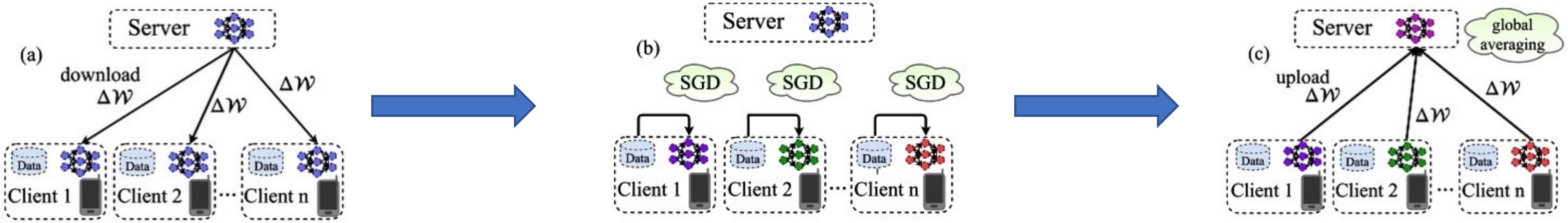
Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data

privacy-preserving collaborative learning, comes at the cost of a significant communication overhead during training



The total amount of bits that have to be uploaded and downloaded by every client during training is given by

$$b^{up/down} \sim O(N_{iter} * f * |W| * (H(\Delta W^{up/down}) + \eta))$$

Where,

N_{iter} = total number of training iteration
f = communication frequency
$|W|$ = size of the model
$\Delta W = W_{new} - W_{old}$
$H(\Delta W^{up/down})$ = entropy of the weight updates exchanges during the upload & download
η = inefficiency of the encoding

Possible Solutions

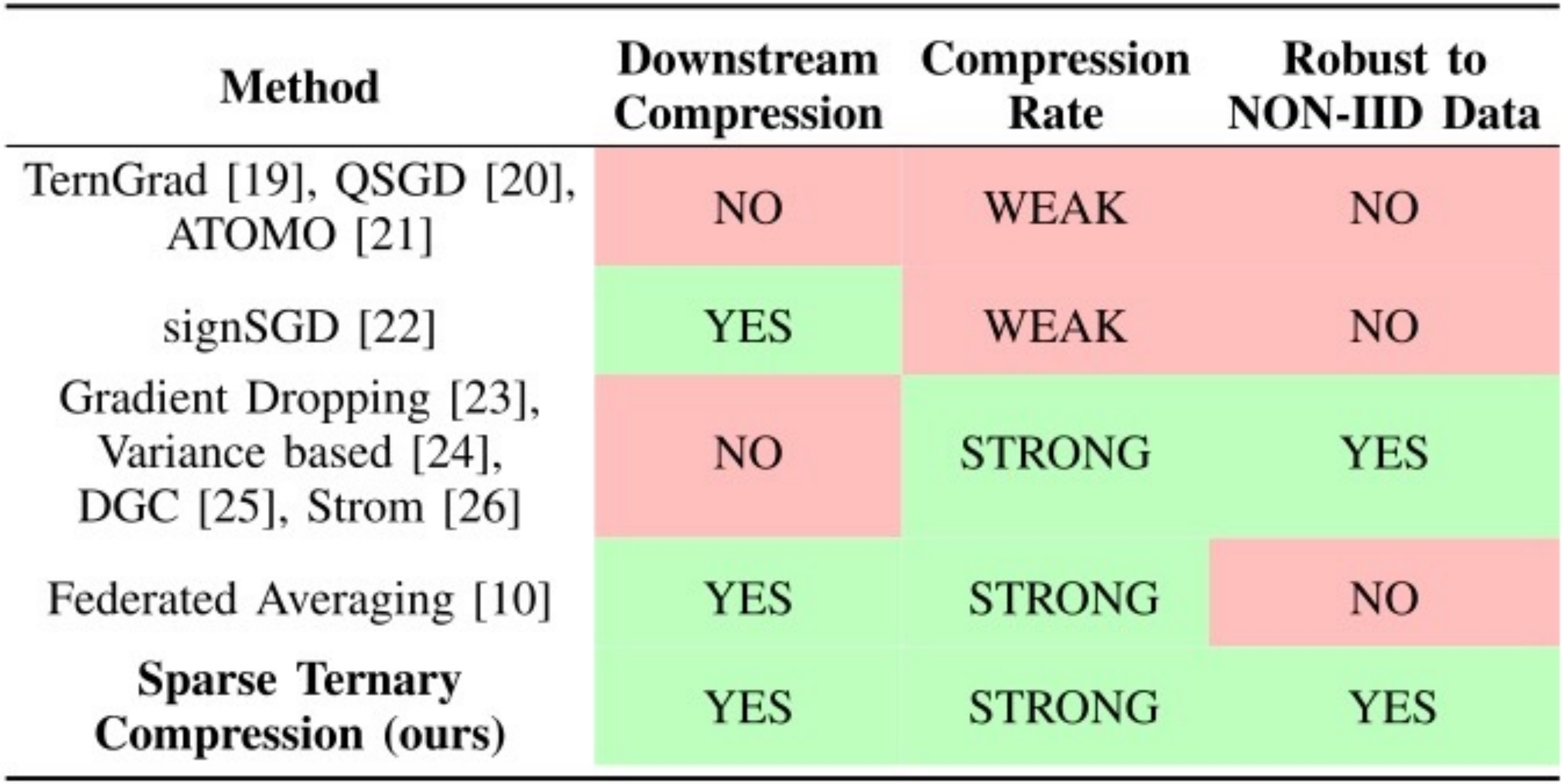
We have three options to reduce the communication keeping in mind N_{iter} & $|W|$ to be fixed:

