\boldsymbol{u}) - \boldsymbol{u}\|_2 \le \|\Pi_{\mathcal{P}}(\boldsymbol{v}) - \boldsymbol{u}\|_2$$

Proof. We first prove that $\Pi_{\mathcal{P}}(\boldsymbol{v}) - \Pi_{\mathcal{P}}(\boldsymbol{u})$ and $\Pi_{\mathcal{P}}(\boldsymbol{u}) - \boldsymbol{u}$ are orthogonal.

$$\begin{aligned}
(\Pi_{\mathcal{P}}(\boldsymbol{v}) - \Pi_{\mathcal{P}}(\boldsymbol{u}))^\top(\Pi_{\mathcal{P}}(\boldsymbol{u}) - \boldsymbol{u}) &= [(\boldsymbol{P}(\boldsymbol{v} - \boldsymbol{m}) + \boldsymbol{m}) - (\boldsymbol{P}(\boldsymbol{u} - \boldsymbol{m}) + \boldsymbol{m})]^\top[\boldsymbol{P}(\boldsymbol{u} - \boldsymbol{m}) + \boldsymbol{m} - \boldsymbol{u}] \\
&= [\boldsymbol{P}(\boldsymbol{v} - \boldsymbol{u})]^\top[(\boldsymbol{P} - \boldsymbol{I})(\boldsymbol{u} - \boldsymbol{m})] \\
&= (\boldsymbol{v} - \boldsymbol{u})^\top[\boldsymbol{P}^\top(\boldsymbol{P} - \boldsymbol{I})](\boldsymbol{u} - \boldsymbol{m}) \\
&= (\boldsymbol{v} - \boldsymbol{u})^\top\boldsymbol{0}(\boldsymbol{u} - \boldsymbol{m}) \\
&= 0
\end{aligned}$$

With this result,

$$\begin{aligned}
\|\Pi_{\mathcal{P}}(\boldsymbol{v}) - \boldsymbol{u}\|_2 &= \sqrt{\|\Pi_{\mathcal{P}}(\boldsymbol{v}) - \Pi_{\mathcal{P}}(\boldsymbol{u})\|_2^2 + \|\Pi_{\mathcal{P}}(\boldsymbol{u}) - \boldsymbol{u}\|_2^2} \\
&\ge \|\Pi_{\mathcal{P}}(\boldsymbol{u}) - \boldsymbol{u}\|_2
\end{aligned}$$

□

Lemma B.7 (1-contraction)**.** *For any projection function $\Pi_{\mathcal{P}} : \mathbb{R}^d \to \mathbb{R}^d$ and two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$,*

$$\|\Pi_{\mathcal{P}}(\boldsymbol{u}) - \Pi_{\mathcal{P}}(\boldsymbol{v})\|_2 \le \|\boldsymbol{u} - \boldsymbol{v}\|_2$$

Proof.

$$\begin{aligned}
\|\Pi_{\mathcal{P}}(\boldsymbol{u}) - \Pi_{\mathcal{P}}(\boldsymbol{v})\|_2 &= \|[\boldsymbol{P}(\boldsymbol{u} - \boldsymbol{m}) + \boldsymbol{m}] - [\boldsymbol{P}(\boldsymbol{v} - \boldsymbol{m}) + \boldsymbol{m}]\|_2 \\
&= \|\boldsymbol{P}(\boldsymbol{u} - \boldsymbol{v})\|_2 \\
&\le \|\boldsymbol{P}\|_2\|\boldsymbol{u} - \boldsymbol{v}\|_2 \\
&\le \|\boldsymbol{u} - \boldsymbol{v}\|_2
\end{aligned}$$

□

Lemma B.8 (Commutativity of affine projection and affine combination)**.** *For any projection function $\Pi_{\mathcal{P}}$: $\mathbb{R}^d \to \mathbb{R}^d$, a set of n vectors $\{\boldsymbol{u}_i\}_{i=1}^n \in \mathbb{R}^d$ and coefficients $\{\lambda_i\}_{i=1}^n$ subject to $\sum_{i=1}^n \lambda_i = 1$,*

$$\Pi_{\mathcal{P}}\left(\sum_{i=1}^n \lambda_i \boldsymbol{u}_i\right) = \sum_{i=1}^n \lambda_i \Pi_{\mathcal{P}}(\boldsymbol{u}_i)$$

Proof.

$$\begin{aligned}\Pi_{\mathcal{P}}\left(\sum_{i=1}^{n}\lambda_i\boldsymbol{u}_i\right) &= \boldsymbol{P}\left(\sum_{i=1}^{n}\lambda_i\boldsymbol{u}_i-\boldsymbol{m}\right)+\boldsymbol{m}\\ &= \sum_{i=1}^{n}\lambda_i[\boldsymbol{P}(\boldsymbol{u}_i-\boldsymbol{m})+\boldsymbol{m}]\\ &= \sum_{i=1}^{n}\lambda_i\Pi_{\mathcal{P}}(\boldsymbol{u}_i)\end{aligned}$$

□

Remark. The sum-to-one constraint on coefficients is crucial as the projection is affine, not linear.

Lemma B.9. *For any two projection functions $\Pi_{\mathcal{P}_1},\Pi_{\mathcal{P}_2}:\mathbb{R}^d\to\mathbb{R}^d$, a set of n vectors $\{\boldsymbol{u}_i\}_{i=1}^n\in\mathbb{R}^d$, coefficients $\{\lambda_i\}_{i=1}^n$ subject to $\sum_{i=1}^n\lambda_i=1$ and an affine combination $\boldsymbol{v}=\sum_{i=1}^n\lambda_i\boldsymbol{u}_i$,*

$$\|\partial\boldsymbol{v}\|_2\le\|\partial\boldsymbol{U}\|_2\cdot\|\boldsymbol{\lambda}\|_2$$

where $\boldsymbol{\lambda}=[\lambda_1,\cdots,\lambda_n]^\top\in\mathbb{R}^n$, $\partial\boldsymbol{v}=\Pi_{\mathcal{P}_1}(\boldsymbol{v})-\Pi_{\mathcal{P}_2}(\boldsymbol{v})\in\mathbb{R}^d$ and $\partial\boldsymbol{U}=[\Pi_{\mathcal{P}_1}(\boldsymbol{u}_1)-\Pi_{\mathcal{P}_2}(\boldsymbol{u}_1),\cdots,\Pi_{\mathcal{P}_1}(\boldsymbol{u}_n)-\Pi_{\mathcal{P}_2}(\boldsymbol{u}_n)]\in\mathbb{R}^{d\times n}$.

Proof.

$$\begin{aligned}\|\partial\boldsymbol{v}\|_2 &= \|\Pi_{\mathcal{P}_1}(\boldsymbol{v})-\Pi_{\mathcal{P}_2}(\boldsymbol{v})\|_2\\ &= \left\|\Pi_{\mathcal{P}_1}\left(\sum_{i=1}^{n}\lambda_i\boldsymbol{u}_i\right)-\Pi_{\mathcal{P}_2}\left(\sum_{i=1}^{n}\lambda_i\boldsymbol{u}_i\right)\right\|_2\\ &= \left\|\sum_{i=1}^{n}\lambda_i\Pi_{\mathcal{P}_1}\left(\boldsymbol{u}_i\right)-\sum_{i=1}^{n}\lambda_i\Pi_{\mathcal{P}_2}\left(\boldsymbol{u}_i\right)\right\|_2 && \text{(Lemma B.8)}\\ &= \left\|\sum_{i=1}^{n}\lambda_i[\Pi_{\mathcal{P}_1}\left(\boldsymbol{u}_i\right)-\Pi_{\mathcal{P}_2}\left(\boldsymbol{u}_i\right)]\right\|_2\\ &= \|\partial\boldsymbol{U}\boldsymbol{\lambda}\|_2\\ &\le \|\partial\boldsymbol{U}\|_2\cdot\|\boldsymbol{\lambda}\|_2\end{aligned}$$

□

Remark. This lemma shows how to bound the projection error of another vector based on the "basis" vectors. (Strictly speaking, $\{\boldsymbol{u}_i\}_{i=1}^n$ are not basis, as they can be dependent. In this case, we can find a $\boldsymbol{\lambda}$ with the smallest norm to get the tightest bound of $\|\Pi_{\mathcal{P}_1}(\boldsymbol{v})-\Pi_{\mathcal{P}_2}(\boldsymbol{v})\|_2$.)

B.2.4 Bounded Trimmed Reconstruction Loss

BOBA stage 1 minimizes the *trimmed reconstruction loss* among clients' gradients. In this subsubsection, we prove that for both BOBA-ES (which use exhaustive searching) and BOBA (which use more efficient alternating optimization) can fit an affine subspace with upper bounded trimmed reconstruction loss in expectation, while this loss is not affected by outer deviation. We first formally define trimmed reconstruction loss in Definition B.10, and then derive upper bounds of the trimmed reconstruction losses for BOBA-ES and BOBA in Lemma B.11 and B.12, respectfully. Finally, we empirically compare their trimmed reconstruction loss.

Definition B.10 (Trimmed reconstruction loss)**.** Given gradients $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n$ and Byzantine tolerance f, the trimmed reconstruction loss of an affine subspace $\mathcal{P}$ is

$$\ell_t(\mathcal{P}) = \min_{\substack{\boldsymbol{r} \in \{0,1\}^n \\ \sum_{i=1}^n r_i = n-f}} \sum_{i=1}^n r_i \left\| \boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i) \right\|_2^2$$

which is the sum of squared distance from $\mathcal{P}$ to its $n - f$ nearest neighbors.

Lemma B.11 (Trimmed reconstruction loss of BOBA-ES)**.** *Let $\hat{\mathcal{P}}$ denote the subspace fitted by BOBA-ES (exhaustive searching) stage 1 and $\ell_t(\hat{\mathcal{P}})$ be its corresponding trimmed reconstruction loss. We have*

$$\ell_t(\hat{\mathcal{P}}) \leq \frac{n-f}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2$$

Meanwhile, if we take expectation at both sides

$$\mathbb{E}\ell_t(\hat{\mathcal{P}}) \leq (n-f)\epsilon^2$$

Proof. BOBA-ES iterates through all subsets of gradients with cardinality $n - f$, and pick the subset with we it fits an affine subspace with smallest trimmed reconstruction loss. For any affine subspace $\mathcal{P}$ fitted by $n - f$ gradients, denote $\mathcal{F}(\mathcal{P})$ as the $n - f$ gradients with which $\mathcal{P}$ is fitted, and $\mathcal{N}(\mathcal{P})$ as the $n - f$ nearest neighbors of $\mathcal{P}$. Also, denote $\mathcal{P}^*$ as the honest subspace. For any $\mathcal{P}'$ fitted by $n - f$ *honest* gradients denoted as $\mathcal{F}(\mathcal{P}')$ (notice that $n - f \leq |\mathcal{H}|$),

$$\begin{aligned}
\ell_t(\hat{\mathcal{P}}) &\leq \ell_t(\mathcal{P}') && \text{(Optimality of BOBA-ES)} \\
&= \sum_{i \in \mathcal{N}(\mathcal{P}')} \|\Pi_{\mathcal{P}'}(\boldsymbol{g}_i) - \boldsymbol{g}_i\|_2^2 \\
&\leq \sum_{i \in \mathcal{F}(\mathcal{P}')} \|\Pi_{\mathcal{P}'}(\boldsymbol{g}_i) - \boldsymbol{g}_i\|_2^2 \\
&\leq \sum_{i \in \mathcal{F}(\mathcal{P}')} \|\Pi_{\mathcal{P}^*}(\boldsymbol{g}_i) - \boldsymbol{g}_i\|_2^2 && \text{(Optimality of SVD)} \\
&\leq \sum_{i \in \mathcal{F}(\mathcal{P}')} \|\Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_i) - \boldsymbol{g}_i\|_2^2 && \text{(Lemma B.6)} \\
&= \sum_{i \in \mathcal{F}(\mathcal{P}')} \|\mathbb{E}\boldsymbol{g}_i - \boldsymbol{g}_i\|_2^2
\end{aligned}$$

Finally, we iterate all subset of honest gradients with cardinality $n - f$. There will be $\binom{|\mathcal{H}|}{n-f}$ subsets, while each honest gradient is chosen for $\binom{|\mathcal{H}|-1}{n-f-1}$ times. Therefore,

$$\binom{|\mathcal{H}|}{n-f} \ell_t(\hat{\mathcal{P}}) \leq \binom{|\mathcal{H}|-1}{n-f-1} \sum_{i \in \mathcal{H}} \|\mathbb{E}\boldsymbol{g}_i - \boldsymbol{g}_i\|_2^2 \quad \Rightarrow \quad \ell_t(\hat{\mathcal{P}}) \leq \frac{n-f}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \|\mathbb{E}\boldsymbol{g}_i - \boldsymbol{g}_i\|_2^2$$

□

Lemma B.12 (Trimmed reconstruction loss of BOBA)**.** *Let $\hat{\mathcal{P}}$ denote the subspace fitted by BOBA (alternating optimization) stage 1 and $\ell_t(\hat{\mathcal{P}})$ be its corresponding trimmed reconstruction loss. We have*

$$\ell_t(\hat{\mathcal{P}}) \leq 2\frac{n-f}{|\mathcal{H}|}\left(\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + \sum_{i\in\mathcal{H}}\sum_{z=1}^{c} p_{iz}\|\mathbb{E}\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_z\|_2^2\right)$$

Meanwhile, if we take expectation at both sides

$$\mathbb{E}\ell_t(\hat{\mathcal{P}}) \leq 2(n-f)(\epsilon^2 + \epsilon_s^2)$$

Proof. We denote $\hat{\mathcal{P}}_0$ as the affine subspace initialized by server gradients $\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_c$. Since the trimmed reconstruction loss is monotone non-increasing during the alternating optimization, we have

$$\ell_t(\hat{\mathcal{P}}) \leq \ell_t(\hat{\mathcal{P}}_0)$$

For each honest client $i \in \mathcal{H}$, its expected gradient can be expressed as a *convex* combination of expected server gradients, i.e.,

$$\mathbb{E}\boldsymbol{g}_i = \sum_{z=1}^{c} p_{iz}\mathbb{E}\boldsymbol{\gamma}_z \qquad \text{(Proposition 3.3)}$$

where $[p_{i1}, \cdots, p_{iz}]^\top$ is the label distribution of client i. We have

$$\begin{aligned}
\left\|\mathbb{E}\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\mathbb{E}\boldsymbol{g}_i)\right\|_2^2 &= \left\|\Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_i) - \Pi_{\hat{\mathcal{P}}_0}(\mathbb{E}\boldsymbol{g}_i)\right\|_2^2 \\
&= \left\|\Pi_{\mathcal{P}^*}\left(\sum_{z=1}^{c} p_{iz}\mathbb{E}\boldsymbol{\gamma}_z\right) - \Pi_{\hat{\mathcal{P}}_0}\left(\sum_{z=1}^{c} p_{iz}\mathbb{E}\boldsymbol{\gamma}_z\right)\right\|_2^2 \\
&= \left\|\sum_{z=1}^{c} p_{iz}\left(\Pi_{\mathcal{P}^*}\left(\mathbb{E}\boldsymbol{\gamma}_z\right) - \Pi_{\hat{\mathcal{P}}_0}\left(\mathbb{E}\boldsymbol{\gamma}_z\right)\right)\right\|_2^2 && \text{(Lemma B.8)} \\
&\leq \sum_{z=1}^{c} p_{iz}\|\Pi_{\mathcal{P}^*}\left(\mathbb{E}\boldsymbol{\gamma}_z\right) - \Pi_{\hat{\mathcal{P}}_0}\left(\mathbb{E}\boldsymbol{\gamma}_z\right)\|_2^2 && \text{(Convexity of } \|\boldsymbol{x}\|_2^2\text{)} \\
&= \sum_{z=1}^{c} p_{iz}\|\mathbb{E}\boldsymbol{\gamma}_z - \Pi_{\hat{\mathcal{P}}_0}\left(\mathbb{E}\boldsymbol{\gamma}_z\right)\|_2^2 \\
&\leq \sum_{z=1}^{c} p_{iz}\|\mathbb{E}\boldsymbol{\gamma}_z - \Pi_{\hat{\mathcal{P}}_0}\left(\boldsymbol{\gamma}_z\right)\|_2^2 && \text{(Lemma B.6)} \\
&= \sum_{z=1}^{c} p_{iz}\|\mathbb{E}\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_z\|_2^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\boldsymbol{g}_i)\|_2^2 &\leq \|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\mathbb{E}\boldsymbol{g}_i)\|_2^2 && \text{(Lemma B.7)} \\
&= \|(\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i) + (\mathbb{E}\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\mathbb{E}\boldsymbol{g}_i))\|_2^2 \\
&\leq 2\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + 2\|\mathbb{E}\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\mathbb{E}\boldsymbol{g}_i)\|_2^2
\end{aligned}$$

Finally, denote $\mathcal{N}(\hat{\mathcal{P}}_0)$ as the $n-f$ nearest neighbors of $\hat{\mathcal{P}}_0$

$$\begin{aligned}
&\ell_t(\hat{\mathcal{P}}) \leq \ell_t(\hat{\mathcal{P}}_0) \\
&\quad= \sum_{i\in\mathcal{N}(\hat{\mathcal{P}}_0)} \|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\boldsymbol{g}_i)\|_2^2 \\
&\quad\leq \frac{n-f}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}_0}(\boldsymbol{g}_i)\|_2^2 \\
&\quad\leq 2\frac{n-f}{|\mathcal{H}|}\left(\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + \sum_{i\in\mathcal{H}}\sum_{z=1}^{c} p_{iz}\|\mathbb{E}\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_z\|_2^2\right)
\end{aligned}$$

□

Remark. When $\epsilon_s = \mathcal{O}(\epsilon)$, $\mathbb{E}\ell_t(\hat{\mathcal{P}}) = \mathcal{O}(n\epsilon^2)$ for both BOBA-ES and BOBA.

