Critical Learning Periods Augmented Model Poisoning Attacks to Byzantine-Robust Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Existing attacks in federated learning (FL) control a set of malicious clients and share a *fixed* number of malicious gradients with the central server in each training round, to achieve a desired tradeoff between attack impact and resilience against defenses. In this paper, we show that such a tradeoff is not fundamental and an *adaptive* attack budget not only improves the impact of attack $\mathcal{A}$ but makes it more resilient to defenses. Inspired by recent findings on *critical learning periods* (CLP), where small gradient errors have irrecoverable impact on model accuracy, we advocate CLP augmented model poisoning attacks $\mathcal{A}$-`CLP`, which merely augment attack $\mathcal{A}$ with an adaptive attack budget scheme. $\mathcal{A}$-`CLP` inspects the changes in federated gradient norms to identify CLP and adaptively adjusts the number of malicious clients that share their malicious gradients with the central server in each round, leading to dramatically improved attack impact compared to $\mathcal{A}$ itself by up to 6.85×, with a smaller attack budget and hence improved resilience of $\mathcal{A}$ by up to 2×. Based on understandings on $\mathcal{A}$-`CLP`, we further relax the inner attack subroutine $\mathcal{A}$ in $\mathcal{A}$-`CLP`, and propose `SimAttack-CLP`, a lightweight CLP augmented similarity-based attack, which is more flexible and impactful.

1 Introduction

Federated learning (FL) McMahan et al. (2017) has emerged as an attractive distributed learning paradigm that leverages a large number of *untrusted clients* to collaboratively learn a joint model, called the *global model*, with decentralized training data on each client. A *central server* repeatedly coordinates clients and collects their local model updates computed using their local data, aggregates the clients' updates using an *aggregation rule*, and finally uses the aggregated updates to tune the global model, which is broadcast to a subset of clients at the beginning of each FL training round.

Unfortunately, FL is susceptible to *poisoning* Kairouz et al. (2019); Shejwalkar et al. (2022) by malicious clients compromised by an adversary. Most existing untargeted model poisoning attacks (see Section B.1 for details), such as Fang Fang et al. (2020), LIE Baruch et al. (2019), Min-Sum and Min-Max Shejwalkar & Houmansadr (2021), control a set of malicious clients $\mathcal{M}$. In each FL training round, attack $\mathcal{A}$ crafts the gradients of a fixed number of malicious clients (i.e., a subset of $\mathcal{M}$), and shares their malicious gradients with the central sever for global model update.

However, choosing the number of malicious clients that share malicious gradients[1] with the central server in each FL training round presents a seemingly inherent tradeoff between *the attack impact* (measured by the reduction in model accuracy) and *the attack budget* (the average number of malicious clients per round). For example, when the FL model is trained using the Multi-krum aggregation rule (see Section B.2 for details) on CIFAR-10 with AlexNet, the Fang, LIE, Min-Sum and Min-Max attacks with an attack budget of 25% of total clients are 3.75×, 4.2×, 2.7× and 3.8× more impactful than those with an attack budget of 10% of total clients Shejwalkar & Houmansadr (2021). However, such attack impact improvements are at the cost of sharing more malicious gradients in each round, which in turn causes these poisoning attacks to be more detectable by existing defenses.

[1] Unless otherwise specified, in the rest of this paper, we refer to "malicious clients" only as the malicious clients that share malicious gradients with the central sever for global model update in each FL training round. Such malicious clients are a subset of the total compromised clients controlled by the adversary.

This raises a fundamental question: *Is this observed tradeoff between attack impact and attack budget, and hence the adversary's resilience to defenses fundamental?* In this paper, we show that such a tradeoff is *not* fundamental but a mere artifact of using a *fixed* attack budget throughput the FL training process. In other words, if the attack budget is adaptively tuned, i.e., the number of malicious clients is adaptively tuned over rounds, then both the attack impact and the adversary's resilience can be significantly improved, compared against a fixed number of malicious clients in each round.

We attribute the power of adaptive attack budget to the existing of *critical learning periods* (CLP), i.e., the final quality of a deep neural network (DNN) model is determined by the first few training rounds, in which deficits such as low quality or quantity of training data will cause irreversible model degradation. This phenomenon was revealed in the latest series of works Achille et al. (2019); Jastrzebski et al. (2019); Golatkar et al. (2019); Frankle et al. (2020); Jastrzebski et al. (2021); Yan et al. (2022). We build upon them and extend this notion to poisoning attacks to Byzantine-robust FL.

We advocate **CLP augmented model poisoning attacks**, dubbed as $\mathcal{A}$−CLP, which *merely* augments a state-of-the-art attack $\mathcal{A}$ with an adaptive scheme for attack budget in each FL training round. Hence, $\mathcal{A}$−CLP is *orthogonal* to attack $\mathcal{A}$ since it does not change the way how attack $\mathcal{A}$ crafts malicious gradients. Specifically, $\mathcal{A}$−CLP first identifies CLP in an online manner using an easy-to-compute federated gradient norm metric, and then adaptively adjusts the number of malicious clients in each FL training round. We show that a larger attack budget is *only* required during the CLP. As a result, $\mathcal{A}$−CLP significantly improves the impact of $\mathcal{A}$ attack itself while maintaining a smaller attack budget, and hence improves the resilience of attack $\mathcal{A}$ and makes it less easier to be defeated by state-of-the-art defenses. Extensive experimental results show that when augmenting the strongest state-of-the-art attacks, our $\mathcal{A}$−CLP results in up to 6.85× more accuracy reduction compared to $\mathcal{A}$ (i.e., without being augmented by CLP). In return, $\mathcal{A}$−CLP improves the resilience of $\mathcal{A}$ by up to 2×.

To achieve the above desired tradeoff, one needs to specify the inner attack subroutine $\mathcal{A}$ in $\mathcal{A}$−CLP. The goal of most existing attacks is to derivate the global model parameter *the most* towards the inverse of the direction along which the global model parameter would change without attacks in each FL training round. However, optimizing such a global objective becomes difficult due to highly non-linear constraints, large state space of local models and non-IID local data Li et al. (2020). To address these challenges, we propose SimAttack-CLP, a CLP augmented similarity-based poisoning attack based on above understandings on $\mathcal{A}$−CLP. Specifically, we first adopt a simple cosine similarity as a proximity between clients' gradients, and relax the adversary's goal to compromise a set of malicious clients such that the cosine similarity between after-attacked aggregated gradient and that of before-attack is beyond an *attack threshold* τ. Hence, SimAttack-CLP reduces the computational complexity of $\mathcal{A}$−CLP, and achieves an improved attack impact by up to 1.4× compared to $\mathcal{A}$−CLP.

Our key contributions are summarized as follows:

• We advocate CLP augmented model poisoning attack $\mathcal{A}$−CLP that enable an existing attack $\mathcal{A}$ to adaptively determine the number of malicious clients in each FL training round by identifying CLP via an easy-to-compute federated gradient norm. Our $\mathcal{A}$−CLP avoids the tradeoff between attack impact and attack budget, and hence makes $\mathcal{A}$ not only more impactful but also more resilient.

• We further propose SimAttack-CLP, a CLP augmented similarity-based attack, which crafts malicious gradients based on an attack threshold, and hence is more flexible and easy to implement.

2 Background

Federated learning solves $\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \sum_{i\in\mathcal{N}} p_i F_i(\mathbf{w})$ over $\mathcal{N}=\{1,\cdots,N\}$ clients, where $F_i(\mathbf{w})=\frac{1}{|\mathcal{D}_i|}\sum_{\xi\in\mathcal{D}_i}\ell_i(\mathbf{w};\xi)$ is the local loss function associated with client i's dataset $\mathcal{D}_i$, $p_i = |\mathcal{D}_i|/\sum_i|\mathcal{D}_i|$ is the relative sample size. The training process is orchestrated by repeating two steps in each round t: (1) ***Local training***. The sever randomly selects $\mathcal{N}(t)$ clients for training. For ease of illustration, let $|\mathcal{N}(t)|=n, \forall t$. Client $i\in\mathcal{N}(t)$ pulls the latest global model $\mathbf{w}_i(t-1)$ from the sever, and then performs local updates $\mathbf{w}_i^k(t)\leftarrow\mathbf{w}_i^{k-1}(t-1)-\eta\mathbf{g}(\mathbf{w}_i^{k-1}(t-1),\mathcal{D}_i)$, where η is the learning rate and $k=1,\cdots,K$ is the index of local iterations. (2) ***Model aggregation***. Participants push their local updates to the sever for aggreation to obtain a new global model $\mathbf{w}(t)$: $\mathbf{w}(t)\leftarrow\mathcal{H}(\mathbf{w}_1^K(t),\cdots,\mathbf{w}_n^K(t))$, where $\mathcal{H}$ is the aggregation rule, e.g., the most widely used FedAvg McMahan et al. (2017) performs a weighted average as $\mathbf{w}(t)\leftarrow\sum_{i\in\mathcal{N}(t)}\frac{|\mathcal{D}_i|}{|\cup_{i\in\mathcal{N}(t)}\mathcal{D}_i|}\mathbf{w}_i^K(t)$.

Poisoning attacks on FL. An attack can be either *untargeted*, *targeted*, or *backdoor* based on the goal of the adversary; or be either *model* or *data* poisoning based on the capabilities of the adversary. Detailed discussions are provided in Appendix B.1. To better understand the severity of poisoning attacks to FL, *we mainly focus on the stronger untargeted model poisoning attacks to FL* in this paper.

