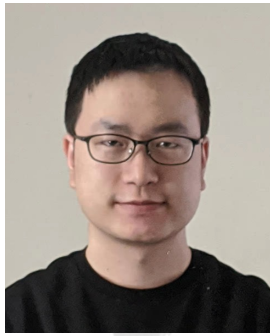
BOBA: Byzantine-Robust Federated Learning with Label Skewness

Wenxuan Bao



Jun Wu

Jingrui He

University of Illinois Urbana-Champaign

`{wbao4,junwu3,jingrui}@illinois.edu`

`baowenxuan.github.io, publish.illinois.edu/junwu3, hejingrui.org`

Federated Learning

- **Federated Learning** (FL): n clients collaborate to train a machine learning model under the orchestration of a central server, without sharing their raw data.

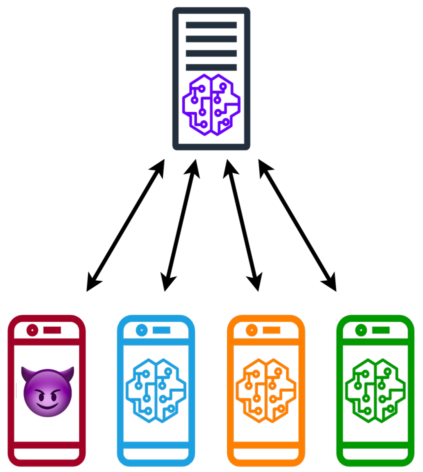
- *FL systems are vulnerable to attacks and failures* [1,2].
  - Some clients may have corrupted data or upload malicious gradients.
  - These behaviors can cause sub-optimal convergence, or even divergence.

[1] Peter Kairouz, et al.: Advances and Open Problems in Federated Learning. Found. Trends Mach. Learn. 2021.
[2] Lingjuan Lyu, et al.: Privacy and robustness in federated learning: Attacks and defenses. 2020.

FedSGD: A Prototype Framework of FL

- **FedSGD** [1]: In each communication rounds,
  1. The server broadcast the parameter $\boldsymbol{w}_G$ to all clients.
  2. Each **honest client** $i \in \mathcal{H}$ computes the gradient $\boldsymbol{g}_i$ based on local data and sends the honest gradient to the server.
  3. Each **Byzantine client** $i \in \mathcal{B}$ sends arbitrary *Byzantine gradient* to the server, due to failures or attacks.
  4. The server aggregates all n gradients $\hat{\boldsymbol{\mu}} = \mathrm{Agg}(\{\boldsymbol{g}_i\}_{i=1}^{n})$ and update the model $\boldsymbol{w}_G \leftarrow \boldsymbol{w}_G - \eta\hat{\boldsymbol{\mu}}$

     Example: Average aggregation: $\mathrm{Agg}(\{\boldsymbol{g}_i\}_{i=1}^{n}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i$



[1] Brendan McMahan, et al.: Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017.

Robust Aggregation Rules (AGRs)

- Robust AGRs replaces the Average aggregation with a robust estimator of the true gradient $\mathbb{E}\boldsymbol{\mu} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \mathbb{E}\boldsymbol{g}_i$.
- Example: One-dimensional aggregation,
  - Data: -0.4, -0.2, -0.1, 0.0, 0.1, 0.2, 0.4, 100.0 (Byzantine)
  - True mean: 0.0 Average: 12.5 $\gg$ 0.0 Median: 0.05 $\approx$ 0.0
- Previous works mostly assume IID clients: $\mathbb{E}\boldsymbol{g}_i = \mathbb{E}\boldsymbol{g}_j, \forall i, j \in \mathcal{H}$
- Our work: Non-IID clients with different label distributions,

$$\mathbb{E}\boldsymbol{g}_i \neq \mathbb{E}\boldsymbol{g}_j$$

c-Label Skewness

- **c-label skew distribution**: Data distribution for each honest client $i \in \mathcal{H}$ can be expressed as

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

where

- $P_i(\xi)$ is the data distribution of client i,
- The label z can take c finite values,
- $p_{iz} \geq 0$ is the label distribution of client i subject to $\sum_{z=1}^{c} p_{iz} = 1$,
- $Q_z(\xi) = P_i(\xi \mid z)$ represents the conditional distribution given z.
- Different clients share the same $\{Q_z(\xi)\}_{z=1}^{c}$ but different $\boldsymbol{p}_i = [p_{i1}, \cdots, p_{ic}]^\top$.