Empirical comparison

In Lemma B.11 and B.12, we derive upper bounds of the trimmed reconstruction loss, and show that they have the same order. Additionally, we empirically compare the trimmed reconstruction loss of two algorithms. Specifically, for each type of attack, we employed both BOBA and BOBA-ES in every round and recorded their respective trimmed reconstruction losses. Since BOBA always yield trimmed reconstruction losses greater than or equal to those of BOBA-ES, we plotted the ratio of their losses, i.e. $\frac{\ell_t(\text{BOBA})}{\ell_t(\text{BOBA-ES})}$.

We have organized the results in Figure 5. It is noteworthy that when facing Gauss/IPM/MinMax/MinSum attacks, BOBA can achieve the same trimmed reconstruction loss as BOBA-ES. However, when not subjected to attacks or when facing LIE/Mimic attacks, due to the existence of multiple subspaces that can yield similar trimmed reconstruction loss, BOBA converges to a slightly higher trimmed reconstruction loss compared to BOBA-ES. Nevertheless, their convergence results are highly similar, with the loss ratio seldom exceeding 1.2. This demonstrates that BOBA can yield very similar results to BOBA-ES. It is important to note that in the main text, we also provide a comprehensive comparison of the performance of both algorithms.

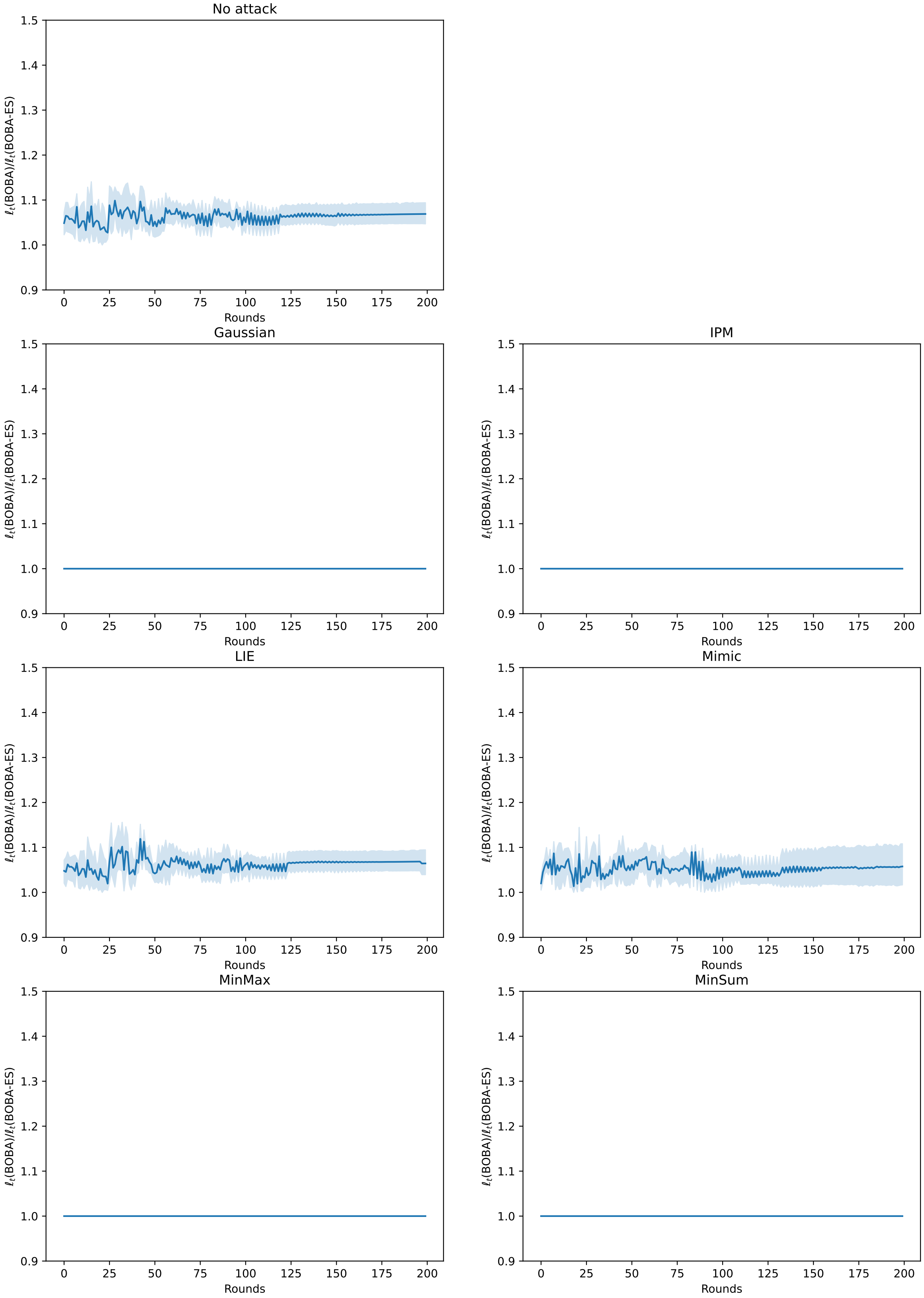

Figure 5: Comparison of trimmed reconstruction loss of BOBA and BOBA-ES

B.2.5 Robustness of BOBA Stage 1

In BOBA stage 1, we fit an affine subspace close to all honest gradients, and project all gradient into this fitted subspace. In this subsubsection, we prove the robustness of stage 1, i.e., honest gradients will only be slightly perturbed in stage 1. Specifically

- Lemma B.13 proves that each honest gradient's projection is close enough to its expectation.
- Lemma B.14 proves that the average of honest gradients' projections is close enough to the expectation of the average of honest gradients.
- Lemma B.15 proves that each server gradient's projection is close enough to its expectation.

Lemma B.13 (Robustness of stage 1)**.** *Let $\hat{\mathcal{P}}$ denote the subspace fitted by BOBA stage 1 and $\ell_t(\hat{\mathcal{P}})$ be its corresponding trimmed reconstruction loss. For any honest gradient $\boldsymbol{g}_h$, $\forall h \in \mathcal{H}$, we have*

$$\left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \mathbb{E}\boldsymbol{g}_h\right\|_2^2 \leq 2\|\boldsymbol{g}_h - \mathbb{E}\boldsymbol{g}_h\|_2^2 + 4\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)$$

Meanwhile, if we take expectation at both sides,

$$\begin{aligned}\mathbb{E}\left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \boldsymbol{g}_h\right\|_2^2 &\leq 2\epsilon^2 + 4\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)\left(\mathbb{E}\ell_t(\hat{\mathcal{P}}) + |\mathcal{H}|\epsilon^2\right) \\ &\leq \begin{cases} \left(2 + 4\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)(n - f + |\mathcal{H}|)\right)\epsilon^2 & \text{(BOBA-ES)} \\ \left(2 + 4\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)(2(n-f) + |\mathcal{H}|)\right)\epsilon^2 + 8\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)(n-f)\epsilon_s^2 & \text{(BOBA)} \end{cases}\end{aligned}$$

Proof. The core of the proof is Lemma B.9. We split the full proof into four steps.

- Step 1: Find $n-2f$ expected gradients $\mathbb{E}\boldsymbol{g}_{s_1}, \mathbb{E}\boldsymbol{g}_{s_2}, \cdots, \mathbb{E}\boldsymbol{g}_{s_{n-2f}}$ that affinely span the honest subspace. Their projections to the fitted subspace and the honest subspace are close.
- Step 2: Express $\mathbb{E}\boldsymbol{g}_h$ as a affine combination of $\mathbb{E}\boldsymbol{g}_{s_1}, \mathbb{E}\boldsymbol{g}_{s_2}, \cdots, \mathbb{E}\boldsymbol{g}_{s_{n-2f}}$ with coefficient $\boldsymbol{\lambda}$. Derive an upper bound for $\|\boldsymbol{\lambda}\|_2$.
- Step 3: Use Lemma B.9 to show that $\mathbb{E}\boldsymbol{g}_h$'s projections to the fitted subspace and the honest subspace are close.
- Step 4: Use triangle inequality to postprocess the inequality.

Step 1. Let $\mathcal{N}(\hat{\mathcal{P}})$ denote the $n-f$ nearest neighbors of $\hat{\mathcal{P}}$. Among these $n-f$ gradients, at least $n-2f$ gradients are honest. We use $\boldsymbol{g}_{s_1}, \boldsymbol{g}_{s_2}, \cdots, \boldsymbol{g}_{s_{n-2f}}$ to denote these $n-2f$ honest gradients and $\mathcal{S} = \{s_1, s_2, \cdots, s_{n-2f}\}$ denote their indices. Since $\mathcal{S} \subset \mathcal{N}(\hat{\mathcal{P}})$, we have

$$\sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \boldsymbol{g}_i\right\|_2^2 \leq \sum_{i\in\mathcal{N}(\hat{\mathcal{P}})} \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \boldsymbol{g}_i\right\|_2^2 = \ell_t(\hat{\mathcal{P}})$$

Let $\mathcal{P}^*$ denote the honest subspace, i.e., the subspace on which expected gradients $\{\mathbb{E}\boldsymbol{g}_i\}_{i\in\mathcal{H}}$ lie. Then,

$$\begin{aligned}
\sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_i) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_i)\right\|_2^2 &= \sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_i) - \mathbb{E}\boldsymbol{g}_i\right\|_2^2 \\
&\le \sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \mathbb{E}\boldsymbol{g}_i\right\|_2^2 && \text{(Lemma B.7)} \\
&\le \sum_{i\in\mathcal{S}} \left(\left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \boldsymbol{g}_i\right\|_2 + \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2\right)^2 \\
&\le \sum_{i\in\mathcal{S}} \left(2\left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \boldsymbol{g}_i\right\|_2^2 + 2\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right) \\
&\le 2\sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \boldsymbol{g}_i\right\|_2^2 + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 \\
&= 2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2
\end{aligned}$$

Define $\partial\boldsymbol{S} = [\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_{s_1}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_{s_1}), \cdots, \Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_{s_{n-2f}}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_{s_{n-2f}})] \in \mathbb{R}^{d\times(n-2f)}$, we have

$$\|\partial\boldsymbol{S}\|_2^2 \le \|\partial\boldsymbol{S}\|_F^2 = \sum_{i\in\mathcal{S}} \left\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_i) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_i)\right\|_2^2 \le 2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2$$

Step 2. By Assumption 5.2(1), $\mathbb{E}\boldsymbol{g}_{s_1}, \mathbb{E}\boldsymbol{g}_{s_2}, \cdots, \mathbb{E}\boldsymbol{g}_{s_{n-2f}}$ can affinely span the honest subspace. Therefore we can express $\mathbb{E}\boldsymbol{g}_h$ as an affine combination of them, i.e., there exists $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_{n-2f}]^\top \in \mathbb{R}^{n-2f}$, s.t.,

$$\mathbb{E}\boldsymbol{g}_h = \sum_{i=1}^{n-2f} \lambda_i \mathbb{E}\boldsymbol{g}_{s_i}, \quad \sum_{i=1}^{n-2f} \lambda_i = 1$$

With bounded singular values (Assumption 5.3(1)), we can find a $\boldsymbol{\lambda}$ with small bounded norm. Although the solution of $\boldsymbol{\lambda}$ is usually not unique (when $n - 2f > c$), we only need one solution with small norm. We first "centralize" gradients. Let $\mathbb{E}\boldsymbol{m}_s = \frac{1}{n-2f}\sum_{i=1}^{n-2f} \mathbb{E}\boldsymbol{g}_{s_i}$, we need to solve the following centralized linear system

$$\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{m}_s = \sum_{i=1}^{n-2f} \theta_i(\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{m}_s) = \boldsymbol{A}_s\boldsymbol{\theta}$$

where $\boldsymbol{A}_s = [\mathbb{E}\boldsymbol{g}_{s_1} - \mathbb{E}\boldsymbol{m}_s, \cdots, \mathbb{E}\boldsymbol{g}_{s_{n-2f}} - \mathbb{E}\boldsymbol{m}_s] \in \mathbb{R}^{d\times(n-2f)}$. One solution of $\boldsymbol{\theta}$ is $\theta = \boldsymbol{A}_s^+(\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{m}_s)$, where $\boldsymbol{A}_s^+$ is the Moore-Penrose inverse of $\boldsymbol{A}_s$. The norm of this solution is bounded by

$$\begin{aligned}
\|\boldsymbol{\theta}\|_2 &= \|\boldsymbol{A}_s^+(\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{m}_s)\|_2 \\
&\le \|\boldsymbol{A}_s^+\|_2 \cdot \|\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{m}_s\|_2 \\
&\le \frac{1}{\sigma} \cdot \|\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{m}_s\|_2 && \text{(Assumption 5.3(1))} \\
&\le \frac{1}{\sigma} \cdot \left\|(\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{\mu}) - \frac{1}{n-2f}\sum_{i=1}^{n-2f}(\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu})\right\|_2 \\
&\le \frac{1}{\sigma} \cdot \left(\|\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{\mu}\|_2 + \frac{1}{n-2f}\sum_{i=1}^{n-2f}\|\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu}\|_2\right) \\
&\le \frac{2\delta}{\sigma} && \text{(Assumption 5.2(2))}
\end{aligned}$$

Each solution of the centralized linear system $\boldsymbol{\theta}$ corresponds to a solution of the original linear system

$$\boldsymbol{\lambda} = \frac{1}{n-2f}\mathbf{1} + \left(\boldsymbol{I} - \frac{1}{n-2f}\mathbf{1}\mathbf{1}^\top\right)\boldsymbol{\theta}$$

Therefore,

$$
\begin{aligned}
\|\boldsymbol{\lambda}\|_2 &= \left\| \frac{1}{n-2f}\mathbf{1} + \left(\boldsymbol{I} - \frac{1}{n-2f}\mathbf{1}\mathbf{1}^\top\right)\boldsymbol{\theta} \right\|_2 \\
&= \sqrt{\left\| \frac{1}{n-2f}\mathbf{1} \right\|_2^2 + \left\| \left(\boldsymbol{I} - \frac{1}{n-2f}\mathbf{1}\mathbf{1}^\top\right)\boldsymbol{\theta} \right\|_2^2} && \text{(Orthogonality)} \\
&\le \sqrt{\left\| \frac{1}{n-2f}\mathbf{1} \right\|_2^2 + \left\| \boldsymbol{I} - \frac{1}{n-2f}\mathbf{1}\mathbf{1}^\top \right\|_2^2 \cdot \|\boldsymbol{\theta}\|_2^2} \\
&\le \sqrt{\left\| \frac{1}{n-2f}\mathbf{1} \right\|_2^2 + \|\boldsymbol{\theta}\|_2^2} \\
&\le \sqrt{\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}}
\end{aligned}
$$

It is also easy to verify that $\mathbf{1}^\top\boldsymbol{\lambda} = 1$.

