Robust Distributed Gradient Aggregation Using Projections onto Gradient Manifolds

Kwang In Kim
POSTECH
kimkin@postech.ac.kr

Abstract
We study the distributed gradient aggregation problem where individual clients contribute to learning a central model by sharing parameter updates constructed from local losses. However, errors in some gradients, caused by low-quality data or adversaries, can degrade the learning process when naively combined. Existing robust gradient aggregation approaches assume that local data represent the global data-generating distribution, which may not always apply to heterogeneous (non-i.i.d.) client data. We propose a new algorithm that can robustly aggregate gradients from potentially heterogeneous clients. Our approach leverages the manifold structure inherent in heterogeneous client gradients and evaluates gradient anomaly degrees by projecting them onto this manifold. This algorithm is implemented as a simple and efficient method that accumulates random projections within the subspace defined by the nearest neighbors within a gradient cloud. Our experiments demonstrate consistent performance improvements over state-of-the-art robust aggregation algorithms.

1 Introduction
The effectiveness of deep learning depends on diverse and rich data that accurately represent diverse facets of real-world applications. However, collecting and maintaining such large datasets can be prohibitively expensive. To address this, in distributed collaborative learning, such as federated learning, local clients (devices or sites that participate in the learning process) individually manage data. Often, such client datasets are kept undisclosed. For instance, medical or healthcare organizations would want to contribute to building a neural network model while ensuring the privacy of patients and experimental participants.

Training a model in such privacy-preserving environments is coordinated by a centralized server that stores only the model parameters. During each training step, the server distributes the parameters to participating clients who then use their respective local data to calculate their updates, i.e., the loss gradients with respect to the parameters. These local updates are then sent back to the server to be combined and update the global model.

Existing collaborative learning approaches typically assume that all clients possess clean and high-quality data. However, in real-world applications, data quality can vary significantly. For example, some clients have highly noisy labels due to the limited availability of experienced annotators, while other clients may be compromised by adversaries. Additionally, some clients might attempt to free-ride, i.e., they wish to use the trained model without contributing adequate data (Fang and Ye 2022). Such affected clients can transfer arbitrarily erroneous or nonsensical gradients, which can severely degrade the final model’s performance if combined straightforwardly.

Robustness against noisy examples and adversarial attacks has been extensively studied in traditional centralized learning environments. However, existing techniques typically require access to the training data, e.g., to build a noise model or design noise-robust losses; therefore, they are not directly applicable to privacy-preserving distributed learning (see Sec. 2 for examples). For collaborative learning, (Turan et al. 2022) presented a robust gradient aggregation scheme assuming a known number of affected clients, while (Fang and Ye 2022) proposed to store a separate dataset in the server to identify outliers. Recently, (Kim 2022) formulated the gradient aggregation problem into an iteratively reweighted averaging process. The resulting algorithm does not require separate datasets or information on affected clients but is limited to cases where client data are roughly homogeneous (Sec. 3.2).

In this paper, we present a novel gradient aggregation algorithm that can handle situations where there is no known information on the client data or the types of noise or attacks, and it can be applied to both heterogeneous and homogeneous clients. Our algorithm leverages the manifold structure of client gradients and assesses the degrees of anomaly of the gradients by projecting them onto this manifold. This is achieved through an intuitive and computationally efficient algorithm that accumulates random projections onto the nearest neighbors in the point cloud of sample gradients. In the experiments with six benchmarks, our algorithm consistently outperformed both the existing robust gradient aggregation algorithm and a baseline uniform gradient averaging approach.

2 Related Work
Robust (Centralized) Learning. State-of-the-art approaches include the use of robust losses (Xu et al. 2019; van Rooyen, Menon, and Williamson 2015; Brooks 2011), regularizers (Liu et al. 2020), filtering out noisy examples (Huang...
The task of aggregating gradients can be viewed as a robust averaging problem, where the objective is to estimate the mean of a set of observations in the presence of outliers. Various robust mean estimators have been proposed as alternatives to the conventional uniform average. For example, one can use the median, which is less sensitive to outliers, as the mean estimator (Pillutla, Kakade, and Harchaoui 2019). (Turan et al. 2022) introduced a method that identifies clean gradients by selecting the nearest neighbors of the median gradient. (Kim 2022)’s algorithm performs iterative weighted averaging of the local client gradients, gradually suppressing outliers. However, in our experiments, we demonstrate that these algorithms are competitive only when clients are homogeneous, and they can perform significantly worse than uniform averaging when clients are heterogeneous (with non-i.i.d. data) in general.

