Appendix

A Notations

Notation	Description	Notation	Description
n	Total number of clients	c	Number of clusters
q	Number of faulty clients	$\mathcal{S}$	set of all clients
K	Number of local SGD epochs	$[m]$	The set of integers $\{1, \ldots, m\}$
T	Number of global epochs	$\{\mathcal{M}_j\}_{j\in[c]}$	Set of client clusters
R	Number of resampling epochs	n_i	Number of samples on worker i
b	Trim parameter for defense	m	Number of clients in each cluster
η_l, η	Local and global learning rates	$\|\cdot\|$	All norms in this paper are l_2-norms

Table 1: Notations utilized in this paper

B Proofs

In this section, we elaborate on theoretical guarantees of SHARE. We define the following quantity to aid the proofs that follow

Definition B.1 (Clustered Client Update)**.** We define clustered client update as average model updates from all the clients assigned to a particular cluster. Mathematically, the clustered client update in a reclustering round $r \leq R$ is given by $g_i^r = \sum_{j\in\mathcal{M}_k} \Delta_j$ where Δ_j denotes the model update from client j belonging to cluster $\mathcal{M}_k$ after K steps of SGD.

Lemma 5. *If robust aggregation is replaced by averaging, output of Algorithm 1 is identical to Federated Averaging[15].*

Proof. In each re-clustering round, the update with benign averaging becomes $g^r = \sum_{i\in[c]} \sum_{j\in\mathcal{M}_i} \Delta_j^K = \sum_{l\in[n]} \Delta_l^K$ where n is the total number of clients and $c = \frac{n}{m}$ is the total number of clusters, as this update is independent of the random cluster division, the global update at round t becomes $x^t = x^{t-1} + \eta \sum_{l\in[n]} \Delta_l^K$ which is identical to federated averaging. □

Lemma 6. *In presence of robust aggregation, Algorithm 1 is robust to $q = \frac{q_0}{m}$ where q_0 is the tolerance limit of the robust aggregation oracle followed and m is the cluster size.*

Proof. We consider the worst case scenario of each malicious client being in different clusters, hence spreading the attack to the maximum possible number of clients. Although randomization beats this and might offer better clusters in multiple random rounds, there might still exist such attack favorable rounds. Allowing for this worst case sets the threshold to $q = \frac{q_0}{m}$ if the original robustness oracle has a threshold of q_0. □

B.1 Distance based robust aggregation

Theorem 7. *Consider a function F(x) satisfying Assumptions 5.1,5.2 assume a robust aggregation scheme that picks up b updates and satisfies Assumption 5.3, further, assume (G,B)-Bounded gradient dissimilarity, σ_g^2 variance in client updates and σ^2 variance in gradient estimation, there exists η, η_l such that output of Algorithm 1 after T rounds, x^T, satisfies,*

$$\mathbb{E}\left[\|\nabla F(x^T)\|^2\right] \leq \mathcal{O}\left(\frac{LM\sqrt{F}}{\sqrt{TKn}} + \frac{F^{2/3}(LG^2)^{1/3}}{(T+1)^{2/3}} + \frac{B^2LF}{T} + 2L^2V_2 + \frac{\sigma_g^2}{bm}\left(\frac{n-q-bm}{R(n-q)-1}\right)\right)$$

where $M^2 := \sigma^2(1 + \frac{n}{\eta^2})$ and $F := F(x^0) - F(x^*)$

Proof. Firstly, we bound the distance between global model update from Algorithm 1 and expected benign mean model update in each global iteration. In particular, let the expected benign mean model update be denoted by μ_t and global model update in each

iteration is given by $\frac{1}{R}\sum_r \texttt{RobustAggr}(\{g_i^r\}_{i\in[c]})$. We determine an upper bound on $\|\mathbb{E}[\frac{1}{R}\sum_r \texttt{RobustAggr}(\{g_i^r\}_{i\in[c]})] - \mu_t\|$. This is illustrated below

$$\begin{aligned}
\|\mathbb{E}[\frac{1}{R}\sum_r \texttt{RobustAggr}(\{g_i^r\}_{i\in[c]})] - \mu_t\|^2 &\leq \|\mathbb{E}[\frac{1}{R}(\sum_r \texttt{RobustAggr}(\{g_i^r\}_{i\in[m]}) - \sum_{r,i\in B} g_i^r)] \\
&\quad + \mathbb{E}[\frac{1}{R}\sum_{r,i\in B} g_i^r] - \mu_t\|^2 \\
&\leq \mathcal{O}(V_2) + 2\|\mathbb{E}[\frac{1}{R}\sum_{r,i\in B} g_i^r] - \mu_t\|^2 \\
&\leq \mathcal{O}(V_2) + 2\mathbb{E}\|\texttt{Resample}(\{\Delta_i^K\}_{i\in C}) - \mu_t\|^2 \\
&\leq \mathcal{O}(V_2) + \frac{\sigma_g^2}{Rbm}(1 - \frac{R(bm)-1}{R(n-q)-1}) \\
&\leq \mathcal{O}(V_2) + \frac{\sigma_g^2}{bm}(\frac{n-q-bm}{R(n-q)-1})
\end{aligned}$$