Step 3. Since then, we construct a $\boldsymbol{\lambda}$ satisfying the condition of Lemma B.9. Therefore

$$
\begin{aligned}
\left\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_h) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_h)\right\|_2^2 &\le \|\partial \boldsymbol{S}\|_2^2 \cdot \|\boldsymbol{\lambda}\|_2^2 && \text{(Lemma B.9)} \\
&\le \left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)\left(2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)
\end{aligned}
$$

Step 4. Finally,

$$
\begin{aligned}
\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \mathbb{E}\boldsymbol{g}_h\|_2^2 &= \|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_h)\|_2^2 \\
&\le 2\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_h)\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_h) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_h)\|_2^2 \\
&\le 2\|\boldsymbol{g}_h - \mathbb{E}\boldsymbol{g}_h\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{g}_h) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{g}_h)\|_2^2 && \text{(Lemma B.7)} \\
&\le 2\|\boldsymbol{g}_h - \mathbb{E}\boldsymbol{g}_h\|_2^2 + 2\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)\left(2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right) \\
&= 2\|\boldsymbol{g}_h - \mathbb{E}\boldsymbol{g}_h\|_2^2 + 4\left(\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)
\end{aligned}
$$

□

Lemma B.13 shows that for each honest gradient, its projection is close to its expectation. Next, we demonstrate in Lemma B.14 that a similar property holds for the average of honest gradients.

Lemma B.14. *Let $\hat{\mathcal{P}}$ denote the subspace fitted by BOBA stage 1 and $\ell_t(\hat{\mathcal{P}})$ be its corresponding trimmed reconstruction loss. Denote $\boldsymbol{\mu} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \boldsymbol{g}_i$ and $\hat{\boldsymbol{\mu}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i)$, we have*

$$
\|\hat{\boldsymbol{\mu}}_{\mathcal{H}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \le 2\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + 4\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)
$$

Meanwhile, if we take expectation at both sides,

$$
\begin{aligned}
\mathbb{E}\,\|\hat{\boldsymbol{\mu}}_{\mathcal{H}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 &\le 2\epsilon^2 + 4\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\mathbb{E}\ell_t(\hat{\mathcal{P}}) + |\mathcal{H}|\epsilon^2\right) \\
&\le \begin{cases} \left(2 + 4\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n - f + |\mathcal{H}|)\right)\epsilon^2 & \text{(BOBA-ES)} \\ \left(2 + 4\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(2(n-f) + |\mathcal{H}|)\right)\epsilon^2 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n-f)\epsilon_s^2 & \text{(BOBA)} \end{cases}
\end{aligned}
$$

Proof. The average of honest gradients can also be seen as a honest gradient. Therefore, we can use the same proof framework as Lemma B.13, while providing tighter bound.

Step 1. Omitted, identical to the step 1 in the proof of B.13.

Step 2. Similar to step 2 in the proof of B.13, while we can derive a tighter bound of $\|\boldsymbol{\lambda}\|_2$

$$\begin{aligned}
\|\boldsymbol{\theta}\|_2 &= \|\boldsymbol{A}_s^+(\mathbb{E}\boldsymbol{\mu} - \mathbb{E}\boldsymbol{m}_s)\|_2 \\
&\leq \|\boldsymbol{A}_s^+\|_2 \cdot \|\mathbb{E}\boldsymbol{\mu} - \mathbb{E}\boldsymbol{m}_s\|_2 \\
&\leq \frac{1}{\sigma} \cdot \|\mathbb{E}\boldsymbol{\mu} - \mathbb{E}\boldsymbol{m}_s\|_2 && \text{(Assumption 5.3(1))} \\
&\leq \frac{1}{\sigma} \cdot \left\| -\frac{1}{n-2f} \sum_{i=1}^{n-2f} (\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu}) \right\|_2 \\
&\leq \frac{1}{\sigma} \cdot \frac{1}{n-2f} \sum_{i=1}^{n-2f} \|\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu}\|_2 \\
&\leq \frac{\delta}{\sigma} && \text{(Assumption 5.2(2))}
\end{aligned}$$

And therefore $\|\boldsymbol{\lambda}\|_2 \leq \sqrt{\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}}$.

Step 3. Also identical to the step 3 in the proof of B.13.

$$\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 \leq \left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)$$

Step 4. Finally,

$$\begin{aligned}
\|\hat{\boldsymbol{\mu}}_{\mathcal{H}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 &= \left\| \frac{1}{|\mathcal{H}|} \sum_{i\in\mathcal{H}} \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) - \mathbb{E}\boldsymbol{\mu} \right\|_2^2 \\
&= \left\| \Pi_{\hat{\mathcal{P}}}\left(\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \boldsymbol{g}_i\right) - \mathbb{E}\boldsymbol{\mu} \right\|_2^2 && \text{(Lemma B.8)} \\
&= \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\mu}) - \mathbb{E}\boldsymbol{\mu}\right\|_2^2 \\
&= \|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 \\
&\leq 2\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\mu}) - \Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 \\
&\leq 2\|\boldsymbol{\mu} - \mathbb{E}\boldsymbol{\mu}\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 && \text{(Lemma B.7)} \\
&= 2\left\| \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} (\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i) \right\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 \\
&\leq 2\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu}) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\mu})\|_2^2 && \text{(Convexity of } \|\boldsymbol{x}\|_2^2\text{)} \\
&= 2\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + 4\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)
\end{aligned}$$

□

Lemma B.15 (Robustness of Stage 1 for server gradients)**.** *Let $\hat{\mathcal{P}}$ denote the subspace fitted by BOBA stage 1 and $\ell_t(\hat{\mathcal{P}})$ be its corresponding trimmed reconstruction loss. For server gradients $\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_c$,*

$$\|\Delta\boldsymbol{\Gamma}\|_2^2 \leq 2\sum_{z=1}^{c} \|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 + 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}} \|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)$$

where $\mathbf{\Delta\Gamma} = [\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_1) - \mathbb{E}\boldsymbol{\gamma}_1, \cdots, \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_c) - \mathbb{E}\boldsymbol{\gamma}_c] \in \mathbb{R}^{d\times c}$*. Meanwhile, if we take expectation at both sides,*

$$\begin{aligned}\mathbb{E}\|\mathbf{\Delta\Gamma}\|_2^2 &\leq 2c\epsilon_s^2 + 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)\left(\mathbb{E}\ell_t(\hat{\mathcal{P}}) + |\mathcal{H}|\epsilon^2\right) \\ &\leq \begin{cases} 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)(n-f+|\mathcal{H}|)\epsilon^2 + \left(2c + 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)(n-f)\right)\epsilon_s^2 & \text{(BOBA-ES)} \\ 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)(2(n-f)+|\mathcal{H}|)\epsilon^2 + \left(2c + 8c\left(\frac{1}{n-2f} + \frac{\delta+\delta_s)^2}{\sigma^2}\right)(n-f)\right)\epsilon_s^2 & \text{(BOBA)} \end{cases}\end{aligned}$$

Proof. Each server gradient can also be seen as a honest gradients. Therefore, we can use the same proof framework as Lemma B.13. We first derive upper bound of $\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2$ for each $z \in \{1, \cdots, c\}$.

Step 1. Omitted, identical to the step 1 in the proof of B.13.

Step 2. Similar to step 2 in the proof of B.13, while the bound of $\|\boldsymbol{\lambda}\|_2$ need to be updated:

$$\begin{aligned}\|\boldsymbol{\theta}\|_2 &= \|\boldsymbol{A}_s^+(\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{m}_s)\|_2 \\ &\leq \|\boldsymbol{A}_s^+\|_2 \cdot \|\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{m}_s\|_2 \\ &\leq \frac{1}{\sigma}\cdot\|\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{m}_s\|_2 && \text{(Assumption 5.3(1))} \\ &\leq \frac{1}{\sigma}\cdot\left\|(\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\mu}) - \frac{1}{n-2f}\sum_{i=1}^{n-2f}(\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu})\right\|_2 \\ &\leq \frac{1}{\sigma}\cdot\left(\|\mathbb{E}\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\mu}\|_2 + \frac{1}{n-2f}\sum_{i=1}^{n-2f}\|\mathbb{E}\boldsymbol{g}_{s_i} - \mathbb{E}\boldsymbol{\mu}\|_2\right) \\ &\leq \frac{\delta+\delta_s}{\sigma} && \text{(Assumption 5.2(2) and (4))}\end{aligned}$$

And therefore $\|\boldsymbol{\lambda}\|_2 \leq \sqrt{\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}}$.

Step 3. Also identical to the step 3 in the proof of B.13.

$$\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\gamma}_z) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\gamma}_z)\|_2^2 \leq \left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)\left(2\ell_t(\hat{\mathcal{P}}) + 2\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)$$

Step 4. Finally,

$$\begin{aligned}\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 &= \|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\gamma}_z)\|_2^2 \\ &\leq 2\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) - \Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\gamma}_z)\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\gamma}_z) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\gamma}_z)\|_2^2 \\ &\leq 2\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 + 2\|\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\gamma}_z) - \Pi_{\mathcal{P}^*}(\mathbb{E}\boldsymbol{\gamma}_z)\|_2^2 && \text{(Lemma B.7)} \\ &= 2\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 + 4\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)\end{aligned}$$

$$\begin{aligned}\|\mathbf{\Delta\Gamma}\|_2^2 &\leq \|\mathbf{\Delta\Gamma}\|_F^2 \\ &= \sum_{z=1}^{c}\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z) - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 \\ &\leq 2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 + 4c\left(\frac{1}{n-2f} + \frac{(\delta+\delta_s)^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)\end{aligned}$$

□

B.2.6 Robustness of BOBA Stage 2

In BOBA stage 2, we estimate the label distribution for each client and discard abnormal clients with strongly negative elements in their label distribution. We use a filtering strategy with a hyper-parameter $p_{\min}$. In this subsubsection, we show that we can find a hyper-parameter $p_{\min}$ such that $|p_{\min}| \geq \|\hat{\boldsymbol{p}}_h - \boldsymbol{p}_h\|_2$, where $\boldsymbol{p}_h$ is the true label distribution and $\hat{\boldsymbol{p}}_h$ is the estimated label distribution, for a honest client $h \in \mathcal{H}$.

Lemma B.16 (Weyl's perturbation bound for singular values)**.** *Let $\boldsymbol{A}$ be a matrix with singular value $\sigma_1 \geq \cdots \geq \sigma_n$ and $\hat{\boldsymbol{A}} = \boldsymbol{A} + \Delta\boldsymbol{A}$ be a perturbation of $\boldsymbol{A}$, with corresponding singular value $\hat{\sigma}_1, \cdots, \hat{\sigma}_n$, we have*

$$|\hat{\sigma}_i - \sigma_i| \leq \|\Delta\boldsymbol{A}\|_2$$

Proof. See proof by Stewart (1990). □

We re-introduce some useful notation. Let

$$\begin{aligned}
\mathbb{E}\boldsymbol{\Gamma} &= [\mathbb{E}\boldsymbol{\gamma}_1, \cdots, \mathbb{E}\boldsymbol{\gamma}_c] \in \mathbb{R}^{d\times c} \\
\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) &= [\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_1), \cdots, \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_c)] \in \mathbb{R}^{d\times c} \\
\Delta\boldsymbol{\Gamma} &= \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma} = [\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_1) - \mathbb{E}\boldsymbol{\gamma}_1, \cdots, \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_c) - \mathbb{E}\boldsymbol{\gamma}_c] \in \mathbb{R}^{d\times c} \\
\Delta\boldsymbol{g}_h &= \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \mathbb{E}\boldsymbol{g}_h \in \mathbb{R}^d
\end{aligned}$$

The true and estimated label distributions of honest client $h \in \mathcal{H}$ are denoted as $\boldsymbol{p}_h, \hat{\boldsymbol{p}}_h$, which follow

$$\mathbb{E}\boldsymbol{g}_h = (\mathbb{E}\boldsymbol{\Gamma})\boldsymbol{p}_h, \quad \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\hat{\boldsymbol{p}}_h$$

Lemma B.17 (Robustness of stage 2)**.** *For any honest gradient $\boldsymbol{g}_h$, we have*

$$\|\Delta\boldsymbol{p}_h\|_2 \leq \frac{1}{\sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2} \cdot \left[\|\Delta\boldsymbol{g}_h\|_2 + \sqrt{2}\|\Delta\boldsymbol{\Gamma}\|_2\right]$$

where $\Delta\boldsymbol{p}_h = \hat{\boldsymbol{p}}_h - \boldsymbol{p}_h$.

Proof. We compare two linear systems:

$$(\mathbb{E}\boldsymbol{\Gamma})\boldsymbol{p}_h = \mathbb{E}\boldsymbol{g}_h, \quad \boldsymbol{1}^\top\boldsymbol{p}_h = 1 \tag{System 1}$$

$$(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}))\hat{\boldsymbol{p}}_h = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h), \quad \boldsymbol{1}^\top\hat{\boldsymbol{p}}_h = 1 \tag{System 2}$$

Different from solving the affine combination at step 3 of Lemma B.13, the solutions here to both linear systems are *unique*. Therefore, we can use any method to express $\Delta\boldsymbol{p}_h = \hat{\boldsymbol{p}}_h - \boldsymbol{p}_h$ and then get a corresponding bound of its 2-norm.

It is also worth noting that the linear system in the algorithm/code is solved in latent space $\mathbb{R}^{c-1}$ instead of original space $\mathbb{R}^d$, which is much more efficient. However in this proof, we consider the problem in $\mathbb{R}^d$ to compare the fitted projection with the ideal projection. We still get the same solution of $\boldsymbol{p}_h$ and $\hat{\boldsymbol{p}}_h$, thus the bound is valid.

