BOBA: Byzantine-Robust Federated Learning with Label Skewness

Wenxuan Bao, Jingrui He
University of Illinois at Urbana-Champaign
{wbao4,jingrui}@illinois.edu

Abstract

In federated learning, most existing techniques for robust aggregation against Byzantine attacks are designed for the IID setting, i.e., the data distributions for clients are 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 techniques suffer from selection bias, leading to significant performance drop for particular classes; they are also more vulnerable to Byzantine attacks due to the increased deviation 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 optimal order. Empirically, we verify the superior unbiasedness and robustness of BOBA across a wide range of models and data sets against various baselines.

1 Introduction

Federated learning (FL) (McMahan et al. 2017) has attracted much research attention in the past decade. It refers to a machine learning system where multiple clients collaboratively train a shared machine learning model under the orchestration of a central server, thus avoiding the leakage of their own private and sensitive data. During training, each client downloads a copy of the shared model from the server, trains the model with local data, and submits a model update to the server for aggregation. In FL, each client’s data is stored locally and invisible to both the server and other clients. FL has wide applications in sales, finance, healthcare (Yang et al. 2019), etc.

FL systems can be vulnerable to attacks and failures (Kairouz et al. 2021; Lyu et al. 2020). In particular, Byzantine attacks can lead to the convergence to an unsatisfactory model or even divergence (Blanchard et al. 2017). A common defending mechanism against Byzantine attacks is to replace the gradients averaging with robust estimation of the center (Chen, Su, and Xu 2017; Yin et al. 2018). These methods have proven Byzantine-robustness when data from different clients are independent and identically distributed (IID). However, the IID assumption is usually violated in real applications. For example, if each client corresponds to a specific user, the data distribution usually varies from user to user. It is unclear how existing robust aggregators are affected by the violation of the IID assumption.

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, each client has the same conditional data distribution given labels, but their label distributions can vary a lot. Label skewness degrades the model in two ways. First, it introduces a selection bias of clients. Aggregators tend to select some clients over others across iterations and thus bias the model. Second, it increases the deviation among honest gradients, making existing aggregators more vulnerable to Byzantine attacks. It requires advanced techniques to tackle these challenges in FL with label skewness.

Focusing on label skewness, we find that the gradients of honest clients are distributed near a \((c-1)\)-simplex, where \(c\) is the number of classes. Based on this, we propose BOBA (Byzantine-rO bust and unBiased Aggregator), a two-stage algorithm to estimate this simplex. In the first stage, we robustly estimate the low dimensional affine subspace that the simplex is in and project all gradients to 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 possible Byzantines. With our algorithm, all honest gradients will be preserved with small perturbation, while Byzantine gradients will either be discarded or weakened. We evaluate BOBA theoretically and empirically, proving that it can alleviate selection bias and achieve robustness. We summarize our contributions below:

- We make a systematic analysis of the robustness of FL with label skewness. Specifically, we summarize two novel challenges and propose optimal order robustness to measure the robustness of an aggregator. (Section 3)
- Based on our finding that honest gradients distribute near a \((c-1)\)-simplex, we propose BOBA, including an objective addressing both label skewness and robustness, and an efficient optimization algorithm. (Section 4)
- We theoretically derive the gradient estimation error of BOBA and make a convergence analysis. Theoretical results show that BOBA achieves optimal order robustness while baseline aggregators do not. (Section 5)
- We empirically evaluate the unbiasedness and robustness of BOBA across a wide range of models, data sets, and attacks. We show that BOBA outperforms various baseline aggregators, even when they are allowed to leverage
2 Related Works

Robustness in FL

There are extensive works of robust aggregators with IID data (Blanchard et al. 2017; Chen, Su, and Xu 2017; Yin et al. 2018; Pillutla, Kakade, and Harachouri 2022), most of which replace the averaging step on the server with a robust mean estimator. These methods are theoretically proved 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. A few works have studied robustness with non-IID data by making the clients “more IID”. (Karimireddy, He, and Jaggi 2022) applies bucketing to make the input of the aggregator more homogeneous, while sacrificing some robustness. (Ghosh et al. 2019) divides clients into IID groups and learns 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) introduces 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.

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. FedAWeS (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 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.

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 H) or Byzantine (in Byzantine set B). In each communication round, the server broadcasts the parameter \( w_G \in \mathbb{R}^d \) to all clients. Each honest client computes the gradient with its own data \( \{x_{ij}\}_{j=1}^m \) sampled from \( P_i \) and sends back the honest gradient \( g_i = \nabla_{w_G} L_i(w_G) \), where \( L_i(w_G) = \frac{1}{m} \sum_{j=1}^m L(w_G; x_{ij}) \). Each Byzantine client can send arbitrary Byzantine gradient to the server. Finally, the server aggregates all n gradients \( \mu = \text{Agg}(\{g_i\}_{i=1}^n) \) and updates the parameter \( w_G \leftarrow w_G - \eta \mu \), where Agg(·) is the aggregation function, and \( \eta \) is the learning rate. In training phase, the system minimizes the empirical risk, \( \frac{1}{|H|} \sum_{i \in H} L_i(w_G) \). Let \( |H|, |B| \) be the real number of honest and Byzantine clients respectively. As the server does not know the exact number of Byzantines, let \( f \) be the declared number of Byzantines, i.e. the max number of Byzantines that the aggregator is robust to.

For each honest client \( i \in H \), let \( E\mathbb{g}_i \) be its expected gradient, where the expectation is taken on data sampling from \( P_i \). Let \( \mu = \frac{1}{|H|} \sum_{i \in H} g_i \) denote the gradient of empirical risk and \( E\mu = \frac{1}{|H|} \sum_{i \in H} E\mathbb{g}_i \) denote its expectation, also the gradient of population risk. Aggregators aim to estimate \( E\mu \) robustly under arbitrary behavior of Byzantine clients.

Distribution of Honest Gradients

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

Definition 1 (Inner, outer and total deviations). For an honest client \( i \in H \), its inner deviation is \( g_i - Eg_i \); its outer deviation is \( Eg_i - E\mu \), and its total deviation is \( g_i - E\mu \).

Inner deviation measures the randomness of sampling data from clients’ data distribution \( P_i \), while outer deviation measures the difference from a client’s data distribution \( P_i \) to the global distribution \( P = \frac{1}{|H|} \sum_{i \in H} P_i \), without randomness.

In the IID setting, the outer deviation is zero, implying that honest gradients \( \{g_i\}_{i \in H} \) are distributed around the same center \( E\mu \). When the local batch size \( B \to +\infty \), each \( g_i \overset{D}{\to} E\mu \). Such implication does not hold with label skewness, since outer deviations are non-zero. We formally define label skewness and analyze how honest gradients distribute.

Definition 2 (c-label skew distribution). Honest clients’ data distributions \( \{P_i\}_{i \in H} \) are c-label skew distribution if

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

where \( P_i(\xi) \) is the data distribution of client \( i \). The label \( z \) can take \( c \) finite values. The mixture weight \( p_{iz} \) is the label distribution of client \( i \) satisfying \( \sum_{z=1}^{c} p_{iz} = 1 \) and the mixture component \( Q_z(\xi) = P_i(\xi|z) \) is the conditional distribution given \( z \). Different clients share the same \( \{Q_z(\xi)\}_{z=1}^c \) but different label distribution \( p_i = [p_{i1}, \ldots, p_{ic}]^\top \).

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 1 (Distribution of expected honest gradients). With c-label skew distribution, for any honest client \( i \in H \),

\[
E\mathbb{g}_i = \sum_{\xi} P_i(\xi) \nabla_{w} L(w; \xi) = \sum_{\xi} \sum_{z=1}^{c} p_{iz} Q_z(\xi) L(w; \xi)
\]

\[
= \sum_{z=1}^{c} p_{iz} Q_z(\xi) L(w; \xi) = \sum_{z=1}^{c} p_{iz} E\gamma_z
\]

where \( E\gamma_z = \sum_{\xi} Q_z(\xi) L(w; \xi) \) is the expected gradient computed with data from class \( z \).
Figure 1: Comparison of aggregation result. Orange set is the range of aggregation results when Byzantines are different. The left three aggregators can have very large aggregation error, while BOBA is much more accurate. See details in Appendix B.2.

Proposition 1 shows that each expected honest gradient is a convex combination of \( \{\mathbb{E}\gamma_z\}_{z=1}^{c}\), whose range forms a \((c-1)\)-simplex. We define the honest simplex to be \(\{\sum_{z=1}^{c}p_{z}\mathbb{E}\gamma_z : \sum_{z=1}^{c}p_{z} = 1, p_{z} \geq 0\}\), and the honest subspace to be \(\{\sum_{z=1}^{c}p_{z}\mathbb{E}\gamma_z : \sum_{z=1}^{c}p_{z} = 1\}\).

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 principle component analysis on honest gradients, the variance should concentrate in the first \((c-1)\) principle components. Figure 2 verifies our finding on MNIST. Appendix B.2 gives details of this preliminary experiment.

Notice that the conclusion can be extended to a general mixture distribution (Marfoq et al. 2021) when \(z\) is a general latent variable other than the class label, which gives more flexibility. Our paper focuses on label skewness for clarity.

Figure 2: PCA of honest gradients on MNIST (\(c = 10\))

**Challenges of Label Skewness**

In the IID setting, every honest gradient is an unbiased estimation of \(\mathbb{E}\mu\). It is easy to design a robust aggregator since it only needs to find one honest gradient (or a close Byzantine gradient). However, in label skewness settings, each honest gradient can deviate a lot from \(\mathbb{E}\mu\). It introduces two challenges to aggregators designed for IID settings: selection bias and increased vulnerability.

**Selection bias** Many robust aggregators, including Krum (Blanchard et al. 2017), select a subset of gradients and aggregate. With label skewness, such aggregators select some clients much more often than others. Since each honest gradient is only a biased estimation of \(\mathbb{E}\mu\), the aggregation result is also biased, even when there is no Byzantines. In Figure 1 where honest gradients form two clusters, three baseline aggregators all choose the cluster with more clients, resulting in a highly biased aggregation (the red square). Considering each cluster belongs to one class, the model will be trained with one class of data only, resulting in a model predicting every sample as this class.

**Increased vulnerability** We observe that the aggregation result can deviate more from the true center \(\mathbb{E}\mu\) when the total deviations become larger. Figure 1 shows that the range of aggregation results is huge for IID aggregators, even along the direction where honest gradients do not diverge.

For the two reasons above, although IID aggregators are theoretically robust in IID settings, they have large gradient estimation error even with the absence of Byzantines, and also suffer more with their existence. It is necessary to reconsider the theory of robustness in FL with label skewness.

**Optimal Order Robustness**

In IID settings, robustness is usually explained as bounded gradient estimation error (Chen, Su, and Xu 2017; Blanchard et al. 2017; Yin et al. 2018). However, such bounds confuse inner and outer deviations, ignore the bias introduced by robust aggregators, and thus can be vacuous (e.g., in Figure 1). Different from them, a good aggregator for label skewness should not bias the aggregation when there are no Byzantines. We define this property as unbiasedness.

**Definition 3 (Unbiasedness).** An aggregator \(\text{Agg}(\cdot)\) is unbiased if \(\text{Agg}([g_i]_{i \in \mathcal{H}}) \rightarrow \mathbb{E}\mu\) when \(g_i \rightarrow \mathbb{E}g_i, \forall i \in \mathcal{H}\); i.e., the aggregator should not introduce additional bias when there are no Byzantines.

However, when there are Byzantine clients, bias is inevitable in label skewness settings. For example, Byzantine clients can mimic the behavior of honest clients (Karimireddy, He, and Jaggi 2022). In this case, any unbiased aggregator should keep them since they are the same as honest ones. With label skewness, such Byzantines can bias the average label distribution by \(\|\Delta p\|_1 \leq 2|\mathcal{B}|/n\), introducing an inevitable gradient estimation error of \(\mathcal{O}(|\mathcal{B}|/n)\). Based on this fact, we propose optimal order robustness, an adaptive upper bound of gradient estimation error.

**Definition 4 (Optimal order robustness).** An aggregator \(\text{Agg}(\cdot)\) has optimal order robustness if \(\|\text{Agg}([g_i]_{i=1}^{\mathcal{H}}) - \mathbb{E}\mu\|_2 \rightarrow \mathcal{O}(\beta)\) when \(g_i \rightarrow \mathbb{E}g_i, \forall i \in \mathcal{H}\). \(\beta = |\mathcal{B}|/n\) is the fraction of Byzantines.
Algorithm 1: BOBA (stage 1: line 1-5, stage 2: line 6-10)

Input: $G = [g_1, \cdots, g_n]$, $\Gamma = [\gamma_1, \cdots, \gamma_c]$, $n, f, c, p_{\text{min}}$

Output: Aggregation result $\hat{\mu}$

1: Initialize subspace $\mathcal{P}: m, U, \Sigma, V = \text{TrSVD}_{c-1}(\Gamma)$
2: while not converge do
3:   Update $r$: $G'_{[n-f]} = \{n - f \text{ gradients in } G \text{ with smallest } \|g_i - \Pi \hat{g}(g_i)\|_2\}$ where $\Pi \hat{g}(g_i) = UU^T (g_i - m) + m$
4:   Update $\mathcal{P}$: $m, U, \Sigma, V = \text{TrSVD}_{c-1}(G'_{[n-f]})$
5: end while
6: Encode: $\hat{g}_i = U^T (g_i - m_i)$, $\forall i$; $\hat{\Gamma} = U^T (\Gamma - m 1^T)$
7: Estimate: $\hat{\mu}_i = \frac{1}{m} \sum_{i=1}^m g_i$
8: Filter: $\mathcal{A}(\hat{\mu}_i)$
9: Aggregate: $\hat{\mu} = \sum_{i=1}^n a_i \hat{g}_i / \sum_{i=1}^n a_i$
10: Decode: $\hat{\mu} = U \hat{G} + m$

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, $m, U, \Sigma, V = \text{TrSVD}_{c-1}(G)$, s.t. $U \Sigma V^T \approx G - m 1^T$ where $G = [g_1, \cdots, g_n] \in \mathbb{R}^{d \times n}$ is the client gradient matrix, $m = \frac{1}{n} \sum_i^m G_i \in \mathbb{R}^d$ is their average, $U \in \mathbb{R}^{d \times (c-1)}$, $V \in \mathbb{R}^{n \times (c-1)}$ are column-orthogonal and $\Sigma \in \mathbb{R}^{(c-1) \times (c-1)}$ is diagonal. TrSVD fits a $(c - 1)$-dimensional affine subspace $\mathcal{P}$ minimizing the reconstruction loss

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

where $\Pi \hat{g}_i = UU^T (g_i - m) + m$ is a projection function that projects vectors to $\mathcal{P} = \{UX : x \in \mathbb{R}^c\}$. Although vanilla TrSVD is robust to small perturbations (Stewart 1990), it is not Byzantine-robust. 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 $\{y = x\}$.