- we can reduce the communication frequency f
- reduce the entropy of the weight updates $H(\Delta W^{up/down})$ via lossy compression schemes
- use more efficient encodings to communicate the weight updates, thus reducing η.

they come up with a framework
STC (sparse ternary compression)
which can address

- It should compress both upstream and downstream communications.
- It should be robust to non-i.i.d., small batch sizes, and unbalanced data.
- It should be robust to large numbers of clients and partial client participation.

Comparison from Existing methods



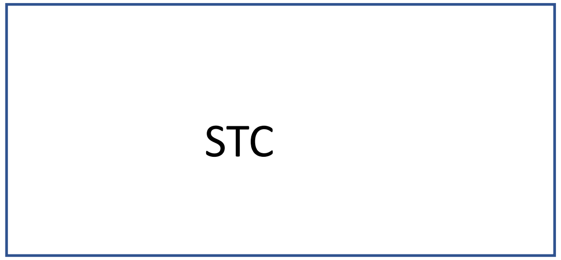
Method	Downstream Compression	Compression Rate	Robust to NON-IID Data
TernGrad [19], QSGD [20], ATOMO [21]	NO	WEAK	NO
signSGD [22]	YES	WEAK	NO
Gradient Dropping [23], Variance based [24], DGC [25], Strom [26]	NO	STRONG	YES
Federated Averaging [10]	YES	STRONG	NO
Sparse Ternary Compression (ours)	YES	STRONG	YES

Sparsification

- a methods reduce the entropy *H*(ΔW) of the updates by restricting changes to only a small subset of the parameters.
- only gradients with a magnitude greater than a certain predefined threshold are sent to the server. All other gradients are accumulated in a residual.
- This method is shown to achieve upstream compression.

Method (STC):