The strategy of filtering examples based on their training losses, originally used in centralized learning (Zhu, Dong, and Liu 2022; Huang et al. 2019; Shen and Sanghavi 2019), can also be applied in collaborative learning by detecting affected clients based on the model’s losses averaged on their respective client data, assuming that each client faithfully communicates their average loss (which may not always be the case). (Fang and Ye 2022) proposed a framework that learns multiple local models for heterogeneous clients, where their client re-weighting algorithm monitors the average losses and their temporal differences to weigh the local learning objectives. We performed experiments with an adaptation of this algorithm for our problem setting, showing that our algorithm provides a more robust alternative.

Several existing algorithms rely on certain assumptions, such as additional clean datasets (Regatti, Chen, and Gupta 2021; Pan et al. 2020; Han and Zhang 2019) or access to local client datasets from the server (Cao and Lai 2018), or the knowledge of the number of affected clients (Mhamdi, Guerraoui, and Rouault 2018; Blanchard et al. 2017). Since these assumptions do not align with our problem, these algorithms are not directly applicable in our context.

3 Robust Gradient Aggregation Algorithm

3.1 Distributed Gradient Aggregation Scheme

The primary objective of training neural networks is to minimize the sum of losses $L$ with respect to the model parameters $w$ on a dataset $P = \{(x_i, y_i)\}_{i=1}^L$: $F(w) = \frac{1}{L} \sum_{i=1}^L L(x_i, y_i; w)$. Following (McMahan et al. 2017)’s learning framework, we assume that $P$ is distributed across $K$ clients. In this case, $F$ can be rewritten as

$$F(w) = \frac{1}{K} \sum_{k=1}^K \frac{L_k}{L} F_k(w),$$

where

$$F_k(w) = \frac{1}{L_k} \sum_{(x_i, y_i) \in P_k} L(x_i, y_i; w),$$

$P_k$ is the subset of $P$ corresponding to client $k$, and $L_k = |P_k|$. Since the client data $\{P_k\}$ may be privacy-sensitive, they are kept hidden in individual clients. Generally, the local datasets $\{P_k\}$ can be non-i.i.d. and they can differ substantially from one another (McMahan et al. 2017). Training
in this scenario can be facilitated by exploiting the additive property of the gradient operator: A single step of gradient descent-based minimization of $\mathcal{F}$ can be stated as

$$w^{t+1} = w^t - \eta \sum_{k=1}^{K} \frac{L_k}{L} g_k^t, \text{ where } g_k^t = \nabla_w \mathcal{F}_k(w^t)$$

and $\eta$ is the learning rate. A server can coordinate these updates by only maintaining the model parameters $w$: At step $t$, it broadcasts $w^t$ to individual clients, and collects and aggregates the resulting gradients $\{g_k^t\}$ to compute $w^{t+1}$.

Collaborative learning can enhance privacy and reduce security risks by limiting access to the entire dataset (McMahan et al. 2017). However, this approach can also increase the risk of attacks or data corruption at local clients, especially for those with limited security resources. Such attacks could alter the data or gradients of affected clients, leading to misleading updates if these gradients are naively aggregated in Eq. 2. In practice, the identities and numbers of affected clients may not be known in advance, making it challenging to detect and mitigate such attacks.

3.2 Estimating Gradient Aggregation Weights

Inspired by recent progress in robust collaborative learning (Fang and Ye 2022; Turan et al. 2022; Kim 2022), we introduce a method to regulate the contribution of individual gradients using convex combination weights $\{\alpha_k^t\}_{k=1}^K$:

$$w^{t+1} = w^t - \eta K \bar{g}^t, \text{ where } \bar{g}^t = \sum_{k=1}^{K} \alpha_k^t \frac{L_k}{L} g_k^t \quad \alpha_k^t \geq 0, \quad \text{and } \sum_{k=1}^{K} \alpha_k^t = 1.$$  

When $\{\alpha_k^t\}$ are identical (i.e. $\alpha_k^t = \frac{1}{K}$), Eq. 3 simplifies to the standard uniform update (Eq. 2) while in the ideal case, $\alpha_k^t = 0$ for the affected clients $k$'s. Between these two extremes, various robust learning algorithms are created by specifying how $\{\alpha_k^t\}$ are determined. For instance, (Turan et al. 2022) proposed constructing the median $\bar{g}^t$ of $\{g_k^t\}$ as a robust alternative to the average. Subsequently, clients that do not belong to the nearest neighbors of $\bar{g}^t$ are assigned zero $\alpha$-values. On the other hand, (Kim 2022) proposed an algorithm involving iteratively reweighted averaging of $\{g_k^t\}$: Initially (at $t = 1$), $\{\alpha_k^t\}$ are uniform. From step $t = 2$ onwards, $\alpha_k^{t+1}$ is inversely proportional to the distance between $g_k^t$ and $\bar{g}^t$.