Objective Since reconstruction loss is vulnerable, we design trimmed reconstruction loss to robustify TrSVD.

$$\ell_t(\mathcal{P}) = \min_{r_{i} \in [0,1]^{m}} \sum_{i=1}^n r_i \|g_i - \Pi \hat{g}_i\|^2_2$$

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 Byzantine 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 Condition 1). We show in Appendix A.5 that stage 1 is theoretically guaranteed to estimate the honest subspace robustly.

Strongest colluding Byzantines may focus on another dimension different from the $c - 1$ honest dimensions. But BOBA stage 1 will not identifies the Byzantine dimension as honest. If it makes such 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). Converting TrSVD to them results in non-negligible loss $\propto \delta^2$, which can be further optimized. Meanwhile, correctly identifying all honest dimensions results in a loss unrelated to outer deviations, which is much smaller. In our experiments, we also show that BOBA can resist such colluding Byzantines, e.g. Signflip (Xie, Koyejo, and Gupta 2019) and Little (Baruch, Baruch, and Goldberg 2019).

4 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 it. 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). Ideally, 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 impact of Byzantine gradients is limited.
Optimization To minimize trimmed reconstruction loss, we solve a joint minimization problem

\[
\hat{\mathcal{P}}, \hat{r} = \arg\min_{\mathcal{P}, r} \ell_t(\mathcal{P}, r) = \sum_{i=1}^{n} r_i \|g_i - \Pi_{\mathcal{P}}(g_i)\|_2^2
\]

Fixing \(\mathcal{P}\), the optimal \(r\) selects the \(n-f\) nearest neighbors of \(\mathcal{P}\); while fixing \(r\), 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, which iterates every possible value of \(r\), 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 an exponentially high computational complexity.

Instead, we use alternating optimization, with details in Line 2 - 5 in Algorithm 1. It alternately updates \(\mathcal{P}\) and \(r\) 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, which 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 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.

Stage 2: Finding the Honest Simplex

Stage 2 aims to estimate the honest simplex and rule out gradients outside it. Since we have no information on the distribution of clients' label distributions, it is impossible to estimate the honest simplex with only clients' gradients. We use a small amount of server data to estimate \(c\) vertices of honest simplex, and reconstruct the label distribution of each client. Clients with abnormal label distribution will be discarded.

Proposition 1 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 \(\{\gamma_i\}_{i=1}^{c}\) with the same process of honest clients. To estimate the label distribution of a client \(i\), we solve

\[
\sum_{z=1}^{c} \hat{p}_{iz} \Pi_{\mathcal{P}}(\gamma_z) = \Pi_{\mathcal{P}}(g_i), \quad \sum_{z=1}^{c} \hat{p}_{iz} = 1
\]

Solving this linear system in the gradient space is inefficient. Therefore, we split the projection into two steps: encoding \((\hat{g}_{i} = U^\top (g_i - m))\) and decoding \((\Pi_{\mathcal{P}}(g_i) = U \hat{g}_{i} + m)\), and solve the linear system in the latent space. The explicit solution is shown in line 6-7 in Algorithm 1.

If our estimation is perfect (e.g. when \(g_i \rightarrow E \hat{g}_{i}, \gamma_z \rightarrow E \gamma_z\)), \(\hat{p}_{i}\) will lie in the probability simplex, i.e. \(\{p : 1^t p = 1, p \geq 0\}\) if client \(i\) is honest, while it can be arbitrary if client \(i\) is Byzantine. So we can discard clients with negative entries in \(p_i\), since they must be Byzantines. However in practice, our estimation has bounded error (Appendix A.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 criteria

\[
a = A(\{p_i\}_{i=1}^{n}), \quad \text{where } a_i = 1 \text{ if.f. } \min p_{iz} \geq p_{min}
\]

Here \(p_{min} \leq 0\) is a hyper-parameter deciding the threshold of rejecting Byzantines. In our implementation, we will accept \(n-f\) clients with largest \(\min_{i=1}^{n} p_{iz}\) if \(\sum_{i=1}^{n} a_i \leq n-f\), i.e., our acceptance criteria 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.

Computational complexity The computational complexity of BOBA is \(O(kend)\) if it conducts TrSVD for \(k\) times. The complexity of TrSVD is \(O(cnd)\) (Halko, Martinsson, and Tropp 2011), where \(c\) is the number of classes, \(n\) is the number of clients and \(d\) is the dimension of gradient. 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 three settings with MNIST, CIFAR-10 and AG-News (see Section 6), \(k = 3.29, 3.20, 4.77\) on average.

5 Theoretical Analysis

This section provides theoretical results of the convergence of BOBA and shows that BOBA has optimal order robustness. For space limits, all proofs are deferred to Appendix A. Similar to previous works (Blanchard et al. 2017; Chen, Su, and Xu 2017; Yin et al. 2018), we start with a classical convergence analysis connecting convergence to the gradient estimation error.

Proposition 2. With \(L\)-smooth and \(\mu\)-strongly convex population risk, conducting SGD with gradient \(\hat{\mu}\) with estimation error \(\|\hat{\mu} - E\mu\|_2 \leq \Delta\), and step size \(\frac{1}{L}\), after \(t\) steps,

\[
\|w(t) - w^*\|_2 \leq \left(1 - \frac{\mu}{L^2}\right)^t \|w(0) - w^*\|_2 + \frac{\mu \Delta}{L^2 - L \sqrt{L^2 - \mu^2}}
\]

where \(w^*\) is the optimal model parameter.

Proposition 2 shows that the parameters exponentially converge to the optimal parameters with an error rate of \(O(\Delta)\). In other words, the gradient estimation error is a good indicator of the error of the model. The following part derives the gradient estimation error of BOBA and test whether it has optimal order robustness. We start with assumptions.

Assumption 1 (Bounded deviations).

1. Honest client inner deviation: with large probability \(1-p\) where \(p = O(\frac{1}{\sqrt{L}}), \frac{1}{\sqrt{L}} \sum_{i \in \mathcal{H}} \|g_i - E g_i\|_2^2 \leq c^2\).
2. Honest client outer deviation: \(\|E g_i - E \mu\|_2 \leq \delta^2, \forall i \in \mathcal{H}\).
3. Server inner deviation: with large probability \(1-p_s\) where \(p_s = O(\frac{1}{\sqrt{c}}), \frac{1}{c} \sum_{i=1}^{c} \|\gamma_z - E \gamma_z\|_2^2 \leq c^2\).
4. Server outer deviation: \(\|E \gamma_z - E \mu\|_2 \leq \delta^2, \forall z = 1, \cdots, c\).
Assumption 3 (Satisfactory optimizer). There exists a projection function $P'$ fitted by TrSVD on $n-f$ honest gradients with $\ell_i(P') \geq \ell_i(P)$.

Assumption 1 is a deterministic version of bounded inner and outer variation (Wu et al. 2020; Peng, Wu, and Ling 2020), and can be derived from it with Markov inequality. Assumption 2 reformulates Condition 1 with SVD, since the number of non-zero singular values indicates the dimension of a subspace. Assumption 3 says that the optimizer can find a solution that is not too bad (not necessarily the global minimum). This assumption is satisfied in practice when the initialized subspace is fitted by server gradients, which are computed similarly as honest gradients and thus near the honest subspace $P^\star$.

Theorem 3. With probability at least $1 - p - p_s$,

$$\|\hat{\mu} - \mu\|_2 \leq \left[ 1 + 2 \sqrt{\frac{1}{n - 2f}} + \frac{\delta^2}{\sigma^2} \sqrt{|H|} \right] \epsilon + \beta(1 + c|\mu_{\min}|)(\sqrt{c\epsilon_s + \delta_s})$$

where $\beta = |B|/n$ is the real fraction of Byzantine clients. When the outer deviation increases $t$ times, both $\delta$ and $\sigma$ increase $t$ times. When all clients are duplicated, $\delta^2$ does not change but $\sigma^2$ doubled. Thus generally we have $\frac{\epsilon_s}{\sigma^2} \propto \frac{1}{n}$.

When $\epsilon_s = O(\epsilon)$, $\delta_s = O(\delta)$, $c = O(1)$, $\frac{n-2f}{n} = O(\frac{1}{n})$, $|H| = O(n)$, and $|\mu_{\min}| = O(1)$, we have $\|\hat{\mu} - \mu\|_2 = O(\epsilon + \beta\delta)$. Let $\epsilon \rightarrow 0$, we conclude that BOBA has optimal order robustness, which baseline aggregators do not achieve (see Appendix A.9).

### 6 Experiments

In this section, we conduct experiments to evaluate the unbiasedness and robustness of BOBA. Specifically, we aim to answer the following research questions.

- **RQ1**: Is BOBA unbiased when there are no Byzantines?
- **RQ2**: Is BOBA robust enough to Byzantine attacks?
- **RQ3**: How do different components contribute to the effectiveness, robustness and efficiency of BOBA?

We experiment 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, Zhao, and LeCun 2015). We partition training sets to $|H| = 100, 100, 100$ honest clients with the method in (McMahan et al. 2017), respectively. For page limits, we give detailed experimental settings and all error bars in Appendix B.1 and B.2.

#### Byzantines

We consider three types of Byzantines:

- **Gaussian** (Blanchard et al. 2017), a non-colluding attack uploading large-scale vectors from Gaussian distribution.

- **Signflip** (Xie, Koyejo, and Gupta 2019), a strong colluding attack uploading $g_i = -\gamma \cdot \frac{1}{|H|} \sum_{i \in H} g_i$, with large $\gamma$ to make Average perform gradient ascent. We set $\gamma = 10$ for AG-News and $\gamma = 20$ for MNIST and CIFAR-10 (since there are less Byzantines).

- **Little** (Baruch, Baruch, and Goldberg 2019), a moderate 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., $z = \phi^{-1}((n - |n/2 + 1|))/(n - |B|)$.

#### Aggregators

We compare a wide range of baselines:

- **Average aggregator** (McMahan et al. 2017) simply averages all gradients. It is unbiased but vulnerable to attacks.

- **Server aggregator** only uses server data to fit a model. We use it to verify that one cannot train a good model with server data only.

- **IID aggregators** include coordinate median (CooMed) / trimmed mean (TrMean) (Yin et al. 2018), Krum / multi-Krum (MKrum) (Blanchard et al. 2017), and geometric median (GeoMed) (Chen, Su, and Xu 2017).

- **Bucketing aggregators** (Karimireddy, He, and Jaggi 2022) feed buckets (means) of gradients to IID aggregators. We consider bucketing + Krum (B-Krum) and bucketing + MKrum (B-MKrum) with $s = 2$.

- **Loss-based rejection aggregators** (Fang et al. 2020) detect possible Byzantine clients based on the performance on server data. SelfRej selects $n-f$ clients whose local models $w_i = w_G - \eta g_i$ have smallest loss, while AvgRej selects $n-f$ clients whose gradients can lower the loss of averaged model the most.

All aggregators are set to be robust to $f = 16$ Byzantines on MNIST/CIFAR-10 and $f = 60$ on AG-News. BOBA uses $\mu_{\min} = -0.5$. 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.

#### Unbiasedness and Robustness (RQ1, 2)

**Evaluation of unbiasedness** We evaluate the unbiasedness with $|B| = 0$. Besides accuracy, we introduce max-recall-drop (MRD) as a compliment. It computes how the recall scores of each class differ from the model trained with Average (with $|B| = 0$) and picks the largest absolute drop. Smaller MRD indicates a less biased aggregator. As selection bias may dramatically decrease some classes’ recalls.
while increasing the others. MRD can reflect selection bias better than using accuracy only. Results are shown in Table 1. We observe that BOBA has accuracy very close to Average, and the smallest MRD among all robust aggregators.

**Evaluation of robustness** We evaluate the robustness with $|S| = 15, 15, 54$ on three data sets respectively. Results are shown in Table 2. Notice that Byzantines can choose their attack methods to reduce the model accuracy as much as possible. Therefore, we summarize the worst-case accuracy for a clear comparison in the “worst” column. We observe that BOBA significantly improves the worst-case accuracy by 1.3%, 13.4%, 10.7% on three data sets, respectively.

**Discussion of aggregators** Models trained with CooMed and Krum suffer a lot from selection bias, totally fail on some classes. As their natural interpolation with Average, TrMean and MKrum select more clients and thus perform better. MKrum usually has good overall accuracy but still suffers from selection bias and has non-negligible recall drops. GeoMed can be expressed as a positive weighted averaging of all gradients, avoiding dropping useful clients. This property alleviates the selection bias, but also makes GeoMed more vulnerable to Byzantine attacks compared to MKrum. We also observe that bucketing aggregators can alleviate the selection bias of a robust aggregator at the cost of robustness. SelfRej is vulnerable to colluding attacks, while AvgRej is vulnerable to non-colluding Gaussian attacks. Among all aggregators, BOBA is the only method that achieves superior unbiasedness and robustness. It gives uniformly good performance under all settings.

**Leveraging server data** Noticing that additional server data may introduce advantages to BOBA, we also study whether server data can enhance baseline aggregators for a fair comparison. We enhance the baselines with $\mu = (1 - \lambda)\text{Agg}(\{g_i\}_{i=1}^n) + \lambda(\sum\gamma_z)\text{Agg}$ and test different $\lambda$. Results are given in Appendix B.4, showing that additional server data cannot enhance these aggregators.

**Ablation Study (RQ3)**
In this part, we test how each components of BOBA contribute to the aggregation. Since we need to test exhaustive searching, we train a 2-layer MLP on Spambase (Dua and Graff 2017) with $|H| = 12$ honest clients, most of which have only one class of data. We let $f = 3, p_{\min} = -0.5$ and assume 30 samples per class. Results are shown in Table 3.

**Two Stages** BOBA w.o. stage 1 skips the subspace optimization and uses the subspace initialized with server gradients. It fails to utilize clients’ data, and thus has a worse performance similar to training with server data only. It shows that stage 1 can effectively use clients’ data and thus is necessary. BOBA w.o. stage 2 does not discard any gradients in stage 2. Instead, it averages all projected gradients. It is not robust to large scale Byzantine gradients.

**Optimizer** BOBA-ES uses exhaustive searching instead of alternating optimization to fit the honest subspace, which globally minimizes the trimmed reconstruction loss. We observe that BOBA has performance comparable to BOBA-ES. However, it only conducts TrSVD for $< 3$ times in average in each rounds, compared to $n_f$ for BOBA-ES. We can conclude that (1) both stages in BOBA are necessary to guarantee performance and robustness, and (2) alternation optimization significantly improves the efficiency while introducing only marginal performance loss.