We first centralized both systems to remove the affine constraint. Let

$$\begin{aligned}
\boldsymbol{A} &= \mathbb{E}\boldsymbol{\Gamma}\left(\boldsymbol{I} - \frac{1}{c}\boldsymbol{1}\boldsymbol{1}^\top\right) & \hat{\boldsymbol{A}} &= \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\boldsymbol{I} - \frac{1}{c}\boldsymbol{1}\boldsymbol{1}^\top\right) & \Delta\boldsymbol{A} &= \hat{\boldsymbol{A}} - \boldsymbol{A} \\
\boldsymbol{b} &= \mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{\Gamma} \cdot \frac{1}{c}\boldsymbol{1} & \hat{\boldsymbol{b}} &= \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) \cdot \frac{1}{c}\boldsymbol{1} & \Delta\boldsymbol{b} &= \hat{\boldsymbol{b}} - \boldsymbol{b}
\end{aligned}$$

Previously, we have bounded $\|\Delta\boldsymbol{g}_h\|_2$ and $\|\Delta\boldsymbol{\Gamma}\|_2$ in Lemma B.13 and B.15, respectively. We use them to give bounds of $\|\Delta \boldsymbol{A}\|_2$ and $\|\Delta \boldsymbol{b}\|_2$.

$$
\begin{aligned}
\|\Delta \boldsymbol{A}\|_2 &= \left\|\hat{\boldsymbol{A}} - \boldsymbol{A}\right\|_2 \\
&= \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\boldsymbol{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^\top\right) - \mathbb{E}\boldsymbol{\Gamma}\left(\boldsymbol{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^\top\right)\right\|_2 \\
&= \left\|\left(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma}\right)\left(\boldsymbol{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^\top\right)\right\|_2 \\
&\leq \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma}\right\|_2 \cdot \left\|\boldsymbol{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^\top\right\|_2 \\
&\leq \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma}\right\|_2 \\
&= \|\Delta\boldsymbol{\Gamma}\|_2
\end{aligned}
$$

and similarly,

$$
\begin{aligned}
\|\Delta \boldsymbol{b}\|_2 &= \|\hat{\boldsymbol{b}} - \boldsymbol{b}\|_2 \\
&= \left\|\left(\Pi_{\Pi_{\hat{\mathcal{P}}}}(\boldsymbol{g}_h) - \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) \cdot \frac{1}{c}\mathbf{1}\right) - \left(\mathbb{E}\boldsymbol{g}_h - \mathbb{E}\boldsymbol{\Gamma} \cdot \frac{1}{c}\mathbf{1}\right)\right\|_2 \\
&= \left\|\left(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \mathbb{E}\boldsymbol{g}_h\right) - \left(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma}\right) \cdot \frac{1}{c}\mathbf{1}\right\|_2 \\
&\leq \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_h) - \mathbb{E}\boldsymbol{g}_h\right\|_2 + \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma}) - \mathbb{E}\boldsymbol{\Gamma}\right\|_2 \cdot \left\|\frac{1}{c}\mathbf{1}\right\|_2 \\
&= \|\Delta\boldsymbol{g}_h\|_2 + \frac{1}{\sqrt{c}}\|\Delta\boldsymbol{\Gamma}\|_2
\end{aligned}
$$

Then, instead of the original systems, we analyze the centralized systems

$$
\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}}
$$

with

$$
\boldsymbol{x} = \boldsymbol{p}_h - \frac{1}{c}\mathbf{1} \qquad \hat{\boldsymbol{x}} = \hat{\boldsymbol{p}}_h - \frac{1}{c}\mathbf{1} \qquad \Delta\boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x}
$$

By standard perturbation analysis of linear system,

$$
\begin{aligned}
\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x} &= \hat{\boldsymbol{b}} - \boldsymbol{b} \\
\hat{\boldsymbol{A}}(\Delta\boldsymbol{x}) + (\Delta\boldsymbol{A})\boldsymbol{x} &= \Delta\boldsymbol{b} \\
\hat{\boldsymbol{A}}(\Delta\boldsymbol{x}) &= \Delta\boldsymbol{b} - (\Delta\boldsymbol{A})\boldsymbol{x}
\end{aligned}
$$

On the left hand side, $\Delta\boldsymbol{x} \in \mathbb{R}^c$ but the rank of $\hat{\boldsymbol{A}}$ is only $c-1$. Usually, this results in an unbounded norm of $\Delta\boldsymbol{x}$, as it can grow arbitrarily in the direction of the c-th right singular vector of $\hat{\boldsymbol{A}}$. However, the c-th right singular vector of $\hat{\boldsymbol{A}}$ is $\frac{1}{\sqrt{c}}\mathbf{1}$.

$$
\hat{\boldsymbol{A}}\mathbf{1} = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\boldsymbol{I} - \frac{1}{c}\mathbf{1}\mathbf{1}^\top\right)\mathbf{1} = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\mathbf{1} - \mathbf{1}\right) = \mathbf{0}
$$

But $\Delta\boldsymbol{x}$ cannot grow in the direction of $\mathbf{1}$

$$
\mathbf{1}^\top\Delta\boldsymbol{x} = \mathbf{1}^\top\left[\left(\hat{\boldsymbol{p}}_h - \frac{1}{c}\mathbf{1}\right) - \left(\boldsymbol{p}_h - \frac{1}{c}\mathbf{1}\right)\right] = \mathbf{1}^\top\hat{\boldsymbol{p}}_h - \mathbf{1}^\top\boldsymbol{p}_h = 1 - 1 = 0
$$

Thus, we can still bound $\Delta\boldsymbol{x}$ with the $(c-1)$-th singular value of $\hat{\boldsymbol{A}}$ (instead of the smallest singular value, 0).

We have

$$\begin{aligned}\hat{\sigma}_{c-1}\|\Delta\boldsymbol{x}\|_2 &\leq \left\|\hat{\boldsymbol{A}}(\Delta\boldsymbol{x})\right\|_2 \\ &= \|\Delta\boldsymbol{b} - (\Delta\boldsymbol{A})\boldsymbol{x}\|_2 \\ &\leq \|\Delta\boldsymbol{b}\|_2 + \|\Delta\boldsymbol{A}\|_2 \cdot \|\boldsymbol{x}\|_2 \\ \|\Delta\boldsymbol{x}\|_2 &\leq \frac{1}{\hat{\sigma}_{c-1}}\left(\|\Delta\boldsymbol{b}\|_2 + \|\Delta\boldsymbol{A}\|_2 \cdot \|\boldsymbol{x}\|_2\right)\end{aligned}$$

$\|\Delta\boldsymbol{A}\|_2$ and $\|\Delta\boldsymbol{b}\|_2$ are already bounded, we still need to bound $\frac{1}{\hat{\sigma}_{c-1}}$ and $\|\boldsymbol{x}\|_2$.

$\hat{\sigma}_{c-1}$ is the $(c-1)$-th singular value of $\hat{\boldsymbol{A}}$, and is perturbed from σ_{c-1}, the $(c-1)$-th singular value of $\boldsymbol{A}$. By Assumption 5.3 and Weyl's perturbation bound for singular value (Lemma B.16)

$$\begin{aligned}\hat{\sigma}_{c-1} &\geq \sigma_{c-1} - |\hat{\sigma}_{c-1} - \sigma_{c-1}| \\ &\geq \sigma_{c-1} - \|\Delta\boldsymbol{A}\|_2 && \text{(Lemma B.16)} \\ &\geq \sigma_s - \|\Delta\boldsymbol{A}\|_2 && \text{(Assumption 5.3)} \\ &\geq \sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2\end{aligned}$$

The 2-norm of $\boldsymbol{x}$ can also be bounded,

$$\begin{aligned}\|\boldsymbol{x}\|_2 &= \left\|\boldsymbol{p}_h - \frac{1}{c}\boldsymbol{1}\right\|_2 \\ &= \sqrt{\left(\boldsymbol{p}_h - \frac{1}{c}\boldsymbol{1}\right)^\top \left(\boldsymbol{p}_h - \frac{1}{c}\boldsymbol{1}\right)} \\ &= \sqrt{\boldsymbol{p}_h^\top \boldsymbol{p}_h - \frac{1}{c}} \\ &\leq \sqrt{1 - \frac{1}{c}}\end{aligned}$$

Putting everything together, we have

$$\begin{aligned}\|\Delta\boldsymbol{p}_h\|_2 &= \|\Delta\boldsymbol{x}\|_2 \\ &\leq \frac{1}{\hat{\sigma}_{c-1}} \cdot \left(\|\Delta\boldsymbol{b}\|_2 + \|\Delta\boldsymbol{A}\|_2 \cdot \|\boldsymbol{x}\|_2\right) \\ &\leq \frac{1}{\sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2} \cdot \left(\|\Delta\boldsymbol{g}_h\|_2 + \frac{1}{\sqrt{c}}\|\Delta\boldsymbol{\Gamma}\|_2 + \sqrt{1 - \frac{1}{c}}\|\Delta\boldsymbol{\Gamma}\|_2\right) \\ &= \frac{1}{\sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2} \cdot \left[\|\Delta\boldsymbol{g}_h\|_2 + \left(\sqrt{\frac{1}{c}} + \sqrt{1 - \frac{1}{c}}\right)\|\Delta\boldsymbol{\Gamma}\|_2\right] \\ &\leq \frac{1}{\sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2} \cdot \left[\|\Delta\boldsymbol{g}_h\|_2 + \sqrt{2}\|\Delta\boldsymbol{\Gamma}\|_2\right]\end{aligned}$$

when $\sigma_s - \|\Delta\boldsymbol{\Gamma}\|_2 > 0$.

□

Remark. We consider the case where $\|\Delta\boldsymbol{g}_h\|_2 = \mathcal{O}(\epsilon)$ and $\|\Delta\boldsymbol{\Gamma}\|_2 = \mathcal{O}(\sqrt{c}\epsilon)$ (see remarks of Lemma B.13 and B.15). When the outer deviation dominates the inner deviation, $\sigma_s \gg \|\Delta\boldsymbol{\Gamma}\|_2$, thus $\|\Delta\boldsymbol{p}_h\|_2 = \mathcal{O}(\frac{\sqrt{c}\epsilon}{\sigma_s})$. This means that we can set a small $|p_{\min}|$ and still preserve all honest gradients.

B.2.7 Robustness of BOBA

So far, we have already proved two things.

- In stage 1, all honest gradients are only slightly perturbed.
- In stage 2, all honest gradients are preserved, given

In this subsubsection, we wrap up the theoretical results and provide the unbiasedness and robustness of BOBA. Specifically,

- When there are no Byzantine attack, BOBA is *unbiased.*
- When there are Byzantine attack, BOBA is *robust* and has gradient estimation error of optimal order matching with the theoretical lower bound.

Theorem 5.5. *Let $\hat{\boldsymbol{\mu}}$ denote the aggregation result of BOBA. We have,*

$$\begin{aligned}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq 4\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}}) + \sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i - \mathbb{E}\boldsymbol{g}_i\|_2^2\right)\\ + \beta^2 8(1+c|p_{\min}|)^2\left(2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z - \mathbb{E}\boldsymbol{\gamma}_z\|_2^2 + 2\delta_s^2\right)\end{aligned}$$

Then we take expectation on both sides,

$$\begin{aligned}\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 &\leq 4\epsilon^2 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\mathbb{E}\ell_t(\hat{\mathcal{P}}) + |\mathcal{H}|\epsilon^2\right) + 8\beta^2(1+c|p_{\min}|)^2(2c\epsilon_s^2 + 2\delta_s^2)\\ &\leq \begin{cases}\left(4 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n-f+|\mathcal{H}|)\right)\epsilon^2 + 16c(1+c|p_{\min}|)^2\beta^2\epsilon_s^2 & \\ \qquad +16(1+c|p_{\min}|)^2\beta^2\delta_s^2 & \text{(BOBA-ES)}\\ \left(4 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(2(n-f)+|\mathcal{H}|)\right)\epsilon^2 & \\ \qquad + \left(16\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n-f) + 16c(1+c|p_{\min}|)^2\beta^2\right)\epsilon_s^2 & \\ \qquad +16(1+c|p_{\min}|)^2\beta^2\delta_s^2 & \text{(BOBA)}\end{cases}\end{aligned}$$

Proof. When some Byzantine clients are accepted, they can affect the aggregation result via biasing the average of estimated label distribution. Without loss of generality, we consider the worst case: all Byzantine gradients are accepted by BOBA stage 2.

We first decompose the gradient estimation error into two parts. Define $\hat{\boldsymbol{\mu}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i)$, we have

$$\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq 2\|\hat{\boldsymbol{\mu}}_{\mathcal{H}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 + 2\|\hat{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}_{\mathcal{H}}\|_2^2$$

The first term is already bounded in Lemma B.14. We further bound the second term. Notice that,

$$\hat{\boldsymbol{\mu}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\hat{\boldsymbol{p}}_i = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\hat{\boldsymbol{p}}_i\right)$$

where $\hat{\boldsymbol{p}}_i$ is the estimated label distribution of client i. Similarly,

$$\hat{\boldsymbol{\mu}} = \Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{p}}_i\right)$$

We define

$$\hat{\boldsymbol{p}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\hat{\boldsymbol{p}}_i, \quad \hat{\boldsymbol{p}}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\hat{\boldsymbol{p}}_i, \quad \hat{\boldsymbol{p}} = \frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{p}}_i$$

Then,

$$\begin{aligned}
\|\hat{\boldsymbol{\mu}}-\hat{\boldsymbol{\mu}}_{\mathcal{H}}\|_2^2 &= \left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})\left(\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\right)\right\|_2^2 \\
&= \left\|\left(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\Gamma})-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\mathbf{1}^\top\right)\left(\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\right)\right\|_2^2 \\
&= \left\|\sum_{z=1}^{c}(\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}})_z\cdot\left(\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\right)\right\|_2^2 \\
&\le \left(\sum_{z=1}^{c}|(\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}})_z|\cdot\left\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\right\|_2\right)^2 \\
&\le \left(\|\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\|_1\cdot\left(\max_z\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\|_2\right)\right)^2 \\
&= \|\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\|_1^2\cdot\left(\max_z\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\|_2^2\right)
\end{aligned}$$

We first derive a bound for $\|\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\|_1$. In BOBA stage 2, a gradient will be accepted if and only if its estimated label distribution lies in the $(c-1)$-simplex of

$$\hat{\boldsymbol{p}}_i \in \{\boldsymbol{q} : \boldsymbol{q} \ge p_{\min}\mathbf{1}, \mathbf{1}^\top\boldsymbol{q}=1\}$$

Since both $\hat{\boldsymbol{p}}_{\mathcal{H}}$ and $\hat{\boldsymbol{p}}_{\mathcal{B}}$ are averages of some $\hat{\boldsymbol{p}}_i$ that lie inside the simplex above, we have

$$\hat{\boldsymbol{p}}_{\mathcal{H}}, \hat{\boldsymbol{p}}_{\mathcal{B}} \in \{\boldsymbol{q} : \boldsymbol{q} \ge p_{\min}\mathbf{1}, \mathbf{1}^\top\boldsymbol{q}=1\}$$

Therefore, denote $\beta=\frac{|\mathcal{B}|}{n}$, we have

$$\begin{aligned}
\|\hat{\boldsymbol{p}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\|_1 &= \left\|\frac{|\mathcal{H}|}{n}\hat{\boldsymbol{p}}_{\mathcal{H}}+\frac{|\mathcal{B}|}{n}\hat{\boldsymbol{p}}_{\mathcal{B}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\right\|_1 \\
&= \frac{|\mathcal{B}|}{n}\|\hat{\boldsymbol{p}}_{\mathcal{B}}-\hat{\boldsymbol{p}}_{\mathcal{H}}\|_1 \\
&\le \frac{|\mathcal{B}|}{n}\left(\|\hat{\boldsymbol{p}}_{\mathcal{B}}-p_{\min}\mathbf{1}\|_1+\|\hat{\boldsymbol{p}}_{\mathcal{H}}-p_{\min}\mathbf{1}\|_1\right) \\
&= \beta 2(1+c|p_{\min}|)
\end{aligned}$$

Then we derive a bound for $\max_z\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\|_2^2$. For each server gradient $\boldsymbol{\gamma}_z$,

$$\begin{aligned}
\max_z\|\Pi_{\hat{\mathcal{P}}}(\boldsymbol{\gamma}_z)-\Pi_{\hat{\mathcal{P}}}(\mathbb{E}\boldsymbol{\mu})\|_2^2 &\le \max_z\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\mu}\|_2^2 && \text{(Lemma B.7)} \\
&\le \max_z\left(2\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\gamma}_z\|_2^2+2\|\mathbb{E}\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\mu}\|_2^2\right) \\
&\le 2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\gamma}_z\|_2^2+\max_z 2\|\mathbb{E}\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\mu}\|_2^2 \\
&\le 2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\gamma}_z\|_2^2+2\delta_s^2 && \text{(Assumption 5.2)}
\end{aligned}$$

Therefore,

$$\|\hat{\boldsymbol{\mu}}-\hat{\boldsymbol{\mu}}_{\mathcal{H}}\|_2^2 \le \beta^2 4(1+c|p_{\min}|)^2\left(2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\gamma}_z\|_2^2+2\delta_s^2\right)$$

Put all together

$$\begin{aligned}
\|\hat{\boldsymbol{\mu}}-\mathbb{E}\boldsymbol{\mu}\|_2^2 \le{}& 4\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i-\mathbb{E}\boldsymbol{g}_i\|_2^2+8\left(\frac{1}{n-2f}+\frac{\delta^2}{\sigma^2}\right)\left(\ell_t(\hat{\mathcal{P}})+\sum_{i\in\mathcal{H}}\|\boldsymbol{g}_i-\mathbb{E}\boldsymbol{g}_i\|_2^2\right) \\
&+\beta^2 8(1+c|p_{\min}|)^2\left(2\sum_{z=1}^{c}\|\boldsymbol{\gamma}_z-\mathbb{E}\boldsymbol{\gamma}_z\|_2^2+2\delta_s^2\right)
\end{aligned}$$

Then we take expectation on both sides,

$$\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq 4\epsilon^2 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)\left(\mathbb{E}\ell_t(\hat{\mathcal{P}}) + |\mathcal{H}|\epsilon^2\right) + 8\beta^2(1+c|p_{\min}|)^2(2c\epsilon_s^2 + 2\delta_s^2)$$

$$\leq \begin{cases} \left(4 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n - f + |\mathcal{H}|)\right)\epsilon^2 + 16c(1+c|p_{\min}|)^2\beta^2\epsilon_s^2 \\ \qquad +16(1+c|p_{\min}|)^2\beta^2\delta_s^2 & \text{(BOBA-ES)} \\ \left(4 + 8\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(2(n-f) + |\mathcal{H}|)\right)\epsilon^2 \\ \qquad + \left(16\left(\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2}\right)(n-f) + 16c(1+c|p_{\min}|)^2\beta^2\right)\epsilon_s^2 \\ \qquad +16(1+c|p_{\min}|)^2\beta^2\delta_s^2 & \text{(BOBA)} \end{cases}$$

□

Remark. We analyze the order of the gradient estimation error. When the outer variation increases t times, i.e., $\mathbb{E}\boldsymbol{g}_i \leftarrow \mathbb{E}t\boldsymbol{g}_i$, both δ and σ increase t times. When all clients are duplication, i.e., $\boldsymbol{G} \leftarrow [\boldsymbol{G}, \boldsymbol{G}]$, and $f \leftarrow 2f$, we have that δ^2 does not change but σ^2 is doubled. Thus generally we have $\frac{\delta^2}{\sigma^2} \propto \frac{1}{n}$. When $\epsilon_s = \mathcal{O}(\epsilon), \delta_s = \mathcal{O}(\delta)$, $c = \mathcal{O}(1)$, $\frac{1}{n-2f} = \mathcal{O}(\frac{1}{n})$, $|\mathcal{H}| = \mathcal{O}(n)$, and $|p_{\min}| = \mathcal{O}(1)$, we have $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathcal{O}(\epsilon^2 + \beta^2\delta^2)$. We can conclude that

$$\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathcal{O}(\epsilon^2 + \beta^2\delta^2)$$

Especially, when $\beta = 0$, we have

$$\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \mathcal{O}(\epsilon^2)$$

Comparison to Bucketing Karimireddy et al. (2022) also have a similar claim in their Theorem II. However, their theorem heavily relies on the assumption that AGR is aware of β, the real fraction of Byzantine clients, as defined in their Definition A. (Their paper use δ.) However, in practical FL systems, the fraction of Byzantine clients can be dynamic, and the AGR usually do not have precise knowledge of it. On the contrary, our BOBA algorithm does not require exact estimation of β; it only needs to satisfy Assumption 5.3 and the condition $f \geq |\mathcal{B}|$. We also show in Appendix C.4 that BOBA has consistent performance under a wide range of f and $|\mathcal{B}|$.