Byzantine-robust aggregation rules have been proposed to defend against model poisoning attacks. We focus on several representative including Multi-krum Blanchard et al. (2017), Bulyan El El Mhamdi et al. (2018), Trimmed-mean and Median Yin et al. (2018); Xie et al. (2018), and Adaptive federated average (AFA) Muñoz-González et al. (2019) with details relegated to Appendix B.2.

Threat model. The adversary's goal is to craft malicious gradients such that the global model accuracy reduces indiscriminately on any test inputs, known as *untargeted model poisoning attack.*

$\triangleright$ ***Adversary's capabilities.*** The adversary has control over M out to N total clients which is assumed to be less than 50%, i.e., $M/N < 50\%$ Fang et al. (2020); Shejwalkar & Houmansadr (2021); otherwise, no Byzantine-robust aggregation rule will be able to defeat poisoning attacks. Following previous works Bhagoji et al. (2019); Baruch et al. (2019), the adversary can access the global model parameters in each round and directly craft the gradients on malicious clients.

$\triangleright$ ***Adversary's knowledge.*** We characterize the adversary's knowledge along two dimensions: aggregation rule and gradient updates shared by benign clients. Many previous works assume full access to both knowledge, which has limited practical significance. For example, to protect the security of proprietary global models, FL platforms conceal details and/or parameters of their robust aggregation rule, and hence the assumption of full knowledge of aggregation rule is not realistic. We consider a more practical and challenging setting where the adversary is *agnostic* to the aggregation rule and the gradient updates shared by benign clients. In other words, the adversary only knows gradient updates on malicious clients. Since the adversary does not know the aggregation rule, we need to manipulate local model parameters for malicious clients based on a certain aggregation rule. To the best of our knowledge, only Shejwalkar & Houmansadr (2021) considered a similar setting as ours. Similar to them, we will also consider a setting where the adversary has gradients of benign clients but not the aggregation rule of the server. This setting provides an *empirical upper bound* of the severity of our agnostic attacks, for which we relegate the corresponding experimental results to Appendix C.6.

3 $\mathcal{A}$-CLP: CLP Augmented Model Poisoning Attacks

In this section, we advocate the CLP augmented poisoning attack $\mathcal{A}$-$\texttt{CLP}$, which is *orthogonal* to attack $\mathcal{A}$. This is due to the fact that $\mathcal{A}$-$\texttt{CLP}$ does *not* change the way how $\mathcal{A}$ crafts malicious gradients, instead $\mathcal{A}$-$\texttt{CLP}$ merely augments $\mathcal{A}$ with *an adaptive scheme* to determine the number of malicious clients $m(t)$ in each round t in $\mathcal{A}$, rather than using a fixed $m(t) \equiv m, \forall t$ in all rounds.

3.1 The Design of $\mathcal{A}$-$\texttt{CLP}$

As motivated by aforementioned works, it is clear that to adaptively determine the number of malicious clients in each round is akin to identifying CLP in FL training process. Prior works use the changes in eigenvalues of the Hessian or approximating the Hessian using Fisher information Achille et al. (2019); Jastrzebski et al. (2019); Yan et al. (2022) as an indicator to identify CLP. We deviate from these works and develop an approach based on federated gradient norm (FGN), which can be efficiently computed. Considering the difference in training loss for an individual data sample ξ, let $g(\mathbf{w};\xi) = \frac{\partial}{\partial w}\ell(\mathbf{w};\xi)$ denote the gradient of the loss function evaluated on ξ. After performing a step SGD on this sample, the training loss $\Delta\ell = \ell(\mathbf{w} - \eta g(\mathbf{w};\xi);\xi) - \ell(\mathbf{w};\xi)$ can be approximated by its gradient norm using Taylor expansion, i.e., $\Delta\ell \approx -\eta\|g(\mathbf{w};\xi)\|^2$. As a result, the overall training loss at the t-th round, which we define as the FGN, can be approximated using the weighted average of training loss across all selected clients, i.e., $\text{FGN}(t) = \sum_{i\in\mathcal{N}(t)} \frac{|\mathcal{D}_i|}{\sum_{i\in\mathcal{N}(t)}|\mathcal{D}_i|}\Delta\ell_i(t)$. We then develop a simple threshold-based rule to identify the CLP as follows if $\frac{\text{FGN}(t)-\text{FGN}(t-1)}{\text{FGN}(t-1)} \geq \delta$, then the current training round t is in CLP, where δ is the threshold used to declare CLP.

Per our discussions on CLP, the final model accuracy will be permanently impaired if not enough clients are involved in CLP no matter how much additional training is performed after the period Yan et al. (2022). Therefore, for simplicity and usability, $\mathcal{A}$-$\texttt{CLP}$ automatically switches between *a*

Algorithm 1 $\mathcal{A}$-CLP: CLP Augmented Model Poisoning Attacks

```
for $t = 0, 1, \cdots, T-1$ do
    if $\frac{\text{FGN}(t)-\text{FGN}(t-1)}{\text{FGN}(t-1)} \geq \delta$ then
        A larger (i.e., $2m$) number of malicious clients share gradients with the server
    else
        A smaller (i.e., $m/2$) number of malicious clients share gradients with the server
    end if
end for
```

larger (i.e., $2m$) number and *a smaller* (i.e., $m/2$) number of malicious clients that attack $\mathcal{A}$ shares their malicious gradients with the central server in each round by identifying CLP in FL, given that the attack $\mathcal{A}$ without being CLP augmented always selects m malicious clients in each round throughout the FL training process. Therefore, once the CLP is identified, $\mathcal{A}$-CLP increases the number of malicious clients that $\mathcal{A}$ shares their malicious gradients with the central server from m to $2m$, implying that more clients now are being compromised to improve the attack impact on the global model during the CLP. To save the attack budget and also make $\mathcal{A}$ more resilient to defenses, $\mathcal{A}$-CLP changes to share a smaller number of malicious gradients (i.e. $m/2$ clients) after the CLP. Algorithm 1 summarizes $\mathcal{A}$-CLP on top of any state-of-the-art attack $\mathcal{A}$.

From a high-level perspective, $\mathcal{A}$-CLP exploits more malicious clients in the initial phase of the learning procedure than a fixed number of malicious clients for $\mathcal{A}$ itself in each FL training round, to promptly craft the global model with a higher attack impact since the initial learning phase plays a critical role in FL performance. However, the performance improvement comparison is unfair since more malicious clients are used during the CLP. To address these issues, we decrease the number of malicious clients after the CLP. Our empirical results show that this improves the attack budget without hurting the final attack impact. The key point is that more malicious clients should be involved in the global model update in the initial learning phase of FL, and only a smaller number of malicious clients is needed after the CLP.

3.2 Experimental Evaluation

We experimentally verify the performance of $\mathcal{A}$-CLP when paired with four state-of-the-art model poisoning attacks, i.e., Fang, LIE, Min-Sum and Min-Max. We call the corresponding CLP augmented attacks as Fang-CLP, LIE-CLP, Min-Sum-CLP and Min-Max-CLP, respectively.

3.2.1 Experimental Setup

Datasets. We use CIFAR-10 Krizhevsky et al. (2009), MNIST and Fashion-MNIST LeCun et al. (1998) as the evaluation datasets, which are widely used in prior works. We simulate a heterogeneous partition into N clients by sampling $\boldsymbol{p}_i \sim \text{Dir}_N(\alpha)$, where α is the parameter of the Dirichlet distribution. We choose $\alpha = 0.5$ as the default parameter in our experiments as done in Fang et al. (2020); Wang et al. (2020b;c); Cao & Gong (2022), and will numerically investigate its impact. Additional results on Shakespeare dataset McMahan et al. (2017) are provided in Appendix C.8.

Machine learning models. We consider three representative DNN models: AlexNet Krizhevsky et al. (2012), VGG-11 Simonyan & Zisserman (2015) and a fully connected network (FC) with layer sizes $\{784, 512, 10\}$. In particular, we use AlexNet and VGG-11 as the global model architecture for CIFAR-10, FC for MNIST and AlexNet for Fashion-MNIST, respectively. We note that our goal is not to achieve the largest attack impact or rates for considered datasets using the DNN architectures, but rather to show that augmenting existing attacks $\mathcal{A}$ with CLP via our $\mathcal{A}$-CLP can significantly improve the attack impact of a state-of-the-art poisoning attack $\mathcal{A}$ of the learned DNN classifiers.