These algorithms have demonstrated state-of-the-art performance when the distributions of local data $\{P_k\}$ are homogeneous (i.e., i.i.d.). In this case, if a client $k$ is not affected, $P_k$ provides a good approximation of $P$, and the corresponding client gradient $g_k^t$ should be similar to the (weighted) average $\bar{g}^t$ or median $\tilde{g}^t$ calculated based on $P$. However, when the client datasets are sampled from heterogeneous distributions, even clean gradients may exhibit significant differences from each other. Therefore, the affected gradients may not be adequately characterized by their weak similarities to their average or median, as illustrated in Fig. 1.

Our algorithm leverages the manifold structure $M$ of client gradients and evaluates the (degree of) anomaly of a gradient based on its projection distance onto $M$.

![Figure 1: Two approaches to assessing client anomaly. Each data point in the plots represents a client gradient. In the case of relatively i.i.d. client data (Left), the corresponding client gradients tend to be closely clustered around their weighted average (black points) or median (green points). Here, detecting outlier clients (orange points) can be achieved by measuring their distance from the average or median. Alternatively, when clients are heterogeneous in nature, a client may be similar to only a small subset of other clients while significantly differing from the remaining ones (Right). In this scenario, even outlier clients may be closer to the average or median, and therefore client distances to their projections onto the underlying manifold could provide more reliable estimates of anomalies.](image-url)

1More generally, $T_PM$ is a vector space of derivative operators on smooth functions around $p \in M$ (Jost 2011).
We consider normalized gradient vectors $\mathbf{G} = \{\mathbf{g}_k\}_{k=1}^K$ that are not directly given as elements of an underlying manifold $M$ but rather presented as a potentially noisy sample in the ambient space $X \subset \mathbb{R}^N$: $||\mathbf{g}_k|| = 1$ and $\sum_j (\mathbf{g}_k)_j^2 = 0$ where $(\mathbf{g}_k)_j^2$ is the $j$-th component of $\mathbf{g}_k$. Since we do not have direct access to $M$, we estimate the projection $P$ from $G$:

Given a gradient vector $\mathbf{z} \in X$, we first find its nearest neighbors $N(\mathbf{z})$ in $G$. Arranging the elements of $N(\mathbf{z})$ column-wise in a matrix, we obtain a local design matrix $\mathbf{G}_x = [\mathbf{g}_q(1), \ldots, \mathbf{g}_q(H)]$ where $q$ assigns the global indices of $G$ from the local indices of $N(\mathbf{z})$. The span of vectors in $\mathbf{G}_x$ can be considered as an estimate of the tangent space $d\mathcal{I}(T_pM)$ at the point $\mathcal{I}(p)$ closest to $\mathbf{z}$. Our empirical projection is then defined as the orthogonal projection onto this $d\mathcal{I}(T_pM)$-estimate as a proxy geometry of $M$ around $p$:

$$
\mathbf{P}_N(\mathbf{z}) = \mathbf{G}_x (\mathbf{G}_x^\top \mathbf{G}_x)^{-1} \mathbf{G}_x^\top \mathbf{z}.
$$

While $\mathbf{P}_N(\mathbf{z})$ is usually well-defined, in the unlikely case that $\mathbf{G}_x$ is not full rank, we can perform column selection using the F-statistic (Mood, Graybill, and Boes 1974) in $\mathbf{G}_x$ before constructing $\mathbf{P}_N(\mathbf{z})$. Once $\mathbf{P}_N(\mathbf{z})$ is calculated, the anomaly of $\mathbf{z}$ is obtained as $A(\mathbf{z}) = ||\mathbf{P}_N(\mathbf{z}) - \mathbf{z}||^2$.