Expectation of Honest Gradients

- **Proposition 3.3.** With c-label skew distribution we have

$$\mathbb{E}\boldsymbol{g}_i = \sum_{z=1}^{c} p_{iz}\mathbb{E}\boldsymbol{\gamma}_z, \quad \forall i \in \mathcal{H}$$

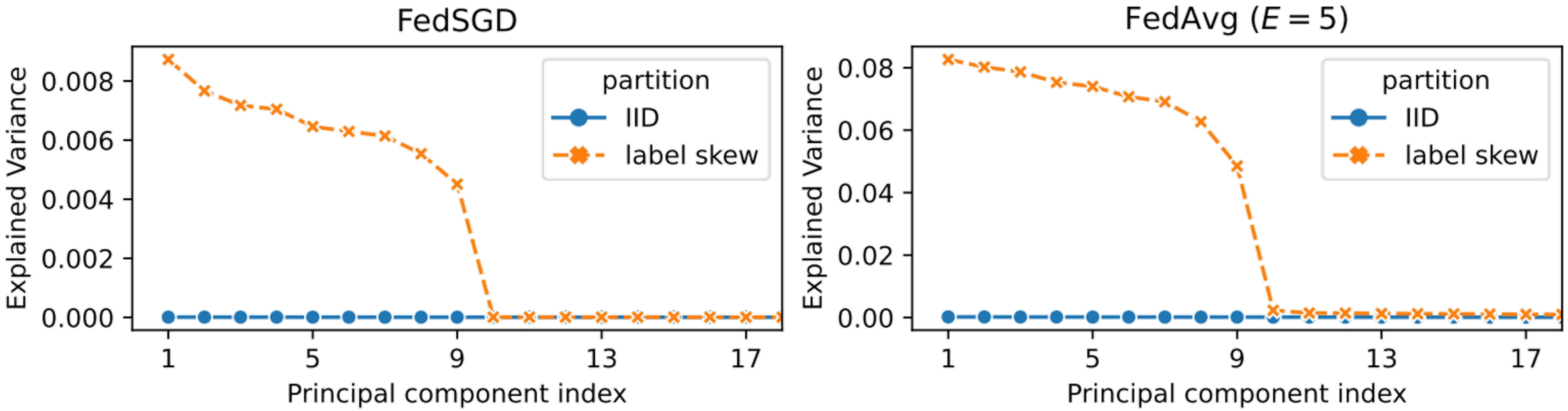
  where

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

- We define
  - **Honest simplex**: $\{\sum_{z=1}^{c} p_z \mathbb{E}\boldsymbol{\gamma}_z : \sum_{z=1}^{c} p_z = 1, p_z \geq 0\}$
  - **Honest subspace**: $\{\sum_{z=1}^{c} p_z \mathbb{E}\boldsymbol{\gamma}_z : \sum_{z=1}^{c} p_z = 1\}$