Different CLP augmented schemes. To illustrate the importance of being CLP augmented and take attack budget into account, we consider four schemes: (1) **Tradition**: Attack $\mathcal{A}$ always shares m malicious gradients in all FL training rounds, which is the default setting in $\mathcal{A}$; (2) **CL** (*The CLP augmented scheme*): As in Algorithm 1, attack $\mathcal{A}$ shares $2m$ malicious gradients with the central server for model update in each round during the CLP, and then $m/2$ malicious gradients after the CLP; (3) **RCL** (*The reverse CLP augmented scheme*): In contrast to **CL**, attack $\mathcal{A}$ shares $m/2$

Dataset (Model)	Aggregation Rule	No Attack (Accuracy)	Fang		LIE		Min-Max		Min-Sum	
			Tradition	CL	Tradition	CL	Tradition	CL	Tradition	CL
CIFAR-10 (AlexNet)	Multi-krum	57.57	10.4	**20.02**	5.73	**11.86**	12.03	**26.47**	11.32	**24.37**
	Bulyan	56.34	9.4	**20.69**	7.5	**12.99**	7.98	**20.9**	6.53	**16.95**
	Trimmed-mean	57.33	10.32	**22.44**	7.36	**17.23**	9.5	**22.85**	8.35	**19.44**
	Median	55.46	11.73	**22.62**	10.89	**18.44**	9.1	**20.48**	7.91	**18.44**
	AFA	57.89	6.99	**11.81**	2.98	**7.41**	9.27	**19.05**	7.73	**14.83**
CIFAR-10 (VGG-11)	Multi-krum	62.63	9.13	**16.03**	6.24	**12.82**	9.94	**17.94**	9.5	**18.07**
	Bulyan	63.37	15.16	**22.53**	13.46	**19.56**	14.91	**21.85**	14.54	**21.52**
	Trimmed-mean	62.9	11.62	**18.88**	11.2	**17.02**	13.14	**20.89**	10.09	**20.95**
	Median	60.13	15.23	**23.58**	12.8	**15.98**	15.05	**23.0**	14.38	**23.34**
	AFA	62.75	7.21	**10.58**	6.26	**8.55**	8.54	**11.55**	7.87	**11.09**
MNIST (FC)	Multi-krum	97.02	1.59	**2.06**	0.26	**0.96**	1.51	**2.32**	1.47	**2.25**
	Bulyan	97.21	1.36	**1.88**	0.84	**1.18**	1.32	**2.14**	1.23	**2.06**
	Trimmed-mean	97.24	1.49	**2.05**	0.24	**0.93**	1.35	**2.28**	1.35	**2.23**
	Median	96.93	1.51	**2.03**	0.31	**1.0**	1.31	**2.15**	1.25	**2.12**
	AFA	97.2	1.27	**1.7**	0.13	**0.89**	1.28	**2.06**	1.28	**2.08**
Fashion MNIST (AlexNet)	Multi-krum	83.24	5.97	**11.05**	3.51	**6.3**	5.06	**15.05**	4.64	**12.1**
	Bulyan	83.12	7.79	**20.58**	3.95	**7.42**	6.8	**13.24**	5.51	**12.88**
	Trimmed-mean	83.53	6.1	**9.39**	4.46	**11.62**	5.21	**8.75**	4.93	**8.57**
	Median	81.81	5.34	**8.88**	5.84	**10.65**	4.27	**8.25**	4.14	**8.72**
	AFA	83.97	4.04	**6.46**	2.96	**5.09**	4.91	**9.49**	3.62	**7.57**

Table 1: The attack impact for state-of-the-art model poisoning attack $\mathcal{A}$ and the corresponding CLP augmented attack $\mathcal{A}$-CLP under various threat models using non-IID partitioned datasets.

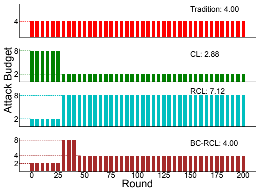


Figure 1: The attack budget.

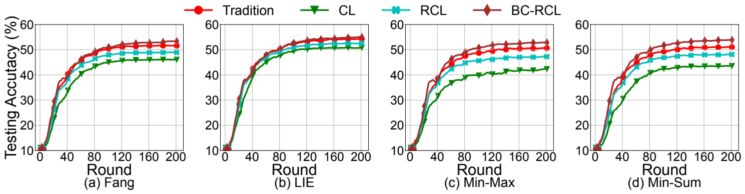


Figure 2: Comparisons of different CLP augmented attacks to FL.

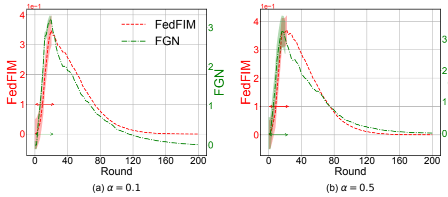
malicious gradients with the central server for model update in each round during the CLP, and then $2m$ malicious gradients after the CLP; and (4) **BC-RCL** (*The budget-constrained RCL scheme*): The total number of clients selected by attack $\mathcal{A}$ is the same as that of **Tradition** throughout the whole FL training process. An illustrative example on the average attack budget per round (i.e., the number of malicious clients per round) under these schemes is presented in Figure 1 (also see Appendix C.4).

Parameter settings. We implement our attacks in PyTorch Paszke et al. (2017) on Python 3 with three NVIDIA RTX A6000 GPUs. We run each experiments for 100 independent trials and report the average results. For ease of presentation, we omit the variances which are observed to be small in the experiments. By default, we consider a total number of $N = 128$ clients in our experiments and the adversary controls $M = 32$ clients. In each round, the FL central server randomly selects $n = 16$ clients to participate in the global model update, in which $m = 4$ are malicious clients. Each client applies 20 iterations of the stochastic gradient descent to update its local model and the central server aggregates local model updates from all selected clients. We set 200 rounds for all DNN classifiers on all datasets considered in this paper. The local learning rate η is initialized as 0.01 and decayed by a constant factor after each communication round. The batch size is set to be 16. We set the weight decay to be 10^{-4} and the detection threshold $\delta = 0.01$ in all of our experiments. An ablation study is presented in Section 3.2.3. The Trimmed-mean aggregation rule prunes the largest and smallest β parameters, where $m \leq \beta \leq n/2$. By default, we consider $\beta = m$ as in Yin et al. (2018).

3.2.2 Significance of CLP Augmented

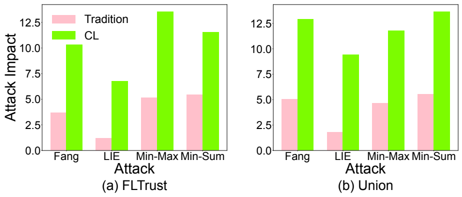
We evaluate $\mathcal{A}$-CLP in terms of attack impact, attack budget and resilience against defenses for all state-of-the-art attacks $\mathcal{A}$ considered in this paper using different DNN models and aggregation rules over several datasets, when $\mathcal{A}$ has no knowledge about the aggregation rule and benign gradient updates (see Section 2). The impacts of $\mathcal{A}$ attack and its corresponding CLP augmented attacks are summarized in Table 1. For ease of readability, we only present the testing accuracy using AlexNet on non-IID partitioned CIFAR-10 when the underlying aggregation rule is Multi-Krum in Figure 2.

Detecting CLP. We compare the CLP identified by our FGN with the federated Fisher information (FedFIM) approach in Yan et al. (2022). When training AlexNet on non-IID CIFAR-10, we observe that these two approaches yield similar results as shown in Figure 3, where the shade and double-arrows indicate identified CLP. However, our FGN approach is much more computationally efficient



(a) $\alpha = 0.1$ (b) $\alpha = 0.5$

Figure 3: Detecting CLP via FGN and FedFIM.



(a) FLTrust (b) Union

Figure 4: Attack impacts of $\mathcal{A}$ and $\mathcal{A}$-`CLP`.

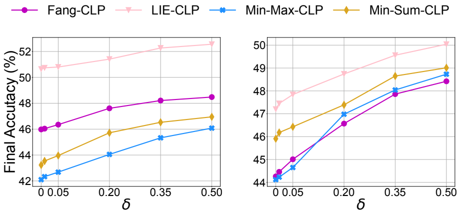
(being orders of magnitude faster to compute) and can be easily leveraged for determining the number of malicious clients in each round during the FL training process in an online manner. More discussions and results on the robustness of our FGN to identify CLP can be found in Appendix C.5.

Improved impact of attack. For any attack $\mathcal{A}$, when augmented by CLP (i.e., the **CL** columns in Table 1 and the **CL** curves in Figure 2), the attack impact is dramatically improved compared to attack $\mathcal{A}$ itself (i.e., the **Tradition** columns in Table 1 and the **Tradition** curves in Figure 2). $\mathcal{A}$-`CLP` attack is 1.25× to 6.85× more impactful than $\mathcal{A}$ attack itself. For example, when running AlexNet on non-IID CIFAR-10 with the underlying aggregation rule of Bulyan, the Fang attack has an attack impact of 9.4, while augmenting it with the CLP, the impact of `Fang-CLP` attack is 20.69, i.e., our $\mathcal{A}$-`CLP` for Fang (i.e., `Fang-CLP`) is 2.2× more effective than the Fang attack itself. Take AlexNet on non-IID Fashion-MNIST with Multi-krum aggregation rule as another example, the impact of Min-Sum attack is 4.64, while the corresponding impact of `Min-Sum-CLP` is 12.1, i.e., `Min-Sum-CLP` is 2.6× impactful than Min-Sum itself without being augmented by CLP. The improvements are further pronounced when benign gradients are known to attack $\mathcal{A}$ (see Appendix C.6).

Improved resilience against defenses. The benefit of being CLP augmented for improved attack impact is reflected in attack budget as shown in Figure 1. It is clear that the $\mathcal{A}$-`CLP` attack is more impactful than the attack $\mathcal{A}$ itself with even a smaller number of attack budget, i.e., $\mathcal{A}$-`CLP` achieves a higher attack impact and a smaller attack budget at the same time. This property is highly desirable, since a smaller attack budget will improve the resilience of the attack against defenses, i.e., $\mathcal{A}$-`CLP` improves the resilience of $\mathcal{A}$. For example, FLTrust Cao et al. (2021) and Union Fang et al. (2020) are two state-of-the-art defenses against model poisoning attacks. We present the attack impact of Fang, LIE, Min-Sum and Min-Max when they are defended by FLTrust and Union with Trimmed-mean using AlexNet on non-IID partitioned CIFAR-10 in Figure 4. It is clear that being CLP augmented significantly improves the attack effectiveness, e.g., the impact of Min-Sum attack is 5.6% when defended by Union, while the impact of `Min-Sum-CLP` attack is 13.2%, i.e., the CLP makes Min-Sum 2.36× impactful and hence makes Union 2.36× less effective to defeat `Min-Sum-CLP`.