Where B denotes indices of benign clusters (clusters with uncorrupted device updates. Mathematically, let $\mathcal{C}_t$ denote set of benign clients among all n clients. $\{B : i \in [c]$ such that $\forall k \in [m], \Delta_k \in \mathcal{M}_i, \Delta_k \in \mathcal{C}_t\}$), $r \leq R$ as mentioned in the text denote reclustering rounds. The second inequality follows from Assumption 5.3. Since each reclustering round randomly groups clients together, the set $\{\Delta_k : \Delta_k \in \mathcal{M}_i, i \in B\}$ is a random resample of bm benign client updates from the available $n-q$, where b is the number of updates available after filteration through robust aggregation. With R resampling rounds, this is equivalent to resampling Rbm updates from $R(n-q)$ benign updates. Following Rice [18](Chapter 7, Theorem B), we obtain the scaled down variance bound.

Using L-smoothness of $F(x)$,

$$\begin{aligned}
\mathbb{E}[\|\nabla F(x^t)\|^2] &\leq 2\mathbb{E}[\|\nabla F(x^t) - \nabla F(\mu_t)\|^2] + 2\mathbb{E}[\|F(\mu_t)\|^2] \\
&\leq 2L^2(\mathcal{O}(V_2) + \frac{\sigma_g^2}{bm}(\frac{n-q-bm}{R(n-q)-1})) + 2\mathbb{E}[\|F(\mu_t)\|^2]
\end{aligned}$$

The rest follows a similar approach as Karimireddy et al. [10] hence we get

$$\mathbb{E}\left[\|\nabla F(x^T)\|^2\right] \leq \mathcal{O}\left(\frac{LM\sqrt{F}}{\sqrt{TKn}} + \frac{F^{2/3}(LG^2)^{1/3}}{(T+1)^{2/3}} + \frac{B^2LF}{T} + 2L^2V_2 + \frac{\sigma_g^2}{bm}\left(\frac{n-q-bm}{R(n-q)-1}\right)\right)$$

where $M^2 := \sigma^2(1 + \frac{n}{\eta^2})$ and $F := F(x^0) - F(x^*)$. □

B.2 Zeno++ as robust aggregation

We first illustrate a modified Zeno++ algorithm and adapt it to Federated Learning setting from its original asynchronous SGD paradigm. Firstly, we define a score that helps filter out updates if they fall below a threshold. Intuitively, the score denotes trustworthiness of a clustered update.

Definition B.2 (Approximated model update score)**.** Denote $f_s(x) = \frac{1}{n_s}\sum_i^{n_s} f(x; z_i)$, where $z_j's$ are drawn independent and identically from $\mathcal{D}_s \neq \mathcal{D}_i, \forall i \in [n]$ and n_s is the batch size of $f_s(\cdot)$, for a clustered client update g, model parameter x, global learning rate η and constant weight $\rho > 0$, we define model update score as

$$Score_{\eta,\rho} \approx -\eta\langle\nabla f_s(x), g\rangle - \rho\|g\|^2$$

where x is the current model available on the server.

Using this approximated model update score, we set hard thresholding parameterized by ϵ to filter client cluster updates. Algorithm 2 illustrates SHARE framework with Zeno++ as robust aggregation. We analyze the convergence of Algorithm 2 in the following theorem.

Theorem 8. *Consider L-smooth and potentially non-convex functions $F(x)$ and $f_s(x)$, Assume validation set is close to training set, implying a bounded variance given by $\mathbb{E}[\|\nabla f_s(x) - \nabla F(x)\|^2] \leq$*