Distribution of Honest Gradients

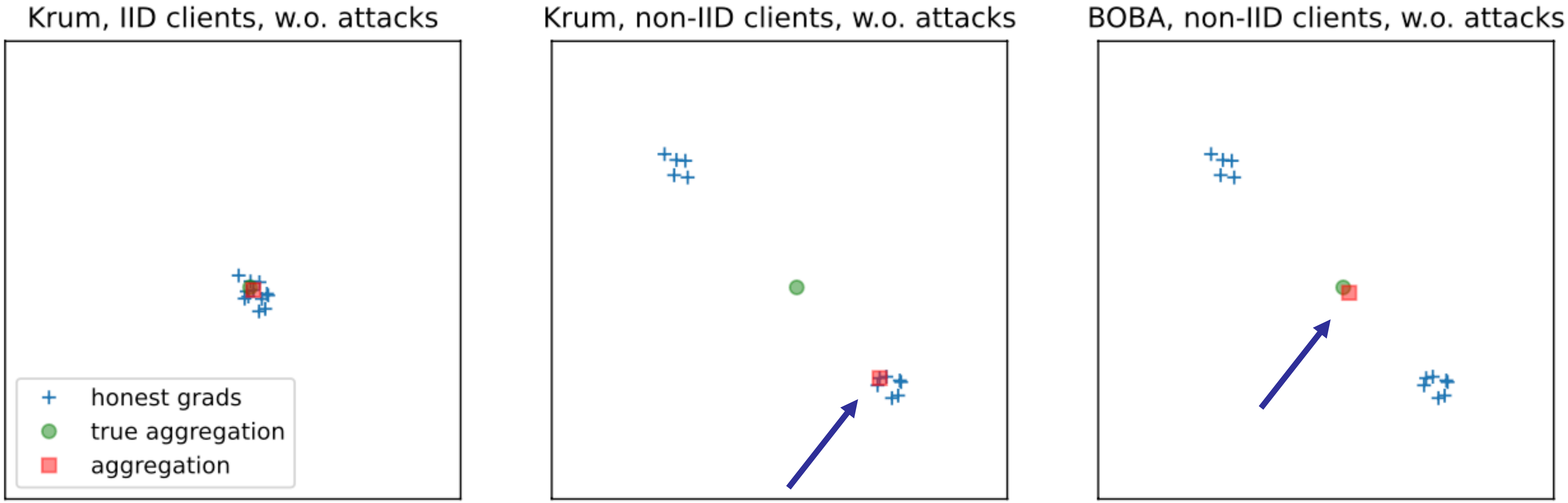
- Our findings:
  - Honest gradient's expectations distribute on the honest simplex.
  - Honest gradients distribute near the honest simplex.



- Empirical verification: PCA of honest clients on MNIST ($c = 10$). Over 99% of the variance concentrate on the first $(c-1)$ principal components

Challenge 1: Selection Bias

- **Selection Bias**: Robust AGRs are biased to certain clients, even in the absence of any attacks.