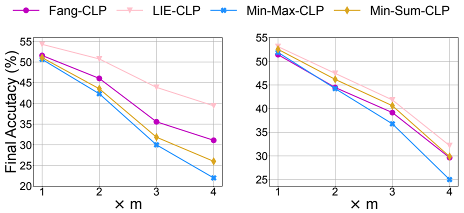
Importance of properly leveraging CLP. To further advocate the importance of being CLP augmented, we consider two variants of $\mathcal{A}$-`CLP`, i.e., **RCL** and **BC-RCL** in Figure 2. On the one hand, **RCL** *only* exhibits a slight better or even similar attack impact as the **Tradition**. In other words, if attack $\mathcal{A}$ only shares malicious gradients from a smaller number of malicious clients during the CLP, it will require $\mathcal{A}$ to share malicious gradients from a much larger number of malicious clients after the CLP in order to achieve a slighter better or even similar attack impact as the **Tradition**. This finding on the importance of being CLP augmented in the FL training process and properly leveraging CLP to adaptively determine the number of malicious clients is consistent with recently reported observations that the initial learning phase plays a key role in determining the outcome of the training process Achille et al. (2019); Jastrzebski et al. (2019); Yan et al. (2022). Further exacerbating the importance of properly leveraging CLP is the fact that **RCL** achieves similar attack performance as **Tradition** at the cost of a significant increase in the attack budget, i.e., a 78% attack budget increase (i.e., average 7.12 vs. 4 attack per round as shown in Figure 1). On the other hand, if we reduce the attack budget, i.e., keeping the total attack budget the same as that of **Tradition**, we observe that the impact of **BC-RCL** attack is significantly worse that **Tradition**. This coincides with the intuition and the functionality of CLP periods in FL training that if the learning models cannot be sufficiently crafted in early training phases, additional attacks cannot improve the attack impact.

Takeaway 1. *We show that attack $\mathcal{A}$ should be augmented by CLP to determine the number of malicious clients to share gradients with the central server for global model update in each FL training round to avoid the tradeoff between attack impact and attack budget, and hence the vulnerability to defenses. In other words, $\mathcal{A}$-`CLP` dramatically improves both the attack impact and the vulnerability of $\mathcal{A}$, and hence make it harder to be defeated by existing defenses. In addition, CLP should be leveraged in a proper manner, i.e., more malicious clients are only needed during the CLP.*



(a) Multi-krum (b) Trimmed-mean

Figure 5: CLP detection threshold δ.



(a) Multi-krum (b) Trimmed-mean

Figure 6: Participated malicious client number.

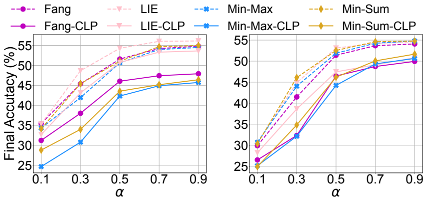
3.2.3 Effect of Model Parameters

We conduct an ablation study to investigate the impacts of hyperparameters. Due to space constraints, we relegate some results to Appendix C.7.

Sensitivity of CLP detection threshold. As discussed in Section 3, the CLP can be efficiently identified using our proposed FGN metric via the simple threshold-type rule. We now evaluate the sensitivity of the threshold value δ to declare CLP. Figure 5 illustrates the final model accuracy of `Fang-CLP`, `LIE-CLP`, `Min-Sum-CLP` and `Min-Max-CLP` attacks to FL with the Multi-krum and Trimmed-mean aggregation rules using AlexNet on non-IID partitioned CIFAR-10. It is clear that the CLP declaration determined by δ has an observable effect on the impact of $\mathcal{A}$-`CLP` attack. This is because as δ becomes larger, fewer rounds in the initial training phases are declared as CLP. As a result, $\mathcal{A}$-`CLP` only uses a larger number of malicious clients to participate in the global model update in fewer rounds according to Algorithm 1, and hence the effect of being CLP augmented on the attack impact is shallowed. However, we observe that even using a large threshold, e.g., $\delta = 0.5$, $\mathcal{A}$-`CLP` is more impactful than $\mathcal{A}$ itself. For example, when $\delta = 0.5$ with Trimmed-mean aggregation rule, the impact of `Min-Sum-CLP` is 27.5% on CIFAR-10, while that of Min-Sum is 16.7%. For ease of simplicity and from Figure 5, we set $\delta = 0.01$ in all of our experiments.

Participated malicious client number. As discussed in Section 3.1, our $\mathcal{A}$-`CLP` automatically switches between a larger and a smaller number of malicious clients that attack $\mathcal{A}$ shares their malicious gradients with the central server during the FL training. For simplicity, we set the number to be $2m$ and $m/2$ in Algorithm 1 given that $\mathcal{A}$ without being CLP augmented always selects m malicious clients in each round. We now vary the larger (resp. smaller) number to be $m, 2m, 3m, 4m$ (resp. $m, m/2, m/3, m/4$). As shown in Figure 6, as a larger number of malicious clients is selected during the CLP, the final accuracy are more severely degraded, which is consistent with above observations (see Takeaway 1). However, as more malicious clients are involved during the CLP, the average attack budget is also increased (see Figure 1). To balance the tradeoff between attack impact and attack budget, and for simplicity, we choose the larger number to be $2m$ in our experiments.

Non-IID degrees of data distribution. We simulate a heterogeneous data partition into N clients using the Dirichlet distribution with parameter α. As observed in Figure 7 using AlexNet on CIFAR10, when the non-IID degree increases, the impacts of `Fang-CLP`, `LIE-CLP`, `Min-Sum-CLP` and `Min-Max-CLP` attacks increase. This is quite intuitive since a higher degree of non-IID data makes the Byzantine-robust aggregation rule harder to detect and remove malicious gradients. As a result, the attack crafts more malicious gradients without being detected and hence improve their attack impacts. In addition, we observe that our $\mathcal{A}$-`CLP` consistently outperforms its counterpart $\mathcal{A}$ across all settings. Without loss of generality, we set $\alpha = 0.5$ in all of our experiments.



(a) Multi-krum (b) Trimmed-mean

Figure 7: Non-IID degree.

4 CLP Augmented Similarity-based Attack

As discussed in Section 1, the inner attack subroutine (i.e., $\mathcal{A}$) in $\mathcal{A}$-`CLP` crafts malicious gradients via solving a difficult optimization problem to deviate the global model parameter *the most* towards the inverse of the direction along which the global model parameter would change before-attack (see Appendix D for details). To avoid solving a complex optimization, we relax the inner attack subroutine and propose SimAttack-CLP, a lightweight CLP augmented similarity-based poisoning attack. For ease of presentation, we present SimAttack-CLP with all benign gradients known.

Intuition. Most Byzantine-robust FL aggregation rules are distance-based, i.e., removing gradients that lie outside of the clique formed by benign gradients. In particular, the distances could be from benign gradients Alistarh et al. (2018), or difference between ℓ_p-norms of benign and malicious clients Sun et al. (2019), or distributional differences with benign gradients Bhagoji et al. (2019). A natural idea to maximize the performance of the adversary is to ensure that malicious gradients lie within the clique of benign gradients. However, to guarantee such a similarity is *far from* trivial. As discussed earlier, optimizing a complex global objective is often difficult Fang et al. (2020). Instead of solving a complex global optimization problem to determine the changing directions, *why not simply craft the malicious clients' gradients based on the proximities between their local models?*

The design of SimAttack-CLP. Our key insight is that it is sufficient to *approximate* an inverse direction that deviates the malicious gradient updates based on proximities between the adversary's local updates, *but not necessarily* the most towards the inverse direction of global model update as in existing attacks $\mathcal{A}$. The number of such malicious clients in each round is determined by the identified CLP as in Algorithm 1. This naturally leads to two questions: *(i) how to measure the proximity or distance?* and *(ii) how to determine the attack goal of the adversary in each communication round?*

For the choice of measure, the ℓ_p distance has been used has a heuristic between models. However, this often suffers from huge computational overheads due to the large state space of local models to search. The cosine similarity between the gradients calculated by updates of model parameters is an alternative measure, which is lightweight. Specifically, the cosine similarity between gradient updates of any two clients i and i' is defined by $\mathcal{F}(\mathbf{g}_i(t), \mathbf{g}_{i'}(t)) := \frac{\langle \mathbf{g}_i(t), \mathbf{g}_{i'}(t)\rangle}{\|\mathbf{g}_i(t)\|\|\mathbf{g}_{i'}(t)\|}$. The expectation of $\mathcal{F}(\cdot,\cdot)$ remains asymptotically constant as dimensionality increases Radovanović et al. (2010).

Using this similarity measure, the goal of the adversary boils down to craft the gradients of m^{CLP} malicious clients such that the cosine similarity between the after-attacked aggregated gradient computed by the adversary and that of before-attack is $\tau \in [-1, 1]$, where τ is a system-wide control knob, which the adversary can set to tradeoff between the severity of attacks and possibility to be defensed. The number of malicious clients is $m^{\text{CLP}} = 2m$ if the current round is in CLP, and otherwise $m/2$ as in $\mathcal{A}$-CLP. We call τ as the *attack threshold.* The choice of τ is very much dependent on the adversary, and having τ as an adversary input adds to the "flexibility" of the overall attacking framework and ultimately, shows the wide applicability of our SimAttack-CLP.