Algorithm 2 SHARE (Secure Hierarchical Robust Aggregation) with Zeno++ defense

Server:
for $t = 0, \ldots, T-1$ **do**
  for $r = 1, \ldots, R$ **do**
    Assign clients to clusters $\mathcal{S} = \mathcal{M}_1 \cup \ldots \mathcal{M}_i \ldots \cup \mathcal{M}_c$ with $|\mathcal{M}_i| = |\mathcal{M}_j| \forall i, j \in [c]$
    Compute secure average $g_j^r \leftarrow \texttt{SecureAggr}(\{\Delta_i\}_{i \in \mathcal{M}_j}) = \sum_{i \in \mathcal{M}_j} u_i, \forall j \in [c]$
    Randomly sample $z_j \sim S, \forall j \in [n_s]$ to compute f_s
    for $i = 1, \ldots, c$ **do**
      if score$(g_i^r, x^{t-1} \geq -\eta\epsilon)$ **then**
        $g^r \leftarrow g^r + g_i^r$,
      end if
    end for
  end for
  if stopping criteria met **then**
    break
  end if
  Push $x^t = x^{t-1} + \eta \frac{1}{R} \sum_r g^r$ to the clients
end for
Client:
for each client $i \in \mathcal{S}$ (if honest) **in parallel do**
  $x_{i,0}^t \leftarrow x^t$
  for $k = 0, \ldots, K-1$ **do**
    Compute an unbiased estimate $g_{i,k}^t$ of $\nabla f_i(x_{i,k}^t)$
    $x_{i,k+1}^t \leftarrow \texttt{ClientOptimize}(x_{i,k}^t, g_{i,k}^t, \eta_l, k)$
  end for
  $\Delta_i = \frac{n_i}{n}(x_{i,K}^t - x^t)$
  Push Δ_i to the assigned clusters using secure aggregation
end for
return x^T

$V_1, \forall x$, Assume $\|f_s(x)\|^2 \leq V_3, \forall x$. Further assuming bounded gradient dissimilarity as stated in 5.1,variance between client updates of σ_g^2 and variance in gradient estimation at each client be σ, with global and local learning rates of $\eta \leq \frac{1}{2L}$ and $\rho \geq \frac{\alpha\sqrt{\eta}}{6K^2\eta_l^2 B^2} + \eta$, after T global updates, let $F := F(x^0) - F(x^*)$, Algorithm 2 with Zeno++ as robust aggregation converges at a critical point:

$$\frac{\mathbb{E}[\sum_{t \in [T]} \|\nabla F(x_{t-1})\|^2]}{T} \leq \frac{\mathbb{E}[F]}{\alpha\sqrt{\eta}T} + \frac{\sqrt{\eta}}{\alpha}\mathcal{O}\left(\frac{\sigma_g^2}{m}\left(\frac{n-q-m}{R(n-q)-1}\right) + G^2 + \sigma^2 + V_1 + V_3 + \epsilon\right)$$

Proof. Since for any cluster update g that passes the test of Zeno++, it follows that

$$-\langle \nabla f_s(x_t), \eta g \rangle - \rho \|g\|^2 \geq -\eta\epsilon.$$

Thus, we have

$$\begin{aligned}
&\langle \nabla F(x_{t-1}), \eta \mathbb{E}[g^r] \rangle \\
&\leq \langle \nabla F(x_{t-1}) - \nabla f_s(x_t), \eta \mathbb{E}[g^r] \rangle - \rho \mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \frac{\eta}{2}\|\nabla F(x_{t-1}) - \nabla f_s(x_t)\|^2 + \frac{\eta}{2}\|\mathbb{E}[g^r]\|^2 - \rho \mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \frac{\eta}{2}\|\nabla F(x_{t-1}) - \nabla f_s(x_t)\|^2 + (\frac{\eta}{2} - \rho)\mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \frac{\eta}{2}\|\nabla F(x_{t-1}) - \nabla F(x_t) + \nabla F(x_t) - \nabla f_s(x_t)\|^2 + (\frac{\eta}{2} - \rho)\mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \eta\|\nabla F(x_{t-1}) - \nabla F(x_t)\|^2 + \eta\|\nabla F(x_t) - \nabla f_s(x_t)\|^2 + (\frac{\eta}{2} - \rho)\mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \eta\|\nabla F(x_{t-1}) - \nabla F(x_t)\|^2 + \eta V_1 + (\frac{\eta}{2} - \rho)\mathbb{E}[\|g^r\|^2] + \eta\epsilon \\
&\leq \eta^3 L^2 \mathbb{E}[\|g^r\|^2] + \eta V_1 + (\frac{\eta}{2} - \rho)\mathbb{E}[\|g^r\|^2] + \eta\epsilon
\end{aligned}$$

Where g^r is the model update in reclustering round $r \leq R$. From L smoothness, we have

$$\|\nabla F(x_{t-1}) - \nabla F(x_t)\|^2 \leq L^2 \|x_{t-1} - x_t\|^2 \leq L^2 \eta^2 \mathbb{E}[\|g^r\|^2]$$