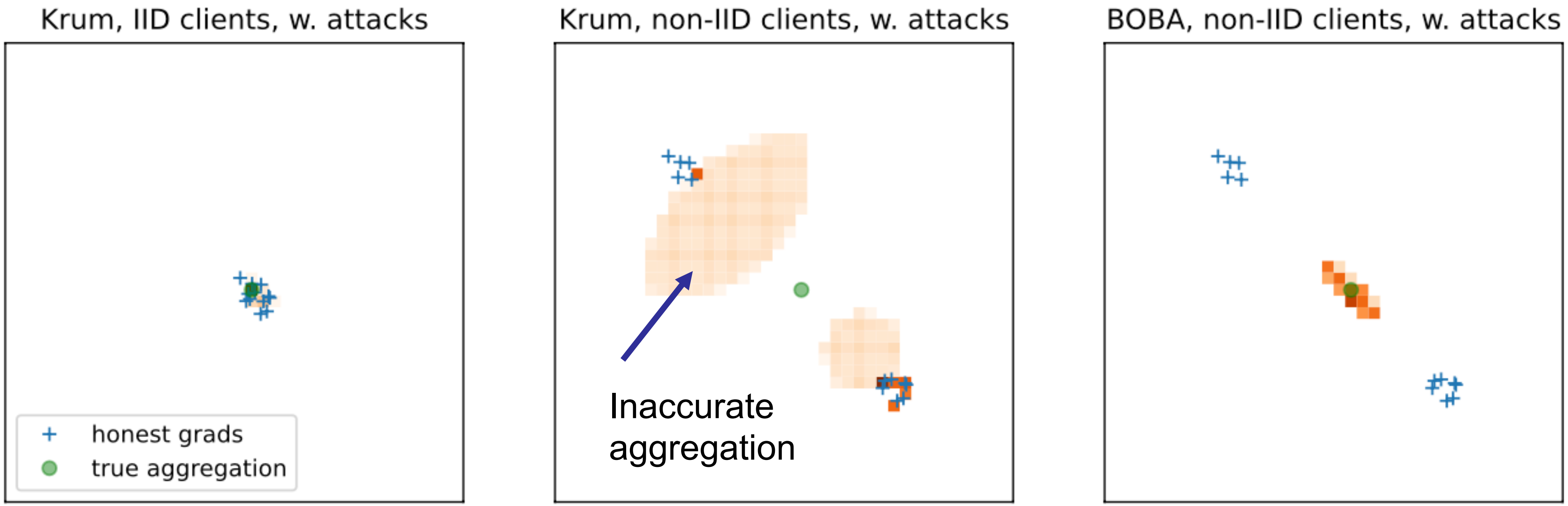


When applied to non-IID clients, Krum [1] is biased to the majority of clients. BOBA is unbiased.

[1] Peva Blanchard, et al.: Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NeurIPS 2017.

Challenge 2: Increased Vulnerability

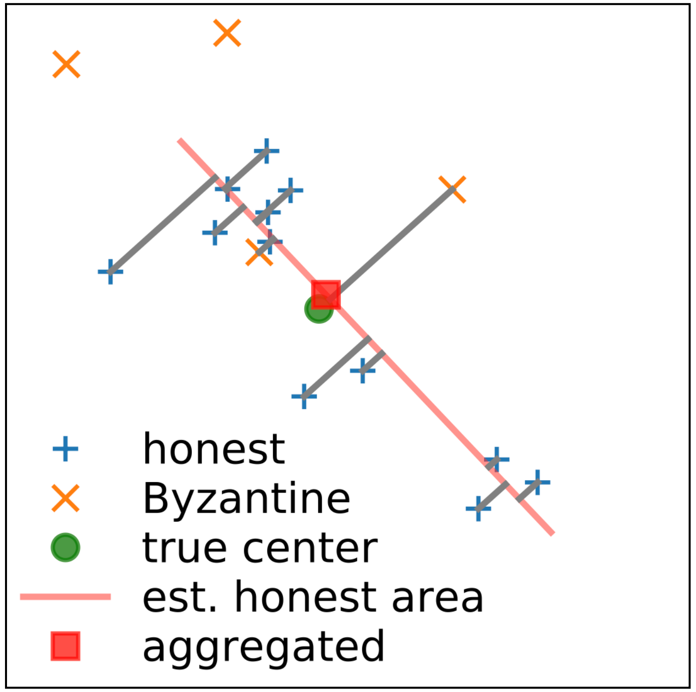
- Increased vulnerability: Robust AGRs can deviate more from the center in all directions.