Therefore, for a given attack threshold τ, *the goal of the adversary* is to find changing directions via $\lambda_i, \forall i$ to craft gradients of each of m^{CLP} malicious clients by solving

$$\mathcal{F}(\mathbf{g}(t), \tilde{\mathbf{g}}(t)) = \tau, \tag{1}$$

where $\mathbf{g}(t)$ and $\tilde{\mathbf{g}}(t)$ are given in (5) and (6), respectively, and $\tilde{\mathbf{g}}_i(t) = \mathbf{g}_i(t) + \lambda_i \mathbf{s}_t, \forall i = 1, \cdots, m^{\text{CLP}}$. The key challenge is that the adversary does not know the aggregation rule. To address this, we make one approximation. Specifically, we assume that the adversary adopts an "average rule" to approximate the aggregation rule of the server, i.e., $\mathbf{g}(t) \approx \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}_i(t)$, $\tilde{\mathbf{g}}(t) \approx \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathbf{g}}_i(t)$, and $\lambda \triangleq \sum_{i=1}^{m^{\text{CLP}}} \lambda_i$. Then we have $\tilde{\mathbf{g}}(t) \approx \mathbf{g}(t) + \lambda \mathbf{s}(t)$, from which we can easily solve a so-called "global" λ that is common for all m^{CLP} malicious clients (See Proposition 1 in Appendix D).

Since in practical FL systems, clients often have heterogeneous data distributions and system capabilities Kairouz et al. (2019); Bonawitz et al. (2019); Hsieh et al. (2020), and hence a heterogeneous changing direction determined by λ_i, $\forall i$ is more preferable than a "global" one λ across all malicious clients. To achieve this goal, we first leverage the definition of cosine similarity to obtain $\mathcal{F}(\mathbf{g}(t), \tilde{\mathbf{g}}(t)) = \frac{1}{m^{\text{CLP}}} \frac{\sum_{i=1}^{m^{\text{CLP}}} \langle \mathbf{g}(t), \tilde{\mathbf{g}}_i(t)\rangle}{\|\mathbf{g}(t)\|\|\tilde{\mathbf{g}}(t)\|}$. Then the changing direction to craft each malicious gradient $\forall i$ can be approximated by $\mathcal{F}(\mathbf{g}(t), \tilde{\mathbf{g}}_i(t)) = \frac{\langle \mathbf{g}(t), \tilde{\mathbf{g}}_i(t)\rangle}{\|\mathbf{g}(t)\|\|\tilde{\mathbf{g}}(t)\|} \approx \tau$. When combined with $\tilde{\mathbf{g}}_i(t) = \mathbf{g}_i(t) + \lambda_i \mathbf{s}_t$, we can determine $\lambda_i, \forall i$:

Lemma 1. *Suppose that λ_i is the changing direction to craft malicious gradient of the malicious client i, $\forall i = 1, \cdots, m^{CLP}$. Then for any given attack threshold τ, the value of λ_i satisfies*

$$\lambda_i = \frac{\langle \boldsymbol{g}(t), \boldsymbol{g}_i(t)\rangle - \tau\|\boldsymbol{g}(t)\|\|\tilde{\boldsymbol{g}}(t)\|}{\boldsymbol{g}(t)^{\intercal}\boldsymbol{s}(t)}, \ \forall i = 1, \cdots, m^{CLP}. \tag{2}$$

Remark 1. *Our SimAttack-CLP can be easily generalized to the case where benign gradients are unknown to the adversary. Since the adversary does not have benign gradients, the changing directions $\mathbf{s}(t), \forall t$ are not known and hence we cannot directly solve for λ_i using (2). However, the before-attack local models on malicious clients are known to the adversary. Hence, similar to Fang et al. (2020); Shejwalkar & Houmansadr (2021), we estimate the changing directions using the mean before-attack local model of malicious clients. In other words, if the mean of the j-parameter is larger than the j-th global model parameter received from the central sever in the current round, then $s_j(t)$ is approximated to be 1, and otherwise -1. Using this approximation, we can obtain the changing directions, which we denote as $\tilde{\mathbf{s}}$, and hence the $\tilde{\lambda}_i, \forall i$ using (2).*

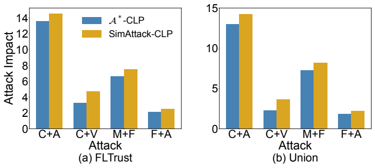
Experimental evaluation. We compare the performance of SimAttack-CLP with state-of-the-art CLP augmented attacks (see Section 3). We consider the same experimental setup as in Section 3.2. Note that Fang requires the knowledge of the aggregation rule. Also, Min-Sum and Min-Max outperform Fang when the aggregation rule is unknown Shejwalkar & Houmansadr (2021). Hence we exclude the comparisons with Fang here since our SimAttack-CLP is also fully agnostic to the aggregation rule. For ease of readability, we relegate the results of testing accuracy to Appendix D.1.

▷ ***Impact of attacks.*** The impacts of SimAttack-CLP attack and that of the best among LIE-CLP, Min-Sum-CLP and Min-Max-CLP attacks (see Table 1), which we denote as $\mathcal{A}^*$-CLP, are reported in Table 2. Our SimAttack-CLP is consistently more impactful than the strongest of existing poisoning attacks. For example, SimAttack-CLP is 1.1× more impactful than $\mathcal{A}^*$-CLP attack for AlexNet and VGG-11 models on non-IID partitioned CIFAR-10. Combined with results in Table 1, SimAttack-CLP is up to 2.9× more impactful than the strongest state-of-the-art LIE, MIN-Sum and Min-Max attacks. Similarly, SimAttack-CLP is 1.4× and 1.1× more impactful than $\mathcal{A}^*$-CLP attack for FC model on non-IID MNIST and AlexNet model on non-IID Fashion-MNIST, respectively, and hence is 9.6× and 3.3× more impactful than the strongest state-of-the-art attacks, respectively.

Dataset (Model)	Aggregation Rule	$\mathcal{A}^*$-CLP	SimAttack-CLP
CIFAR-10 (AlexNet)	Multi-krum	26.47	**28.55**
	Bulyan	20.9	**22.11**
	Trimmed-mean	22.85	**24.31**
	Median	22.62	**23.6**
	AFA	19.05	**20.27**
CIFAR-10 (VGG-11)	Multi-krum	18.07	**20.13**
	Bulyan	22.53	**24.03**
	Trimmed-mean	20.95	**22.62**
	Median	23.58	**24.21**
	AFA	11.55	**13.27**
MNIST (FC)	Multi-krum	2.32	**2.86**
	Bulyan	2.14	**3.02**
	Trimmed-mean	2.28	**2.85**
	Median	2.15	**2.72**
	AFA	2.08	**2.55**
Fashion MNIST (AlexNet)	Multi-krum	15.05	**16.37**
	Bulyan	20.58	**21.81**
	Trimmed-mean	11.62	**12.41**
	Median	10.65	**11.35**
	AFA	9.49	**10.68**

Table 2: Attack impacts of SimAttack-CLP and $\mathcal{A}^*$-CLP.

▷ ***Resilience against defenses.*** We further compare the resilience of SimAttack-CLP and $\mathcal{A}^*$-CLP. Figure 8 illustrates the attack impact of SimAttack-CLP and $\mathcal{A}^*$-CLP when defended by FLTrust Cao et al. (2021) and Union Fang et al. (2020), respectively, where C+A, C+V, M+F, and F+A represent AlexNet on CIFAR-10, VGG-11 on CIFAR-10, FC on MNIST, and AlexNet on Fashion-MNIST, respectively. We observe that SimAttack-CLP is more resilient than $\mathcal{A}^*$-CLP in all settings. For example, SimAttack-CLP makes FLTrust 1.6× less effective to be defeated than $\mathcal{A}^*$-CLP using VGG-11 on CIFAR-10 with unknown benign gradients.



(a) FLTrust (b) Union

Figure 8: Attack impacts of SimAttack-CLP and $\mathcal{A}^*$-CLP when defended by FLTrust and Union under various threat models using non-IID partitioned datasets.

Takeaway 2. *Two fundamental differences between SimAttack-CLP and existing attacks contribute to the superior performance of SimAttack-CLP. First, rather than solving a complex optimization problem to maximize the difference in the direction between malicious and benign gradients, a key insight in the design of SimAttack-CLP is that it is sufficient to approximate the largest derivation via an attack threshold. This flexible control knob relaxes the assumptions (see Section 1) needed in state of the arts Fang et al. (2020); Shejwalkar & Houmansadr (2021), whose performance largely depend on these hyperparameters. In addition, SimAttack-CLP carefully crafts the gradient of each malicious client (i.e., using different λ_i) due to the practical heterogeneity among FL clients, rather than a single attack across all malicious clients, e.g., Fang et al. (2020).*

Second, SimAttack-CLP leverages CLP to adaptively determine the number of malicious clients in each round. Though being CLP augmented also significantly improves the impact of these attacks (see Section 3.2), our SimAttack-CLP is still superior than their $\mathcal{A}$-CLP counterparts. We conjecture that a flexible attack threshold, rather than a maximal attack, fits better with CLP, which contributes to the superior performance of SimAttack-CLP. Building a better theoretical understanding of SimAttack-CLP is an avenue for future work.

References

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical Learning Periods in Deep Networks. In *Proc. of ICLR*, 2019.

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *Proc. of NeurIPS*, 2018.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proc. of AISTATS*, 2020.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Proc. of NeurIPS*, 2019.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proc. of ICML*, 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proc. of NeurIPS*, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. In *Proc. of MLSys*, 2019.

Xiaoyu Cao and Neil Zhenqiang Gong. Mpaf: Model poisoning attacks to federated learning based on fake clients. *arXiv preprint arXiv:2203.08669*, 2022.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *Proc. of NDSS*, 2021.

Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. In *Proc. of NIPS*, 2012.

Mahdi El El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *Proc. of ICML*, 2018.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proc. of USENIX Security*, 2020.