B.3 Lower Bounds of Gradient Estimation Error

B.3.1 Lower Bounds of Gradient Estimation Error for Any AGR

In the IID setting, as the inner variation approaches zero (i.e., $\epsilon \to 0$), the gradient estimation error of robust AGR typically also tends to zero (i.e., $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \to 0$). Consider the extreme case where each honest client uploads the same vector. In this scenario, robust AGR only needs to select the mode, i.e., the vector with the highest frequency from the collected vectors. This implies that the aggregation result is entirely immune to the influence of Byzantine clients.

However, this intuition does not hold in non-IID settings, including cases with label skewness. On the contrary, for any AGR, as long as it remains unaware of the identity of Byzantine clients (i.e., which clients are honest and which are Byzantine), the best-case gradient estimation error can only be guaranteed to be $\mathcal{O}(\beta^2\delta^2)$, rather than approaching zero, even when ϵ is zero. We rigorously state the above proposition as Proposition 5.4.

Proposition 5.4 (Lower bound of gradient estimation error for any AGR)**.** *Given any AGR, we can find $|\mathcal{H}|$ honest gradients and $|\mathcal{B}|$ Byzantine gradients, such that $\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \geq \Omega(\beta^2\delta^2)$.*

Proof. W.l.o.g., we assume $n = |\mathcal{H}| + |\mathcal{B}|$ is even. We consider the following two sets of gradients, both with $|\mathcal{H}|$ honest clients, $|\mathcal{B}|$ Byzantine clients, zero inner variation, and outer variation bounded by δ

Gradient set 1:

$$\boldsymbol{g}_i = \begin{cases} +\frac{|\mathcal{H}|}{n}\delta, & i = 1, \cdots, \frac{n}{2} \\ -\frac{|\mathcal{H}|}{n}\delta, & i = \frac{n}{2}+1, \cdots, n \end{cases}, \quad \mathcal{H} = 1, \cdots, |\mathcal{H}|, \quad \mathcal{B} = |\mathcal{H}|+1, \cdots, n$$

Gradient set 2:

$$\boldsymbol{g}_i = \begin{cases} +\frac{|\mathcal{H}|}{n}\delta, & i = 1, \cdots, \frac{n}{2} \\ -\frac{|\mathcal{H}|}{n}\delta, & i = \frac{n}{2}+1, \cdots, n \end{cases}, \quad \mathcal{H} = |\mathcal{B}|+1, \cdots, n, \quad \mathcal{B} = 1, \cdots, |\mathcal{B}|$$

For the gradient set 1,

$$\mathbb{E}\boldsymbol{\mu}^{(1)} = \frac{1}{|\mathcal{H}|}\left(\frac{n}{2}\left(\frac{|\mathcal{H}|}{n}\delta\right) + \left(|\mathcal{H}| - \frac{n}{2}\right)\left(-\frac{|\mathcal{H}|}{n}\delta\right)\right) = \frac{|\mathcal{B}|}{n}\delta$$

while for the gradient set 2, $\mathbb{E}\boldsymbol{\mu}^{(2)} = -\frac{|\mathcal{B}|}{n}\delta$.

Notice that the two gradient sets have the *same gradient values*; the only difference is the identity of Byzantine clients. Since the input is identical, any AGR will give identical aggregation result for both gradient sets. Thus,

$$\begin{aligned} \max\{\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(1)}\|_2, \|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(2)}\|_2\} &\geq \frac{1}{2}\left(\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(1)}\|_2, \|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(2)}\|_2\right) \\ &\geq \frac{1}{2}\|\mathbb{E}\boldsymbol{\mu}^{(1)} - \mathbb{E}\boldsymbol{\mu}^{(2)}\|_2 \\ &= \frac{|\mathcal{B}|}{n}\delta \\ &= \beta\delta \end{aligned}$$

equivalently,

$$\max\{\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(1)}\|_2^2, \|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}^{(2)}\|_2^2\} \geq \beta^2\delta^2$$

which means that there exists one gradient set among set 1 and 2, such that the gradient estimation error is at least $\beta^2\delta^2$. □

Remark. Notice that this result is *not* in contradiction with Theorem III in Karimireddy et al. (2022). Theorem III in Karimireddy et al. (2022) gives an lower bound of $\Omega(\delta\zeta^2)$, where δ represents the fraction of Byzantines (equivalent to our β) and ζ represents the expected norm of outer variation (similar to our δ). The discrepancy arises from a slight difference in the definition of outer variation. We define δ as the maximum norm of outer variation, while Karimireddy et al. (2022) define ζ as the expected norm of outer variation.

B.3.2 Lower Bounds of Gradient Estimation Error for Krum, CooMed and GeoMed

In this subsubsection, we prove that the gradient estimation error for Krum (Blanchard et al., 2017), CooMed (Yin et al., 2018) and GeoMed (Chen et al., 2017) cannot be better than $\mathcal{O}(\epsilon^2 + \delta^2)$. To prove this, we construct example where the gradient estimation error for the above three AGRs are all $\Omega(\epsilon^2 + \delta^2)$, even when $\beta = 0$ (no attacks).

We construct a simple 3-client setting:

$$\begin{aligned}
\boldsymbol{g}_1 &= \frac{\delta}{2} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \epsilon(2Z_1 - 1) \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{where } Z_1 \sim \text{Bernoulli}(0.5) \\
\boldsymbol{g}_2 &= \frac{\delta}{2} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \epsilon \cdot \frac{1}{\sqrt{2}}(2Z_2 - 1) \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{where } Z_2 \sim \text{Bernoulli}(0.5) \\
\boldsymbol{g}_3 &= -\delta \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\end{aligned}$$

Their expectations:

$$\mathbb{E}\boldsymbol{g}_1 = \mathbb{E}\boldsymbol{g}_2 = \frac{\delta}{2} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbb{E}\boldsymbol{g}_3 = -\delta \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbb{E}\boldsymbol{\mu} = \frac{\mathbb{E}\boldsymbol{g}_1 + \mathbb{E}\boldsymbol{g}_2 + \mathbb{E}\boldsymbol{g}_3}{3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Moreover, we consider $\delta > 2\epsilon$. We can easily verify the bounded inner/outer variations.

Krum No matter how Z_1, Z_2 are chosen, we always have $\|\boldsymbol{g}_1 - \boldsymbol{g}_2\|_2 < \|\boldsymbol{g}_1 - \boldsymbol{g}_3\|_2$ and $\|\boldsymbol{g}_1 - \boldsymbol{g}_2\|_2 < \|\boldsymbol{g}_2 - \boldsymbol{g}_3\|_2$. Therefore, Krum will always choose from $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. In both case, $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 = \epsilon^2 + \frac{\delta^2}{4}$.

CooMed

$$\hat{\boldsymbol{\mu}} = \begin{cases} \left[-\frac{\delta - 2\epsilon}{2\sqrt{2}}, \frac{\delta - 2\epsilon}{2\sqrt{2}}\right]^\top, & (Z_1, Z_2) \in \{(1,0),(0,1)\} \\ \left[-\frac{\delta - 2\epsilon}{2\sqrt{2}}, \frac{\delta + 2\epsilon}{2\sqrt{2}}\right]^\top, & (Z_1, Z_2) = (1,1) \\ \left[-\frac{\delta + 2\epsilon}{2\sqrt{2}}, \frac{\delta - 2\epsilon}{2\sqrt{2}}\right]^\top, & (Z_1, Z_2) = (0,0) \end{cases}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 &= \frac{1}{2} \cdot 2\left(\frac{\delta - 2\epsilon}{2\sqrt{2}}\right)^2 + \frac{1}{2} \cdot \left[\left(\frac{\delta + 2\epsilon}{2\sqrt{2}}\right)^2 + \left(\frac{\delta - 2\epsilon}{2\sqrt{2}}\right)^2\right] \\
&= \frac{(\delta - 2\epsilon)^2}{8} + \frac{\delta^2}{8} + \frac{\epsilon^2}{2} \\
&> \frac{\delta^2}{8} + \frac{\epsilon^2}{2}
\end{aligned}$$

GeoMed

$$\hat{\boldsymbol{\mu}} = \begin{cases} -(\frac{\delta}{2} - \frac{\epsilon}{\sqrt{3}}) \cdot \frac{1}{\sqrt{2}}[-1, 1]^\top, & (Z_1, Z_2) \in \{(1,0),(0,1)\} \\ \left[-\frac{\delta - 2\epsilon}{2\sqrt{2}}, \frac{\delta + 2\epsilon}{2\sqrt{2}}\right]^\top, & (Z_1, Z_2) = (1,1) \\ \left[-\frac{\delta + 2\epsilon}{2\sqrt{2}}, \frac{\delta - 2\epsilon}{2\sqrt{2}}\right]^\top, & (Z_1, Z_2) = (0,0) \end{cases}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 &= \frac{1}{2} \cdot \left(\frac{\delta}{2} - \frac{\epsilon}{\sqrt{3}}\right)^2 + \frac{1}{2} \cdot \left[\left(\frac{\delta + 2\epsilon}{2\sqrt{2}}\right)^2 + \left(\frac{\delta - 2\epsilon}{2\sqrt{2}}\right)^2\right] \\
&= \frac{1}{2} \cdot \left(\frac{\delta}{2} - \frac{\epsilon}{\sqrt{3}}\right)^2 + \frac{\delta^2}{8} + \frac{\epsilon^2}{2} \\
&> \frac{\delta^2}{8} + \frac{\epsilon^2}{2}
\end{aligned}$$

B.3.3 Impossible Unbiasedness and Robustness without Server Data

As mentioned in Subsection 3.2, AGR in label skewness faces two challenges: selection bias and increased vulnerability. In other words, we aim for AGR to possess unbiasedness and robustness, which are rigorously defined in Definition B.18 and B.19,

Definition B.18 (Unbiasedness)**.** An AGR is unbiased if for any $\boldsymbol{w}_G$, $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \to 0$ when $\epsilon \to 0$ and $\beta = 0$ (i.e., no attacks).

Definition B.19 (Robustness)**.** An AGR is robust if there exist $\Delta > 0$ such that for any $\boldsymbol{w}_G$, $\|\hat{\boldsymbol{\mu}} - \mathbb{E}\boldsymbol{\mu}\|_2^2 \leq \Delta^2$.

Proposition B.20 (Trade-Off Between Unbiasedness and Robustness)**.** *For any AGR, if it can only utilize n gradients without relying on any other information, then it is impossible for it to be both unbiased and robust.*

Proof. We consider the following machine learning task:

$$\min_{b\in\mathbb{R}} \mathcal{L} = \mathbb{E}_{(x,y)}\ell(x,y|w), \quad \text{where } \ell(x,y|w) = [y-(x-w)]^2$$

whose gradient w.r.t. w is $\frac{\partial \ell}{\partial w} = 2(w-(x-y))$. We let $x, w \in \mathbb{R}$ and $y \in \{-1, +1\}$. We start with $w_G^{(0)} = 0$.

We consider the following two sets of gradients.

Gradient set 1:

- Client 1 and 2 are honest, with data $(x, y) = (0.5, 1)$
- Client 3 and 4 are honest, with data $(x, y) = (-0.5, -1)$
- Client 5 is Byzantine

This results in the following gradients:

$$g_1 = g_2 = +1, \quad g_3 = g_4 = -1, \quad g_5 = k, \quad \mathcal{H} = \{1,2,3,4\}, \quad \mathcal{B} = \{5\}$$

Gradient set 2:

- Client 5 is honest with $(x, y) = (1 - \frac{k}{2}, 1)$
- Client 3 and 4 are honest, with data $(x, y) = (-0.5, -1)$
- Client 1 and 2 are honest, with $\frac{1}{k+1}$ of their data as $(x, y) = (1 - \frac{k}{2}, 1)$ amd $\frac{k}{k+1}$ of their data as $(x, y) = (-0.5, -1)$

This results in the following gradients:

$$g_1 = g_2 = +1, \quad g_3 = g_4 = -1, \quad g_5 = k, \quad \mathcal{H} = \{1,2,3,4,5\}, \quad \mathcal{B} = \emptyset$$

For gradient set 1, we have inner variation upper bound $\epsilon = 0$, outer variation bound $\delta = 1$; for gradient set 2, we have inner variation upper bound $\epsilon = 0$, outer variation bound $\delta = \frac{4}{5}k$.

Notice that the two gradient sets have the *same gradient values*; the only difference is the identity of Byzantine clients. Therefore, any AGR only utilizing n gradients will give identical aggregation results for two gradient sets. To achieve unbiasedness in gradient set 2, for all $k > 1$, the aggregation result must be $\hat{\mu}^{(1)} = \frac{1}{5}k$. Its aggregation result on gradient set 1 will also be $\hat{\mu}^{(2)} = \frac{1}{5}k$. Let $k \to \infty$, then the gradient estimation error on gradient set 1 will be unbounded, which violates robustness. □

Remark. This unbiasedness-robustness trade-off can be circumvented by using additional server data. Consider that the two server gradients are $\gamma_1 = +1.5, \gamma_2 = -1.5$, then for any $k \gg 1.5$, the AGR can guarantee that it must be a Byzantine gradient, i.e., gradient set 2 is not valid.