Using smoothness again, considering a global step size as $\eta L \leq \frac{1}{2}$we get

$$\mathbb{E}[F(x_t)] \tag{1}$$

$$\leq F(x_{t-1}) + \langle \nabla F(x_{t-1}), \eta \mathbb{E}[g^r] \rangle + \frac{L\eta^2}{2} \mathbb{E}[\|g^r\|^2] \tag{2}$$

$$\leq F(x_{t-1}) + (\eta^3 L^2 + \frac{\eta}{2} - \rho + \frac{L\eta^2}{2}) \mathbb{E}[\|g^r\|^2] + \eta V_1 + \eta \epsilon \tag{3}$$

$$\leq F(x_{t-1}) + (\eta - \rho) \mathbb{E}[\|g^r\|^2] + \eta V_1 + \eta \epsilon \tag{4}$$

Now we will bound the term $\mathbb{E}[\|g^r\|^2]$. Further, $\mathbb{E}[\|g^r\|^2] \leq 2(\frac{V_3}{2} + \eta\epsilon) + \mathbb{E}\|\tilde{g}^r\|^2$ where $\|\nabla f_s(x)\|^2 \leq V_3$ and $\tilde{g}^r$ is benign average obtained through sampling of benign clients

$$\mathbb{E}\|\tilde{g}^r\|^2 \leq \mathbb{E}\|x_i^K - x_{t-1}\|^2 + \frac{\sigma_g^2}{m} (\frac{n-q-m}{R(n-q)-1}) \tag{5}$$

Where x_i^K corresponds to model parameters after K rounds of SGD on ith device, σ_g^2 corresponds to variance between device updates. Since sampling successive $g_i's$ can be seen as sampling with replacement, and at least one cluster is selected each time, this has a maximum variance of single cluster selection case. (m is the cluster size, n is the total number of devices). The mean is equal to client drift, which can be bounded as shown below (for notational brevity, present global model x_{t-1} is denoted as x). Let us assume gradients at each data point $g_i(x_i^{k-1}) = \nabla f_i(x_i^{k-1}) +$ error, where error has mean 0 and σ standard deviation as stated in Assumption 5.3. For $k \leq K$ steps of local SGD, we get

$$\begin{aligned}
\mathbb{E}\|x_i^k - x\|^2 &= \mathbb{E}\|x_i^{k-1} - x - \eta_l g_i(x_i^{k-1})\|^2 \\
&\leq \mathbb{E}\|x_i^{k-1} - x - \eta_l \nabla f_i(x_i^{k-1})\|^2 + \eta_l^2 \sigma^2 \\
&\leq (1 + \frac{1}{K-1}) \mathbb{E}\|x_i^{k-1} - x\|^2 + K\eta_l^2 \|\nabla f_i(x_i^{k-1})\|^2 + \eta_l^2 \sigma^2 \\
&\leq (1 + \frac{1}{K-1}) \mathbb{E}\|x_i^{k-1} - x\|^2 + 2K\eta_l^2 \|\nabla f_i(x_i^{k-1}) - \nabla f_i(x)\|^2 + 2K\eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2 \\
&\leq (1 + \frac{1}{K-1} + 2K\eta_l^2 L^2) \mathbb{E}\|x_i^{k-1} - x\|^2 + 2K\eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2
\end{aligned}$$

Where the first inequality uses mean and variance in gradient estimation, the second one follows from relaxed triangle inequality as stated in Karimireddy et al. [10](Lemma 3). Taking appropriate local step size $\eta_l^2 \leq \frac{1}{2L^2 K(K-1)}$ and telescoping the sum, we get

$$\begin{aligned}
\mathbb{E}\|x_i^k - x\|^2 &\leq \sum_{\tau=1}^{k-1} (2K\eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2)(1 + \frac{2}{K-1})^\tau \\
&\leq (2K\eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2) \sum_\tau (1 + \frac{2}{K-1})^\tau \\
&\leq (2K\eta_l^2 \|\nabla f_i(x)\|^2 + \eta_l^2 \sigma^2) 3K
\end{aligned}$$

The last inequality follows from the fact that $\tau < K$ and $(1 + x/n)^n < \exp(x)$. Substituting this back into (5) and averaging over all $i's$ (client devices), we get

$$\begin{aligned}
\mathbb{E}\|g^r\|^2 &\leq \frac{1}{N} 6K^2 \eta_l^2 \sum_i \|\nabla f_i(x)\|^2 + 3K\eta_l^2 \sigma^2 + \frac{\sigma_g^2}{m} (\frac{n-q-m}{R(n-q)-1}) + V_3 + 2\eta\epsilon \\
&\leq 6K^2 \eta_l^2 G^2 + 6K^2 \eta_l^2 B^2 \|\nabla F(x)\|^2 + 3K\eta_l^2 \sigma^2 + \frac{\sigma_g^2}{m} (\frac{n-q-m}{R(n-q)-1}) + V_3 + 2\epsilon\eta
\end{aligned}$$