Jonathan Frankle, David J Schwab, and Ari S Morcos. The Early Phase of Neural Network Training. In *Proc. of ICLR*, 2020.

Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto. Time Matters in Regularizing Deep Networks: Weight Decay and Data Augmentation Affect Early Learning Dynamics, Matter Little Near Convergence. *Proc. of NeurIPS*, 2019.

William C Guenther. Some remarks on the runs test and the use of the hypergeometric distribution. *The American Statistician*, 32(2):71–73, 1978.

William L Harkness. Properties of the extended hypergeometric distribution. *The Annals of Mathematical Statistics*, 36(3):938–945, 1965.

Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling. 2020.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *Proc. of ICML*, 2020.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *Proc. of IEEE S&P*, 2018.

Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos J Storkey. On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length. In *Proc. of ICLR*, 2019.

Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic Fisher Explosion: Early Phase Fisher Matrix Impacts Generalization. In *Proc. of ICML*, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977*, 2019.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Proc. of NIPS*, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the Convergence of FedAvg on Non-IID Data. In *Proc. of ICLR*, 2020.

Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *Proc. of ICML*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. of AISTATS*, 2017.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 27–38, 2017.

Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. On the existence of obstinate results in vector space models. In *Proc. of ACM SIGIR*, 2010.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Proc. of NDSS*, 2021.

Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *Proc. of IEEE S&P*, 2022.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. In *Proc. of ICLR*, 2015.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Proc. of NeurIPS*, 2020a.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated Learning with Matched Averaging. In *Proc. of ICLR*, 2020b.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. *Proc. of NeurIPS*, 2020c.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *Proc. of ICLR*, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Proc. of UAI*, 2020.

Gang Yan, Hao Wang, and Jian Li. Seizing Critical Learning Periods in Federated Learning. In *Proc. of AAAI*, 2022.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proc. of ICML*, 2018.

A Summary of Notations

We summarize the major notations used in the paper in Table 3.

Table 3: Notations.

Symbol	**Meaning**
$\mathcal{A}$	the poisoning attack/adversary
$\mathcal{H}$	the aggregation rule of FL
i	client i, where $i \in \mathcal{N} = \{1, \cdots, N\}$
N	the total number of clients in the system
M	the total number of compromised clients controlled by adversary $\mathcal{A}$
n	the number of clients selected by the central server for model update in each FL training round
m	the number of malicious clients that share malicious gradients with the central server in each FL training round
$\mathbf{w}_i(t)$	model parameter of client i before attack
$\mathbf{g}_i(t)$	gradient of client i before attack
$\tilde{\mathbf{g}}_i(t)$	gradient of client i after attack
$\mathcal{F}(\cdot, \cdot)$	the cosine similarity between two gradients
$s_j(t)$	the j-th global model parameter's changing direction
τ	an adversary defined attack threshold

B Extensive Review on Background

B.1 Poisoning Attacks on Federated Learning

FL is vulnerable to various poisoning attacks Bagdasaryan et al. (2020); Baruch et al. (2019); Bhagoji et al. (2019); Blanchard et al. (2017); Fang et al. (2020); Jagielski et al. (2018); Shejwalkar & Houmansadr (2021), which can be categorized into two classes based on the adversary's goal and capabilities. On the one hand, an attack can be either *untargeted*, *targeted*, or *backdoor* based on the goal of the adversary. The goal of untargeted attacks is to minimize the accuracy of the global model on *any* test input El El Mhamdi et al. (2018); Mahloujifar et al. (2019); Baruch et al. (2019); Fang et al. (2020); Xie et al. (2020); Shejwalkar & Houmansadr (2021); the goal of targeted attacks is to minimize the accuracy on *specific* test inputs, and maintaining high accuracies on the rest of test inputs Bhagoji et al. (2019); Sun et al. (2019); while *backdoor* attacks target on reducing the utility on test inputs that contain a specific signal called the trigger Bagdasaryan et al. (2020); Wang et al. (2020a); Xie et al. (2019). To this end, untargeted attacks can completely cripple the global model and hence pose more severe threats to FL.

On the other hand, an attack can be either *model* or *data* poisoning based on the capabilities of the adversary. In model positioning attacks El El Mhamdi et al. (2018); Bhagoji et al. (2019); Bagdasaryan et al. (2020); Fang et al. (2020); Xie et al. (2020); Shejwalkar & Houmansadr (2021), the adversary directly manipulates the gradients on malicious devices before sharing them with the server in each communication round, while in data positioning attacks Muñoz-González et al. (2017); Jagielski et al. (2018), the adversary can only indirectly manipulate the gradients on malicious clients by poisoning training datasets on the clients. As a result, model poisoning attacks often achieve higher attack impacts on FL. Therefore, to better understand the severity of poisoning attacks to FL, *we mainly focus on the stronger untargeted model poisoning attacks to FL* in this paper.

B.2 Byzantine-Robust Aggregation Rules

The mean aggregation rule has been widely used in non-adversarial settings Dean et al. (2012); Konečnỳ et al. (2016); McMahan et al. (2017), which, however, is not robust and can be manipulated by even a single malicious client Blanchard et al. (2017); Yin et al. (2018); Bagdasaryan et al. (2020); Bhagoji et al. (2019). Therefore, multiple Byzantine-robust aggregation rules Blanchard et al. (2017);

El El Mhamdi et al. (2018); Yin et al. (2018); Xie et al. (2018); Muñoz-González et al. (2019); Alistarh et al. (2018); Chang et al. (2019); He et al. (2020); Cao et al. (2021) have been proposed to defend against poisoning attacks. In the following, we review several representative Byzantine-robust aggregation rules that will be used in this paper.

Multi-krum Blanchard et al. (2017). Krum Blanchard et al. (2017) selects gradients from the set of its input gradients that is close to its $n-m-2$ neighbor gradients in squared Euclidean norm space with m being an upper bound on the number of malicious clients and n being the number of participated clients in each FL training round. The intuition is that malicious gradients need to be far from benign ones in order to poison the global model. To effectively utilize the knowledge shared by the clients in each round, Multi-krum selects a gradient using Krum from a remaining set, adds it to a selection set and removes it from the remaining set. It has been shown that when $m \leq n/2-1$, Multi-krum converges for certain objective functions. Since Multi-krum significantly outperforms Krum in terms of the global model accuracy, we will focus on Multi-krum in this paper.

Bulyan El El Mhamdi et al. (2018). Mhamdi et al. El El Mhamdi et al. (2018) showed that a malicious gradient remains close to benign gradients while having a single gradient dimension with a large value and thus prevent convergence of the global model. As such, Bulyan was proposed which first iteratively applies Multi-krum to select θ ($\theta \leq n-2m$) local models, and then use a variant of trimmed mean to aggregate θ local models. To guarantee robustness, it requires $n \geq 4m+3$ to hold, see El El Mhamdi et al. (2018) for more details.

Trimmed-mean Yin et al. (2018); Xie et al. (2018) coordinate-wisely aggregates each dimension of input gradients separately. Specifically, for a given dimension j, the j-th parameters of n local models $\{\mathbf{g}_i^j\}_{i=1,\cdots,n}$, Trimmed-mean removes the largest and smallest β of them, and computes the mean of the remaining $n-2\beta$ parameters as the j-th dimension of the global model. It is shown that Trimmed-mean achieves order-optimal error rates when $m \leq \beta \leq n/2$ for strongly convex objective functions.

Median Yin et al. (2018); Xie et al. (2018) is also a coordinate-wise aggregation rule. Unlike Trimmed-mean, it takes the median as the j-th dimension of the global model. Like Trimmed-mean, it has also been shown that Median achieves an order-optimal error rate when the objective function is strongly convex.

Adaptive federated average (AFA) Muñoz-González et al. (2019). AFA first computes a weighted average of collected gradients in each communication round. Then it computes cosine similarities between the weighted average and each of the collected gradients. Finally, AFA discards the gradients with similarities out of a range, which is a simple function of mean, median and standard deviation of the similarities.

C Experimental Details and Additional Results of $\mathcal{A}$-CLP

We experimentally verify the performance of $\mathcal{A}$-CLP when paired with four state-of-the-art poisoning attacks, i.e., Fang, LIE, Min-Sum and Min-Max. We call the corresponding CLP-aware poisoning attacks as Fang-CLP, LIE-CLP, Min-Sum-CLP and Min-Max-CLP, respectively.

C.1 Datasets

We use CIFAR-10 Krizhevsky et al. (2009), MNIST and Fashion-MNIST LeCun et al. (1998) as the evaluation datasets, which are widely used in prior works. The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, where 50,000 samples are for training and the other 10,000 samples for testing. The MNIST and Fashion-MNIST datasets contain handwritten digits with 60,000 samples for training and 10,000 samples for testing, where each sample is an 28×28 grayscale images over 10 classes. We simulate a heterogeneous partition into N clients by sampling $\boldsymbol{p}_i \sim \text{Dir}_N(\alpha)$, where α is the parameter of the Dirichlet distribution. We choose $\alpha = 0.5$ as the default parameter in our experiments as done in Fang et al. (2020); Wang et al. (2020b;c); Cao & Gong (2022), and will numerically investigate its impact.

We further investigate the task of next-character prediction on the dataset of *The Complete Works of William Shakespeare* (Shakespeare) (McMahan et al., 2017), which consists of 74 characters with 734,057 training data and 70,657 testing data.