- Input : flattened tensor $T \in R^n$, sparsity p

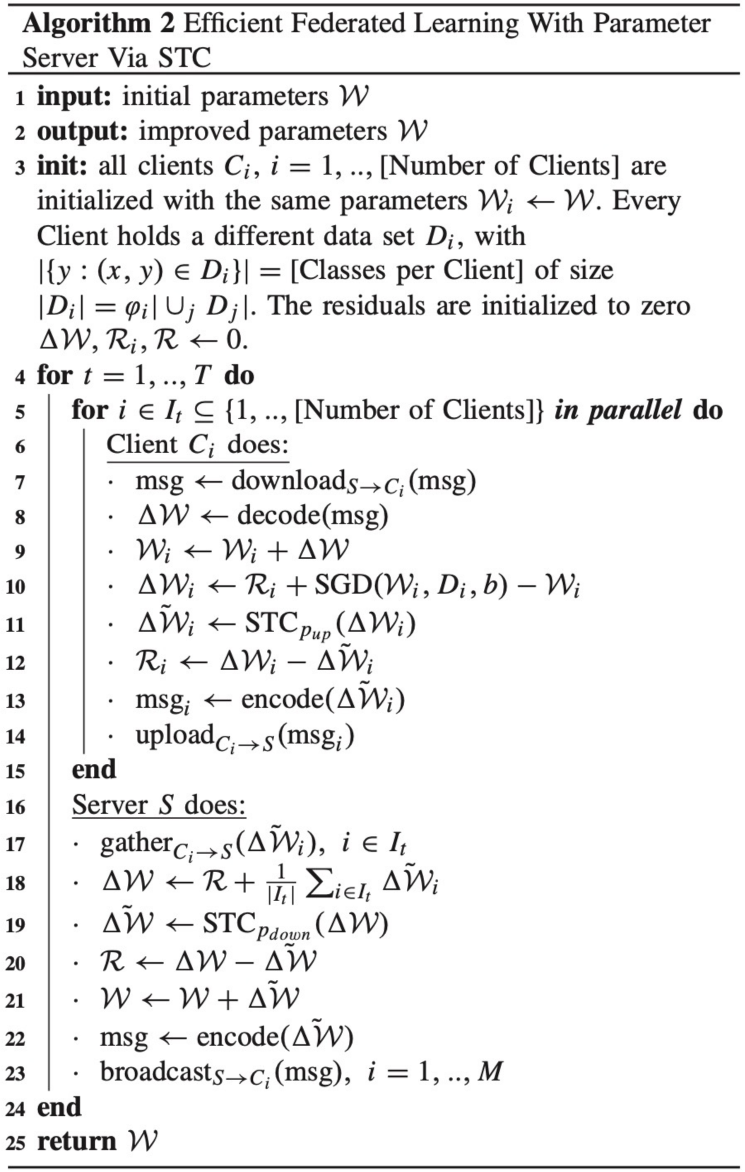


STC

- Output : flattened tensor $T^* \in \{-\mu, 0, \mu\}^n$

Note : To communicate a set of sparse ternary tensors produced by STC, we only need to transfer the positions of the nonzero elements in the flattened tensors, along with one bit per nonzero update to indicate the mean sign μ or $-\mu$.

Process



Algorithm 2 Efficient Federated Learning With Parameter Server Via STC

```
input: initial parameters $\mathcal{W}$
output: improved parameters $\mathcal{W}$
init: all clients $C_i$, $i = 1, .., $[Number of Clients] are
initialized with the same parameters $\mathcal{W}_i \leftarrow \mathcal{W}$. Every
Client holds a different data set $D_i$, with
$|\{y : (x, y) \in D_i\}| =$ [Classes per Client] of size
$|D_i| = \varphi_i |\cup_j D_j|$. The residuals are initialized to zero
$\Delta\mathcal{W}, \mathcal{R}_i, \mathcal{R} \leftarrow 0$.
for $t = 1, .., T$ do
    for $i \in I_t \subseteq \{1, .., $[Number of Clients]$\}$ in parallel do
        Client $C_i$ does:
        · msg $\leftarrow$ download$_{S \rightarrow C_i}$(msg)
        · $\Delta\mathcal{W} \leftarrow$ decode(msg)
        · $\mathcal{W}_i \leftarrow \mathcal{W}_i + \Delta\mathcal{W}$
        · $\Delta\mathcal{W}_i \leftarrow \mathcal{R}_i + \text{SGD}(\mathcal{W}_i, D_i, b) - \mathcal{W}_i$
        · $\Delta\tilde{\mathcal{W}}_i \leftarrow \text{STC}_{p_{up}}(\Delta\mathcal{W}_i)$
        · $\mathcal{R}_i \leftarrow \Delta\mathcal{W}_i - \Delta\tilde{\mathcal{W}}_i$
        · msg$_i \leftarrow$ encode($\Delta\tilde{\mathcal{W}}_i$)
        · upload$_{C_i \rightarrow S}$(msg$_i$)
    end
    Server $S$ does:
    · gather$_{C_i \rightarrow S}(\Delta\tilde{\mathcal{W}}_i)$, $i \in I_t$
    · $\Delta\mathcal{W} \leftarrow \mathcal{R} + \frac{1}{|I_t|} \sum_{i \in I_t} \Delta\tilde{\mathcal{W}}_i$
    · $\Delta\tilde{\mathcal{W}} \leftarrow \text{STC}_{p_{down}}(\Delta\mathcal{W})$
    · $\mathcal{R} \leftarrow \Delta\mathcal{W} - \Delta\tilde{\mathcal{W}}$
    · $\mathcal{W} \leftarrow \mathcal{W} + \Delta\tilde{\mathcal{W}}$
    · msg $\leftarrow$ encode($\Delta\tilde{\mathcal{W}}$)
    · broadcast$_{S \rightarrow C_i}$(msg), $i = 1, .., M$
end
return $\mathcal{W}$
```

Check List

- Data heterogeneity : yes
- System heterogeneity : Yes
- Model accuracy : improves on validation set.
- Use of synthetic data : No, uses already existing datasets (CIFAR-10, MNIST)
- Data transfer : No (weights are transfer)

Conclusion

- This approach can be understood as an alternative paradigm for communication-efficient federated optimization that relies on high-frequent low-volume instead of low-frequent high- volume communication.
- STC converges faster than federated averaging both with respect to the number of training iterations and the amount of communicated bits even
- STC, a communication protocol that compresses both the upstream and downstream communications via sparsification, ternarization, error accumulation, and optimal Golomb encoding