Where $\frac{1}{N}\sum_i \|\nabla f_i(x)\|^2 \leq G^2 + B^2\|\nabla F(x)\|^2$ follows from bounded gradient assumption. Combining this with (4), we get

$$\mathbb{E}[F(x_t)] \leq F(x_{t-1})$$
$$+ (\eta - \rho)(6K^2\eta_l^2 G^2 + 6K^2\eta_l^2 B^2\|\nabla F(x)\|^2 + 3K\eta_l^2\sigma^2 + \frac{\sigma_g^2}{m}(\frac{n-q-m}{R(n-q)-1})) + \eta V_1 + V_3 + 3\eta\epsilon$$

Taking $\rho \geq \frac{\alpha\sqrt{\eta}}{6K^2\eta_l^2 B^2} + \eta$, we have

$$\|\nabla F(x_{t-1})\|^2 \leq \frac{\mathbb{E}(F(x_{t-1}) - F(x_t)))}{\alpha\sqrt{\eta}} + \frac{\sqrt{\eta}}{\alpha}\mathcal{O}\left(\frac{\sigma_g^2}{m}\left(\frac{n-q-m}{R(n-q)-1}\right) + G^2 + \sigma^2 + V_1 + V_3 + \epsilon\right)$$

Telescoping and using expectation after T global epochs, we get

$$\frac{\mathbb{E}[\sum_{t\in[T]}\|\nabla F(x_{t-1})\|^2]}{T} \leq \frac{\mathbb{E}[F]}{\alpha\sqrt{\eta}T} + \frac{\sqrt{\eta}}{\alpha}\mathcal{O}\left(\frac{\sigma_g^2}{m}\left(\frac{n-q-m}{R(n-q)-1}\right) + G^2 + \sigma^2 + V_1 + V_3 + \epsilon\right)$$

□

C Security

To summarize, the security protocol operates in multiple rounds as is the case with any secure aggregation oracle in distributed learning. Firstly, keys are shared among every pair of clients in a cluster, this is followed by collection of masked inputs among each cluster by the server, which are then averaged within the cluster after a consistency check to make sure enough participants have participated in the round. Since model parameters are 32-bit floating points we convert them to integers and perform the masking modulo 2^{32}.

In particular, each client masks its private update with random vectors such that the server, even if curious, does not learn anything more than the sum of updates from a client cluster. For a given cluster $\mathcal{M}_k, k \in [c]$ assume that $\Delta_i, \sum_{i\in\mathcal{M}_k}\Delta_j \in \mathbb{Z}_P$, for some P. Consider an order on all the clients within a cluster and each pair of users $i, j(i < j)$ agree on a random vector $r_{i,j}$. If i adds this to its updates (Δ_i) and j subtracts it from its update (Δ_j), adding them would cancel and server would learn just the average but not individual updates. Hence, each client $i \in \mathcal{M}_k$ would compute $u_i = \Delta_i + \sum_{j\in\mathcal{M}_k, i<j} r_{i,j} - \sum_{j\in\mathcal{M}_k, i>j} r_{i,j}$ (mod P). If no clients drop in the computation round, it can be seen that $\sum_{i\in\mathcal{M}_k} u_i = \sum_{i\in\mathcal{M}_k}\Delta_i$ (mod P). Further, this can be made communication efficient by coordinating a common agreement on seeds for pseudorandom generator.

C.1 Communication efficiency

Notice that sharing a whole mask has a communication cost that increases linearly with the model size and hence prevents dealing with large models. This can be circumvented by clients agreeing on common seeds for a pseudorandom generator (PRG). PRG takes in a random seed as input and generates uniformly random numbers in $[0, P)^d$ where d is the model dimension. Engaging in a key agreement after broadcasting Diffie–Hellman public keys, as stated in [3] is a way to compute these shared seeds. Hence, each client belonging to a cluster $i \in \mathcal{M}_k$ would compute $u_i = \Delta_i + \sum_{j\in\mathcal{M}_k, i<j}\text{PRG}(s_{i,j}) - \sum_{j\in\mathcal{M}_k, i>j}\text{PRG}(s_{i,j})_{i,j}$ (mod P), where $s_{i,j}$ is the shared seed between clients i, j.

C.2 Handling dropped users

Since masks are shared between clients in a cluster, dropping of a client in a round causes incorrect computation of average as the masks do not exactly cancel each other. This problem is resolved by utilizing Samir's t out of n Secret Sharing [19] to share each clients' Diffie–Hellman secret with others and hence server can retrieve masks for the dropped client. Optionally, double masking as noted in Bonawitz et al. [3] can be used to enhance security.