C.2 Machine learning models

For the classification problems, we consider three representative DNN models: AlexNet Krizhevsky et al. (2012), VGG-11 Simonyan & Zisserman (2015) and a fully connected network (FC) with layer sizes $\{784, 512, 10\}$. In particular, we use AlexNet and VGG-11 as the global model architecture for CIFAR-10, FC for MNIST and AlexNet for Fashion-MNIST, respectively. For the language task, we train a stacked character-level LSTM language model as in (McMahan et al., 2017). We summarize the details of AlexNet, VGG-11, FC and LSTM architectures used in our experiments in Tables 4, 5, 6 and 7, respectively. We note that our goal is not to achieve the largest attack impact or rates for considered datasets using the DNN architectures, but rather to show that augmenting existing attacks $\mathcal{A}$ with CLP via our $\mathcal{A}$-`CLP` can significantly improve the attack impact of a state-of-the-art poisoning attack $\mathcal{A}$ of the learned DNN classifiers.

Parameter	Shape	Layer hyper-parameter
layer1.conv1.weight	$3 \times 64 \times 3 \times 3$	stride:2; padding: 1
layer1.conv1.bias	32	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer2.conv2.weight	$64 \times 192 \times 3 \times 3$	stride:1; padding: 1
layer2.conv2.bias	64	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer3.conv3.weight	$192 \times 384 \times 3 \times 3$	stride:1; padding: 1
layer3.conv3.bias	128	N/A
layer4.conv4.weight	$384 \times 256 \times 3 \times 3$	stride:1; padding: 1
layer4.conv4.bias	128	N/A
layer5.conv5.weight	$256 \times 256 \times 3 \times 3$	stride:1; padding: 1
layer5.conv5.bias	256	N/A
pooling.max	N/A	kernel size:2; stride: 2
dropout	N/A	p=5%
layer6.fc6.weight	1024×4096	N/A
layer6.fc6.bias	512	N/A
dropout	N/A	p=5%
layer7.fc7.weight	4096×4096	N/A
layer7.fc7.bias	512	N/A
layer8.fc8.weight	4096×10	N/A
layer8.fc8.bias	10	N/A

Table 4: Detailed information of the AlexNet architecture used in our experiments. All non-linear activation function in this architecture is ReLU. The shapes for convolution layers follow (C_{in}, C_{out}, c, c).

C.3 Baseline attacks

We consider the following four strongest model poisoning attacks in the literature, i.e., Fang Fang et al. (2020), LIE Baruch et al. (2019), Min-Sum and Min-Max Shejwalkar & Houmansadr (2021). Besides Fang, the other three attacks are agnostic to the aggregation rule.

• ***Fang:*** An optimization based model poisoning attack model has been proposed in Fang et al. Fang et al. (2020), which can be tailored to several aggregation rules such as Krum, Trimmed-mean and Median. Specifically, the adversary computes the average μ of benign gradients that she has access to. Then the adversary computes $-\text{sign}(\mu)$ and computes a malicious update by solving for a global coefficient λ. The adversary then attacks all malicious clients and change their gradient updates' direction based on λ.

• ***LIE:*** Small amounts of noises are added to each dimension of the average of benign gradients in the LIE attack Baruch et al. (2019). The small noises can be sufficiently large to adversely impact the global model and can be sufficiently small to evade detection by the Byzantine-robust aggregation rules. In particular, the adversary computes the average μ and standard deviation σ of benign gradients that she has access to. Furthermore, the adversary computes a coefficient z based on the total number of benign and malicious clients, and hence obtains the malicious update as $\mu + z\sigma$.

• ***Min-Sum:*** The Min-Sum attack ensures that the sum of squared distances of the malicious gradients from all the benign gradients is upper bounded by the sum of squared distances of any benign gradient

Parameter	Shape	Layer hyper-parameter
layer1.conv1.weight	$3 \times 64 \times 3 \times 3$	stride:1; padding: 1
layer1.conv1.bias	64	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer2.conv2.weight	$64 \times 128 \times 3 \times 3$	stride:1; padding: 1
layer2.conv2.bias	128	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer3.conv3.weight	$128 \times 256 \times 3 \times 3$	stride:1; padding: 1
layer3.conv3.bias	256	N/A
layer4.conv4.weight	$256 \times 256 \times 3 \times 3$	stride:1; padding: 1
layer4.conv4.bias	256	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer5.conv5.weight	$256 \times 512 \times 3 \times 3$	stride:1; padding: 1
layer5.conv5.bias	512	N/A
layer6.conv6.weight	$512 \times 512 \times 3 \times 3$	stride:1; padding: 1
layer6.conv6.bias	512	N/A
pooling.max	N/A	kernel size:2; stride: 2
layer7.conv7.weight	$512 \times 512 \times 3 \times 3$	stride:1; padding: 1
layer7.conv7.bias	512	N/A
layer8.conv8.weight	$512 \times 512 \times 3 \times 3$	stride:1; padding: 1
layer8.conv8.bias	512	N/A
pooling.max	N/A	kernel size:2; stride: 2
dropout	N/A	p=20%
layer9.fc9.weight	4096×512	N/A
layer9.fc9.bias	512	N/A
layer10.fc10.weight	512×512	N/A
layer10.fc10.bias	512	N/A
dropout	N/A	p=20%
layer11.fc11.weight	512×10	N/A
layer11.fc11.bias	10	N/A

Table 5: Detailed information of the VGG-11 architecture used in our experiments. All non-linear activation function in this architecture is ReLU. The shapes for convolution layers follow (C_{in}, C_{out}, c, c).

Parameter	Shape	Layer hyper-parameter
layer1.fc1.weight	1024×256	N/A
layer1.fc1.bias	256	N/A
layer2.fc2.weight	256×256	N/A
layer2.fc2.bias	256	N/A
layer3.fc3.weight	256×10	N/A
layer3.fc3.bias	10	N/A

Table 6: Detailed information of the FC architecture used in our experiments. All non-linear activation function in this architecture is ReLU. The shapes for convolution layers follow (C_{in}, C_{out}, c, c).

from the other benign gradients. All malicious gradients are kept the same for the maximum attack impact.

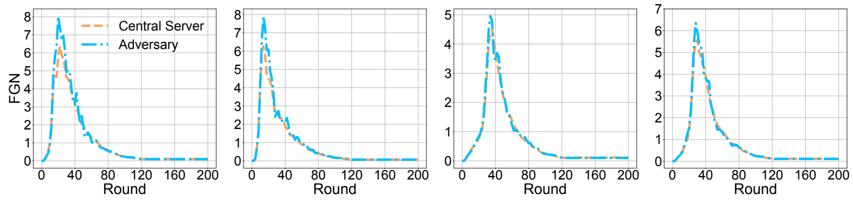
• ***Min-Max:*** The Min-Max attack computes the malicious gradients such that its maximum distance from any other gradient is upper bounded by the maximum distance between any two benign gradients. As a result, the malicious gradients lie close to the clique of the benign gradients.

C.4 Different CLP-awareness schemes

To illustrate the importance of being CLP augmented and take attack budget into account, we consider the following four schemes for comparisons: (1) **Tradition** (*The original scheme*): Attack $\mathcal{A}$ always shares malicious gradients from m malicious clients throughout all FL training rounds, which is the default setting in $\mathcal{A}$ itself; (2) **CL** (*The CLP-based scheme*): As in Algorithm 1, attack $\mathcal{A}$ shares malicious gradients from $2m$ malicious clients with the central server for model update in each round

Parameter	Shape	Layer hyper-parameter
layer1.embeding	80 × 256	N/A
layer2.lstm	256 × 512	num_layers=2, batch_first=True
dropout	N/A	p=5%
layer3.fc.weight	512 × 80	N/A
layer3.fc.bias	80	N/A

Table 7: Detailed information of the LSTM architecture used in our experiments.



(a) Fang (b) LIE (c) Min-Sum (d) Min-Max

Figure 9: Comparisons of detecting CLP from the perspectives of the FL central server and the adversary $\mathcal{A}$ using AlexNet on Fashion-MNIST with the Multi-krum aggregation rule.

during the CLP, and then from $m/2$ malicious clients after the CLP; (3) **RCL** (*The reverse CL-based scheme*): In contrast to **CL**, attack $\mathcal{A}$ shares malicious gradients from $m/2$ malicious clients with the central server for model update in each round during the CLP, and then from $2m$ malicious clients after the CLP; and (4) **BC-RCL** (*The budget-constrained RCL-based scheme*): The total number of clients selected by attack $\mathcal{A}$ is the same as that of **Tradition** throughout the whole FL training process.

For example, we illustrate the average attack budget per round (i.e., the number of malicious clients per round) under these schemes in Figure 1, where we run AlexNet on non-IID partitioned CIFA-10 over 200 rounds. In **Tradition**, attack $\mathcal{A}$ always selects $m = 4$ malicious clients in each round. In **CL**, attack $\mathcal{A}$ selects $2m = 8$ malicious clients during the CLP and $m/2 = 2$ malicious clients afterwards, resulting an average attack budget of 2.88 malicious clients per round. Similarly, the average attack budgets under **RCL** and **BC-RCL** are 7.12 and 4, respectively.

C.5 Robustness of Identifying CLP

We propose a lightweight FGN metric to identify CLP in federated settings in Section 3.1. Our numerical results show that our FGN approach yields similar results as that using the state-of-the-art FIM approach Yan et al. (2022) as shown in Figure 3, where we implemented our attacks in PyTorch Paszke et al. (2017) on Python 3 with three NVIDIA RTX A6000 GPUs, 48GB with128GB RAM. However, our FGN approach is much more computationally efficient than FIM approach as shown in Figure 11.