### 7 Conclusion
This paper focuses on Byzantine-robustness in FL with label skewness. We show that existing aggregators suffer from selection bias and increased vulnerability, and propose BOBA to alleviate these problems. We believe robustness has been overlooked in FL, and hope to inspire future works on analyzing robustness with various flexible FL frameworks.

---

### Table 2: Evaluation of robustness (mean Acc % over 10 runs)

| Method     | MNIST   | CIFAR-10 | AG-News |
|------------|---------|----------|---------|
|            | Gaussian| Signflip | Little  |
| Average    | 9.8     | 9.8      | 90.9    | 9.8     | 10.0    | 10.0    | 65.4    | 10.0    | 27.5    | 25.0    | 87.6    | 25.0    |
| CooMed     | 62.1    | 42.4     | 87.3    | 42.4    | 20.7    | 7.7     | 23.6    | 7.7     | 86.0    | 55.2    | 81.5    | 55.2    |
| TrMean     | 88.6    | 57.1     | 83.6    | 57.1    | 54.6    | 13.1    | 29.7    | 13.1    | 88.1    | 62.4    | 85.2    | 62.4    |
| Krum       | 42.4    | 42.4     | 89.7    | 42.4    | 36.1    | 34.7    | 39.1    | 34.7    | 65.8    | 65.7    | 80.4    | 65.7    |
| MKrum      | 90.8    | 89.2     | 90.1    | 89.2    | 70.0    | 54.8    | 63.7    | 54.8    | 88.3    | 77.0    | 86.7    | 77.0    |
| GeoMed     | 90.2    | 78.0     | 89.8    | 78.0    | 69.9    | 46.6    | 42.0    | 42.0    | 88.4    | 76.2    | 83.6    | 76.2    |
| B-Krum     | 73.6    | 77.3     | 89.4    | 73.6    | 55.9    | 56.1    | 36.5    | 36.5    | 88.3    | 73.3    | 87.6    | 73.3    |
| B-MKrum    | 90.6    | 88.8     | 90.1    | 88.8    | 70.3    | 24.3    | 63.4    | 24.3    | 88.3    | 9.3     | 85.9    | 9.3     |
| SelfRej    | 90.7    | 68.2     | 90.2    | 68.2    | 70.2    | 23.3    | 63.3    | 23.3    | 88.3    | 24.8    | 86.5    | 24.8    |
| AvgRej     | 9.8     | 90.7     | 89.6    | 9.8     | 10.0    | 68.3    | 64.1    | 10.0    | 40.8    | 87.2    | 84.9    | 40.8    |
| BOBA       | 90.7    | 90.5     | 90.7    | 90.5    | 69.9    | **68.8**| **68.2**| **68.2**| **88.3**| **87.9**| **88.4**| **87.9**|

### Table 3: Ablation study (mean Acc % over 10 runs)

| Method     | $|S| = 0$ | $|S| = 3$ |
|------------|----------|----------|
|            | Gaussian | Signflip | Little  |
| Average    | 93.6     | 83.3     | 60.6    | 93.5    |
| Server     | 84.2     | 84.2     | 84.2    | 84.2    |
| BOBA       | 93.4     | **93.6** | 93.0    | 93.4    |
| BOBA-ES    | 93.3     | **93.6** | **93.3**| **93.5**|
| BOBA w.o. stage 1 | 84.6 | 85.6 | 85.5 | 85.1 |
| BOBA w.o. stage 2 | 93.4 | 57.5 | 39.4 | 93.4 |
References

Baruch, G.; Baruch, M.; and Goldberg, Y. 2019. A Little Is Enough: Circumventing Defenses For Distributed Learning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 8632–8642.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Chen, Y.; Su, L.; and Xu, J. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. Proc. ACM Meas. Anal. Comput. Syst., 1(2).

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.

Fang, M.; Cao, X.; Jia, J.; and Gong, N. Z. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In Capkun, S.; and Roesner, F., eds., 29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020, 1605–1622. USENIX Association.

Ghosh, A.; Hong, J.; Yin, D.; and Ramchandran, K. 2019. Robust Federated Learning in a Heterogeneous Environment. CoRR, abs/1906.06629.

Halko, N.; Martinsson, P. G.; and Tropp, J. A. 2011. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Review, 53(2): 217–288.

Hu, Z.; Shaloudegi, K.; Zhang, G.; and Yu, Y. 2020. FedMGDA+: Federated Learning meets Multi-objective Optimization. CoRR, abs/2006.11489.

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

Karimireddy, S. P.; He, L.; and Jaggi, M. 2022. Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing. In International Conference on Learning Representations.

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

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

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and Robust Federated Learning Through Personalization. In Meila, M.; and Zhang, T., eds., 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, 6357–6368. PMLR.

Li, X.; and Zhan, D. 2021. FedRS: Federated Learning with Restricted Softmax for Label Distribution Non-IID Data. In Zhu, F.; Ooi, B. C.; and Miao, C., eds., KDD ’21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, 995–1005. ACM.

Lyu, L.; Yu, H.; Ma, X.; Sun, L.; Zhao, J.; Yang, Q.; and Yu, P. S. 2020. Privacy and Robustness in Federated Learning: Attacks and Defenses. CoRR, abs/2012.06337.

Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, I.; and Vidal, R. 2021. Federated Multi-Task Learning under a Mixture of Distributions. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 15434–15447.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, X. J., eds., Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, volume 54 of Proceedings of Machine Learning Research, 1273–1282. PMLR.

Peng, J.; Wu, Z.; and Ling, Q. 2020. Byzantine-Robust Variance-Reduced Federated Learning over Distributed Non-i.i.d. Data. CoRR, abs/2009.08161.

Pillutla, K.; Kakade, S. M.; and Harchaoui, Z. 2022. Robust Aggregation for Federated Learning. IEEE Transactions on Signal Processing, 70: 1142–1154.

Shen, Z.; Cervino, J.; Hassani, H.; and Ribeiro, A. 2022. An Agnostic Approach to Federated Learning with Class Imbalance. In International Conference on Learning Representations.

Stewart, G. W. 1990. Perturbation Theory for the Singular Value Decomposition. Technical report, USA.

Wu, Z.; Ling, Q.; Chen, T.; and Giannakis, G. B. 2020. Federated Variance-Reduced Stochastic Gradient Descent With Robustness to Byzantine Attacks. IEEE Transactions on Signal Processing, 68: 4583–4596.

Xie, C.; Koyejo, O.; and Gupta, I. 2019. Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation. In Globerson, A.; and Silva, R., eds., Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019, volume 115 of Proceedings of Machine Learning Research, 261–270. AUAI Press.
Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.*, 10(2).

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5650–5659. PMLR.

Yoon, T.; Shin, S.; Hwang, S. J.; and Yang, E. 2021. 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.

Yu, F. X.; Rawat, A. S.; Menon, A. K.; and Kumar, S. 2020. 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*, 10946–10956. PMLR.

Zhang, L.; Shen, B.; Barnawi, A.; Xi, S.; Kumar, N.; and Wu, Y. 2021. FedDPGAN: Federated Differentially Private Generative Adversarial Networks Framework for the Detection of COVID-19 Pneumonia. *Inf. Syst. Frontiers*, 23(6): 1403–1415.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated Learning with Non-IID Data. *CoRR*, abs/1806.00582.
**BOBA: Byzantine-Robust Federated Learning with Label Skewness**

**Technical Appendix**

## Contents

### A Theoretical Analysis

- A.1 Convergence Analysis ................................................................. 2
- A.2 Preliminary: Affine Subspace, Affine Projection, and Their Properties .................................................... 4
- A.3 Notation .................................................................................. 6
- A.4 Assumptions ........................................................................... 7
- A.5 Robustness of Stage 1 ................................................................. 8
- A.6 Robustness of Stage 2 ................................................................. 12
- A.7 Gradient Estimation Error without Accepted Byzantines ............. 15
- A.8 Gradient Estimation Error with Accepted Byzantines ................. 17
- A.9 Gradient Estimation Error of Baselines, and Optimal Order Robustness .................................................... 19

### B Experiments

- B.1 Experimental Setup .................................................................... 20
- B.2 Preliminary Experiments ............................................................... 21
- B.3 Omitted Error Bars ..................................................................... 22
- B.4 Baseline Aggregators Leveraging Server Data .............................. 23
- B.5 Hyper-parameter Sensitivity ....................................................... 24
- B.6 Extension to FedAvg ................................................................. 25
A Theoretical Analysis

A.1 Convergence Analysis

This section gives classical convergence analysis connecting convergence to the gradient estimation error. The convergence analysis provided here is similar to (Chen, Su, and Xu 2017) with some modification. It shows that when the loss is strongly

convex and smooth, where there is an unique optimal parameter \( w^* \), the parameter estimation error \( ||w - w^*||^2 \) converge exponentially fast with error \( O(\Delta) \), where \( \Delta \) is the gradient estimation error.

**Definition 1** (\( \mu \)-strong convexity). A function \( f : \mathbb{R}^d \to \mathbb{R} \) is \( \mu \)-strongly convex if for all \( x, y \in \mathbb{R}^d \),

\[
    f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2} ||x-y||^2
\]

**Definition 2** (\( L \)-smoothness). A function \( f : \mathbb{R}^d \to \mathbb{R} \) is \( L \)-smooth if for all \( x, y \in \mathbb{R}^d \),

\[
    ||\nabla f(x) - \nabla f(y)||_2 \leq L||x-y||_2
\]

**Lemma 1** (Convergence without gradient estimation error). With \( L \)-smooth and \( \mu \)-strongly convex expectation risk function \( L(w) \), conducting GD with real gradient \( \nabla L(w) \) with no estimation error, and step size \( \eta = \frac{\mu}{L^2} \), after \( t \) steps,

\[
    ||w(t) - w^*||_2 \leq \left( 1 - \frac{\mu^2}{L^2} \right)^\frac{t}{2} ||w(0) - w^*||_2
\]

where \( w^* \) is the optimal model parameter.

**Proof.** As \( L(w) \) is \( \mu \)-strongly convex, for all \( w^{(t)} \),

\[
    L\left(w^{(t)}\right) \geq L\left(w^*\right) + \nabla L\left(w^*\right)^T\left(w^{(t)} - w^*\right) + \frac{\mu}{2} \left||w^* - w^{(t)}\right||^2_2
\]

Adding up,

\[
    0 \geq \left(-\nabla L\left(w^{(t)}\right) - \nabla L\left(w^*\right)\right)^T\left(w^{(t)} - w^*\right) + \mu \left||w^* - w^{(t)}\right||^2_2
\]

As \( w^* \) is the optimal solution, \( \nabla L\left(w^*\right) = 0 \), thus

\[
    \nabla L\left(w^{(t)}\right)^T\left(w^{(t)} - w^*\right) \geq \mu \left||w^* - w^{(t)}\right||^2_2
\]

Then we consider each step of the gradient descent

\[
    ||w^{(t+1)} - w^*||_2^2 = ||w^{(t)} - \eta \nabla L\left(w^{(t)}\right) - w^*||_2^2
\]

\[
    = ||w^{(t)} - w^*||_2^2 - 2\eta \nabla L\left(w^{(t)}\right)^T\left(w^{(t)} - w^*\right) + \eta^2 \left||\nabla L\left(w^{(t)}\right)\right||^2_2
\]

\[
    \leq ||w^{(t)} - w^*||_2^2 - 2\eta \mu \left||w^* - w^{(t)}\right||_2^2 + \eta^2 \left||\nabla L\left(w^{(t)}\right)\right||^2_2
\]

\[
    \leq ||w^{(t)} - w^*||_2^2 - 2\eta \mu \left||w^* - w^{(t)}\right||_2^2 + \eta^2 L^2 \left||w^{(t)} - w^*\right||_2^2
\]

\[
    = \left(1 - 2\eta \mu + \eta^2 L^2\right) ||w^{(t)} - w^*||_2^2
\]

\[
    = \left(1 - \frac{\mu^2}{L^2}\right) ||w^{(t)} - w^*||_2^2
\]

\[
    \leq \left(1 - \frac{\mu^2}{L^2}\right)^{t+1} ||w^{(0)} - w^*||_2^2
\]

Equivalently,

\[
    ||w^{(t)} - w^*||_2 \leq \left(1 - \frac{\mu^2}{L^2}\right)^\frac{t}{2} ||w^{(0)} - w^*||_2
\]

\( \square \)
Proposition 2 (Convergence with gradient estimation error). With $L$-smooth and $\mu$-strongly convex population risk $\mathcal{L}(w)$, conducting SGD with gradient $\hat{\mu}$ with estimation error $\|\hat{\mu} - E\mu\|_2 \leq \Delta$, and step size $\frac{\mu}{L^2}$, after $t$ steps,

$$\|w^{(t)} - w^*\|_2 \leq \left(1 - \frac{\mu^2}{L^2}\right)^\frac{t}{2} \|w^{(0)} - w^*\|_2 + \frac{\mu}{L^2 - L\sqrt{L^2 - \mu^2}} \Delta$$

where $w^*$ is the optimal model parameter.

Proof. Similarly as before,

$$\|w^{(t+1)} - w^*\|_2 = \|w^{(t)} - \eta \hat{g}(w^{(t)}) - w^*\|_2$$

$$= \|w^{(t)} - \eta \nabla \mathcal{L}(w^{(t)}) - w^* + \eta \left(\nabla \mathcal{L}(w^{(t)}) - \hat{g}(w^{(t)})\right)\|_2$$

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

$$\leq \sqrt{1 - \frac{\mu^2}{L^2}} \|w^{(t)} - w^*\|_2 + \eta \Delta$$

(Lemma 1)

By induction,

$$\|w^{(t)} - w^*\|_2 \leq \left(1 - \frac{\mu^2}{L^2}\right)^\frac{t}{2} \|w^{(0)} - w^*\|_2 + \sum_{\tau=0}^{t-1} \left(\sqrt{1 - \frac{\mu^2}{L^2}}\right)^\tau \eta \Delta$$

$$\leq \left(1 - \frac{\mu^2}{L^2}\right)^\frac{t}{2} \|w^{(0)} - w^*\|_2 + \frac{1}{1 - \sqrt{1 - \frac{\mu^2}{L^2}}} \frac{\mu}{L^2 - L\sqrt{L^2 - \mu^2}} \Delta$$

$$= \left(1 - \frac{\mu^2}{L^2}\right)^\frac{t}{2} \|w^{(0)} - w^*\|_2 + \frac{\mu}{L^2 - L\sqrt{L^2 - \mu^2}} \Delta$$

Remark. The convergence analysis reveals two important facts.