B.4 Computation Complexity of BOBA

Algorithm 1 BOBA Framework

```
Input: $\boldsymbol{G} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n]$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_c]$, $n, f, c, p_{\min}$
Output: Aggregation result $\hat{\boldsymbol{\mu}}$
1: Initialize subspace $\hat{\mathcal{P}}$: $\boldsymbol{m}, \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \text{TrSVD}_{c-1}(\boldsymbol{\Gamma})$
2: while not converge do
3:     Update $\boldsymbol{r}$: $\boldsymbol{G}_{[n-f]} = \{n - f$ gradients in $\boldsymbol{G}$ with smallest $\|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i)\|_2\}$ where $\Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i) = \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m}) + \boldsymbol{m}$
4:     Update $\hat{\mathcal{P}}$: $\boldsymbol{m}, \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \text{TrSVD}_{c-1}(\boldsymbol{G}_{[n-f]})$
5: Encode: $\tilde{\boldsymbol{g}}_i = \boldsymbol{U}^\top(\boldsymbol{g}_i - \boldsymbol{m}), \forall i$; $\tilde{\boldsymbol{\Gamma}} = \boldsymbol{U}^\top(\boldsymbol{\Gamma} - \boldsymbol{m}\boldsymbol{1}^\top)$
6: Estimate: $\hat{\boldsymbol{p}}_i = \begin{bmatrix} \tilde{\boldsymbol{\Gamma}} \\ \boldsymbol{1}^\top \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\boldsymbol{g}}_i \\ 1 \end{bmatrix}, \forall i$
7: Filter: $\boldsymbol{a} = \mathcal{A}(\{\hat{\boldsymbol{p}}_i\}_{i=1}^n)$
8: Aggregate: $\tilde{\boldsymbol{\mu}} = \sum_{i=1}^n a_i \tilde{\boldsymbol{g}}_i / \sum_{i=1}^n a_i$
9: Decode: $\hat{\boldsymbol{\mu}} = \boldsymbol{U}\tilde{\boldsymbol{g}}_G + \boldsymbol{m}$
```

In this subsection, we provide a detail analysis of the complexity of BOBA (Algorithm 1). We use the results that the complexity of TrSVD is $\mathcal{O}(cnd)$ (Halko et al., 2011).

- Line 1: The complexity is $\mathcal{O}(cnd)$.
- Line 3: The complexity is $\mathcal{O}(cnd + n \log n)$, where $\mathcal{O}(cnd)$ comes from computing $\|\boldsymbol{g}_i - \Pi_{\hat{\mathcal{P}}}(\boldsymbol{g}_i)\|_2$ for n gradients $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n$, and $\mathcal{O}(n \log n)$ comes from sorting all n distances and select the smallest $n - f$.
- Line 4: The complexity is $\mathcal{O}(cnd)$.
- Line 5: The complexity is $\mathcal{O}(cnd)$.
- Line 6: The complexity is $\mathcal{O}(c^3 + c^2 n)$, where $\mathcal{O}(c^3)$ comes from computing the inverse matrix $\begin{bmatrix} \tilde{\boldsymbol{\Gamma}} \\ \boldsymbol{1}^\top \end{bmatrix}^{-1}$ and $\mathcal{O}(c^2 n)$ arises from computing $\hat{\boldsymbol{p}}_i$ for $i = 1, \cdots, n$.
- Line 7: The complexity is $\mathcal{O}(cn + n \log n)$, where $\mathcal{O}(cn)$ comes from computing $\min_z p_{iz}$ for each client i, and $\mathcal{O}(n \log n)$ arises from computing the quantile.
- Line 8: The complexity is $\mathcal{O}(cn)$.
- Line 9: The complexity is $\mathcal{O}(cd)$.

Assuming that $c < n < d$, the overall complexity is dominated by Line 3 and 4, which are conducted by k times. Therefore, the total complexity is $\mathcal{O}(kcnd)$.

C ADDITIONAL EXPERIMENTS

C.1 Experimental Setup

In this part, we provide detailed experimental setup.

Table 6: Experimental settings summary

	MNIST	CIFAR-10	AG-News	AG-News (Ablation Study)
# Training Samples	60,000	50,000	120,000	120,000
# Testing Samples	10,000	10,000	7,600	7,600
# Classes c	10	10	4	4
# Rounds	200	2,000	200	200
Initial LR η_0	0.1	0.2	1.0	1.0
LR Decay $(T_s; T_i; \alpha)$	(100; 10; 0.95)	(1,000; 100; 0.8)	(100; 10; 0.95)	(100; 10; 0.95)
# Honest Clients $\lvert\mathcal{H}\rvert$	100	100	160	16
Real # Byzantine Clients $\lvert\mathcal{B}\rvert$	0 or 15	0 or 15	0 or 54	0 or 2
Declared # Byzantine Clients f	16	16	60	2
# Server Samples Per Class	20	20	30	30

Training setup The setup for FL training is summarized in Table 6. Specifically,

- *Data partition.* We use the pathological data partitioning proposed by McMahan et al. (2017). We first sort data samples based on labels and evenly divided the training set into $n_s \cdot |\mathcal{H}|$ shards. As a result, each shard only contains one class of data [1]. We then assign n_s shards to each honest client, so that most clients have only n_s classes of samples. We let $n_s = 2$ for MNIST, CIFAR-10 and AG-News.

- *Models.* For MNIST (Lecun et al., 1998), we train a 3-layer MLP with hidden layers of width 200. This network is the same as "2NN" in (McMahan et al., 2017). For CIFAR-10 (Krizhevsky and Hinton, 2009), we train a 5-layer CNN model as it in the TensorFlow tutorial [2]. For AG News (Zhang et al., 2015), we train a RNN model containing a uni-directional GRU layer with 32 hidden units followed by a global pooling layer and a linear layer.

- *Learning rate strategy.* We use a constant learning rate at the early stage, and exponential learning rate decay at the end to stabilize the training. In detail, we start with an initial learning rate $\eta = \eta_0$ until the T_s round, and then exponentially decrease it with $\eta \leftarrow \alpha\eta$ every T_i rounds.

Attacks We consider six types of attacks.

- *Gauss* (Blanchard et al., 2017), a non-colluding attack uploading large-scale vectors from Gaussian distribution $\mathcal{N}(0, 200\boldsymbol{I})$.

- *IPM* (Xie et al., 2019a), a colluding attack uploading $\boldsymbol{g}_k = -\gamma \cdot \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \boldsymbol{g}_i$. While Xie et al. (2019a) test $\gamma \in \{-10, 0, 0.1, 10\}$, we choose the strongest $\gamma = 10$, making Average to perform gradient ascent.

- *LIE* (Baruch et al., 2019), a colluding attack uploading vectors within the scope of honest ones to bias the model while avoiding being detected. We set the hyper-parameter according to the original paper, i.e., $z = \phi^{-1}\left((n - \lfloor n/2 + 1 \rfloor)/(n - |\mathcal{B}|)\right)$.

- *Mimic* (Karimireddy et al., 2022), a colluding attack inserting consistent bias by always copying the gradient from a particular client with biased label distribution. This attack is on the

- *MinMax* and *MinSum* (Shejwalkar and Houmansadr, 2021), a colluding attack that maximize the effect of attack while not being detected. We use coordinate-wise standard deviation as the perturbation vector ∇^p, and optimize the magnitude according to Algorithm 1 with $\gamma_{\text{init}} = 10$ and $\tau = 10^{-5}$.

[1] For MNIST, since the dataset is not strictly balanced, the size of each shard is slightly different to ensure that each shard only contains one class of data.

[2] https://www.tensorflow.org/tutorials/images/cnn

Baseline AGRs We consider 15 baseline AGRs

- *Average* (McMahan et al., 2017) simply averages all gradients. It is unbiased but vulnerable to attacks.
- *Server* only uses server data to fit a model. We use it to verify that one cannot train a good model with server data only.
- *CooMed* and *TrMean* (Yin et al., 2018) use coordinate-wise median or trimmed mean as the aggregation. For TrMean we trimmed the largest and smallest f entries.
- *Krum* and *Multi-Krum* (Blanchard et al., 2017) find the one or m gradients that is closest to its k nearest neighbors. We use $k = n - f - 2$ and $m = n - f$, according to the original paper.
- *GeoMed* (Chen et al., 2017; Pillutla et al., 2022) computes the geometric median as the aggregation. We use the implementation in *hdmedians* Python package.
- *SelfRej* and *AvgRej* (Fang et al., 2020) evaluate client gradients with their loss on server data. SelfRej selects $n - f$ clients whose local models $\boldsymbol{w}_i = \boldsymbol{w}_G - \eta \boldsymbol{g}_i$ have smallest loss, while AvgRej selects $n - f$ clients whose gradients can lower the loss of averaged model the most.
- *Zeno* (Xie et al., 2019b) considers both loss and gradient scales, select $n - f$ gradients with small loss and small gradient. We optimize ρ on MNIST, and finally use $\rho = 5 \times 10^{-4}$ for all experiments.
- *FLTrust* (Cao et al., 2021) uses server data to estimate one server gradient, and use each gradient' clipped cosine similarity as the weight to re-weight each gradient and aggregate.
- *ByGARS* (Regatti et al., 2022) optimizes the aggregation weights of client gradients with server data as training set. Different from the original implementation, we optimize the aggregation weight $\boldsymbol{q}$ for each communication round independently. We optimize hyperparameters on MNIST and finally use $k = 3$ and $\alpha = 0.05$.
- *Bucketing* (Karimireddy et al., 2022). We consider bucketing ($s = 2$) with Krum (B-Krum) / MKrum (B-MKrum).
- *RAGE* (Data and Diggavi, 2021). Considering that C is usually unknown to the server, we run the while loop for fixed f iterations, to make sure it successfully mitigate the Gauss attack.

Image corruptions We simulate feature skewness by applying different image corruptions to each client. For each client, we randomly choose one kind of corruption (severity = 3) for its local training dataset. We do not add corruptions to the server data and the testing data, in order to make comparison with the corruption-free setting.

Computation We did our experiments with single NVIDIA Tesla V100 GPU.

AGR running time (RQ2) In the main text, we record the running time for all AGRs. For a fair comparison, we run all AGRs with an Intel Core i9-11900 Processor. Specifically,

- For BOBA and FLTrust, we do not include the time taken to compute the server gradient because this computation can be finished simultaneously with the client-side gradient computations.
- However, for SelfRej, AvgRej, Zeno, and ByGARS, we include the time to perform inference or compute gradients using server data, because these computations must occur after the server receives the gradient from each client.

C.2 Majority-based AGRs with Additional Server Data (RQ1)

Reference-based AGRs, including BOBA, use additional server data for aggregation. However, majority-based AGRs do not require server data. To make a fair comparison, we study whether server data can further enhance baseline majority-based AGRs, especially the strongest ones. Specifically, we enhance the baselines majority-based AGRs with

$$\hat{\boldsymbol{\mu}} = (1-\lambda)\mathrm{Agg}(\{\boldsymbol{g}_i\}_{i=1}^n) + \lambda\left(\frac{1}{c}\sum_{z=1}^{c}\boldsymbol{\gamma}_z\right)$$

where $\mathrm{Agg}(\{\boldsymbol{g}_i\}_{i=1}^n)$ is the aggregation given by baseline majority-based AGRs, $\frac{1}{c}\sum_{z=1}^{c}\boldsymbol{\gamma}_z$ is the averaged server gradient, and $\lambda \in [0,1]$ is the hyperparameter for convex combination. The underlying intuition here is that the AGR's output has a smaller variance (due to computing gradients using more data from clients), while the server gradient has a smaller bias (as it is not affected by selection bias). By combining these two outputs, a potentially better bias-variance tradeoff can be achieved, leading to improved aggregation results.

We test $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$. Notice that $\lambda = 0$ refers to vanilla baseline AGRs without server data, and $\lambda = 1$ refers to training with server data only. We test these enhanced AGRs with MNIST dataset.

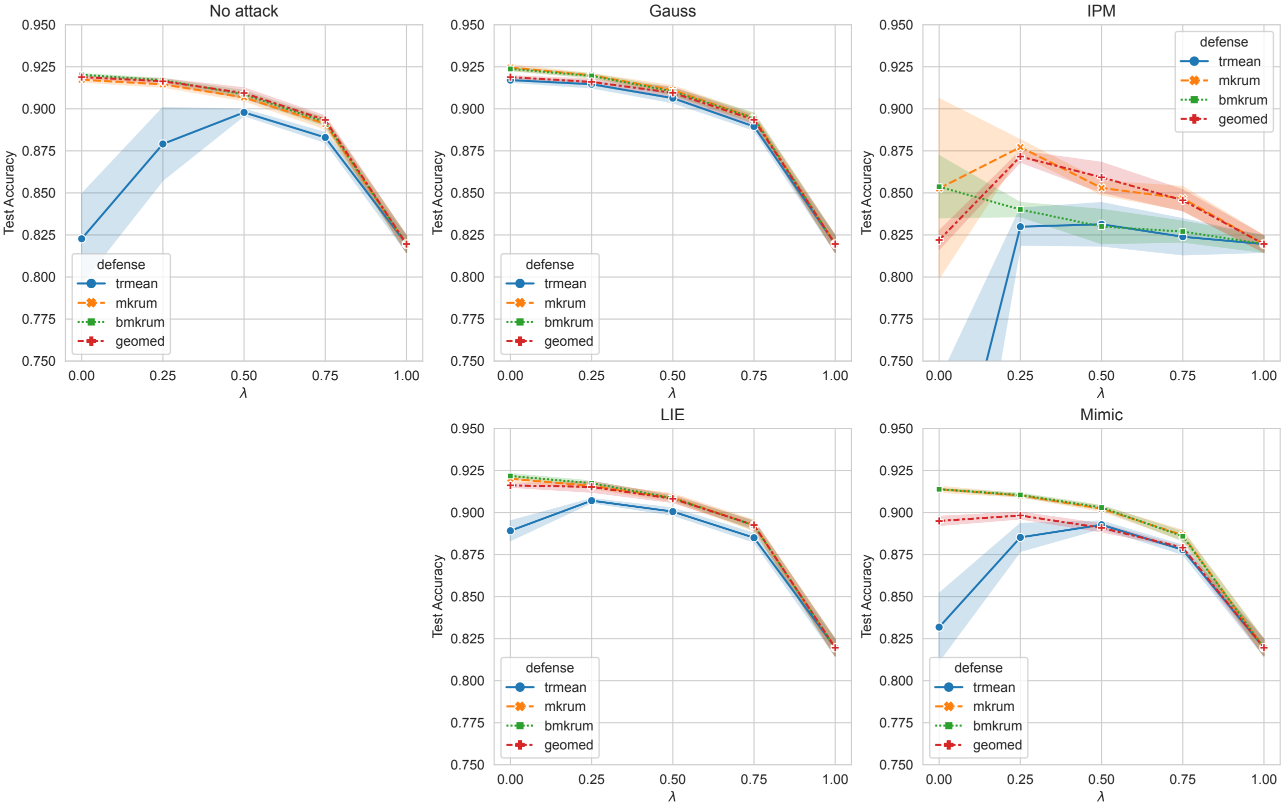

Figure 6: Performance of baseline majority-based AGRs when they use additional server data.

Figure 6 shows that TrMean has better performance when it is combined with server data ($\lambda = 0.25, 0.5$). However, the most competitive baseline majority-based AGRs (MKrum, BMKrum, GeoMed) get no performance improvement in most settings. We also notice that using additional server data can improve the worst-case test accuracy for most robust AGRs (usually under IPM attack). However, they are still significantly worse than BOBA, whose worst-case test accuracy is 91.6%.

C.3 Effect of Server Data (RQ3)

BOBA relies on server data for aggregation. In this subsection, we investigate how the quality and quantity of server data impact BOBA. Specifically, regarding data quality, we examine whether BOBA's performance is affected when server data contains noise (skewed feature distribution) or skewed label distribution. Concerning data quantity, we explore how much server data is sufficient for BOBA to perform robust aggregation.

C.3.1 Server Data with Feature Skewness

We first investigate whether the performance of BOBA is robust to feature skewness of server data. To simulate low-quality data, we introduce four types of random noises to the server data, following the approach proposed by (Hendrycks and Dietterich, 2019). As illustrated in Figure 7, BOBA exhibits remarkable consistency across various noise types, highlighting its robustness to variations in server data quality.

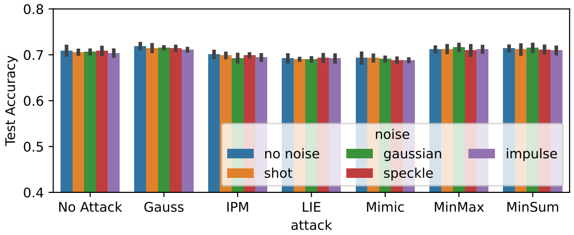

Figure 7: BOBA is robust to corrupted server data

C.3.2 Server Data with Label Skewness

In our experiments in the main text, the server dataset and testing dataset share the same label distribution. This assumption may be violated in real-world FL systems. In this subsubsection, we investigate the impact on reference-based AGR methods, including BOBA, when the server data also exhibits label skewness.