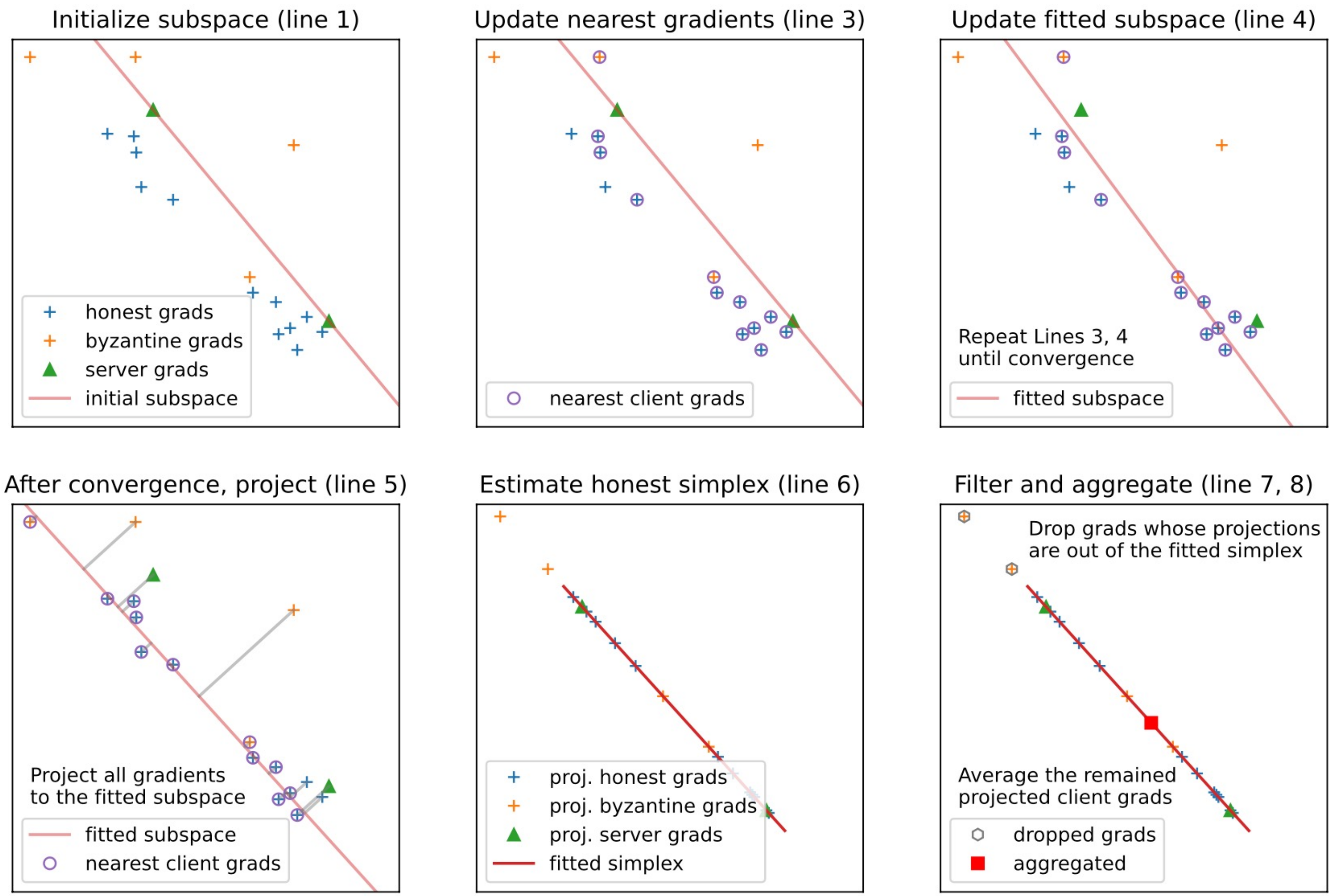
Orange region is the range of aggregations given different Byzantines.

Our Method: BOBA

- Byzantine-rObust and unBiased Aggregator
  - Stage 1: Fitting the honest subspace, and projecting all gradients to this subspace
  - Stage 2: Finding the honest simplex, reconstructing the label distribution for each client, and dropping clients with abnormal label distribution.
- All honest gradients are kept
- Byzantine gradientss are either weakened (in stage 1) or discarded (in stage 2).



BOBA: Overview



BOBA Stage 1: Fitting the Honest Subspace

- Vanilla truncated singular value decomposition (TrSVD) can fit a subspace, by minimizing the reconstruction loss

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

  where

  - $\mathcal{P}$ is the fitted subspace,
  - $\Pi_{\mathcal{P}}$ is a projection function that projects vectors to $\mathcal{P}$.

- However, TrSVD is vulnerable to Byzantine attacks.