### Estimating Gradient Aggregation Weights

Our algorithm generates aggregation weights $\{\alpha_k\}_{k=1}^K$ by first sampling random network parameters $\mathbf{w}^*$ at the server and sharing them with each client. The corresponding client gradients $(\mathbf{g}_k^*)_{k=1}^K$ form a data cloud $G^*$. For each gradient $\mathbf{g}_k^* \in G^*$, we evaluate its anomaly $A(\mathbf{g}_k^*)$ by projecting it to $G_k^*$, formed by the nearest neighbors $N_k^*$ of $\mathbf{g}_k^*$ in $G^* \setminus \{\mathbf{g}_k^*\}$. The gradient aggregation weights at $s$ are then determined as

$$
\alpha_k^s = \frac{\overline{\pi}_k}{\sum_{k=1}^K \alpha_k}, \quad \text{where} \quad \overline{\pi}_k = \exp\left(-\frac{A(\mathbf{g}_k^*)}{\sigma^*_a}\right).
$$

Here, the scale parameter $\sigma^*_a$ is set to the mean of $\{A(\mathbf{g}_k^*)\}_{k=1}^K$. We repeat this process $S = 20$ times with different random parameters $\mathbf{w}^*$, and determine the final aggregation weights $\{\alpha_k\}$ as the average of their corresponding counterparts $\{\alpha_k^s\}$.

The time complexity of each step (out of $S$ steps) in our algorithm is $O(RKH + RK^2 + H^3)$, where $R$ is the size of the gradient vector, $K$ is the number of clients, and $H$ is the local neighborhood size. The calculation of the projection in our algorithm, which involves solving the linear system in Eq. (5), results in negligible computational costs due to the small size of the system ($H \times H$ with $H = 10$; see Sec. 4). The primary bottleneck is twofold: 1) evaluating client gradients from each set of random parameters $\mathbf{w}^*$ as our algorithm requires gradient evaluations before the main network training occurs and 2) finding the neighborhood $N_k$ of each gradient $k$. For a large number $K$ of clients, we can improve the complexity by employing an approximate nearest neighbor search. For CIFAR100 dataset (see Sec. 4), the initial gradient evaluation took around one minute. Our algorithm incurs $O(RK)$ memory complexity to store the

Algorithm 1: Robust collaborative learning algorithm. Training data is distributed across $K$ clients. An unknown number out of $K$ clients will provide erroneous gradients.

**Input:** Gradient neighborhood size $H = 10$ and the number of random projection steps $S = 20$.

for $s = 1, \ldots, S$ do

   Randomly construct $\mathbf{w}^*$ and send it to clients.

   Fetch gradients $G^* = \{\mathbf{g}_k^*\}_{k=1}^K$ from clients.

   for $k = 1, \ldots, K$ do

       Build global indices $\{q\}$ for the neighborhood $N_k^*$ of $\mathbf{g}_k^*$

       and $\mathbf{G}_k^* = [\mathbf{g}_q(1), \ldots, \mathbf{g}_q(H)]$.

       Calculate $\mathbf{G}_k^*$-projection:

       $P_{N_k^*}[\mathbf{g}_k^*] = \mathbf{G}_k^* (\mathbf{G}_k^*^\top \mathbf{G}_k^*)^{-1} \mathbf{G}_k^* \mathbf{g}_k^*$.

       Calculate anomaly degree $A(\mathbf{g}_k^*) = ||P_{N_k^*}[\mathbf{g}_k^*] - \mathbf{g}_k^*||^2$.

   end for

   for $k = 1, \ldots, K$ do

       $\overline{\pi}_k = \exp\left(-\frac{A(\mathbf{g}_k^*)}{\sigma^*_a}\right)$;

       $\alpha_k^s = \frac{\overline{\pi}_k}{\sum_{k=1}^K \overline{\pi}_k}$.

   end for

end for

Perform network training steps in Eq. 3 using $\{\alpha_k\}_{k=1}^K$.

**Output:** Model parameters $\mathbf{w}$ and aggregation weights $\{\alpha_k\}_{k=1}^K$.

### 3.3 Robustness of the Anomaly Estimates

Our method for estimating the tangent space $T_pM$ differs from common approaches: Typically, principal component analysis (PCA) is first performed on $N_k$ to reduce the dimensionality to that of $T_pM$ (Singer and WU 2016). This provides a rigorous convergence guarantee of the resulting tangent space estimate as $|G| \to \infty$ even under noise. However, this method requires a known manifold dimensionality, which can be challenging to estimate from a sampled point cloud. Another significant limitation of this local PCA-based approach applied to our gradient aggregation setting is that $G \setminus \{\mathbf{g}_k\}$ may contain outliers as more than one client can be affected in $G$. Since PCA is susceptible to outliers, the resulting tangent space estimates can be erroneous. The supplemental document shows that our method offers more reliable anomaly estimates compared to the PCA-based alternative.