We conducted experiments on the AG-News dataset, and the results are presented in Figure 8. The "balanced" setting corresponds to the one in the main text, where the server has 30 samples for each class. In the "unbalanced" setting, the server has 40 samples for classes 0 and 1, and 20 samples for classes 2 and 3, thus introducing label skewness between the server data and test data, while the total amount of server data remains the same.

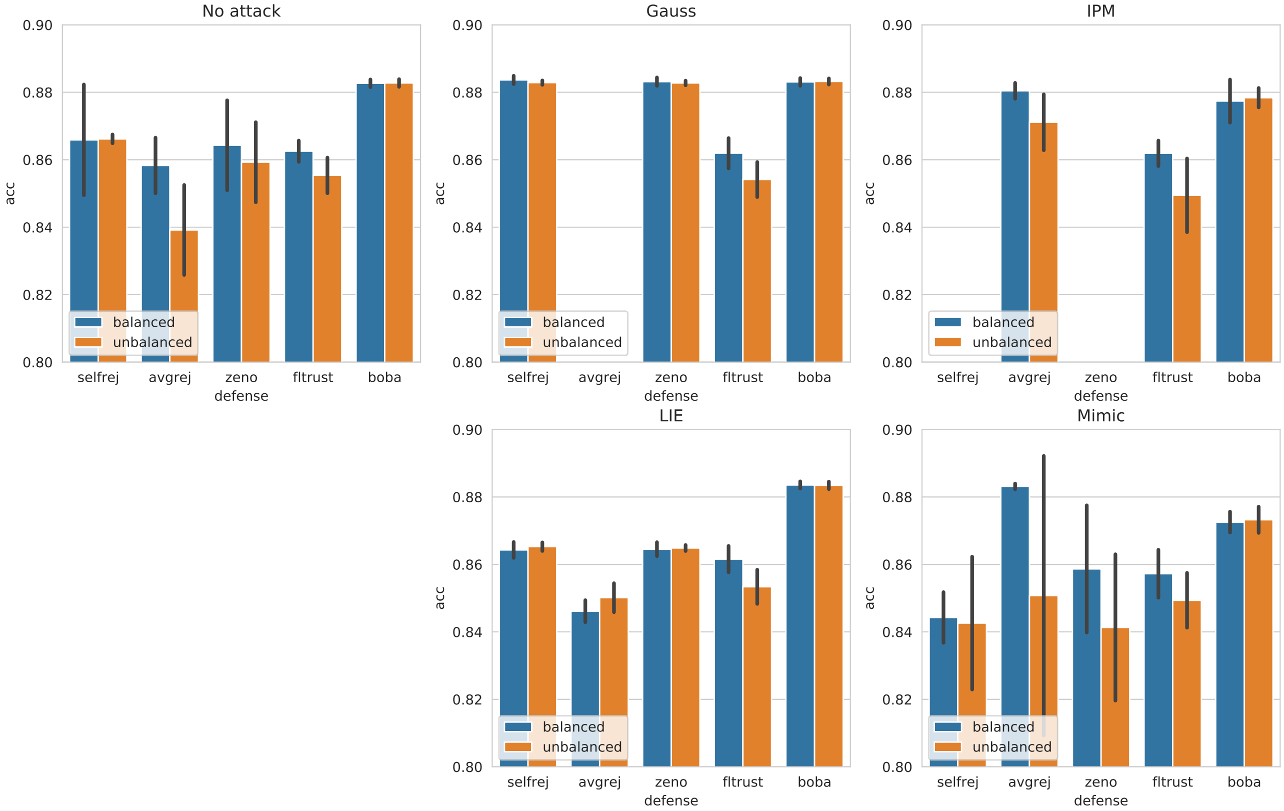

Figure 8: Comparison between aggregators using server data when server data is biased.

As shown in Figure 8, the performance of baseline AGRs generally degrades when the server data becomes unbalanced. However, the performance of BOBA remains almost the same across non-attack settings and four attacks, showing that BOBA is also robust to the label skewness of server data.

C.3.3 Quantity of Server Data

Finally, we investigate the impact of the quantity of the server dataset on BOBA. We test on CIFAR-10 data set when the number of server data per class varies from 1 to 320, with $|\mathcal{B}| = 15$ IPM attackers.

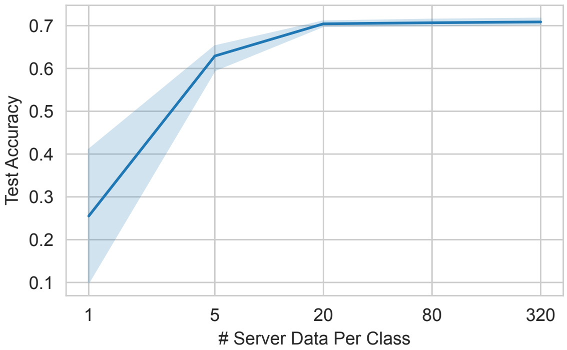

Figure 9: Effect of server data quantity. Error bars represent the s.d. of test accuracy over 5 random seeds.

As depicted in Figure 9, we observe that the test accuracy of our method increases as the number of server data per class rises from 1 to 20. It stabilizes once we have more than 20 samples per class. When the server data is limited to just one sample per class, our method's performance is suboptimal. However, with only 5 samples per class, our method already surpasses the highest-performing baseline AGR (MKrum with accuracy of 50.9%). This demonstrates that our method demands only a small quantity of server data, which is readily achievable in real-world applications.

C.4 Effect of Hyperparameters (RQ3)

C.4.1 Effect of f and $|\mathcal{B}|$

Similar to many robust AGRs (e.g., Krum (Blanchard et al., 2017) and TrMean (Yin et al., 2018)), BOBA incorporates a hyperparameter denoted as f, which signifies Byzantine tolerance, i.e., the maximum number of attackers that the AGR is designed to withstand. Theoretically, an appropriate choice of f should satisfy both $f \geq |\mathcal{B}|$ and Assumption 5.3 simultaneously to achieve robustness. In this section, we empirically evaluate the performance of BOBA under various combinations of $f \in 0, 20, 40, 60, 80, 100$ and $|\mathcal{B}| \in 0, 18, 36, 54$ using the AG-News dataset and the IPM attack. The results are depicted in Figure 10.

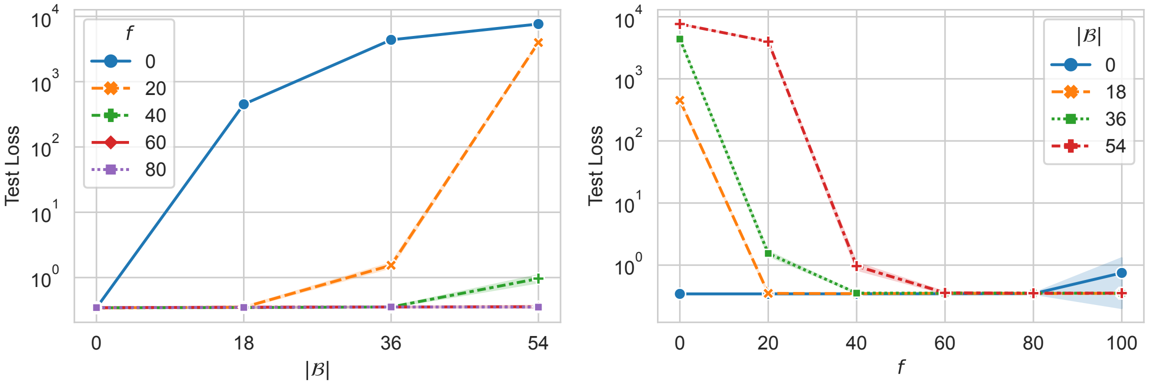

Figure 10: Effect of real number of Byzantines $|\mathcal{B}|$ and Byzantine tolerance f.

Effect of $|\mathcal{B}|$ (given fixed f) In the main text, we investigate the influence of $|\mathcal{B}|$ while keeping f constant. Specifically, we examine two extreme scenarios: when $|\mathcal{B}| = 0$ (where AGRs are most susceptible to selection bias) and when $|\mathcal{B}| \approx f$ (where the AGRs are most susceptible to vulnerability). As depicted in Figure 10 (left), our findings reveal that, within a reasonable range of f ($f \in [0, 80]$), the test loss remains minimal across all $|\mathcal{B}| \in [0, f]$. This demonstrates that BOBA exhibits robustness to the actual number of Byzantines $|\mathcal{B}|$ as long as $|\mathcal{B}| \leq f$.

Effect of f In Figure 10 (right), our observations indicate the following:

- For small values of f, BOBA exhibits robustness to only a limited number of Byzantine clients. For instance, when $f = 20$, BOBA demonstrates robustness when $|\mathcal{B}| = 0$ and $|\mathcal{B}| = 18$ but not when $|\mathcal{B}| = 36$ or $|\mathcal{B}| = 54$.
- As f increases moderately, BOBA becomes more resilient to Byzantine clients while preserving its unbiasedness in scenarios with few or no Byzantines.
- However, when f becomes excessively large (e.g., $f = 100$ in the context of $|\mathcal{H}| = 160$), BOBA loses its unbiasedness in scenarios with small $|\mathcal{B}|$. It's important to note that, when $|\mathcal{H}| = 160$ and $|\mathcal{B}| = 0$, we have $f = 100 > 80 = \frac{n}{2}$, implying an assumption that over half of the clients are Byzantine. Achieving optimal-order robustness under these conditions becomes impossible.

In summary, BOBA exhibits robustness across a broad range of f values, and we recommend selecting a moderately larger f to avoid underestimating $|\mathcal{B}|$.

C.4.2 Effect of $p_{\min}$

In addition to the hyperparameter f, BOBA incorporates another parameter, $p_{\min}$, which is a slightly negative number intended to prevent excessive removal of honest clients during stage 2. In all experiments presented in the main text, spanning various datasets and models, we maintain a consistent value of $p_{\min} = -0.5$. In this section, we explore a range of $p_{\min}$ values to assess the sensitivity of BOBA to this hyperparameter. Specifically, we conduct experiments on the CIFAR-10 dataset with $p_{\min} \in \{-1.0, -0.7, -0.5, -0.2, -0.1, 0.0\}$ under two scenarios: no-attacks and $|\mathcal{B}| = 15$ IPM attackers.

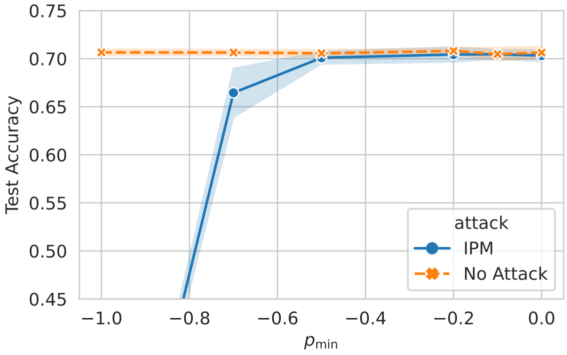

Figure 11: Effect of hyperparameter $p_{\min}$.

The results displayed in Figure 11 illustrate that BOBA consistently delivers robust performance across a wide range of $p_{\min}$ values within the interval $[-0.5, 0.0]$. However, large absolute value for $p_{\min}$, such as $p_{\min} = -1.0$, fails to discard attackers and thus compromise the robustness of BOBA. Therefore, in practical applications, we recommend opting for a small absolute value for $p_{\min}$ to ensure robustness.

C.5 More Label Skewness Settings (RQ3)

In the main test, we focus on pathological partition (McMahan et al., 2017), a very challenging non-IID setting where each client only has two classes of data. In this setting, BOBA has state-of-the-art unbiasedness and robustness. In this part, we test BOBA with two more label skewness settings: step partition (Chen and Chao, 2021) and Dirichlet partition (Yurochkin et al., 2019). We also test BOBA under partial participation.

C.5.1 Step Partition with Various Degrees of Non-IIDness

In this part, we study how the performances of BOBA and other baseline AGRs change when the non-IID degree varies. We focus on the MNIST dataset with *step partition* (Chen and Chao, 2021): each client has 8 minor classes (with less data) and 2 major classes (with more data). We use a parameter α to control the ratio of major and minor class data size. Therefore, larger α indicates a larger non-IID degree. We test $\alpha \in \{1, 2, 4, 8, +\infty\}$. Notice that $\alpha = 1$ refers to the IID setting, and $\alpha = +\infty$ refers to pathological partition in the main text.

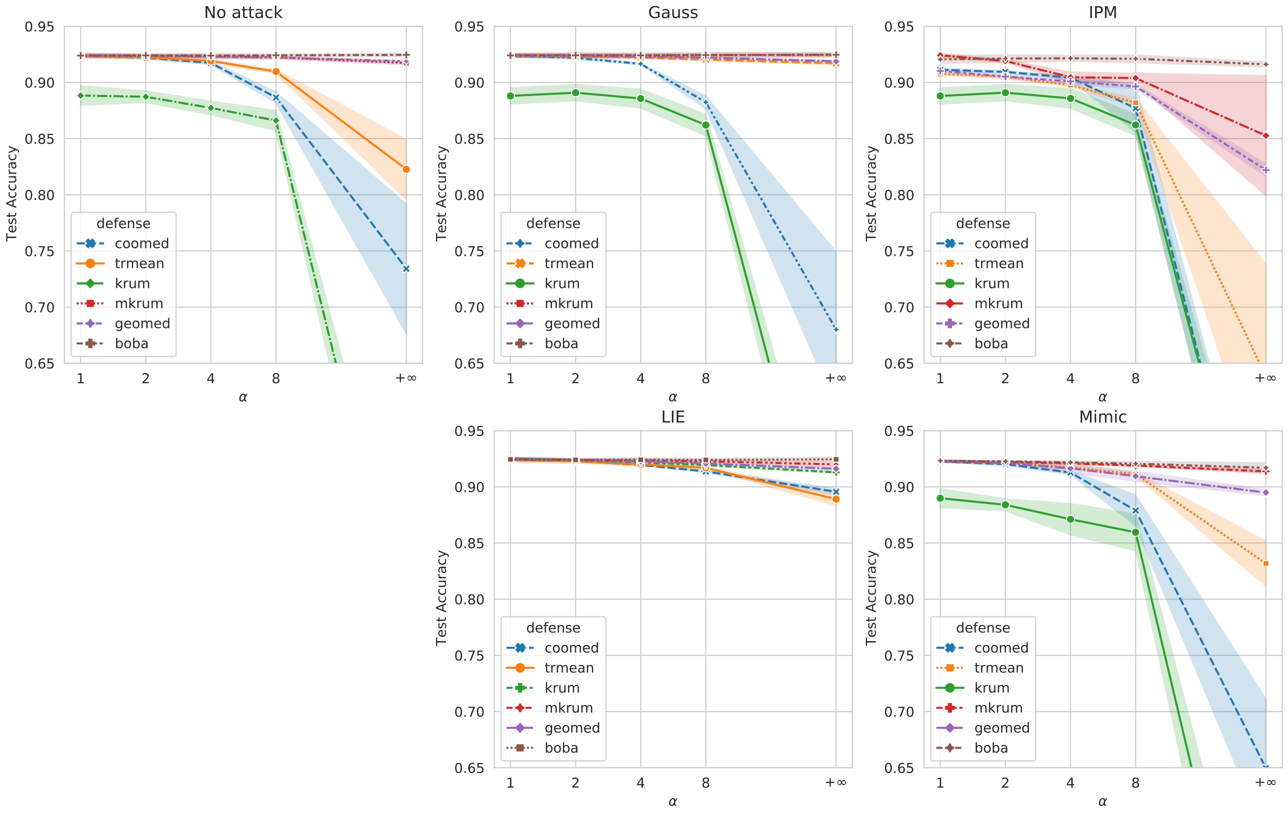

Figure 12: Effect of non-IIDness. Larger α indicates a larger non-IID degree.

We show the test accuracy for both BOBA and selected baselines in Figure 12.

- When the non-IID degree is small (e.g., $\alpha = 1$), almost all the robust AGRs have satisfactory performance under all kinds of attacks. This observation matches the theoretical analysis of their Byzantine-robustness under the IID assumption.