- With bound gradient estimation error, $\|w - w^*\|$ still converge with exponentially fast.
- The parameter estimation error is proportional to the gradient estimation error, $\|w^{(+\infty)} - w^*\|_2 = O(\Delta)$.

What is more, as $\mathcal{L}$ is $L$-smooth, $\mathcal{L}(w^{(t)}) - \mathcal{L}(w^*) \leq \frac{L}{2} \|w^{(t)} - w^*\|_2^2$. Thus, the increment of expectation risk is bounded, and its order is $O(\Delta^2)$. 

A.2 Preliminary: Affine Subspace, Affine Projection, and Their Properties

We frequently use the properties of affine subspace and affine projection in our proof. For clarity, we formally define these notions and present their useful properties.

Definition 3 (Affine Subspace and Affine Projection). \( \mathcal{P} \) is a \( c \)-dimensional affine subspace in \( \mathbb{R}^d \) if there exists a column-orthogonal \( U \in \mathbb{R}^{d \times c} \) and a bias vector \( m \in \mathbb{R}^d \), s.t.

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

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

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

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

Then, we present useful properties of affine projection.

Property 1. An affine projection maps a point to its nearest point on the subspace. For any affine projection function \( \Pi_\mathcal{P} : \mathbb{R}^d \to \mathbb{R}^d \) and two vectors \( u, v \in \mathbb{R}^d \),

\[
\| \Pi_\mathcal{P}(u) - u \|_2 \leq \| \Pi_\mathcal{P}(v) - u \|_2
\]

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

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

With this result,

\[
\| \Pi_\mathcal{P}(v) - u \|_2 = \sqrt{\| \Pi_\mathcal{P}(v) - \Pi_\mathcal{P}(u) \|_2^2 + \| \Pi_\mathcal{P}(u) - u \|_2^2}
\]

\[\geq \| \Pi_\mathcal{P}(u) - u \|_2 \]

Property 2. An affine projection never lengthen a line segment. For any projection function \( \Pi_\mathcal{P} : \mathbb{R}^d \to \mathbb{R}^d \) and two vectors \( u, v \in \mathbb{R}^d \),

\[
\| \Pi_\mathcal{P}(u) - \Pi_\mathcal{P}(v) \|_2 \leq \| u - v \|_2
\]

Proof.

\[
\begin{align*}
\| \Pi_\mathcal{P}(u) - \Pi_\mathcal{P}(v) \|_2 &= \| [P(u - m) + m] - [P(v - m) + m] \|_2 \\
&= \| P(u - v) \|_2 \\
&\leq \| P \|_2 \| u - v \|_2 \\
&\leq \| u - v \|_2
\end{align*}
\]

Property 3. Affine projection and affine combination are interchangeable. For any projection function \( \Pi_\mathcal{P} : \mathbb{R}^d \to \mathbb{R}^d \), a set of vectors \( \{u_i\}_{i=1}^n \in \mathbb{R}^d \) and coefficients \( \{\lambda_i\}_{i=1}^n \) subject to \( \sum_{i=1}^n \lambda_i = 1 \),

\[
\Pi_\mathcal{P} \left( \sum_{i=1}^n \lambda_i u_i \right) = \sum_{i=1}^n \lambda_i \Pi_\mathcal{P}(u_i)
\]

Proof.

\[
\begin{align*}
\Pi_\mathcal{P} \left( \sum_{i=1}^n \lambda_i u_i \right) &= P \left( \sum_{i=1}^n \lambda_i u_i - m \right) + m \\
&= \sum_{i=1}^n \lambda_i [P(u_i - m) + m] \\
&= \sum_{i=1}^n \lambda_i \Pi_\mathcal{P}(u_i)
\end{align*}
\]
Remark. The sum-to-one constraint on coefficients is crucial as the projection is affine, not linear.

**Property 4.** For any two projection functions \( \Pi_{P_1}, \Pi_{P_2} : \mathbb{R}^d \to \mathbb{R}^d \), a set of vectors \( \{u_i\}_{i=1}^n \in \mathbb{R}^d \), coefficients \( \{\lambda_i\}_{i=1}^n \) subject to \( \sum_{i=1}^n \lambda_i = 1 \) and an affine combination \( v = \sum_{i=1}^n \lambda_i u_i \),

\[
\| \partial v \|_2 \leq \| \partial U \|_2 \cdot \| \lambda \|_2
\]

where \( \lambda = [\lambda_1, \cdots, \lambda_n]^\top \in \mathbb{R}^n \),\( \partial v = \Pi_{P_1}(v) - \Pi_{P_2}(v) \in \mathbb{R}^d \) and \( \partial U = [\Pi_{P_1}(u_1) - \Pi_{P_2}(u_1), \cdots, \Pi_{P_1}(u_n) - \Pi_{P_2}(u_n)] \in \mathbb{R}^{d \times n} \).

**Proof.**

\[
\| \partial v \|_2 = \| \Pi_{P_1}(v) - \Pi_{P_2}(v) \|_2 \\
= \left\| \Pi_{P_1} \left( \sum_{i=1}^n \lambda_i u_i \right) - \Pi_{P_2} \left( \sum_{i=1}^n \lambda_i u_i \right) \right\|_2 \\
= \left\| \sum_{i=1}^n \lambda_i \Pi_{P_1}(u_i) - \sum_{i=1}^n \lambda_i \Pi_{P_2}(u_i) \right\|_2 \quad \text{(Property 3)} \\
= \left\| \sum_{i=1}^n \lambda_i [\Pi_{P_1}(u_i) - \Pi_{P_2}(u_i)] \right\|_2 \\
= \| \partial U \lambda \|_2 \\
\leq \| \partial U \|_2 \cdot \| \lambda \|_2
\]

\( \square \)

Remark. This lemma shows how to bound the projection error of another vector based on the “basis” vectors. (Strictly speaking, \( \{u_i\}_{i=1}^n \) are not basis, as they can be dependent. In this case, we can find a \( \lambda \) with the smallest norm to get the tightest bound of \( \| \Pi_{P_1}(v) - \Pi_{P_2}(v) \|_2 \).)
### A.3 Notation

We summarize here the notation used in our proof in Table 1.

| Symbol | Description |
|--------|-------------|
| $d$    | dimensionality of model parameters and gradient |
| $c$    | number of classes |
| $n$    | number of clients |
| $\mathcal{H}$ | set of honest clients |
| $|\mathcal{H}|$ | number of honest clients |
| $\mathcal{B}$ | set of Byzantine clients |
| $|\mathcal{B}|$ | real number of Byzantine clients |
| $f$    | declared number of Byzantine clients. The aggregator is robust when $f \geq |\mathcal{B}|$ |
| $g_i$  | gradient uploaded by client $i$, $i = 1, \ldots, n$ |
| $\mu$  | average of all honest gradients, $\mu = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} g_i$. This is the gradient of empirical loss. |
| $\mathbb{E} g_i$ | expectation of honest gradient $g_i$, $i \in \mathcal{H}$. Note that Byzantine gradient does not have expectation. |
| $\mathbb{E} \mu$ | expectation of $\mu$. This is the gradient of population loss. |
| $\mu$  | aggregation result, $\mu = \text{Agg}((g_i)_{i=1}^n)$ |
| $\epsilon$ | upper bound of client inner deviation, formally defined in Assumption 1 |
| $\delta$ | upper bound of client outer deviation, formally defined in Assumption 1 |
| $\epsilon_s$ | upper bound of server inner deviation, formally defined in Assumption 1 |
| $\delta_s$ | upper bound of server outer deviation, formally defined in Assumption 1 |
| $\sigma$ | lower bound of client singular value, formally defined in Assumption 2 |
| $\sigma_s$ | lower bound of server singular value, formally defined in Assumption 2 |
| $\mathcal{P}$ | an affine subspace |
| $\Pi_{\mathcal{P}}$ | an affine projection function |
| $\mathcal{F}(\mathcal{P})$ | $n - f$ gradients used to fit $\mathcal{P}$ (among $\{g_i\}_{i=1}^n$) |
| $\mathcal{N}(\mathcal{P})$ | $n - f$ nearest neighbors of $\mathcal{P}$ (among $\{g_i\}_{i=1}^n$) |
| $\ell_t(\mathcal{P})$ | trimmed reconstruction loss of $\mathcal{P}$, $\ell_t(\mathcal{P}) = \sum_{i \in \mathcal{N}(\mathcal{P})} \|g_i - \Pi_{\mathcal{P}}(g_i)\|^2$ |
| $\mathcal{P}^*$ | the ideal affine subspace. It goes through the expectation of honest gradients $\{\mathbb{E} g_i\}_{i \in \mathcal{H}}$ |
| $\hat{\mathcal{P}}$ | the projection function fitted by BOBA |
| $S$ | $n - 2f$ clients that is both honest and in $n - f$ nearest neighbors of $\hat{\mathcal{P}}$, $S = \{s_1, \ldots, s_{n-2f}\} \subset (\mathcal{H} \cap \mathcal{N}(\hat{\mathcal{P}}))$ |
| $\partial \mathcal{S}$ | matrix of differences between projections to fitted and ideal affine subspaces of expected gradients in $S$, $\partial \mathcal{S} = [\Pi_{\mathcal{P}}(\mathbb{E} g_{s_1}) - \Pi_{\mathcal{P}}(\mathbb{E} g_i), \ldots, \Pi_{\mathcal{P}}(\mathbb{E} g_{s_{n-2f}}) - \Pi_{\mathcal{P}}(\mathbb{E} g_i)] \in \mathbb{R}^{d \times (n-2f)}$ |
| $\Delta g_i$ | difference between (fitted) projection and expectation of honest gradient $g_i$, $i \in \mathcal{H}$, $\Delta g_i = \Pi_{\mathcal{P}}(g_i) - \mathbb{E} g_i$ |
| $\gamma_z$ | server gradient of class $z$, $z = 1, \ldots, c$ |
| $\mathbb{E} \gamma_z$ | expectation of server gradient $\gamma_z$, $z = 1, \ldots, c$ |
| $\Gamma$ | matrix of server gradients, $\Gamma = [\gamma_1, \gamma_c]$ |
| $\mathbb{E} \Gamma$ | matrix of expectations of server gradients, $\mathbb{E} \Gamma = [\mathbb{E} \gamma_1, \ldots, \mathbb{E} \gamma_c]$ |
| $\Pi_{\mathcal{P}}(\Gamma)$ | matrix of projections of server gradients, $\Pi_{\mathcal{P}}(\Gamma) = [\Pi_{\mathcal{P}}(\gamma_1), \ldots, \Pi_{\mathcal{P}}(\gamma_c)]$ |
| $\Delta \Gamma$ | matrix of differences between (fitted) projection and expectation of server gradients, $\Delta \Gamma = \Pi_{\mathcal{P}}(\Gamma) - \mathbb{E} \Gamma = [\Pi_{\mathcal{P}}(\gamma_1) - \mathbb{E} \gamma_1, \ldots, \Pi_{\mathcal{P}}(\gamma_c) - \mathbb{E} \gamma_c]$ |
| $p_i$ | true label distribution of honest client $i \in \mathcal{H}$ |
| $\hat{p}_i$ | estimated label distribution of client $i$ |
| $p_{\min}$ | hyperparameter of BOBA, $p_{\min} < 0$ in our case |
| $\hat{p}_\mathcal{H}$ | average of all estimated label distributions of honest clients, $\hat{p}_\mathcal{H} = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \hat{p}_i$ |
| $\tilde{p}_\mathcal{B}$ | average of all estimated label distributions of Byzantine clients that evading stage 2, $\tilde{p}_\mathcal{B} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \hat{p}_b$ when all Byzantine gradients evade stage 2 |
A.4 Assumptions

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

**Assumption 1** (Bounded deviations).

1. Honest client inner deviation: with large probability $1 - p$ where $p = O\left(\frac{1}{|\mathcal{H}|}\right)$, $\frac{1}{|\mathcal{H}|}\sum_{i \in \mathcal{H}} \|\mathbf{g}_i - \mathbb{E}\mathbf{g}_i\|_2^2 \leq \epsilon^2$.
2. Honest client outer deviation: $\|\mathbb{E}\mathbf{g}_i - \mathbb{E}\mathbf{\mu}\|_2^2 \leq \delta^2, \forall i \in \mathcal{H}$.
3. Server inner deviation: with large probability $1 - p_s$ where $p_s = O\left(\frac{1}{\gamma^2}\right)$, $\frac{1}{\gamma}\sum_{z=1}^c \|\gamma_z - \mathbb{E}\gamma_z\|_2^2 \leq \epsilon_s^2$.
4. Server outer deviation: $\|\mathbb{E}\gamma_z - \mathbb{E}\mathbf{\mu}\|_2^2 \leq \delta_s^2, \forall z = 1, \cdots, c$.

**Remark.** We explain each sub-assumption one by one.

1. Honest client inner deviation: this assumption is a deterministic version of bounded inner variation, defined in (Wu et al. 2020; Peng, Wu, and Ling 2020). Bounded inner variation is stronger than ours, assuming

   $$\mathbb{E}\|\mathbf{g}_i - \mathbb{E}\mathbf{g}_i\|_2^2 \leq C_{\text{inner}}, \quad \forall i \in \mathcal{H}$$

   Bounded client inner deviation can be derived from bounded inner variation. Let $X = \frac{1}{|\mathcal{H}|}\sum_{i \in \mathcal{H}} \|\mathbf{g}_i - \mathbb{E}\mathbf{g}_i\|_2^2$ be a non-negative random variable. Its expectation

   $$\mathbb{E}X = \frac{1}{|\mathcal{H}|}\sum_{i \in \mathcal{H}} \mathbb{E}\|\mathbf{g}_i - \mathbb{E}\mathbf{g}_i\|_2^2 \leq \frac{1}{|\mathcal{H}|}\sum_{i \in \mathcal{H}} C_{\text{inner}} = C_{\text{inner}}$$

   By Markov’s inequality, we can bound the tail probability of $X$,

   $$p = \Pr\left(X > \epsilon^2\right) \leq \Pr\left(X \geq \epsilon^2\right) \leq \frac{\mathbb{E}X}{\epsilon^2} \leq \frac{C_{\text{inner}}}{\epsilon^2} = O\left(\frac{1}{\epsilon^2}\right)$$

2. Honest client outer deviation: this assumption is similar to bounded outer variation, defined in (Wu et al. 2020; Peng, Wu, and Ling 2020), which assumes

   $$\frac{1}{|\mathcal{H}|}\sum_{i \in \mathcal{H}} \|\mathbb{E}\mathbf{g}_i - \mathbb{E}\mathbf{\mu}\|_2^2 \leq C_{\text{outer}}$$

   Comparing to their assumption, we bound $\left\{\|\mathbb{E}\mathbf{g}_i - \mathbb{E}\mathbf{\mu}\|_2^2\right\}_{i=1}^{|\mathcal{H}|}$ in different way. With our assumption, we can better analyze how the aggregation is gradually biased by adding Byzantine clients.