A deeper insight into the robustness of our approach against outliers in $G \setminus \mathbf{g}_k$ can be gained by noting that all gradients in $G$ are normalized. In this case, the anomaly degree can be restated as

$$
A(\mathbf{g}_k) = \sum_{j=1}^R \left( h_k((\mathbf{G}_k)_{j,j}^*) - (\mathbf{g}_k)_j^2 \right), \quad \text{where} \quad h_k(s) = a_k s, \quad a_k = (\mathbf{G}_k^\top \mathbf{G}_k)^{-1} \mathbf{G}_k^\top \mathbf{g}_k,
$$

and $(\mathbf{G}_k)_{j,j}^*$ is the $j$-th row of $\mathbf{G}_k$. This suggests that the quantity $A(\mathbf{g}_k)$ can be interpreted as the error incurred when attempting to reconstruct the elements of $\mathbf{g}_k$ using linear

client gradients. The resulting robust collaborative learning process is summarized in Algorithm 1.
least-squares estimator $h$. In this method, the rows of $G_k$ are used as training inputs, and the resulting reconstruction is a weighted combination of the columns of $G_k$ as features, each corresponding to a client gradient in $N_k$. Outlier gradients that do not contribute to constructing $g_k$ can be ignored by assigning low-magnitude weight values $\{a_k^i\}$ to them. For instance, on the CIFAR10 dataset (see Sec. 4), the average magnitudes of the elements of $a_k$ corresponding to clean and affected clients were 0.147 and 0.009, respectively.

The above highlights two main factors that contribute to the robustness of the projection operator $P_N$: (1) the assumed manifold structure of clean gradients and (2) considerably smaller neighborhood size $H = |N_k|$ than the size $R$ of the gradient vector. When a clean gradient $g_k$ is present, both $g_k$ and its spatial neighborhood $N_k$ lie within a low-dimensional subspace (tangent space $T_pM$) of $X = \mathbb{R}^R$ (1); therefore combinations of columns in $G_k$ can effectively reconstruct $g_k$ (Eq. 7). When $N_k$ contains outliers, if $H \ll R$ (2), these gradients are unlikely to be coincidentally strongly correlated with $g_k$ (and hence contribute to reconstructing $g_k$), resulting in small $a_k$ weights in the estimator $h_k$. On the other hand, if $g_k$ is an outlier lying outside $T_pM$ and $H \ll R$ (2), both clean and outlier gradients in $N_k$ are likely to deviate significantly from $g_k$. As a result, $G_k$ is likely to fail to reproduce $g_k$.

The robustness of the projection operator $P_N$ is specific to the gradient aggregation setting and should not be considered a general indicator of the robustness of least-squares estimation. It also points out a simple failure case where the number of features, $H$, becomes equal to the number of data instances, $R$, in $G_k$ and those features are in general position. In this case, $G_k$ becomes invertible, leading to constant zero estimation error (Eq. 7). The abnormalities of outliers $g_k$ were substantially reduced when $N_k$ was augmented with 10,000 random vectors. This suggests that as the value of $K$ increases, it is advisable to keep the neighborhood size $H$ small. Accordingly, in all experiments conducted, a fixed neighborhood size of 10 was used (see our supplemental document). While optimizing this parameter for each problem and dataset may enhance performance, it may not be feasible in certain scenarios as it would necessitate separate clean validation sets at the server.

Explicit Robustification. Considering the least-square estimation interpretation of $P_N$ and its empirical robustness against outlier features, it is enticing to explore the possibility of further enhancing its robustness through feature selection approaches. We conducted preliminary experiments with two robust feature selection methods. We conducted preliminary experiments with LASSO as an implicit feature selector. This replaces $A$ by calculating the coefficients $a_k$ of $h_k$ (Eq. 7) as the minimizer of

$$A'(g_k) = \sum_{j=1}^{R} (a_k^j (G_k)^{j,*} - (g_k^j)^{j,*})^2 + \gamma \|a_k\|_1. \quad (8)$$

This should not be confused with the well-known high-sensitivity of least-squares estimation to outlier output instances (rows of $g_k$; see Eq. 7).

We consistently observed that the best gradient aggregation performance was achieved when $\gamma = 0$, resulting in $A'$ being equal to $A$. LASSO promotes sparsity in $a$, helping suppress the contribution of outlier gradients. However, this has an adversarial effect of suppressing potentially beneficial correlated clean gradients since, in this way, it can enhance a’s sparsity while only moderately sacrificing the reconstruction error (the first term in Eq. 8). Consequently, this can lead to increased abnormalities $A'$ for clean clients (see supplemental document for an example).

