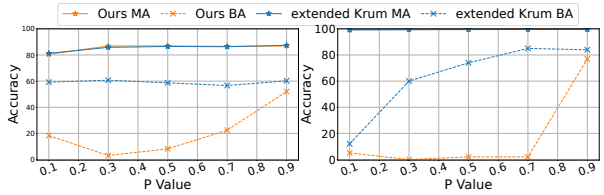
A. Additional expertiment

A.1. Impact of Different Fraction of Clients Selected for Aggregation

In each round of federated training, our method distinguishes the benign gradients from the malicious ones through multiple metrics and dynamic scoring. Then benign gradients are used to perform aggregation while the malicious ones are discarded without impacting the global model, as described in Section 3.4. In practice, we set a fixed ratio $p(p \in [0, 1])$ to denote the fraction of the selected gradients. At each round, p percentage of gradients are deemed benign and participate in the FedAvg aggregation. Intuitively, the performance of the model and the convergence speed of training are positively related to p. In contrast, the relationship between the accuracy of backdoor tasks and p is much more complicated. On the one hand, increasing the value of p would increase the probability of selecting the backdoor gradient for training, which is not beneficial for defending against backdoor attacks. On the other hand, increasing the value of p will mitigate the impact of selecting the backdoor gradient, which is beneficial for the defense. However, because of an absence of knowledge regarding the attacker (*e.g.* the number of attackers), optimal p cannot simply be determined. In this case, what is most essential is that the defense performance is invariant (or often invariant) to the choice of p, which we empirically prove below. By conducting experiments on CIAFR10 and EMNIST with the Edge-case PGD attack, we show that our outstanding defense performance does not heavily rely on the tuning of p. Results in Figure 8 show that the optimal p is at 0.3 but p between 0.1 and 0.7 provides consistently low BA, where our approach has solid defensive effectiveness. Krum and Multi-Krum also select some clients to aggregate, which we make a comparison in terms of defense performance with our proposed method under different p. We demonstrate that our success against this attack does not rely on an optimal p and consistently gives better results against the Krum and Multi-Krum with different p values.



(a) CIFAR10 (b) EMNIST

Figure 8: Accuracy(%) of our defense and extended-Krum under Edge-case PGD attack versus the value of p, where p denotes the fraction of clients selected for aggregation by our method.

A.2. Ablation on Different Definitions for Outlier Detection

We also conduct an ablation study on the definition, as introduced by Equation 3. We perform experiments and compare the results of the simple approach (*i.e.* use the deviation from the mean to detect the outlier as a malicious gradient). As the results in Table 5, we observe that the simple approach is already effective under different attacks. Specifically, with a simple approach, our defense can achieve 7.14% BA under Edge-case PGD which already outperforms the previous state-of-the-art methods by almost 40%, demonstrating the effectiveness of our multi-metrics adaptive defense method. By incorporating the new definition, we achieve the lowest 3.06%. Results under other attacks are also consistent. We note that our new definition increases the overall performance (*i.e.* MA) under the three attacks consistently. This demonstrates that our defense can aggregate the truly benign gradients since this new definition helps further identify the malicious gradients.

Table 5: Impact of different definitions for outlier detection against various attacks on CIFAR10.

Defense	Model Replacement		PGD		Edge-case PGD	
	MA ↑	BA ↓	MA ↑	BA ↓	MA ↑	BA ↓
Mean	85.91	**0.56**	85.83	1.67	85.82	7.14
Ours	**86.34**	**0.56**	**86.44**	**0.56**	**86.86**	**3.06**

Table 6: Computational measured in seconds. We report the increment over FedAvg.

Defense	**FedAvg**	Krum	RFA	Foolsgold
Computational Cost (s)	**24.62**	25.98(+1.36)	31.08(+6.46)	78.67(+54.05)
Defense	Single metric	Dual metrics	Tri metrics (Ours)	Four metrics
Computational Cost (s)	25.55(0.93)	26.14(1.52)	30.19(+5.57)	33.74(+9.12)

A.3. Computational Cost Analysis

We acknowledge that there is a slight increase in computation overhead compared to FedAvg, we emphasize that the additional cost is marginal and significantly lower than the previous SOTA, Foolsgold, as shown in Table 6.

B. Training Hyperparameters

The pixel-pattern backdoor data used in the DBA attack is the same as that in Xie et al. [50]. Following [47], we use the data from Southwest Airlines as the dataset for the semantic backdoor attack. Particularly, edge-case indicates that the backdoor data exists only in the attacker's dataset. For a normal attack (non-edge-case), we distribute 10% of the correctly labelled backdoor data to benign clients.