3. Server inner deviation: server gradients are special honest gradients, as the computation process of them is identical. Thus, this assumption rewrites bounded client inner deviation with updated notation.

4. Server outer deviation: similarly, this assumption rewrites bounded client outer deviation with updated notation.

**Assumption 2** (Bounded singular values).

1. 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.** We claim in Section 3 of main text that Condition 1 is necessary to achieve optimal order robustness. It states that any $n - 2f$ expectation of honest gradients $\{\mathbb{E}\mathbf{g}_i\}$ distributed on a $(c - 1)$-simplex, but not on any $(c - 2)$-simplex. Therefore, if we stack these $n - 2f$ gradients together to form a matrix, the rank of this matrix should be $c - 1$, i.e., with $c - 1$ non-zero singular values. Assumption 2 simply re-expresses Condition 1 with an lower bound for these singular values, which is used in later proofs.

**Assumption 3** (Satisfactory optimizer). There exists a projection function $\mathcal{P}'$ fitted by TrSVD on $n - f$ honest gradients with larger or equal trimmed reconstruction loss than $\mathcal{P}$.

**Remark.** This assumption says that the optimizer can find a solution that is not too bad (not necessary the global optima).

- For exhaustive searching (BOBA-ES), it is satisfied by finding the global optima.
- For alternative optimization (BOBA), the initialization is fitted by server gradients, which are computed similarly as honest gradients and thus near the ideal projection function. Thus, it is natural to believe that this assumption is also satisfied. This assumption is also satisfied in practice.
A.5 Robustness of Stage 1

In BOBA stage 1, we project all gradients to an affine subspace close to all honest gradients. In this part, we prove the robustness of stage 1.

- Lemma 2 shows that the projection of any honest gradient is close enough to its expectation. In other words, honest gradients are almost unaffected in Stage 1.
- Lemma 3 shows that server gradients also have the similar property.

**Lemma 2** (Robustness of stage 1). With probability at least $1 - p$, simultaneously for any honest gradient $g_h$,

$$\|\Delta g_h\|_2 \leq \left[1 + 2\sqrt{\frac{1}{n - 2f} + \frac{4\delta^2}{\sigma^2}}\right] \sqrt{|H|} \epsilon$$

where $\Delta g_h := \Pi_P(g_h) - g_h$.

*Intuition.* We first provide the intuition of the proof. We aim to show that ideal affine subspace $P^*$ (where the expectations are) and the fitted affine subspace $\hat{P}$ (where the projections are) are close enough. There are four main steps.

1. Satisfactory optimizer (Assumption 3) ensures that the trimmed reconstruction loss of the fitted affine subspace $\ell_t(\hat{P})$ is small.
2. Among the $n - f$ nearest neighbors of $\hat{P}$, at least $n - 2f$ gradients ($\{g_i\}_{i \in S}$) are honest. We consider their expectations $\{\E g_i\}_{i \in S}$, the (sum of squared) distance between their projections to ideal and fitted affine subspace $\sum_{i \in S} \|\Pi_{P^*}(\E g_i) - \Pi_P(\E g_i)\|_2^2$ is bounded.
3. By Condition 1, the expectation of these $n - 2f$ gradients can affinely span the ideal affine subspace. Thus the expectation of another honest gradient $g_h$ can be affinely expressed by them, with some coefficients. We bound the coefficients with bounded singular values (Assumption 2). By Property 4, the distance between $\E g_h$’s projections to ideal and fitted affine subspace $\|\Pi_{P^*}(\E g_h) - \Pi_P(\E g_h)\|_2$ is also bounded.
4. By Property 2, $\Pi_P(g_h)$ is close to $\Pi_{\hat{P}}(\E g_h)$. Finally we bound $\|\E g_h - \Pi_{\hat{P}}(\E g_h)\|_2$ with triangle inequality.

*Proof.* We follow the five main steps mentioned in intuition.

1. We first bound the trimmed reconstruction loss of the fitted affine subspace.

$$\ell_t(\hat{P}) \leq \ell_t(P')$$

(Assumption 3)

$$= \sum_{i \in \mathcal{N}(P')} \|\Pi_{P^*}(g_i) - g_i\|_2^2$$

$$\leq \sum_{i \in \mathcal{F}(P')} \|\Pi_{P^*}(g_i) - g_i\|_2^2$$

(Minimization of trimmed reconstruction loss)

$$\leq \sum_{i \in \mathcal{F}(P')} \|\Pi_{P^*}(\E g_i) - g_i\|_2^2$$

(Optimality of SVD)

$$\leq \sum_{i \in \mathcal{F}(P')} \|\Pi_{P^*}(\E g_i) - \E g_i\|_2^2$$

(Property 1)

$$= \sum_{i \in \mathcal{F}(P')} \|\E g_i - g_i\|_2^2$$

(Ideal projection)

$$\leq \sum_{i \in H} \|\E g_i - g_i\|_2^2$$

($\mathcal{F}(P') \subset H$)

$$\leq |H|\epsilon^2$$

(Assumption 1)

2. Among $\mathcal{N}(\hat{P})$ (the $n - f$ nearest neighbors of $\hat{P}$), at least $n - 2f$ gradients ($g_{s_1}, g_{s_2}, \cdots, g_{s_{n-2f}}$) are honest. Since $S = \{s_1, s_2, \cdots, s_{n-2f}\} \subset \mathcal{N}(\hat{P})$,

$$\sum_{i \in S} \|\Pi_{\hat{P}}(g_i) - g_i\|_2^2 \leq \sum_{i \in \mathcal{N}(\hat{P})} \|\Pi_{\hat{P}}(g_i) - g_i\|_2^2 = \ell_t(\hat{P}) \leq |H|\epsilon^2$$

...
For any honest client $i$, $g_i$ can be seen as a perturbation of $E g_i$. Thus,

$$\sum_{i \in S} \left\| P_i (E g_i) - P_i^* (E g_i) \right\|^2 \leq \sum_{i \in S} \left\| P_i (g_i) - E g_i \right\|^2$$

(Property 2)

$$\leq \sum_{i \in S} \left( \left\| P_i (g_i) - g_i \right\|^2 + \left\| g_i - E g_i \right\|^2 \right)$$

$$\leq \sum_{i \in S} \left( 2 \left\| P_i (g_i) - g_i \right\|^2 + 2 \left\| g_i - E g_i \right\|^2 \right)$$

$$= 2 \sum_{i \in S} \left\| P_i (g_i) - g_i \right\|^2 + 2 \sum_{i \in S} \left\| g_i - E g_i \right\|^2$$

$$= 4 |\mathcal{H}| \epsilon^2$$

(Assumption 1)

Define $\partial S = [P_i (E g_s) - P_i^* (E g_s), \ldots, P_i (E g_{s_n-2f}) - P_i^* (E g_{s_n-2f})] \in \mathbb{R}^{d \times (n-2f)}$, we have

$$\| \partial S \|^2 = \sqrt{\sum_{i \in S} \left\| P_i (E g_i) - P_i^* (E g_i) \right\|^2} = 2 \sqrt{|\mathcal{H}|} \epsilon$$

3. By Condition 1, the expectation of these $n - 2f$ gradients can affinely span the ideal affine subspace. Thus the expectation of another honest gradient $g_h$ can be affinely expressed by them, i.e., there exists $\lambda = [\lambda_1, \ldots, \lambda_{n-2f}]^T \in \mathbb{R}^{n-2f}$, s.t.,

$$E g_h = \sum_{i=1}^{n-2f} \lambda_i E g_s, \quad \sum_{i=1}^{n-2f} \lambda_i = 1$$

With bounded singular values (Assumption 2), we can bound the norm of $\lambda$. Although the solution of $\lambda$ is usually not unique (e.g. when $n - 2f > c$), we only need one solution with small norm. We first “centralize” gradients. Let $m_s = \frac{1}{n-2f} \sum_{s=1}^{n-2f} E g_s$, the centralized linear system is

$$E g_h - E m_s = \sum_{i=1}^{n-2f} \theta_i (E g_s_i - E m_s) = A_s \theta$$

where $A_s = [E g_s_i - E m_s, \ldots, E g_{s_n-2f} - E m_s] \in \mathbb{R}^{d \times (n-2f)}$. One solution of $\theta$ is $\theta = A_s^+ (E g_h - E m_s)$, where $A_s^+$ is the Moore-Penrose inverse of $A_s$. The norm of this solution is bounded by

$$\| \theta \|^2 = \| \theta \|^2 = \| A_s^+ (E g_h - E m_s) \|^2$$

$$\leq \| A_s^+ \|^2 \cdot \| E g_h - E m_s \|^2$$

$$= \frac{1}{\| A_s \|^2} \| E g_h - E m_s \|^2$$

$$\leq \frac{1}{\sigma} \| E g_h - E m_s \|^2$$

(Assumption 2)

$$\leq \frac{1}{\sigma} \left\| (E g_h - E \mu) - \frac{1}{n-2f} \sum_{i=1}^{n-2f} (E g_s_i - E \mu) \right\|^2$$

$$\leq \frac{1}{\sigma} \left( \| E g_h - E \mu \|^2 + \frac{1}{n-2f} \sum_{i=1}^{n-2f} \| E g_s_i - E \mu \|^2 \right)$$

$$\leq \frac{2 \delta}{\sigma}$$

(Assumption 1)

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

$$\lambda = \frac{1}{n-2f} I + \left( I - \frac{1}{n-2f} 1 1^T \right) \theta$$
Therefore,
\[
\|\lambda\|_2 = \left\| \frac{1}{n-2f} 1 + \left( I - \frac{1}{n-2f} 11^\top \right) \theta \right\|_2
\]
\[
= \sqrt{\left\| \frac{1}{n-2f} 1 \right\|_2^2 + \left\| \left( I - \frac{1}{n-2f} 11^\top \right) \theta \right\|_2^2}
\]
\[
\leq \sqrt{\left\| \frac{1}{n-2f} 1 \right\|_2^2 + \left\| I - \frac{1}{n-2f} 11^\top \right\|_2^2 \cdot \|\theta\|_2^2}
\]
\[
\leq \sqrt{\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}}
\]

Since then we construct a \(\lambda\) satisfying the condition of Property 4. Therefore
\[
\|\Pi_P(\mathbb{E}g_h) - \Pi_{P^*}(\mathbb{E}g_h)\| \leq \|\partial S\|_2 \cdot \|\lambda\|_2
\]
\[
\leq 2 \sqrt{\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2} \cdot \sqrt{|H|/\epsilon}}
\] (Property 4)

4. Finally, we again use the property that \(g_h\) is close to \(\mathbb{E}g_h\).
\[
\|\Delta g_h\|_2 = \|\Pi_P(g_h) - \mathbb{E}g_h\|_2
\]
\[
= \|\Pi_P(g_h) - \Pi_{P^*}(\mathbb{E}g_h)\|_2
\]
\[
\leq \|\Pi_P(g_h) - \Pi_P(\mathbb{E}g_h)\|_2 + \|\Pi_P(\mathbb{E}g_h) - \Pi_{P^*}(\mathbb{E}g_h)\|_2
\]
\[
\leq \sum_{i \in \mathcal{H}} \|g_h - \mathbb{E}g_h\|_2^2 + \|\Pi_P(\mathbb{E}g_h) - \Pi_{P^*}(\mathbb{E}g_h)\|_2
\]
\[
\leq \sqrt{|H|/\epsilon} + 2 \sqrt{\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2} \cdot \sqrt{|H|/\epsilon}}
\]
\[
= \left[ 1 + 2 \sqrt{\frac{1}{n-2f} + \frac{4\delta^2}{\sigma^2}} \right] \sqrt{|H|/\epsilon}
\] (Assumption 1)

**Remark.** Lemma 2 shows that \(\|\Delta g_h\|_2 \to 0\) when inner deviation \(\epsilon \to 0\), even when outer deviation \(\delta\) is large and with the existence of Byzantine clients. In other words, \(\mathcal{P} \to \mathcal{P}^*\). Consider its order,

- When \(\frac{1}{n-2f} = O\left(\frac{1}{n}\right), \frac{\delta^2}{\sigma^2} = O\left(\frac{1}{n}\right)\) and \(|\mathcal{H}| = O(n)\), we have \(\|\Delta g_h\|_2 = O(\sqrt{n}\epsilon)\).
- If additionally \(\|g_h - \mathbb{E}g_h\|_2 = O(\epsilon)\), the order can be further tightened to \(\|\Delta g_h\|_2 = O(\epsilon)\).
Lemma 3 (Robustness of Stage 1 for server gradients). With probability as least $1 - p - p_s$, for server gradients $\gamma_1, \cdots, \gamma_c$,

$$\|\Delta \Gamma\|_2 \leq \sqrt{2c\varepsilon_s^2 + 8 \left[ \frac{1}{n - 2f} + \frac{(\delta + \delta_s)^2}{\sigma^2} \right]} \cdot |\mathcal{H}|\varepsilon^2$$

where $\Delta \Gamma = [\Pi_{\hat{P}}(\gamma_1) - E\gamma_1, \cdots, \Pi_{\hat{P}}(\gamma_c) - E\gamma_c] \in \mathbb{R}^{d \times c}$.

**Intuition.** The proof of Lemma 3 is adapted from Lemma 2. We only need to replace some notations and express the bound in matrix form.