4 Experiments

To evaluate the effectiveness of our robust gradient aggregation algorithm, we conducted experiments with six benchmark datasets.

Datasets. The CIFAR10 and CIFAR100 datasets consist of 60,000 color images from 10 and 100 classes, respectively (Krizhevsky 2009). For each dataset, 50,000 images were used for training, and the remaining 10,000 images were reserved for testing. TinyImageNet is a subset of the ImageNet 2017 benchmark, consisting of 100,000 training and 10,000 testing images evenly covering 200 object categories (Le and Yang 2015). The Kuzushiji49 dataset provides 223,365 training and 38,547 testing images of 49 Japanese characters (Cinanlou et al. 2018). The Fashion-MNIST (FMNIST) and extended MNIST letters (EMNISTL) datasets provide 70,000 images of 10 cloth categories (Xiao et al. 2017) and 124,800 letter images (LeCun et al. 1998), respectively.

Data Allocation and Implementation Details. We used $K = 100$ clients for all datasets. To distribute each dataset to these clients, we extended (McMahan et al. 2017)’s approach (to more than ten classes), which is commonly used to prepare learning environments with heterogeneous clients (Li et al. 2020, 2022a). Firstly, we partitioned the dataset into $0.2 \times C \times K$ shards, where $C$ is the number of classes. Each shard contained only a single class, and all shards were of equal size. Then, we randomly assigned $0.2 \times C$ shards to each client, ensuring that each client received a subset of up to 20% of all classes. Our supplementary material presents the results obtained when client data allocation was performed by sampling class labels from Dirichlet distributions.

To simulate the affected clients, we employed the class-symmetric flip model (van Rooyen, Menon, and Williamson 2015; Patrini et al. 2017; Han et al. 2018). For an affected client $k$, we generated a transition matrix $T_k \in \mathbb{R}^{C \times C}$ by randomly sampling each row from the uniform distribution on $[0, 1]^C$ and normalizing the results. Then, the label of each data point with class $j$ in $P_k$ was reassigned by sampling from $(T_k)^{j,*}$. For each number of affected clients in $\{10, 20, 30, 40, 50, 60, 70\}$ (out of $K = 100$), we repeat experiments 10 times and report the average results.

We used convolutional neural networks with two convolution layers and two fully-connected layers for Kuzushiji49, FMNIST, and EMNISTL, following the setup of (LeCun et al. 1998). The convolution layers comprised 10 and 20 filters of size $5 \times 5$, followed by $2 \times 2$ max pooling. The fully-connected
### Table 1: Performance of different gradient aggregation methods when client data are allocated using an extension of McMahen et al.’s approach (McMahen et al. 2017). Our supplementary material presents the corresponding results achieved when client data allocation was conducted by sampling class labels from Dirichlet distributions. The best results are highlighted in italic. The results of the statistical significance tests for differences from baseline `Uniform` are represented with bold (significantly better) and underline (significantly worse) fonts using a t-test with α = 0.05. With the exceptions of (Sun et al. 2019) on CIFAR10 with (10,20,30) affected clients and (Kim 2022) on TinyImageNet with 30 affected clients, all existing approaches were statistically equivalent or significantly worse than `Uniform`. In contrast, our algorithm was significantly better than `Uniform` in 36 out of the total 42 cases and was never significantly worse.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Mean accuracy ± standard deviation × 100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Affected clients</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>Uniform</td>
<td>89.58 ± 0.15</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>89.50 ± 0.21</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>89.48 ± 0.15</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>90.60 ± 0.21</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>89.84 ± 0.17</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>89.91 ± 0.16</td>
</tr>
<tr>
<td>CfS100</td>
<td>Uniform</td>
<td>66.64 ± 0.36</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>66.90 ± 0.25</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>66.74 ± 0.27</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>66.73 ± 0.26</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>66.76 ± 0.34</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>66.87 ± 0.35</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td>Uniform</td>
<td>65.66 ± 0.30</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>65.62 ± 0.26</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>65.43 ± 0.25</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>65.55 ± 0.19</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>65.27 ± 0.19</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>65.77 ± 0.23</td>
</tr>
<tr>
<td>FMNIST</td>
<td>Uniform</td>
<td>77.69 ± 0.48</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>76.34 ± 1.03</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>77.06 ± 0.59</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>77.98 ± 0.52</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>77.65 ± 0.51</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>77.87 ± 0.72</td>
</tr>
<tr>
<td>Kazushiji49</td>
<td>Uniform</td>
<td>48.43 ± 0.78</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>48.08 ± 0.58</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>48.40 ± 0.54</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>48.42 ± 0.78</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>48.51 ± 0.62</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>48.50 ± 0.59</td>
</tr>
<tr>
<td>EMNISTL</td>
<td>Uniform</td>
<td>67.91 ± 0.56</td>
</tr>
<tr>
<td></td>
<td>(Kim 2022)</td>
<td>67.41 ± 0.69</td>
</tr>
<tr>
<td></td>
<td>(Turan et al. 2022)</td>
<td>67.46 ± 0.90</td>
</tr>
<tr>
<td></td>
<td>(Sun et al. 2019)</td>
<td>67.55 ± 0.68</td>
</tr>
<tr>
<td></td>
<td>(Fang and Ye 2022)</td>
<td>67.76 ± 0.90</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>68.68 ± 0.78</td>
</tr>
</tbody>
</table>