BOBA Stage 1: Fitting the Honest Subspace

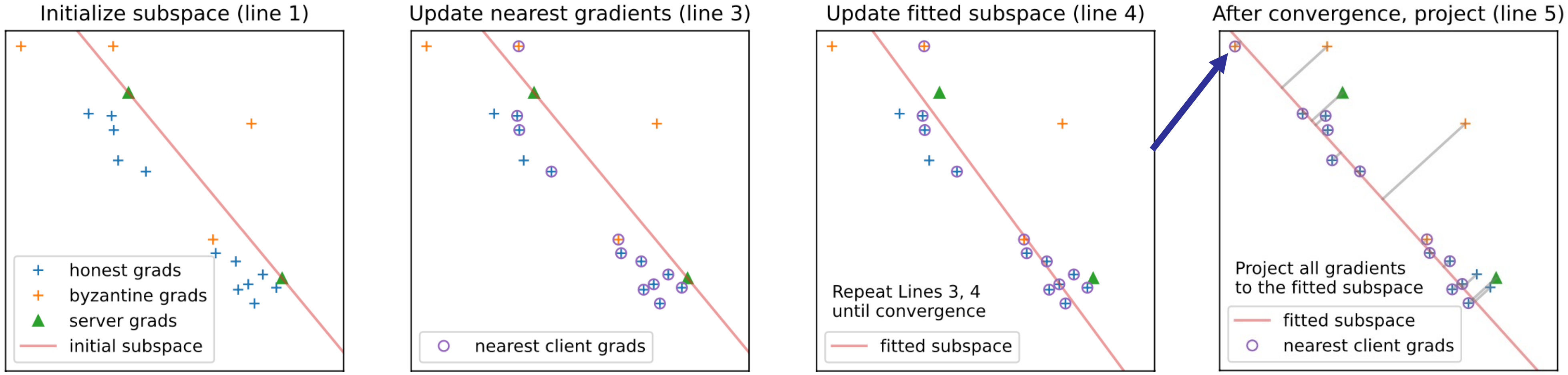
- Instead, we minimize the *trimmed reconstruction loss*

$$\hat{\mathcal{P}}, \hat{\boldsymbol{r}} = \underset{\mathcal{P}, \boldsymbol{r} \in \{0,1\}^n}{\arg\min}\ \ell_t(\mathcal{P}, \boldsymbol{r}) = \sum_{i=1}^{n} r_i \left\| \boldsymbol{g}_i - \Pi_{\mathcal{P}}(\boldsymbol{g}_i) \right\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{n} r_i = n - f$$

  which ensures robustness by dropping f gradients furthest from $\mathcal{P}$.

- We use *alternating optimization* to minimize the objective
  - Update nearest gradients: Fixing $\mathcal{P}$, the optimal $\boldsymbol{r}$ selects the $n - f$ nearest neighbors of $\mathcal{P}$.
  - Update fitted subspace: Fixing $\boldsymbol{r}$, the optimal $\mathcal{P}$ can be fitted by conducting TrSVD on the selected $n - f$ gradients.

BOBA Stage 1: Fitting the Honest Subspace



- After fitting the honest subspace, project all gradients to it.
- Some Byzantine gradients might be close to the honest subspace, but far from the honest simplex…