**Proof.** We first follow the first three steps in the proof of Lemma 2, replacing $g_h$ with $\gamma_z$ for $z = 1, \cdots, c$. Particularly, in step 3, we make one difference:

$$\|E\gamma_z - Em_s\|_2 \leq \left\| (E\gamma_z - E\mu) - \frac{1}{n - 2f} \sum_{i=1}^{n-2f} (Eg_{s_i} - E\mu) \right\|_2$$

$$\leq \|E\gamma_z - E\mu\|_2 + \frac{1}{n - 2f} \sum_{i=1}^{n-2f} \|Eg_{s_i} - E\mu\|_2$$

$$\leq \delta_s + \delta$$  \hspace{1cm} \text{(Assumption 1)}

As a result, for $z = 1, \cdots, c$,

$$\left\| \Pi_{\hat{P}}(E\gamma_z) - \Pi_{\hat{P}^*}(E\gamma_z) \right\| \leq 2\sqrt{\frac{1}{n - 2f} + \frac{(\delta + \delta_s)^2}{\sigma^2}} \cdot \sqrt{|\mathcal{H}|}\varepsilon$$

Finally,

$$\|\Delta \Gamma\|_2 \leq \|\Delta \Gamma\|_F$$

$$= \sqrt{\sum_{z=1}^{c} \left\| \Pi_{\hat{P}}(\gamma_z) - E\gamma_z \right\|_2^2}$$

$$= \sqrt{\sum_{z=1}^{c} \left\| \Pi_{\hat{P}}(\gamma_z) - \Pi_{\hat{P}^*}E\gamma_z \right\|_2^2}$$

$$= \sqrt{\sum_{z=1}^{c} \left\| \Pi_{\hat{P}}(\gamma_z) - \Pi_{\hat{P}^*}(E\gamma_z) + \Pi_{\hat{P}^*}(E\gamma_z) - \Pi_{\hat{P}^*}(E\gamma_z) \right\|_2^2}$$

$$\leq \sqrt{\sum_{z=1}^{c} 2 \left[ \left\| \Pi_{\hat{P}}(\gamma_z) - \Pi_{\hat{P}}(E\gamma_z) \right\|_2^2 + \left\| \Pi_{\hat{P}}(E\gamma_z) - \Pi_{\hat{P}^*}(E\gamma_z) \right\|_2^2 \right]}$$

$$\leq \sqrt{\sum_{z=1}^{c} 2 \left[ \|\gamma_z - E\gamma_z\|_2^2 + \left\| \Pi_{\hat{P}}(E\gamma_z) - \Pi_{\hat{P}^*}(E\gamma_z) \right\|_2^2 \right]}$$  \hspace{1cm} \text{(Property 2)}

$$\leq \sqrt{2c\varepsilon_s^2 + 8c \left[ \frac{1}{n - 2f} + \frac{(\delta + \delta_s)^2}{\sigma^2} \right]} \cdot |\mathcal{H}|\varepsilon^2$$

$\square$
A.6 Robustness of Stage 2

In BOBA stage 2, we adjust the value of the hyper-parameter $p_{\min}$ such that all honest gradients will be accepted. To ensure it, we need $|p_{\min}| \geq \|p_h - p_{\hat{h}}\|_2$, where $p_h$ is the true label distribution and $p_{\hat{h}}$ is the estimated label distribution, for a honest client $h \in \mathcal{H}$. In this part, we claim that this goal is achievable, by given an upper bound of $\|\hat{p}_h - p_h\|_2$ for any honest client $h \in \mathcal{H}$.

Our proof use Lemma 4, a perturbation bound of singular values.

**Lemma 4** (Weyl’s perturbation bound for singular values). Let $A$ be a matrix with singular value $\sigma_1 \geq \cdots \geq \sigma_n$ and $A = A + \Delta A$ be a perturbation of $A$, with corresponding singular value $\hat{\sigma}_1, \cdots, \hat{\sigma}_n$, we have

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

**Proof.** See (Stewart 1990).

We re-introduce some useful notation. Let

$$\mathbb{E}\Gamma = [\mathbb{E}\gamma_1, \cdots, \mathbb{E}\gamma_c] \in \mathbb{R}^{d \times c}$$

$$\Pi_{\hat{p}}(\Gamma) = [\Pi_{\hat{p}}(\gamma_1), \cdots, \Pi_{\hat{p}}(\gamma_c)] \in \mathbb{R}^{d \times c}$$

$$\Delta \Gamma = \Pi_{\hat{p}}(\Gamma) - \mathbb{E}\Gamma = [\Pi_{\hat{p}}(\gamma_1) - \mathbb{E}\gamma_1, \cdots, \Pi_{\hat{p}}(\gamma_c) - \mathbb{E}\gamma_c] \in \mathbb{R}^{d \times c}$$

$$\Delta g_h = \Pi_{\hat{p}}(g_h) - \mathbb{E}g_h \in \mathbb{R}^d$$

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

$$\mathbb{E}g_h = (\mathbb{E}\Gamma)p_h, \quad \Pi_{\hat{p}}(g_h) = \Pi_{\hat{p}}(\Gamma)\hat{p}_h$$

**Lemma 5** (Robustness of Stage 2). With probability at least $1 - p - p_s$, simultaneously for any honest gradient $g_h$,

$$\|\Delta p_h\|_2 \leq \frac{1}{\sigma_s - \|\Delta \Gamma\|_2} \cdot \left[\|\Delta g_h\|_2 + \sqrt{2}\|\Delta \Gamma\|_2\right]$$

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

**Intuition.** True label distribution $p_h$ and estimated label distribution $\hat{p}_h$ are solutions of two linear systems with the same form and slightly different coefficients ($\{\mathbb{E}\Gamma, \mathbb{E}g_h\}$ and $\{\Pi_{\hat{p}}(\Gamma), \Pi_{\hat{p}}(g_h)\}$). We use standard perturbation analysis of linear system to bound the perturbation of $p_h$.

**Proof.** There are two linear systems under consideration:

$$\begin{align*}
(\mathbb{E}\Gamma)p_h &= \mathbb{E}g_h, & 1^\top p_h &= 1 \\
(\Pi_{\hat{p}}(\Gamma))\hat{p}_h &= \Pi_{\hat{p}}(g_h), & 1^\top \hat{p}_h &= 1
\end{align*}$$

(System 1) (System 2)

Different from solving the affine combination at step 3 of Lemma 2, the solutions here to both linear systems are *unique*. Therefore, we can use any method to express $\Delta p_h = \hat{p}_h - p_h$ 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 $p_h$ and $\hat{p}_h$, thus the bound is valid.

We first centralized both systems with the average of (projections or expectations of) server gradients to remove the affine constraint. Let

$$A = \mathbb{E}\Gamma \left( I - \frac{1}{c} 11^\top \right)$$

$$A = \Pi_{\hat{p}}(\Gamma) \left( I - \frac{1}{c} 11^\top \right)$$

$$\Delta A = \hat{A} - A$$

$$b = \mathbb{E}g_h - \mathbb{E}\Gamma \cdot \frac{1}{c}$$

$$b = \Pi_{\hat{p}}(g_h) - \Pi_{\hat{p}}(\Gamma) \cdot \frac{1}{c}$$

$$\Delta b = \hat{b} - b$$

Previously, we have bounded $\|\Delta g_h\|_2$ and $\|\Delta \Gamma\|_2$ in Lemma 2 and 3, respectively. We use them to give bounds of $\|\Delta A\|_2$ and
\[ \|\Delta b\|_2. \]

\[ \|\Delta A\|_2 = \left\| \hat{A} - A \right\|_2 \]
\[ = \left\| \Pi_{\hat{\rho}}(\Gamma) \left( I - \frac{1}{c}11^T \right) - \mathbb{E}\Gamma \left( I - \frac{1}{c}11^T \right) \right\|_2 \]
\[ \leq \left\| \Pi_{\hat{\rho}}(\Gamma) - \mathbb{E}\Gamma \right\|_2 \cdot \left\| I - \frac{1}{c}11^T \right\|_2 \]
\[ \leq \left\| \Pi_{\hat{\rho}}(\Gamma) - \mathbb{E}\Gamma \right\|_2 \]
\[ = \|\Delta \Gamma\|_2 \]

and similarly,
\[ \|\Delta b\|_2 = \|\hat{b} - b\|_2 \]
\[ = \left\| \left( \Pi_{\hat{\rho}}(g_h) - \Pi_{\hat{\rho}}(\Gamma) \cdot \frac{1}{c} \right) - \left( \mathbb{E}g_h - \mathbb{E}\Gamma \cdot \frac{1}{c}1 \right) \right\|_2 \]
\[ \leq \left\| \Pi_{\hat{\rho}}(g_h) - \mathbb{E}g_h \right\|_2 + \left\| \Pi_{\hat{\rho}}(\Gamma) - \mathbb{E}\Gamma \right\|_2 \cdot \left\| \frac{1}{c}1 \right\|_2 \]
\[ = \|\Delta g_h\|_2 + \frac{1}{\sqrt{c}}\|\Delta \Gamma\|_2 \]

Then, instead of the original systems, we analyze the centralized systems
\[ Ax = b, \quad A\hat{x} = \hat{b} \]
with
\[ x = p_h - \frac{1}{c}1 \quad \hat{x} = \hat{p}_h - \frac{1}{c}1 \quad \Delta x = \hat{x} - x \]

By standard perturbation analysis of linear system,
\[ \hat{A}\hat{x} - Ax = \hat{b} - b \]
\[ \hat{A}(\Delta x) + (\Delta A)x = \Delta b \]
\[ \hat{A}(\Delta x) = \Delta b - (\Delta A)x \]

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

\[ \hat{A}1 = \Pi_{\hat{\rho}}(\Gamma) \left( I - \frac{1}{c}11^T \right) 1 = \Pi_{\hat{\rho}}(\Gamma) (1 - 1) = 0 \]

But \( \Delta x \) cannot grow in the direction of \( 1 \)
\[ 1^T \Delta x = 1^T \left[ \left( p_h - \frac{1}{c}1 \right) - \left( \hat{p}_h - \frac{1}{c}1 \right) \right] = 1^T\hat{p}_h - 1^Tp_h = 1 - 1 = 0 \]

Thus, we can still bound \( \Delta x \) with the \( (c - 1) \)-th singular value of \( \hat{A} \) (instead of the smallest singular value, 0). We have
\[ \hat{\sigma}_{c-1}\|\Delta x\|_2 \leq \left\| \hat{A}(\Delta x) \right\|_2 \]
\[ = \|\Delta b - (\Delta A)x\|_2 \]
\[ \leq \|\Delta b\|_2 + \|\Delta A\|_2 \cdot \|x\|_2 \]
\[ \|\Delta x\|_2 \leq \frac{1}{\hat{\sigma}_{c-1}} \left( \|\Delta b\|_2 + \|\Delta A\|_2 \cdot \|x\|_2 \right) \]
\[ \| \Delta A \|_2 \text{ and } \| \Delta b \|_2 \text{ are already bounded, we still need to bound } \frac{1}{\hat{\sigma}_{c-1}} \text{ and } \| x \|_2. \]

\[ \hat{\sigma}_{c-1} \] is the \((c - 1)\)-th singular value of \( \hat{A} \), and is perturbed from \( \sigma_{c-1} \), the \((c - 1)\)-th singular value of \( A \). By Assumption 2 and Weyl’s perturbation bound for singular value (Lemma 4)

\[
\hat{\sigma}_{c-1} \geq \sigma_{c-1} - |\hat{\sigma}_{c-1} - \sigma_{c-1}| \\
\geq \sigma_{c-1} - \| \Delta A \|_2 \tag{Lemma 4} \\
\geq \sigma_s - \| \Delta A \|_2 \tag{Assumption 2}
\]

The 2-norm of \( x \) can also be bounded,

\[
\| x \|_2 = \left\| p_h - \frac{1}{c} 1 \right\|_2 \\
= \sqrt{\left( p_h - \frac{1}{c} 1 \right)^\top \left( p_h - \frac{1}{c} 1 \right)} \\
= \sqrt{p_h^\top p_h - \frac{1}{c}} \\
\leq \sqrt{1 - \frac{1}{c}}
\]

Putting everything together, we have

\[
\| \Delta p_h \|_2 = \| \Delta x \|_2 \\
\leq \frac{1}{\hat{\sigma}_{c-1}} \cdot (\| \Delta b \|_2 + \| \Delta A \|_2 \cdot \| x \|_2) \\
\leq \frac{1}{\sigma_s - \| \Delta \Gamma \|_2} \cdot \left( \| \Delta g_h \|_2 + \frac{1}{\sqrt{c}} \| \Delta \Gamma \|_2 + \sqrt{1 - \frac{1}{c}} \| \Delta \Gamma \|_2 \right) \\
= \frac{1}{\sigma_s - \| \Delta \Gamma \|_2} \cdot \left[ \| \Delta g_h \|_2 + \left( \sqrt{\frac{1}{c}} + \sqrt{1 - \frac{1}{c}} \right) \| \Delta \Gamma \|_2 \right] \\
\leq \frac{1}{\sigma_s - \| \Delta \Gamma \|_2} \cdot \left[ \| \Delta g_h \|_2 + \sqrt{2} \| \Delta \Gamma \|_2 \right]
\]

when \( \sigma_s - \| \Delta \Gamma \|_2 > 0. \) \[ \Box \]

**Remark.** We consider the case where \( \| \Delta g_h \|_2 = O(\epsilon) \) and \( \| \Delta \Gamma \|_2 = O(\sqrt{c}\epsilon) \) (see remarks of Lemma 2 and 3). When the outer deviation dominates the inner deviation, \( \sigma_s \gg \| \Delta \Gamma \|_2 \), thus \( \| \Delta p_h \|_2 = O(\sqrt{c}\epsilon/\sigma_s) \). This means that we can set a small \( |p_{\text{min}}| \) and still preserve all honest gradients.
A.7 Gradient Estimation Error without Accepted Byzantines

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.

In this part, we consider a simple case where all Byzantines are dropped in stage 2. Then the aggregation result of BOBA is

\[ \hat{\mu}_H := \frac{1}{|H|} \sum_{i \in H} \Pi_p(g_i) \]

where \( \hat{\mu}_H \) is the aggregation result when stage 2 perfectly preserve all honest gradients while discard all Byzantines. Since we proved in Lemma 2 that all honest gradients are only slightly perturbed in stage 1, a natural corollary is that the aggregation result will also be close to the real gradient.