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

layers were of 50 and 10. For the remaining datasets, we combined a ResNet50 pre-trained on ImageNet dataset with three fully connected layers of size 300 each, following (Kim 2022). All experiments were conducted on a machine with two Intel Xeon Silver 4210R CPUs and two NVIDIA RTX390 GPUs.

**Baselines.** For comparison, we also conducted experiments with 1) baseline uniform aggregation (i.e. \( \alpha_k = \frac{1}{K}, \forall 1 \leq k \leq K; \text{Uniform}) in (Kim 2022)’s iterative reweighting algorithm, 3) (Turan et al. 2022)’s median-based gradient selection approach, 4) (Sun et al. 2019)’s gradient clipping algorithm, and 5) (Fang and Ye 2022)’s robust federated learning algorithm, which was originally designed for learning personalized models for heterogeneous clients. We adopted their client confidence re-weighting scheme to centralized gradient aggregation by introducing a new client maintaining the combined gradient \( \hat{g}^c \). Turan et al.’s algorithm determines the gradients distinct from the median as outliers and

13156
requires a known number of outlier clients: We used the
ground-truth numbers of affected clients in our experiments
while in practical applications, one would need to determine
them as a hyperparameter. 6) We also performed experiments
with an algorithm that aims to denoise the gradient matrix
\( G = [g_1, \ldots, g_N] \) (Denoising) by considering the affected
gradients as noisy observations of underlying ground truths.
This constructs a new matrix \( (G^*) \) by minimizing
\[
Q(G) = \frac{1}{2} ||G - G^*||_F^2 + \lambda_N ||G||_S,
\]
where \( ||A||_F \) and \( ||A||_S \) are the Frobenius and nuclear norms
of \( A \), respectively. The energy \( Q \) is convex, and its minimum
can be determined through singular value shrinkage (Bach
et al. 2012). Once \( G^* \) is obtained, the corresponding gradi-
ents are uniformly aggregated.

Analysis. The results of our experiments are summarized
in Table 1. Existing gradient combination algorithms demon-
strated comparable or inferior performance to uniform aver-
aging across all datasets: (Fang and Ye 2022)’s algorithm
was on par with Uniform. This algorithm determines the
aggregation weights \( \{\alpha_k\} \) based on the average client
losses and their temporal variations. However, in our setting,
these loss values did not vary notably across the clean and
affected clients, resulting in almost uniform weights. (Sun
et al. 2019)’s algorithm assumes that the outlier gradients
exhibit larger magnitudes than clean ones. This assumption
is similar to that made in (Fang and Ye 2022), as the gradient
magnitudes are directly proportional to the loss scales, and it
also attained an equivalent level of performance as Uniform.
(Kim 2022)’s algorithm performed significantly worse than
Uniform, which can be attributed to the fact that when clients
are heterogeneous, outlier gradients can be closer to the aver-
ages (Fig. 1), and therefore contribute materially to the final
aggregation. (Turan et al. 2022)’s algorithm also achieved
considerably worse results as it filtered out a large portion
of clean gradients, assuming that the affected gradients were
often close to the median.

We set \( \lambda_N \) to \( 10^{-2} \) for Denoising since smaller values led
to nearly identical results as Uniform. Nonetheless, the re-
sulting accuracies were slightly worse than Uniform. In cases
where the affected gradients stem from incorrect training
labels, such as those arising from adversarial attacks, they can
substantially deviate from the ground-truth gradients. In such
scenarios, attempting to denoise \( G^* \) by lowering its nuclear
norm (second term in \( Q \); Eq. 9) can inadvertently alter the
clean gradients, as they are typically more structured: Modifi-
ying them, rather than the affected ones, can more readily
reduce the nuclear norm while retaining low reconstruction
error (first term in \( Q \)).