C.3 Privacy at Server

As noted in Section 5, server learns nothing more than the average of the clustered client updates. However, multiple reclustering rounds poses an additional privacy threat since multiple averages among same clients appear in clear to the server. We note that the server requires at least $R \geq \frac{n}{c}$ to

identify all the updates, this can be used to tune R, c. Further, $R \geq \frac{n}{c}$ does not guarantee that server learns all updates since clusterings can overlap resulting in linearly dependent equations with infinite solutions.

C.4 Privacy from curious clients

As mentioned in Section 5, at least $m-1$ malicious clients are required in a cluster to infer the update of the remaining client. In each reclustering round this probability is upper bounded by $\mathcal{O}(\frac{m(n-m)!}{(q-1-m)!})$ and hence the rest being constant, as cluster size increases, it gets harder to break a client's privacy. Further, we note that just as $n-1$ colluding curious clients can break privacy in traditional Federated Learning, $m-1$ colluding curious clients can in our approach.

C.5 Communication costs

Server: The server communication cost is $\mathcal{O}(Rnm + Rdn)$, where R, m, n, d indicate number of reclustering rounds, number of clients per cluster, total number of clients and model size respectively. Here $\mathcal{O}(Rmn)$ is associated with mediation of pairwise communication between clients in each cluster and $\mathcal{O}(Rdn)$ is for receiving masked data vectors from each user. Although reclustering increases communication costs, clustering helps reduce pairwise communications.

Client: Client communication cost is $\mathcal{O}(Rm + Rd)$. Here $\mathcal{O}(Rm)$ is associated with pairwise key exchange within a cluster over all reclustering rounds and $\mathcal{O}(Rd)$ is for communicating its model to server in every reclustering round.

Note that these are passive adversaries hence while can be curious, they honestly follow the protocol for security. Compared to the two server approach suggested in He et al. [8], our approach can handle attacks on the server, because if an adversary attacks the server(s) or can see communication channels, model updates still remain private.

D Additional Experiments

We use a global learning rate $\eta = 1$, local learning rate $\eta_l = 0.01$, local momentum 0.9, and mini-batch size 64. We run each experiment for 200 global rounds with 2 local rounds each and report top-1 accuracy on the testing set. For all the experiments, unless specified, we use $R = 10$ reclustering rounds.

For Zeno++, we randomly sample 5% of the training data across clients with the same number of samples of each label to use as the server-side validation set as in Xie et al. [24]. We consider batch size of 128, $\rho = 0.0001, \epsilon = 0.2$ as Zeno++ parameters.

In experiments on Shakespear, we use $\eta = 1$, $\eta_l = 1$, local momentum 0.9, and mini-batch size 256. We run each experiment for 100 global rounds with 2 local rounds each. For all the experiments, unless specified, we use $R = 10$ reclustering rounds. Codes for the same would be made available soon.

D.1 Impact of cluster size

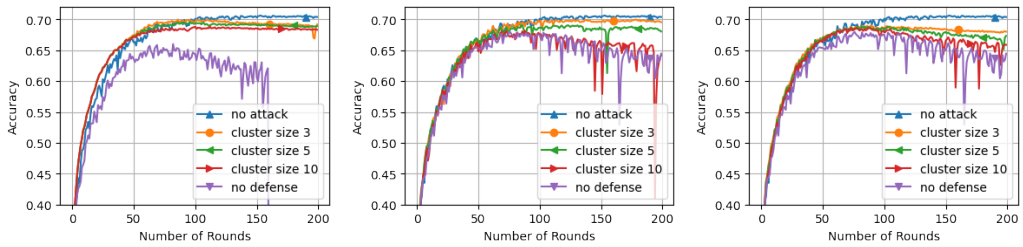
D.1.1 Fall of Empires Attack

We now test the sensitivity of cluster size on Byzantine-tolerance to a stronger attack with colluding adversaries, we utilize a modified version of Fall of Empires (FoE) [23]. In particular, each malicious client sends a negatively scaled averaged model update across all malicious clients. We test scaling these updates by $\beta = -1, -10$. Since Zeno++ can tolerate a greater fraction of clients being Byzantine, we set $q = 6$ while for trimmed mean and Krum, we set $q = 3$. (Parameters for defenses remain the same as in Section 6 unless specified.)

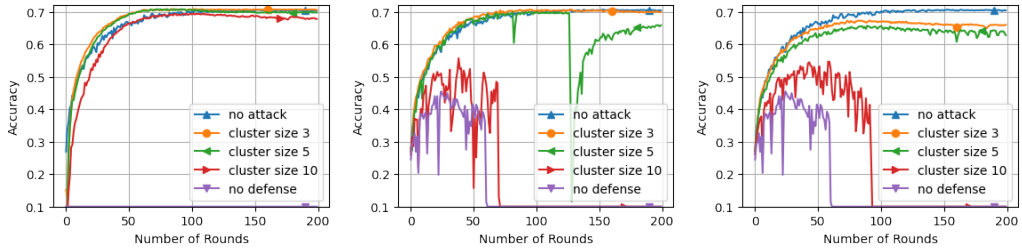
The results are plotted in Figure 5. A similar effect of cluster sizes as discussed in Section 6 can be seen here. Further, Zeno++ ,as expected, is stable even at higher levels of corruption.