Table 7: The training settings for our experiment.

Hyperparameter	backdoor type					
	semantic			trigger		
	CIFAR10	EMNIST	Sentiment140	CIFAR10	EMNIST	LOAN
backdoor data	Southwest Airlines	Ardis	Greek Director	N/A	N/A	N/A
#clients	200	200	2000	100	100	100
#clients selected in each round	10	10	10	10	10	10
#attackers in each round	1	1	1	4	4	4
#attackers local iteration	5	5	2	6	10	5
#benign local iteration	2	2	2	2	1	1
#global iteration	1500	1500	200	300	70	70
batch size	64	64	20	64	64	64
attack interval	10	10	10	1	1	1
no.iid parameter	0.5	0.5	1	0.5	0.5	0.9
benign learning rate	0.02	0.02	0.05	0.1	0.1	0.001
attackers learning rate	0.02	0.02	0.05	0.05	0.05	0.0005

The hyperparameters in FL system can be seen in Table 7, which used in defenses is as follows:

- Multi-krum:In our experiment, we select the hyperparameter $m = n - f$ (where n stands for the number of participating clients and f stands for the number of tolerable attackers);
- RFA:We set $v = 10^{-5}$ (smoothing factor), $\varepsilon = 10^{-1}$(fault tolerance threshold), $T = 500$ (maximum number of iterations);
- Weak-DP:In our experiment, we use $\sigma = 0.0025$ and set the norm difference threshold at 2.
- Flame: In our experiment, we follow the original paper and use small noise $\sigma = 0.001$.

C. Proof of Proposition 1

From the Lemma 1, we can compute the value of M_d and U_d approximately below, where $M_d = Dmax_d^1 - Dmin_d^1$ reflects the discriminating ability of Manhattan distance and $U_d = Dmax_d^2 - Dmin_d^2$ reflects the discriminating ability of Euclidean distance.

$$\lim_{d\to\infty} E\left[\frac{M_d}{d^{\frac{1}{2}}}\right] = C_1. \tag{5}$$

where C1 is a constant.

$$\lim_{d\to\infty} E\left[U_d\right] = C_2. \tag{6}$$

where C2 is a constant. Thus, we can divide them to compare the M_d and U_d as follows:

$$\lim_{d\to\infty} E\left[\frac{M_d}{U_d \cdot d^{\frac{1}{2}}}\right] = \frac{C1}{C2} = C' \tag{7}$$

where C' is a constant.

D. Ranking Score

To facilitate a comprehensive comparison that involves multiple attack methods and two metrics (*i.e*. MA and BA), we base on the ranking score in [52] and design a new scoring rank that considers both the two metrics and the relative improvement over baseline. The original ranking score used in [52] aims to compare the performance over multiple domain generalization datasets while we want to compare the defense over a set of attacks. They first set a baseline method and for every dataset-algorithm pair, depending on whether the attained accuracy is lower or higher than the baseline accuracy on the same dataset, -1 or +1 is assigned and they add up the scores across all datasets to produce the ranking score for each algorithm. **Why don't we use the original ranking score?** The problem with the original score is, despite using +1 and -1 to indicate higher or less, the relative improvement or decrease is not explicitly accounted for in the final score, which can be problematic in our setting as defense performance varies a lot across different attacks (from $\sim$ 10% to $\sim$ 90%) and scales of the two metrics are also significantly different (from $\sim$ 30% to $\sim$ 80%). For example, if one method obtains a small improvement (*e.g*. 1%) on MA, but a much worse BA (decrease by 30%), the original score method gives a +1 and -1 which makes this method as good as the baseline. However, sacrificing a little MA is still a good price to pay for a large BA improvement. Thus, we propose our new ranking score as followed. We first set a baseline method, *i.e*. FedAvg. Then for each defense method, each attack and each metric, we calculate the relative improvement with respect to the

baseline:

$$\text{score}_K = \frac{K - B}{B}$$

where K denotes the MA or BA of some methods and B denotes the MA and BA of the baseline. Adding up the score for MA and BA

$$\text{score}_{\text{MA}} - \text{score}_{\text{BA}}$$

across all attacks produces the ranking score for each defense. Note here we use subtraction instead of addition since a smaller BA means better defense. Another benefit of this metric is that we can show an average of relative improvement over the baseline, which gives us a sense of how good each method is. We compare our defense with previous SOTA on this metric and provide a detailed analysis in Section 5.1.