**Lemma 6** (Robust aggregation without accepted Byzantines). When all Byzantine gradients are dropped in stage 2, with probability at least \( 1 - p - p_s \),

\[ \| \hat{\mu}_H - \mathbb{E}\mu \|_2 \leq \left[ 1 + 2 \sqrt{\frac{1}{n - 2f} + \frac{\delta^2}{\sigma^2} \sqrt{|H|}} \right] \epsilon \]

**Proof.** We first reformulate the form,

\[ \| \hat{\mu}_H - \mathbb{E}\mu \|_2 = \left\| \frac{1}{|H|} \sum_{i \in H} \Pi_p(g_i) - \mathbb{E}\mu \right\|_2 \]

\[ = \left\| \Pi_p \left( \frac{1}{|H|} \sum_{i \in H} g_i \right) - \mathbb{E}\mu \right\|_2 \quad \text{(Property 3)} \]

Noticing that this formula has the same form as each honest gradient \( g_h \) in Lemma 2. We again follow the first three steps in the proof of Lemma 2, replacing \( g_h \) with \( \mu \). Particularly, in step 3, we make one difference:

\[ \| \mathbb{E}\mu - \mathbb{E}m \|_2 \leq \left\| - \frac{1}{n - 2f} \sum_{i=1}^{n-2f} (\mathbb{E}g_{s_i} - \mathbb{E}\mu) \right\|_2 \]

\[ \leq \frac{1}{n - 2f} \sum_{i=1}^{n-2f} \| \mathbb{E}g_{s_i} - \mathbb{E}\mu \|_2 \]

\[ \leq \delta \quad \text{(Assumption 1)} \]

As a result,

\[ \| \Pi_p(\mathbb{E}\mu) - \Pi_p(\mathbb{E}\mu) \| \leq 2 \sqrt{\frac{1}{n - 2f} + \frac{\delta^2}{\sigma^2}} \cdot \sqrt{|H|} \epsilon \]

Finally, we can get a better bound of \( \mu - \mathbb{E}\mu \) than \( g_h - \mathbb{E}g_h \) in Lemma 2

\[ \| \mu - \mathbb{E}\mu \|_2 = \left\| \frac{1}{|H|} \sum_{i \in H} g_i - \mathbb{E} \left( \frac{1}{|H|} \sum_{i \in H} g_i \right) \right\|_2 \]

\[ = \left\| \frac{1}{|H|} \sum_{i \in H} (g_i - \mathbb{E}g_i) \right\|_2 \]

\[ \leq \frac{1}{|H|} \sum_{i \in H} \| g_i - \mathbb{E}g_i \|_2 \]

\[ \leq \epsilon \quad \text{(Assumption 1)} \]
Plug it into the original step 4,
\[ \| \hat{\mu}_H - \mathbb{E}\mu \|_2 = \| \Pi_{\hat{P}}(\mu) - \mathbb{E}\mu \|_2 \]
\[ = \| \Pi_{\hat{P}}(\mu) - \Pi_{P^*}(\mathbb{E}\mu) \|_2 \]
\[ \leq \| \Pi_{\hat{P}}(\mu) - \Pi_{\hat{P}}(\mathbb{E}\mu) \|_2 + \| \Pi_{\hat{P}}(\mathbb{E}\mu) - \Pi_{P^*}(\mathbb{E}\mu) \|_2 \]
\[ \leq \| \mu - \mathbb{E}\mu \|_2 + \| \Pi_{\hat{P}}(\mathbb{E}\mu) - \Pi_{P^*}(\mathbb{E}\mu) \|_2 \quad \text{(Property 2)} \]
\[ \leq \epsilon + 2 \sqrt{\frac{1}{n - 2f}} + \frac{\delta^2}{\sigma^2} \cdot \sqrt{|H|} \epsilon \]
\[ = \left[ 1 + 2 \sqrt{\frac{1}{n - 2f}} + \frac{\delta^2}{\sigma^2} \sqrt{|H|} \right] \epsilon \]

Remark. We first analyze the order of each term in this formula. When the outer deviation increases \( t \) times, both \( \delta \) and \( \sigma \) increase \( t \) times. When all clients are duplicated, \( \delta^2 \) does not change but \( \sigma^2 \) doubled. Thus generally we have \( \frac{\delta^2}{\sigma^2} \propto \frac{1}{n} \). When \( \frac{1}{n - 2f} = O(\frac{1}{n}) \), \( |H| = O(n) \), we have \( \| \hat{\mu}_H - \mathbb{E}\mu \|_2 = O(\epsilon) \).
A.8 Gradient Estimation Error with Accepted Byzantines

In last part we consider a simple case where all Byzantines are dropped in stage 2. However, in reality, some Byzantines can circumvent BOBA stage 2 (and also any other aggregators) since it can mimic the behavior of honest clients. Thus, we cannot assume that the real aggregation result $\hat{\mu} = \hat{\mu}_H$. Although these attacks are inevitable for any aggregators, the influence of such attacks can be bounded by our algorithm.

- During stage 1, all honest and Byzantine gradients are robustly projected to an affine subspace. Thus, Byzantine gradients can only affect the aggregation along the dimensions of fitted affine subspace $\mathcal{P}$, which has very limited dimensions.
- Then in stage 2, our aggregator estimates the label distribution for each client and drops those with large negative components. Thus, Byzantine clients can only affect the estimated label distribution with limited scales.

**Theorem 3.** With probability at least $1 - p - p_s$, 
\[
\|\hat{\mu} - E\mu\|_2 \leq 1 + 2 \sqrt{\frac{1}{n-2f} + \frac{\epsilon^2}{\sigma^2} \sqrt{|H|}} (\epsilon + \beta 2 (1 + c |p_{\text{min}}|) (\sqrt{\epsilon_s} + \delta_s))
\]

where $\beta = |B|/n$ is the real fraction of Byzantine clients.

**Intuition.** Byzantines can only affect the aggregation via biasing the estimated label distribution. The gradient estimation error can be divided into two parts: $\|\hat{\mu}_H - E\mu\|_2$ and $\|\hat{\mu} - \hat{\mu}_H\|_2$. The first term is already bounded in Lemma 6. The second term is introduced by Byzantines that bias the average label distribution. It is the product of fraction of Byzantines $\beta$, maximum perturbation on label distribution $2(1 + c |p_{\text{min}}|)$, and the maximum distance between server gradients and the true averaged gradient $\sqrt{\epsilon_s} + \delta_s$.

**Proof.** When some Byzantine clients are accepted, they can affect the aggregation result via biasing the average of estimated label distribution. We consider the worst case, where all Byzantines collude and are all accepted. We first divide the gradient estimation error into two parts.

\[
\|\hat{\mu} - E\mu\|_2 \leq \|\hat{\mu}_H - E\mu\|_2 + \|\hat{\mu} - \hat{\mu}_H\|_2
\]

The first term is already bounded in Lemma 6. We further bound the second term. Notice that,

\[
\hat{\mu}_H = \frac{1}{|H|} \sum_{i \in H} \Pi_{P_i}(g_i) = \frac{1}{|H|} \sum_{i \in H} \Pi_{P_i}(\Gamma) \hat{p}_i = \Pi_{P_i}(\Gamma) \left( \frac{1}{|H|} \sum_{i \in H} \hat{p}_i \right)
\]

And similarly,

\[
\hat{\mu} = \Pi_{P_i}(\Gamma) \left( \frac{1}{n} \sum_{i=1}^n \hat{p}_i \right)
\]

We define

\[
\hat{p}_H = \frac{1}{|H|} \sum_{i \in H} \hat{p}_i, \quad \hat{p}_B = \frac{1}{|B|} \sum_{i \in B} \hat{p}_i, \quad \hat{p} = \frac{1}{n} \sum_{i=1}^n \hat{p}_i
\]

Then,

\[
\|\hat{\mu} - \hat{\mu}_H\|_2 = \|\Pi_{P_i}(\Gamma) (\hat{p} - \hat{p}_H)\|_2 = \left\| (\Pi_{P_i}(\Gamma) - \Pi_{P_i}(E\mu) \Gamma) (\hat{p} - \hat{p}_H) \right\|_2
\]

\[
\leq \sum_{i=1}^c (\hat{p} - \hat{p}_H)_z \cdot \left\| \Pi_{P_i}(\Gamma) - \Pi_{P_i}(E\mu) \right\|_2
\]

\[
\leq \|\hat{p} - \hat{p}_H\|_1 \cdot \left( \max_z \left\| \Pi_{P_i}(\Gamma) - \Pi_{P_i}(E\mu) \right\|_2 \right)
\]

We first derive a bound for $\|\hat{p} - \hat{p}_H\|_1$. In Lemma 6, we assume that no Byzantine is accepted in stage 2, thus $\hat{p} = \hat{p}_H$. However, when Byzantines are accepted, $\hat{p}$ is biased from $\hat{p}_H$ to $p_B$. In BOBA stage 2, we accepted a gradient if its estimated label distribution is in a $(c - 1)$-simplex of

\[
\hat{p}_i \in \{ q : q \geq p_{\text{min}}1, 1^\top q = 1 \}
\]
Since \( \hat{p}, \hat{p}_H \) and \( \hat{p}_B \) are averages of some \( \hat{p}_i \) that lie inside the simplex above, we have
\[
\hat{p}, \hat{p}_H, \hat{p}_B \in \{ q : q \geq p_{\min}1, 1^\top q = 1 \}
\]
Therefore,
\[
\| \hat{p} - \hat{p}_H \|_1 = \left\| \frac{|H|}{n} \hat{p}_H + \frac{|B|}{n} \hat{p}_B - \hat{p}_H \right\|_1 \\
= \frac{|B|}{n} \| \hat{p}_B - \hat{p}_H \|_1 \\
\leq \frac{|B|}{n} (\| \hat{p}_B - p_{\min}1 \|_1 + \| \hat{p}_H - p_{\min}1 \|_1) \\
= \frac{|B|}{n} 2(1 + c|p_{\min}|)
\]
Then we derive a bound for \( \max_z \| \Pi_{\hat{p}}(\gamma_z) - \Pi_{\hat{p}}(\mathbb{E}\mu) \|_2 \). For each server gradient \( \gamma_z \),
\[
\| \Pi_{\hat{p}}(\gamma_z) - \Pi_{\hat{p}}(\mathbb{E}\mu) \|_2 \leq \| \gamma_z - \mathbb{E}\gamma \|_2 \\
\leq \| \gamma_z - \mathbb{E}\gamma \|_2 + \| \mathbb{E}\gamma - \mathbb{E}\mu \|_2 \\
\leq \sqrt{c} \epsilon_s + \delta_s \\ \\ \text{(Assumption 1)}
\]
Put all together
\[
\| \hat{\mu} - \mathbb{E}\mu \|_2 \leq \left[ 1 + 2 \sqrt{\frac{1}{n-2f} + \frac{\delta^2}{\sigma^2} \sqrt{|H|}} \right] \epsilon + \beta 2(1 + c|p_{\min}|)(\sqrt{c} \epsilon_s + \delta_s)
\]
where \( \beta = \frac{|B|}{n} \). \( \square \)
A.9 Gradient Estimation Error of Baselines, and Optimal Order Robustness

In this part, we aim to test whether BOBA and baseline aggregators achieve optimal order robustness.

**BOBA** In last part, we verify that BOBA generally achieves optimal order robustness. We first analyze the order of $\frac{\delta^2}{\sigma^2}$ with respect to $n$. We have the following observations

- When all gradients are multiplied by $t$, both $\delta$ and $\sigma$ increase for $t$ times.
- When we replicate each client for $t$ times and evaluate totally $tn$ clients, $\delta^2$ does not change but $\sigma^2$ increases for $t$ times.

Therefore, 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 $|p_{\text{min}}| = \mathcal{O}(1)$, we have $\|\hat{\mu} - E\mu\|_2 = \mathcal{O}(\epsilon + \beta \delta)$. Let $\epsilon \to 0$, we conclude that BOBA has optimal order robustness.

**IID Aggregators** Then we show that IID aggregators cannot achieve optimal order robustness. We only need to consider a special case when there is no Byzantines.

This construction is similar to the example in (Karimireddy, He, and Jaggi 2022). Consider there are $n = |\mathcal{H}| \geq 4$ honest clients distribute in $\mathbb{R}$. A majority $\frac{n}{2} + 1$ of them is $Eg_i = +1$, while a minority $\frac{n}{2} - 1$ of them is $Eg_i = -1$. Thus, the true center should be

$$\frac{1}{n} \left( \left( \frac{n}{2} + 1 \right) - \left( \frac{n}{2} - 1 \right) \right) = \frac{2}{n}$$

In this case, $\epsilon = 0$ and $\delta = 1 + \frac{2}{n} \leq 2$. If an aggregator has optimal order robustness (and thus is unbiased), we expect it to have zero gradient estimation error in this case, since $\epsilon = 0$. However the aggregation result of Krum (Blanchard et al. 2017), GeoMed (Chen, Su, and Xu 2017) and CooMed (Yin et al. 2018) will be $+1$. So the gradient estimation error is $1 - \frac{2}{n} \geq \frac{1}{2}$. Such result shows that IID aggregators cannot achieve optimal order robustness.
B Experiments

B.1 Experimental Setup

This part provides detailed experimental setup. Table 2 summa-
```markdown
| Data partitioning | We use the data partitioning in (McMahan et al. 2017). We first sort data samples based on labels and evenly divided the training set into \( n_s \cdot |H| \) shards. As a result, most of the shards only contain one class of data. 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. For Spambase, as there are only two classes, we let \( n_s = 1 \).

Model

- For Spambase (Dua and Graff 2017), we train a 2-layer MLP with hidden layer width 16.
- For MNIST (Lecun et al. 1998), we train a 3-layer MLP with hidden layer width 200, which is the same as (McMahan et al. 2017).
- For CIFAR-10 (Krizhevsky and Hinton 2009), we train a 5-layer CNN model as it in the tensorflow tutorial.
- For AG News (Zhang, Zhao, and LeCun 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 For Spambase and MNIST, we use constant learning rate. For CIFAR-10 and AG-News, we use 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.
```
B.2 Preliminary Experiments

In this part, we explain how Figure 1 and 2 in main text are generated.

Figure 1: Comparison of aggregation result.

Figure 1: Comparison of aggregation results

Figure 1 compares aggregation results of BOBA with three baseline aggregators: GeoMed (Chen, Su, and Xu 2017), Krum (Blanchard et al. 2017), and CooMed (Yin et al. 2018). The setting is similar to the example in Appendix A.9.

- There are $|H| = 12$ honest clients, whose gradients are in $\mathbb{R}^2$. 7 of them (the majority) distributes around $(1, 0)$, 5 of them (the minority) distributes around $(0, 1)$. Blue crosses show the distribution of honest gradients.
- The green circle shows the true center $\mu_E$, i.e. $\left(\frac{7}{12}, \frac{5}{12}\right)$.

For all aggregators, we set $f = 3$. We consider two scenarios similar to our main experiments.

1. We first consider the case without Byzantines. The aggregation results are shown as the red square. We see that all three IID aggregators output an aggregation near $(1, 0)$, the majority class center. This phenomenon results from selection bias, and bias the model a lot.

2. We then consider the case with Byzantines. We add $|B| = 3$ Byzantine gradients that gridly sampled from $[-2, 3] \times [-2, 3]$. Each choice of Byzantine gradients correspond to an aggregation result, and we plot the 2-D histogram of aggregation results as the orange set. The range of the orange set illustrate how the aggregation result can deviate from the true center: larger radius means larger gradient estimation error. We observe that all three IID aggregators have large gradient estimation error in all directions. Meanwhile, BOBA have nearly no gradient estimation error along $\rightarrow$, and bounded error along $\nearrow$.