D.2 Effect of Reclustering with Non-IID data

Empirically, we test the effect of reclustering on a heterogeneous data distribution. In particular, we divide CIFAR-10 among clients such that each one gets data from a few classes. In all experiments we use cluster size of 3.



(a) Zeno++, $\beta = -1, q = 6$ (b) Krum, $\beta = -1, q = 3$ (c) Trmean, $\beta = -1, q = 3$

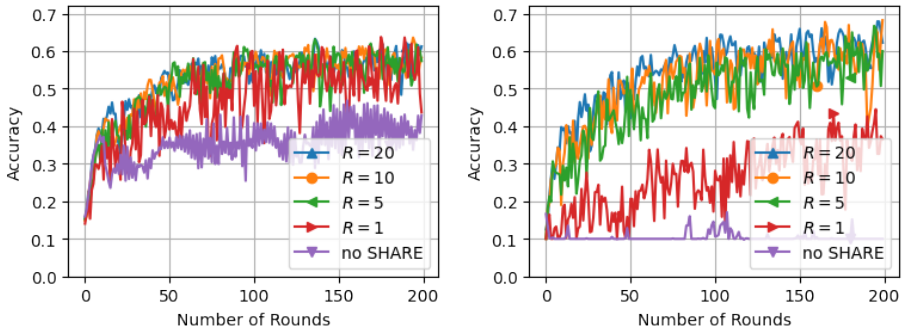


(d) Zeno++, $\beta = -10, q = 6$ (e) Krum, $\beta = -10, q = 3$ (f) Trmean, $\beta = -10, q = 3$

Figure 5: Results of SHARE on CIFAR-10, with varying cluster size across various defenses under FoE attack and q malicious clients out of 60 total clients. Malicious clients send their average update scaled by β as indicated in the subfigures. For trimmed mean (Trmean) we remove 2/3 of input updates and use batch size of 128, $\rho = 0.00001, \epsilon = 0.2$ as Zeno++ parameters.

D.2.1 No Attack

To create heterogeneous data effect, we split CIFAR-10 data across 60 clients such that each client gets only data consisting of 2 labels. As can be seen in Figure 6, robust defenses such as coordinate wise median[16] and Krum[2] fail in this setting, while SHARE with these defenses performs better as the number of reclustering rounds increases. Further, variance reduction is observed as we increase reclustering rounds (R).

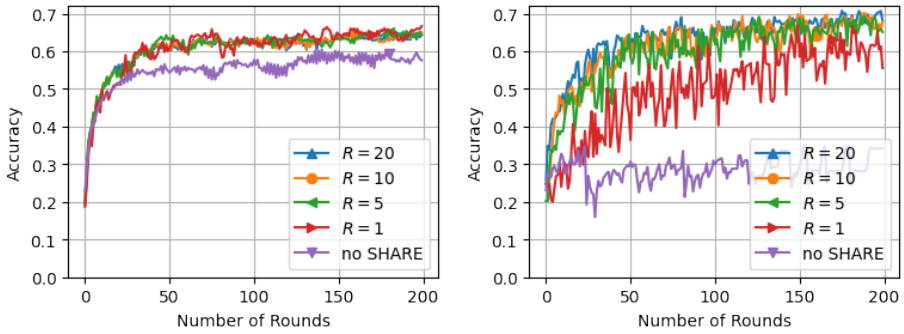


(a) Median (b) Krum

Figure 6: Results of SHARE on non-IID CIFAR-10, where each client has data from 2 labels, across various defenses without attack with 50 clients. We vary the number of reclustering rounds (R). No SHARE indicates the baseline defense without the SHARE framework.

D.2.2 Fall of Empires Attack

We consider a lower level of heterogeneity but with client corruption. In particular, each client gets data consisting of 5 labels and $q = 3$ malicious clients which collude to send their average model update scaled by $\beta = -10$. The results are shown in Figure 7 for various number of reclustering rounds (R).