Our findings suggest that existing robust gradient aggrega-
tion algorithms, which have demonstrated competitive perfor-
ance on homogeneous clients, may not be effective in

\footnote{Based on this observation, we performed preliminary experi-
ments with a new client anomaly degree evaluating how the columns
of the minimizer \( G^* \) deviate from \( G^* \) (with \( G^* \) \( k \)-th being the \( k \)-th
column of \( G^* \): \( A'(g_k) = ||G^*_{:,k} - G^*_{:,k}||^2. \) While this ap-
proach led to measurable improvements over the Uniform approach,
our final design consistently and significantly outperformed it.}

the presence of heterogeneous clients. Specifically, our ex-
periments indicate that the aggregation weights, calculated
based on the gradient distances to the weighted averages or
median, can be misleading in such scenarios. Moreover, the
assumptions typically imposed in prior work, such as the
uniformity of loss values and gradient magnitudes (Fang and
Ye 2022; Sun et al. 2019), may not hold in heterogeneous
settings, rendering those methods ineffective.

Our algorithm consistently outperformed Uniform and state-of-the-art gradient aggregation algorithms by effectively
leveraging the nonlinear structures of heterogeneous gradi-
ents through the assessment of manifold projection distances.
The performance gains were more noticeable in scenarios
with 1) a large number of affected clients and 2) datasets
with a large number of classes. Specifically, for TinyImage-
Net with 200 classes, when 70 clients were affected, ours
achieved a 19.43% accuracy improvement from Uniform,
while the advantage almost disappeared (0.17%; statistically
insignificant) when 10 clients were affected. For CIFAR10
with only ten classes, our algorithm achieved a 4.2% perfor-
ance margin for 70 affected clients. When all clients were
clean, Uniform and ours were statistically equivalent, with
accuracy values of 90.01\( \pm 0.12 \) and 89.94\( \pm 0.18 \), respectively.

In the supplementary material, we present additional ex-
periments on homogeneous clients. Despite our algorithm
not being explicitly designed for this scenario, our results
demonstrate that it is still competitive with or even superior
to existing algorithms, highlighting its general applicability.

5 Conclusions

We have considered the challenge of robustly aggregating
heterogeneous client gradients in scenarios where no informa-
tion about the affected clients, such as their number and types
of attacks, is available. Our thesis posits that the client gradi-
ents lie on a manifold, and we exploit this structure by assessing
the anomalies of given gradients through their projections
onto this manifold. To this end, we have devised an efficient
algorithm that iteratively generates random model parameters
and projects the resulting gradients onto the subspace spanned
by their spatial neighborhoods. This instantiation of our thesis
has produced promising results, outperforming existing
state-of-the-art methods on heterogeneous clients while
delivering comparable outcomes in homogeneous ones.

Further research should focus on developing theoretical
foundations to provide a robust underpinning for these find-
ings. This would entail the development of new convergence
analysis techniques, as individual client gradients may not
provide unbiased estimators of the clean aggregated gradi-
ents in environments that involve heterogeneous clients, un-
like the settings in which existing robust distributed learning
techniques were designed. All robust gradient aggregation
algorithms considered in this paper may fail when client
attacks are coordinated. As a simple example, when local
clients transfer multiple identical erroneous gradients, these
algorithms will mistake them for clean ones. While simple
heuristics can easily detect this particular case, future work
should systematically investigate extending our algorithm to
respond to such coordinated attacks.
Acknowledgments
This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2021R1A2C2012195) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)), funded by the Korean government (MSIT).

References


Cao, X.; and Lai, L. 2018. Robust distributed gradient descent with arbitrary number of Byzantine attackers. In ICASSP.


Fang, X.; and Ye, M. 2022. Robust federated learning with noisy and heterogeneous clients. In CVPR, 10072–10081.


Han, Y.; and Zhang, X. 2019. Robust federated learning via collaborative machine teaching. In AAAI.


Kim, K. I. 2022. Robust combination of distributed gradients under adversarial perturbations. In CVPR.


Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022a. Federated learning on non-IID data silos: an experimental study. In ICDE.


Shen, Y.; and Sanghavi, S. 2019. Learning with bad training data via iterative trimmed loss minimization. In ICML.


van Rooyen, B.; Menon, A. K.; and Williamson, R. C. 2015. Learning with symmetric label noise: the importance of being unhinged. In NIPS.


Zhu, Z.; Dong, Z.; and Liu, Y. 2022. Detecting corrupted labels without training a model to predict. In ICML.