- However, when the non-IID degree (α) increases, all the baseline AGRs degrade rapidly, especially under the IPM attack. This observation matches our claim that IID AGRs degrade under label skewness.

- Finally, we notice that BOBA has almost constant performance under all attacks and non-IID degrees. This verifies our claim that BOBA has superior robustness and unbiasedness under label skewness.

C.5.2 Dirichlet Partition

In this part, we use *Dirichlet partition* ($\alpha = 0.01$) and compare BOBA to the strongest baselines. As shown in Table 7, BOBA consistently outperforms baselines in both unbiasedness and robustness.

Table 7: Performance (mean (s.d.) %) under Dirichlet distribution ($\alpha = 0.01$)

Dataset	Method	$\lvert\mathcal{B}\rvert = 0$		$\lvert\mathcal{B}\rvert = 15$ (for MNIST and CIFAR-10) or 54 (for AG-News) (Acc ↑)						
		Acc ↑	MRD ↓	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst
MNIST ($\lvert\mathcal{H}\rvert = 100$)	Average	**92.3** (0.1)	-	9.8 (0.0)	9.8 (0.0)	**92.3** (0.1)	**92.1** (0.2)	90.4 (0.1)	90.5 (0.3)	9.8
	MKrum	90.0 (1.1)	24.0 (7.6)	**92.4** (0.0)	85.5 (3.5)	91.0 (0.8)	89.5 (1.4)	83.8 (7.9)	85.4 (4.1)	83.8
	FLTrust	85.3 (1.0)	20.1 (3.5)	85.3 (1.0)	85.3 (1.0)	87.9 (0.5)	85.5 (0.6)	85.2 (1.0)	85.4 (0.6)	85.2
	BOBA	**92.3** (0.1)	**1.9** (1.9)	92.3 (0.1)	**92.6** (1.1)	91.7 (0.4)	**92.1** (0.3)	**91.8** (0.4)	**91.8** (0.4)	**91.7**
CIFAR-10 ($\lvert\mathcal{H}\rvert = 100$)	Average	**71.6** (0.4)	-	10.0 (0.0)	10.0 (0.0)	35.6 (2.6)	**70.3** (0.7)	35.3 (3.5)	34.0 (2.4)	34.0
	MKrum	68.3 (0.4)	27.3 (7.8)	**71.7** (0.5)	64.3 (2.7)	43.6 (2.5)	67.2 (1.1)	54.1 (6.9)	41.1 (3.5)	41.1
	FLTrust	49.1 (0.4)	36.9 (6.0)	49.1 (0.5)	47.9 (0.7)	48.2 (0.8)	49.5 (0.7)	48.4 (0.8)	48.3 (0.9)	47.9
	BOBA	69.5 (1.2)	**9.8** (4.3)	**71.7** (0.9)	**70.9** (0.5)	**71.1** (0.8)	67.2 (2.0)	**71.3** (0.9)	**71.2** (0.9)	**67.2**
AG-News ($\lvert\mathcal{H}\rvert = 160$)	Average	**88.3** (0.1)	-	24.3 (4.0)	25.0 (0.0)	87.0 (0.1)	87.5 (0.5)	36.0 (5.0)	31.1 (3.9)	24.3
	MKrum	80.8 (4.2)	38.4 (19.8)	**88.4** (0.1)	30.6 (9.2)	83.2 (0.8)	75.4 (7.3)	85.7 (3.9)	81.7 (1.7)	30.6
	FLTrust	85.3 (0.6)	7.8 (3.0)	85.5 (0.6)	85.3 (0.6)	85.3 (0.2)	85.5 (0.5)	85.3 (0.4)	85.3 (0.4)	85.3
	BOBA	88.2 (0.2)	**1.3** (2.4)	88.3 (0.1)	**88.1** (0.2)	**88.3** (0.1)	**87.6** (0.4)	**88.1** (0.2)	**88.3** (0.2)	**87.6**

C.5.3 Partial Participation

BOBA also works under partial participation, i.e., only a subset of clients are selected for each round. We conduct experiments with AG-News dataset, under participation rate in $\{0.25, 0.50, 0.75, 1.00\}$. As shown in Table 8, BOBA consistently outperforms baselines across different participation rates.

Table 8: Performance (mean (s.d.) %) under partial participation on AG-News ($\lvert\mathcal{H}\rvert = 160$)

Participation Rate	Method	$\lvert\mathcal{B}\rvert = 0$		$\lvert\mathcal{B}\rvert = 54$ (Acc ↑)						
		Acc ↑	MRD ↓	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst
0.25	Average	**88.0** (0.2)	-	25.7 (1.9)	25.0 (0.0)	**87.9** (0.2)	**86.9** (0.5)	33.8 (5.0)	81.3 (0.7)	25.0
	MKrum	87.3 (0.8)	6.0 (2.9)	**88.1** (0.1)	86.2 (0.6)	87.8 (0.1)	82.6 (0.7)	**87.9** (0.2)	86.1 (0.4)	82.6
	FLTrust	86.2 (0.4)	7.6 (1.8)	86.2 (0.6)	86.2 (0.5)	87.1 (0.4)	86.0 (0.4)	85.8 (0.4)	85.9 (0.5)	85.8
	BOBA	87.9 (0.2)	**3.2** (1.5)	**88.1** (0.2)	**87.7** (0.3)	87.7 (0.3)	86.8 (0.5)	87.7 (0.4)	**87.8** (0.3)	**86.8**
0.50	Average	**88.2** (0.2)	-	23.3 (1.9)	25.0 (0.0)	87.9 (0.3)	**87.2** (0.4)	36.4 (4.6)	41.7 (21.1)	23.3
	MKrum	87.6 (0.5)	5.1 (3.1)	**88.2** (0.2)	84.4 (1.2)	87.4 (0.5)	83.2 (1.9)	**88.1** (0.1)	85.8 (0.3)	83.2
	FLTrust	86.3 (0.3)	4.9 (1.5)	86.3 (0.4)	86.0 (1.0)	86.7 (0.7)	86.0 (0.9)	85.7 (0.9)	85.6 (0.9)	85.6
	BOBA	**88.2** (0.2)	**2.9** (2.5)	88.1 (0.3)	**87.8** (0.2)	**88.0** (0.2)	87.1 (0.6)	**88.1** (0.2)	**88.1** (0.2)	**87.1**
0.75	Average	**88.3** (0.1)	-	25.2 (3.6)	25.0 (0.0)	87.9 (0.1)	**87.3** (0.4)	30.5 (3.3)	29.8 (4.7)	25.0
	MKrum	87.8 (0.2)	4.9 (1.6)	88.3 (0.0)	48.8 (32.6)	87.1 (0.6)	83.2 (1.1)	**88.2** (0.1)	86.2 (0.4)	48.8
	FLTrust	86.3 (0.4)	5.4 (1.6)	86.3 (0.5)	86.2 (0.4)	86.4 (0.3)	85.9 (0.7)	86.0 (0.4)	85.9 (0.5)	85.9
	BOBA	**88.3** (0.1)	**1.7** (0.3)	**88.4** (0.1)	**88.0** (0.2)	**88.3** (0.1)	87.1 (0.5)	88.0 (0.3)	**88.1** (0.3)	**87.1**
1.00	Average	**88.3** (0.1)	-	25.4 (2.6)	25.0 (0.0)	87.5 (0.2)	87.2 (0.3)	35.9 (3.6)	30.5 (3.0)	25.0
	MKrum	88.0 (0.1)	4.6 (2.1)	**88.3** (0.2)	80.7 (6.0)	86.6 (0.2)	83.4 (0.6)	**88.3** (0.1)	85.9 (0.3)	80.7
	FLTrust	86.3 (0.4)	5.8 (1.0)	86.2 (0.5)	86.2 (0.4)	86.2 (0.4)	85.7 (0.8)	85.8 (0.9)	85.8 (0.5)	85.7
	BOBA	**88.3** (0.1)	**0.2** (0.1)	**88.3** (0.1)	**87.7** (0.7)	**88.4** (0.1)	**87.3** (0.3)	88.1 (0.1)	**88.3** (0.2)	**87.3**

C.6 Experiments with Both Label and Feature Skewness (RQ4)

BOBA is motivated by label skewness, where each honest client possesses a different label distribution and the same label-conditioned data distribution. However, practical FL systems may have more complex non-IIDness, with both label and feature distribution potentially varying. For example, as mentioned in our introduction with the example of animal image classification, different users may not only capture different prevalent species in their region but also exhibit variations in image appearance due to different camera settings. To validate whether BOBA remains effective in such a more complex non-IID setting, alongside generating label skewness using pathological partition, we inject different image corruption to each client.

Specifically, each client randomly selects one from the 15 common image corruptions (Hendrycks and Dietterich, 2019) and applies it to all of their training data. Consequently, even for images of the same class, there will be varying feature distributions across different clients. To facilitate a comparison with results obtained without image corruption, we do not add image corruptions to the testing data.

Table 9: Performance (mean (s.d.) %) on CIFAR-10 with label skewness and image corruptions

Method	$\lvert\mathcal{B}\rvert = 0$		$\lvert\mathcal{B}\rvert = 15$ (Acc ↑)						
	Acc ↑	MRD ↓	Gauss	IPM	LIE	Mimic	MinMax	MinSum	Wst
Average	68.7 (0.4)	-	10.0 (0.0)	10.0 (0.0)	64.6 (0.7)	67.5 (0.5)	27.9 (4.9)	21.6 (7.5)	10.0
CooMed	19.1 (4.9)	78.4 (2.0)	20.1 (1.5)	9.4 (1.8)	24.0 (2.1)	17.8 (1.8)	17.7 (1.2)	17.7 (1.2)	9.4
TrMean	24.1 (2.9)	78.8 (2.7)	53.8 (1.8)	14.2 (4.4)	30.6 (1.1)	20.1 (5.4)	20.5 (0.6)	22.2 (2.0)	14.2
Krum	33.4 (2.9)	79.3 (3.7)	34.4 (3.0)	33.1 (2.0)	38.7 (2.3)	31.0 (3.2)	34.1 (1.8)	33.2 (2.6)	31.0
MKrum	66.8 (1.1)	16.7 (11.7)	68.2 (0.7)	52.9 (10.2)	63.1 (1.1)	54.9 (25.1)	67.2 (0.4)	62.3 (2.1)	52.9
GeoMed	68.2 (0.6)	5.3 (1.5)	67.9 (0.9)	55.8 (4.6)	42.5 (2.7)	53.6 (5.0)	42.7 (3.0)	42.6 (2.9)	42.5
SelfRej	66.3 (1.4)	21.8 (13.4)	67.8 (0.4)	26.6 (6.5)	63.1 (1.1)	65.9 (0.8)	25.0 (2.5)	25.0 (2.8)	25.0
AvgRej	67.1 (1.7)	25.2 (20.3)	10.0 (0.0)	66.5 (0.8)	63.6 (0.8)	68.1 (0.6)	54.5 (6.2)	53.4 (4.7)	10.0
Zeno	66.3 (1.6)	23.8 (15.6)	67.8 (0.6)	26.7 (6.5)	63.2 (1.3)	66.2 (1.4)	23.8 (4.0)	23.5 (2.3)	23.5
FLTrust	50.1 (0.9)	29.1 (2.1)	50.0 (1.1)	47.8 (1.7)	47.3 (2.5)	49.8 (0.9)	49.0 (1.8)	49.1 (1.9)	47.3
ByGARS	29.2 (2.0)	57.7 (4.2)	29.1 (2.0)	50.9 (0.8)	27.3 (2.8)	29.7 (1.7)	24.4 (0.9)	24.4 (0.7)	24.4
B-Krum	52.4 (2.1)	69.6 (10.0)	58.1 (1.2)	55.8 (1.2)	41.1 (1.2)	43.5 (2.3)	57.7 (1.7)	57.9 (1.1)	41.1
B-MKrum	68.1 (0.4)	6.1 (3.0)	68.2 (0.8)	50.2 (6.6)	63.1 (1.1)	65.6 (2.2)	49.7 (3.4)	45.3 (6.7)	45.3
RAGE	57.2 (4.4)	45.6 (17.6)	65.7 (0.5)	57.5 (2.0)	45.3 (2.8)	59.3 (2.1)	58.5 (3.7)	58.2 (4.9)	45.3
BOBA	66.5 (1.0)	6.7 (2.6)	68.5 (0.3)	66.0 (0.7)	62.8 (1.6)	66.2 (0.7)	67.7 (0.5)	67.5 (0.6)	62.8

We have summarized the experimental results in Table 9. Due to the introduction of perturbation in our training data, the performance of virtually all aggregators has deteriorated. It is worth noting that even in the absence of attacks, the accuracy of the average aggregator has decreased from 71.7 to 68.7. However, in this scenario, BOBA still exhibits better robustness than all the baseline methods, while also being more unbiased than the majority of baselines. This suggests that BOBA can generalize to more complex non-IID settings that exhibit both feature and label skewness.

C.7 Extension to More FL Frameworks (RQ4)

In this part, we empirically show that our BOBA can be integrated with more FL algorithms with different local update function. Specifically, we consider FedAvg (McMahan et al., 2017) with $E = 5$ local epochs (instead of $E = 1$ for FedSGD) and FedProx (Li et al., 2020b) with $E = 5$ and local regularization hyperparameter $\mu = 0.01$.

With multiple local gradient descent steps, slightly abusing notation, we define the pseudo-gradient as follows:

$$\boldsymbol{g}_i = -(\boldsymbol{w}_i - \boldsymbol{w}_G)$$

where $\boldsymbol{w}_G$ is the global parameter send to client i (before local update), and $\boldsymbol{g}_i$ is the local parameter after local update.

First of all, we empirically verify that Proposition 1 still approximately holds, even when using different local update functions. With random initialization, we save all 100 honest (pseudo-)gradients and conduct principal component analysis on them, under both IID and label skew settings. We sort all principal components with their explained variance (from large to small), and plot them in Figure 13.

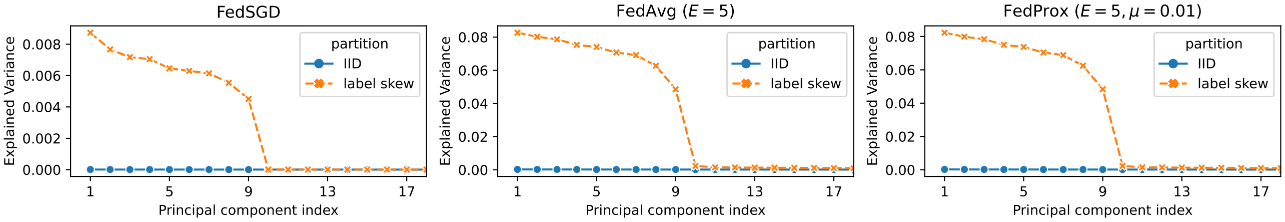

Figure 13: PCA of honest gradients on MNIST ($c = 10$)

It shows that (1) the total variance for label skewness setting is much larger than IID setting, and (2) most of the variances among honest gradients concentrate in the first $c - 1 = 9$ principal components. It verifies that Proposition 3.3 still approximately holds for FedAvg and FedProx.

Then, we evaluate whether BOBA can generalize these two FL frameworks. We run experiments with MNIST data set.

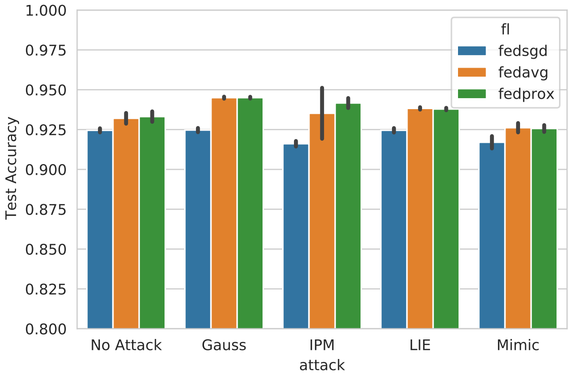

Figure 14: Applying BOBA on more FL frameworks

Figure 14 shows that BOBA can generalize FedAvg and FedProx. With the same number of communication rounds, BOBA + FedAvg/FedProx achieve higher accuracy, indicating their faster convergence. Meanwhile, BOBA remains its unbiasedness and robustness across all attacks.