Figure 2: PCA of honest gradients on MNIST ($c = 10$)

Figure 2 verifies our finding that honest clients are distributed near a $(c-1)$-dimensional subspace. We generate honest gradients with the same setting as MNIST in B.1, and then conduct principle component analysis (PCA) on the matrix of honest gradients. In each principle component (a.k.a., dimension), larger “Explained Variance” indicates that honest gradients diverge along this direction, while smaller “Explained Variance” indicates that honest gradients cluster. The principle components are sorted with their explained variance in descending order, and then plotted in Figure 2.

- The blue line shows the result of PCA when honest clients have IID data. In this case, the explained variances are small and do not concentrate in any dimension. It verifies our analysis that in IID settings, honest clients distribute around the true center.
- The orange line shows the result of PCA when honest clients have label skewed data. In this case, the explained variances for the first $c-1$ dimensions dramatically increases compared to the IID case. And most of the variance concentrate in the first $c-1$ dimensions. It verifies our analysis that in IID settings, honest clients distribute around a $(c-1)$-dimensional subspace.
### B.3 Omitted Error Bars
In our main text, we omit the error bar of numerical results in Table 1, 2, and 3. We provide here results with standard deviations, respectively, in Table 3, 4, and 5.

Table 3: Evaluation of unbiasedness (mean (s.d.) % over 10 runs)

| Method      | MNIST Acc ↑ | MRD ↓ | CIFAR-10 Acc ↑ | MRD ↓ | AG-News Acc ↑ | MRD ↓ |
|-------------|--------------|-------|-----------------|-------|---------------|-------|
| Average     | **90.7 (0.3)** | -     | **70.2 (0.3)** | -     | **88.3 (0.1)** | -     |
| Server      | 82.4 (0.9)   | 17.9 (2.1) | 23.4 (1.3) | 63.0 (4.2) | 81.2 (1.7) | 12.8 (4.2) |
| CooMed      | 64.7 (6.5)   | 75.4 (13.9) | 19.7 (2.6) | 81.4 (1.8) | 82.9 (2.9) | 10.3 (4.6) |
| TrMean      | 76.7 (1.7)   | 72.3 (12.5) | 29.8 (4.4) | 79.2 (2.4) | 87.0 (0.4) | 5.4 (2.5)    |
| Krum        | 42.4 (2.8)   | 93.9 (1.1) | 32.6 (3.6) | 80.8 (1.5) | 65.6 (2.6) | 89.9 (4.4)   |
| MKrum       | 89.3 (0.8)   | 20.1 (11.3) | 69.7 (0.5) | 9.4 (3.9)  | 88.0 (0.2) | 3.3 (2.1)    |
| GeoMed      | 90.2 (0.1)   | 6.1 (1.5)  | 69.9 (0.6) | 4.1 (1.7)  | **88.3 (0.1)** | 0.5 (0.3)    |
| B-Krum      | 70.0 (1.9)   | 91.5 (2.7) | 55.1 (3.4) | 80.0 (2.4) | 87.6 (0.9) | 4.3 (0.3)    |
| B-MKrum     | 90.3 (0.2)   | 5.2 (1.9)  | 69.8 (0.5) | 4.2 (1.7)  | 88.0 (0.3) | 3.5 (2.2)    |
| SellRej     | 88.9 (0.9)   | 24.8 (13.0) | 69.4 (0.7) | 7.3 (3.3)  | 85.2 (2.0) | 20.3 (11.1) |
| AvgRej      | 88.6 (0.8)   | 26.9 (8.5) | 69.5 (0.7) | 14.7 (5.9) | 84.7 (1.3) | 22.2 (8.2)   |
| BOBA        | 90.5 (0.5)   | **4.5 (5.1)** | 69.6 (0.7) | **2.6 (0.8)** | **88.3 (0.1)** | **0.4 (0.6)** |

Table 4: Evaluation of robustness (mean (s.d.) Acc % over 10 runs)

| Method      | Gaussian | MNIST Signflip | Little | Worst | Gaussian | CIFAR-10 Signflip | Little | Worst | Gaussian | AG-News Signflip | Little | Worst |
|-------------|----------|----------------|--------|-------|----------|-------------------|--------|-------|----------|-------------------|--------|-------|
| Average     | 9.8 (0.0) | 9.8 (0.0) | **90.9 (0.1)** | 9.8 | 10.0 (0.0) | 10.0 (0.0) | 65.4 (0.8) | 10.0 | 27.5 (3.7) | 25.0 (0.0) | 87.6 (0.2) | 25.0 |
| CooMed      | 62.1 (5.2) | 42.4 (5.8) | 87.3 (0.5) | 42.4 | 20.7 (2.4) | 7.7 (2.3) | 23.6 (3.1) | 7.7 | 86.0 (1.1) | 55.2 (6.4) | 81.5 (0.3) | 55.2 |
| TrMean      | 88.6 (0.8) | 57.1 (16.8) | 83.6 (3.2) | 57.1 | 54.6 (1.6) | 13.1 (2.1) | 29.7 (3.2) | 13.1 | 88.1 (0.2) | 62.4 (5.7) | 85.2 (0.2) | 62.4 |
| Krum        | 42.4 (7.4) | 42.4 (7.4) | 89.7 (0.2) | 42.4 | 36.1 (3.9) | 34.7 (2.7) | 39.1 (2.3) | 34.7 | 65.8 (2.2) | 65.7 (2.6) | 80.4 (2.8) | 65.7 |
| MKrum       | 90.8 (0.2) | 89.2 (0.8) | 90.1 (0.7) | 89.2 | 70.0 (0.4) | 54.8 (15.7) | 63.7 (0.7) | 54.8 | 88.3 (0.1) | 77.0 (18.7) | 86.7 (0.2) | 77.0 |
| GeoMed      | 90.2 (0.1) | 78.0 (3.5) | 89.8 (0.6) | 78.0 | 69.9 (0.4) | 46.6 (2.2) | 42.0 (2.7) | 42.0 | 88.4 (0.1) | 76.2 (1.5) | 83.6 (0.2) | 76.2 |
| B-Krum      | 73.6 (5.5) | 77.3 (2.9) | 89.4 (0.6) | 73.6 | 55.9 (1.3) | 56.1 (2.3) | 36.5 (2.0) | 36.5 | 88.3 (0.1) | 73.3 (7.7) | 87.6 (0.4) | 73.3 |
| B-MKrum     | 90.6 (0.2) | 88.8 (1.1) | 90.1 (0.7) | 88.8 | 70.3 (0.5) | 24.3 (1.9) | 63.4 (0.6) | 24.3 | 88.3 (0.2) | 9.3 (5.7) | 85.9 (0.2) | 9.3 |
| SellRej     | 90.7 (0.2) | 68.2 (3.1) | 90.2 (0.2) | 68.2 | 70.2 (0.6) | 23.3 (2.3) | 63.3 (1.0) | 23.3 | 88.3 (0.1) | 24.8 (0.7) | 86.5 (0.2) | 24.8 |
| AvgRej      | 9.8 (0.2) | **90.7 (0.1)** | 89.6 (0.4) | 9.8 | 10.0 (0.0) | 68.3 (1.1) | 64.1 (0.9) | 10.0 | 40.8 (2.4) | 87.2 (1.6) | 84.9 (0.8) | 40.8 |
| BOBA        | 90.7 (0.3) | 90.5 (0.4) | 90.7 (0.2) | **90.5** | 69.9 (0.7) | **68.8 (0.5)** | **68.2 (0.6)** | **68.2** | 88.3 (0.1) | **87.9 (0.2)** | **88.4 (0.1)** | **87.9** |

Table 5: Ablation study (mean (s.d.) Acc % over 10 runs)

| Method      | |\(|\mathcal{B}| = 0\) | |\(|\mathcal{B}| = 3\) | Gaussian | Signflip | Little |
|-------------| | | | | | | |
| Average     | 93.6 (0.5) | 83.3 (5.5) | 60.6 (0.0) | 93.5 (0.3) |
| Server      | 84.2 (2.8) | 84.2 (2.8) | 84.2 (2.8) | 84.2 (2.8) |
| BOBA        | 93.4 (0.3) | 93.6 (0.3) | 93.0 (0.6) | 93.4 (0.3) |
| BOBA-ES     | 93.3 (0.3) | 93.6 (0.3) | 93.3 (0.3) | 93.5 (0.2) |
| BOBA w.o. stage 1 | 84.6 (2.4) | 85.6 (2.0) | 85.5 (1.9) | 85.1 (2.2) |
| BOBA w.o. stage 2 | 93.4 (0.3) | 57.5 (12.7) | 39.4 (0.0) | 93.4 (0.3) |
B.4 Baseline Aggregators Leveraging Server Data

Comparing to IID aggregators and bucketing aggregators, BOBA requires a limited amount of server data in its stage 2. One may question additional server data may introduce advantages to BOBA. It is possible that baseline aggregators may have better performance when they utilize these additional server data.

To make a fair comparison, we design a way to enhance these baseline aggregators. In our analysis, baseline aggregators utilize vast client data, but suffer from selection bias and increased vulnerability. Meanwhile, if we only uses server data to fit a model, the model is unbiased and robust, but cannot generalize well since the amount of server data is small. We design a mixed aggregator to balance between these properties. The mixed aggregator outputs interpolation between baseline aggregators and server aggregator.

\[
\hat{\mu} = (1 - \lambda) \text{Agg}(\{g_i\}_{i=1}^n) + \lambda \left( \frac{1}{c} \sum_{z=1}^{c} \gamma_z \right)
\]

where \( \lambda \) is an interpolating hyper-parameter. \( \lambda = 0 \) indicates baseline aggregators and \( \lambda = 1 \) indicates server aggregator. We test different value of \( \lambda \) and see whether baseline aggregators can use these server data to improve their performance.

![Figure 3: IID aggregators leveraging server data](image)

We test this mixed aggregator with strongest baseline aggregators on MNIST, and show the results in Figure 3. It shows that although some aggregators can benefits from server data, mixed aggregators fail to outperform the best baseline aggregators in both setting without or with Byzantines.
B.5 Hyper-parameter Sensitivity

In this part, we aim to answer how different hyper-parameters influence BOBA and how should we choose them. There are two hyper-parameters in BOBA: the declared number of Byzantines $f$ and the threshold of rejecting Byzantines $p_{\text{min}}$.

Hyper-parameter $f$ in stage 1 In label skewness settings, the declared number of Byzantines $f$ should satisfy $f \geq |B|$ and Condition 1 simultaneously, which give both upper and lower bound of $f$. In this part, we examine on AG-News whether BOBA is sensitive to the choice of $f$, under multiple real number of Byzantines $|B|$.

The results are shown on the left of Figure 4. We observe that

- underestimating $|B|$ (i.e., $f < |B|$) results in a highly broken model, while
- overestimating $|B|$ (i.e. $f > |B|$) does not significantly lower the model performance, unless it is obviously too big, e.g., $f = 100$ when there are only $n = |H| + |B| = 160 + 0 = 160$ clients.

We summarize that BOBA is not sensitive to a wide range of $f$, and we recommend choosing a moderately larger $f$ to avoid underestimating $|B|$, since it is more damaging.

Hyper-parameter $p_{\text{min}}$ and amount of server data in stage 2 $p_{\text{min}}$ is the threshold of estimated label distribution to discard Byzantines. When $|p_{\text{min}}|$ is large (e.g., $p_{\text{min}} = -1.5, -2$), both honest and Byzantine clients will be accepted in stage 2, making the aggregation more vulnerable to Byzantines. Especially when $p_{\text{min}} \to -\infty$, stage 2 will be skipped. When $|p_{\text{min}}|$ is small (e.g., $p_{\text{min}} = 0, -0.1$), some honest clients might be dropped. Proposition 5 also reminds us that the choice of $p_{\text{min}}$ can be related to the amount of server data. Thus, we test different $|p_{\text{min}}|$ with different amounts of server data on CIFAR-10. Different from general experimental setup, we use $|B| = 10$ here to guarantee that some honest gradients will be dropped when we under-estimate $|p_{\text{min}}|$.

The results are shown on the right of Figure 4. We observe that unless we use an extremely small amount of data, underestimating $|p_{\text{min}}|$ only introduces marginal performance loss, while over-estimating $|p_{\text{min}}|$ increases the risk more. Thus we recommend choosing a moderately smaller $|p_{\text{min}}|$. Meanwhile, we also find that although 1 sample per class might be too few, BOBA requires only a few samples (e.g., 5) for each class to achieve unbiasedness and robustness.

Figure 4: Hyper-parameter sensitivity

![Figure 4](image_url)
B.6 Extension to FedAvg

Most previous works of robust aggregators only focus on FedSGD where each client takes one local update. However, FedSGD requires lots of communication rounds. To speed up training, in most prevalent FL framework, e.g. FedAvg (McMahan et al. 2017) and FedOpt (Reddi et al. 2021), clients take multiple local gradient descent steps in each communication rounds. It is important to extend robust aggregation to these FL frameworks.

BOBA is designed based on the finding that honest gradients distributed near the honest simplex. In Figure 2 we observe that honest (pseudo)-gradients are still distributed near a subspace, even with multiple local updates. Since the basic assumption is not violated, BOBA has potential to be extended to more FL frameworks.

![Figure 5: Extension to FedAvg](image)

We evaluate BOBA on FedAvg with MNIST, and compares it with strong baseline aggregators under two kinds of attacks. Results are shown in Figure 5. We observe that BOBA uniformly has the best performance, showing that it can generalize to more FL scenarios.
References
Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.
Chen, Y.; Su, L.; and Xu, J. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. Proc. ACM Meas. Anal. Comput. Syst., 1(2).
Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.
Karimireddy, S. P.; He, L.; and Jaggi, M. 2022. Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing. In International Conference on Learning Representations.
Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto.
Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86.
McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, X. J., eds., Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, volume 54 of Proceedings of Machine Learning Research, 1273–1282. PMLR.
Peng, J.; Wu, Z.; and Ling, Q. 2020. Byzantine-Robust Variance-Reduced Federated Learning over Distributed Non-i.i.d. Data. CoRR, abs/2009.08161.
Reddi, S. J.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konecny, J.; Kumar, S.; and McMahan, H. B. 2021. Adaptive Federated Optimization. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.
Stewart, G. W. 1990. Perturbation Theory for the Singular Value Decomposition. Technical report, USA.
Wu, Z.; Ling, Q.; Chen, T.; and Giannakis, G. B. 2020. Federated Variance-Reduced Stochastic Gradient Descent With Robustness to Byzantine Attacks. IEEE Transactions on Signal Processing, 68: 4583–4596.
Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In Dy, J.; and Krause, A., eds., Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, 5650–5659. PMLR.
Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In Advances in Neural Information Processing Systems.