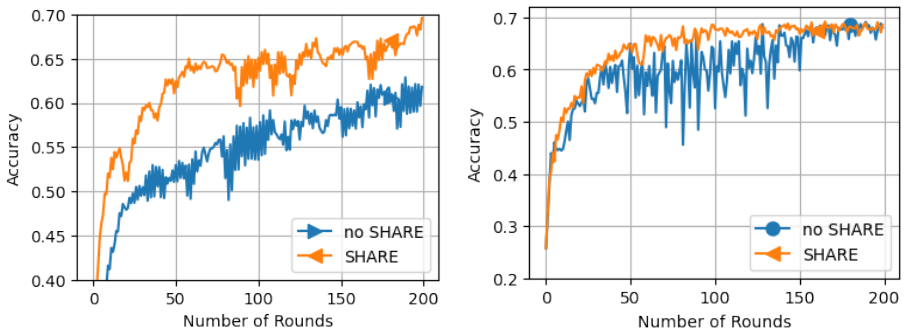


(a) Median (b) Krum

Figure 7: Results of SHARE on non-IID CIFAR-10, where each client has data from 5 labels, across various defenses with FoE attack ($\beta = -10$) and $q = 3$ malicious clients out of 50 total. We vary the number of reclustering rounds (R). No SHARE indicates the baseline defense without the SHARE framework.

D.2.3 Label Flip Attack

We test the effect of SHARE on the label-flip attack with a heterogeneous data split of CIFAR-10. Each client gets data from 5 labels. Malicious clients train on flipped labels, i.e. any label $\in \{0, \ldots, 9\}$ is changed to $9-$label. We test trimmed mean (filtering $2/3$ of input updates) and Zeno++ with batch size of 128, $\rho = 0.00001, \epsilon = 1$ with SHARE and use $\epsilon = 5$ without SHARE framework. These parameters are tuned to achieve good performances within their respective frameworks. We consider $R = 10$ reclustering rounds and $q = 12$ malicious clients. Results as shown in Figure 8 demonstrate the efficacy of SHARE.

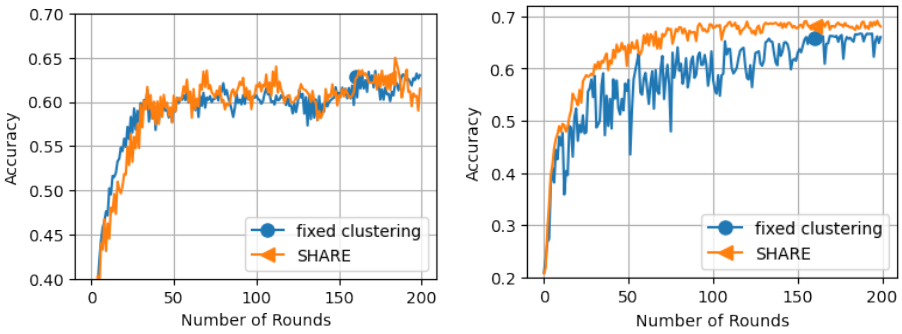


(a) Trimmed mean (b) Zeno++

Figure 8: Results of SHARE on non-IID CIFAR-10, where each client has data from 5 labels, across various defenses with label-flip attack, $q = 12$ malicious clients out of 60 total, and $R = 10$ reclustering rounds. No SHARE indicates the baseline defense without the SHARE framework.

D.2.4 Fixed clustering vs reclustering

Figure 9 compares results between fixed clustering and SHARE. In the former, we fix the client clusters before the learning process starts while the latter allows for random reclustering in every round. CIFAR-10 data is split heterogeneously such that each client receives 5 class labels alone. We use Fall of Empires attack with $\beta = -10$ scaling of the average gradients from the malicious clients. Since Zeno++ can tolerate higher levels of corruption, we consider $q = 6$ malicious clients, while for trimmed mean, we consider $q = 3$. We use $R = 10$ reclustering rounds and 60 clients in total with a cluster size of 3. We test trimmed mean (filtering $2/3$ of input updates) and Zeno++ with batch size of 128, $\rho = 0.00001, \epsilon = 1$ with SHARE and use $\epsilon = 5$ for fixed clustering case. These parameters are tuned to achieve reasonable performances within their respective frameworks.



(a) Trimmed mean, $q = 3$

(b) Zeno++, $q = 6$

Figure 9: Results of SHARE on non-IID CIFAR-10, where each client has data from 5 labels, across various defenses with FoE attack ($\beta = -10$) and q malicious clients out of 60 total. Fixed clustering indicates the baseline defense with fixed clusters of size 5.

E Random Reclustering

In the worst case, a benign client's signal is lost if it is clustered with a malicious client across all reclustering rounds. In particular, the probability that a benign client is effected in a global round is $\left(1 - \frac{\binom{n-q}{\frac{n}{m}-1}}{\binom{n}{\frac{n}{m}-1}}\right)^R$. Hence this probability decreases as R (number of reclustering rounds increase).