BOBA Stage 2: Finding the Honest Simplex

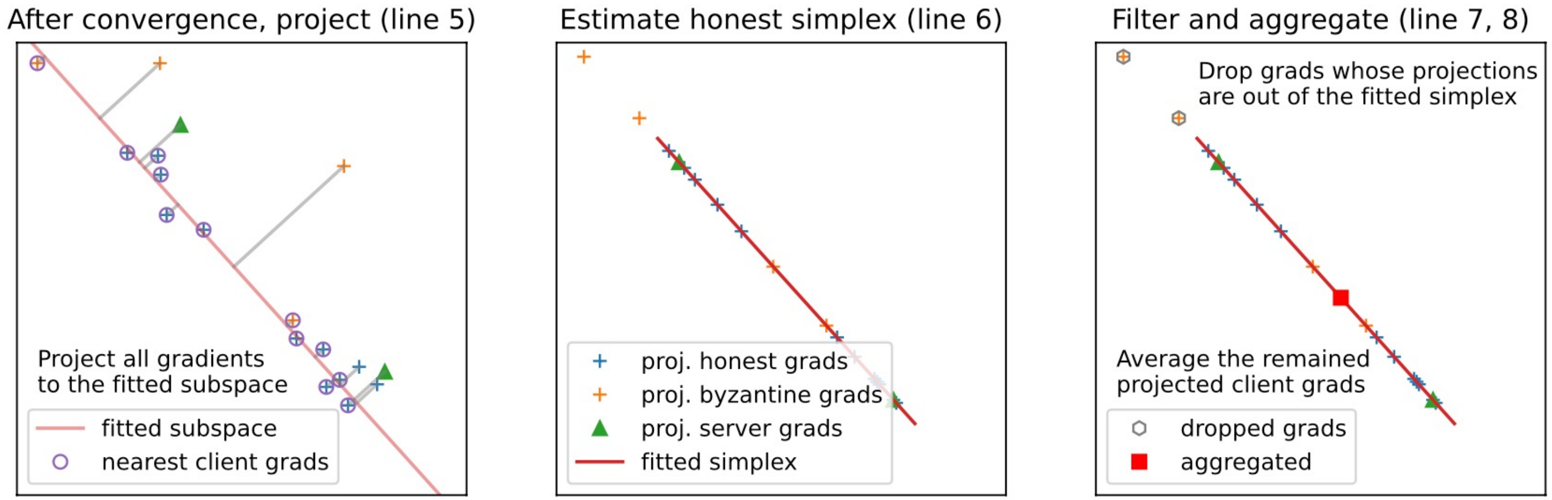
- Use server data to estimate c vertices of the honest simplex.
- Estimate the label distribution for each client i, solve for $\{\hat{p}_{iz}\}_{z=1}^{n}$

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

  - Honest client: $\hat{p}_{iz} \approx p_{iz} \geq 0$, all entries should be positive or close to 0.
  - Byzantine client: $\hat{p}_{iz}$ can be arbitrary.
- We drop a client if its estimated label distribution has strongly negative entries.

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

BOBA Stage 2: Finding the Honest Simplex



- Average the remained projected client gradients as the aggregation.

Computation Complexity

- The computation complexity of BOBA is $\mathcal{O}(kcnd)$, where
  - k is the times conducting TrSVD, which is small in our experiments,
  - c is the number of classes,
  - n is the number of clients,
  - d is the dimension of gradients.

- When k, c are small constants, BOBA has the same order of complexity as vanilla Average.

Bounded Gradient Estimation Error

- **Theorem 5.5.** BOBA has bounded gradient estimation error of

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

  where

  - $\beta = \frac{|\mathcal{B}|}{n}$ is the fraction of Byzantine clients,
  - ϵ, ϵ_s are inner variations (from randomness of data sampling).
  - δ_s is outer variation from non-IIDness.
  - C_1, C_2, C_3 are constants.

- BOBA is unbiased and has optimal order robustness.

Experiment Setup

- Three scenarios:
  - 3-layer MLP for MNIST
  - 5-layer CNN for CIFAR-10
  - GRU network for AG-News

- Pathological partition: each client only has two classes of data.

Experiments: Unbiasedness

- BOBA has accuracy very close to Average, and the smallest MRD among all robust AGRs.
  - MRD: max-recall-drop, smaller is better.

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

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

Experiments: Robustness

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

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

- BOBA significantly improves the worst-case accuracy by 6.1%, 18.3%, 1.6% on three datasets, respectively.

Summary

- **Insights**: We make a systematic analysis of FL robustness challenges under label skewness, including the identification of two key challenges: selection bias and increased vulnerability.
- **Algorithm**: We introduce BOBA which addresses both label skewness and robustness.
- **Theoretical analysis**: We derive bounded gradient estimation error and convergence guarantee for BOBA.
- **Extensive experiments**: We evaluate the unbiasedness of robustness of BOBA across diverse scenarios.