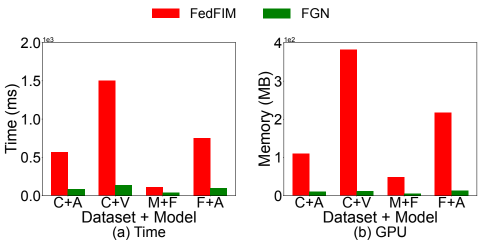
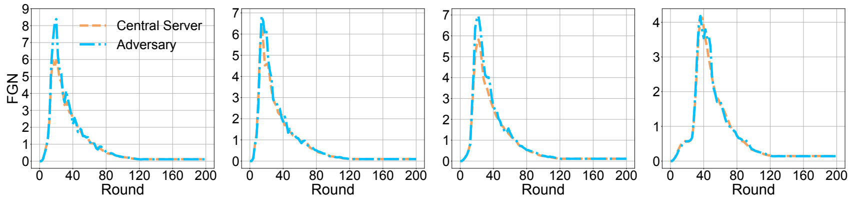


Figure 11: Computation time and memory consumption of FGN and FedFIM approach to detect CLP.

Note that in Figure 3, we compute the FGN from the perspective of the FL central server, which controls a total of N clients and randomly selects n clients for model update in each FL training round. This is the same setting as in Yan et al. (2022) for comparison. Once the central server identifies the CLP, it may broadcast the information to all cients along with the updated global model at the beginning of each round. For the defense purpose, the central server may not want to share such information with the adversary $\mathcal{A}$, as our $\mathcal{A}$-CLP framework shows that the attack $\mathcal{A}$ can leverage the CLP information to significantly improve its attack impact. Since the adversary A controls a set of M clients, and shares m malicious gradients with the central server for model update in each round, a natural question is that can the adversary $\mathcal{A}$ also detect the CLP by itself? If so, is the identified CLP the same as that identified by the central server? Here, we provide affirmative answers to these questions. Specifically, the adversary $\mathcal{A}$ computes the FGN using the information from the set of M clients it controls. For ease of readability, we consider to train AlexNet on Fashion-MNIST dataset



(a) Fang (b) LIE (c) Min-Sum (d) Min-Max

Figure 10: Comparisons of detecting CLP from the perspectives of the FL central server and the adversary $\mathcal{A}$ using AlexNet on Fashion-MNIST with the Trimmed-mean aggregation rule.

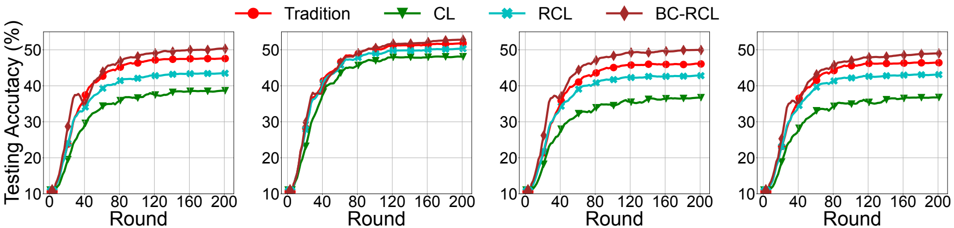
Dataset (Model)	Aggregation Rule	No Attack (Accuracy)	Fang		LIE		Min-Max		Min-Sum	
			Tradition	CL	Tradition	CL	Tradition	CL	Tradition	CL
CIFAR-10 (AlexNet)	Multi-krum	57.57	17.35	**32.88**	10.03	**16.44**	19.88	**36.23**	19.35	**36.12**
	Bulyan	56.34	13.61	**24.22**	12.01	**19.32**	13.11	**25.41**	12.51	**23.32**
	Trimmed-mean	57.33	15.8	**33.57**	14.02	**27.92**	16.43	**31.81**	15.19	**29.14**
	Median	55.46	17.77	**33.41**	18.31	**29.22**	16.48	**29.26**	15.88	**27.37**
	AFA	57.89	11.21	**21.74**	3.66	**8.39**	15.11	**26.37**	14.56	**24.16**
CIFAR-10 (VGG-11)	Multi-krum	62.63	15.75	**25.56**	10.47	**17.62**	19.43	**28.96**	17.08	**27.47**
	Bulyan	63.37	20.73	**31.02**	20.6	**28.84**	21.0	**31.45**	20.65	**31.24**
	Trimmed-mean	62.9	20.14	**31.57**	18.01	**25.42**	22.24	**36.72**	20.68	**34.78**
	Median	60.13	18.94	**34.9**	18.19	**26.09**	19.75	**34.17**	20.02	**33.52**
	AFA	62.75	10.27	**21.67**	6.7	**17.96**	12.87	**24.65**	11.68	**24.5**
MNIST (FC)	Multi-krum	97.02	2.04	**2.55**	0.84	**1.55**	2.06	**2.87**	2.12	**2.97**
	Bulyan	97.21	1.64	**2.24**	1.16	**2.0**	1.53	**2.41**	1.71	**2.53**
	Trimmed-mean	97.24	1.74	**2.4**	0.67	**1.87**	1.76	**2.75**	1.73	**2.71**
	Median	96.93	1.85	**2.5**	0.1	**2.37**	1.8	**2.66**	1.78	**2.77**
	AFA	97.2	1.55	**2.18**	0.28	**1.39**	1.8	**2.36**	1.74	**2.92**
Fashion MNIST (AlexNet)	Multi-krum	83.24	9.43	**20.53**	5.16	**10.43**	9.23	**16.71**	9.41	**17.63**
	Bulyan	83.12	12.59	**26.66**	11.32	**21.4**	11.79	**19.58**	11.75	**23.89**
	Trimmed-mean	83.53	6.74	**12.22**	4.81	**11.73**	6.42	**10.33**	6.74	**11.04**
	Median	81.81	8.66	**15.18**	9.47	**12.54**	5.92	**10.32**	6.18	**12.95**
	AFA	83.97	6.89	**13.19**	2.89	**5.99**	7.6	**12.74**	7.76	**13.52**

Table 8: The attack impact for each state-of-the-art model poisoning attack $\mathcal{A}$ and the corresponding CLP augmented attack, i.e, $\mathcal{A}$-CLP under various threat models when *the benign gradients are known* to attack $\mathcal{A}$. In all settings, the impact of CLP augmented $\mathcal{A}$-CLP attack is significantly higher than that of $\mathcal{A}$ attack itself.

where the server uses the Multi-krum or Trimmed-mean aggregation rules. We evaluate the CLP identified by four attacks considered in this paper. As shown in Figures 9 and 10, where we compute the FGN from both the perspectives of the central server and the adversary $\mathcal{A}$, we observe that the CLPs identified by the central server and the adversary are very similar. Hence, our proposed FGN approach is robust and can be applied to both the FL central server and the adversary.

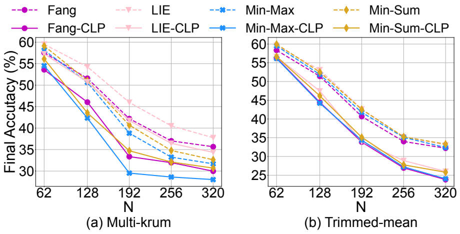
C.6 Significance of CLP-Awareness

Complementary to Table 1, we present the results when the benign gradients are known are known to attack $\mathcal{A}$ in Table 8 and the testing accuracy using AlexNet on non-IID partitioned CIFAR-10 when the underlying aggregation rule is Multi-Krum with known benign gradients in Figure 12. We can draw same conclusions from Table 8 as those from Table 1, and hence we omit further discussions here.



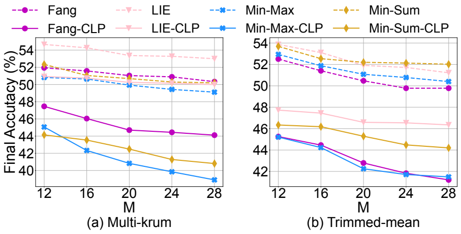
(a) Fang (b) LIE (c) Min-Max (d) Min-Sum

Figure 12: CLP augmented attacks to FL with the Multi-Krum aggregation rule using AlexNet on non-IID partitioned CIFAR-10. All attacks *know the gradients on benign clients*.



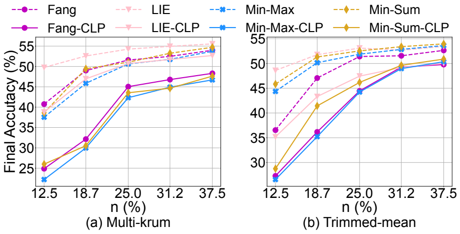
(a) Multi-krum (b) Trimmed-mean

Figure 13: Number of clients: N.



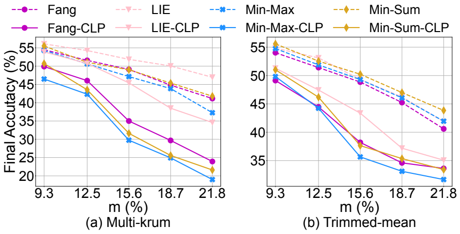
(a) Multi-krum (b) Trimmed-mean

Figure 14: Number of malicious clients: M.



(a) Multi-krum (b) Trimmed-mean

Figure 15: Number of participated clients in each round: n.



(a) Multi-krum (b) Trimmed-mean

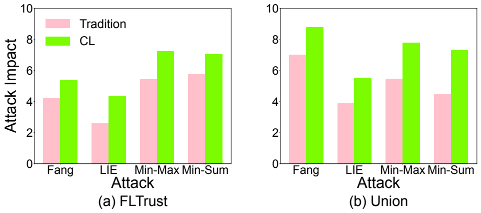
Figure 16: Number of participated malicious clients in each round: m.

C.7 Additional Ablation Study

In our default setting (see Section 3.2.1), we consider a total number of $N = 128$ clients in our experiments and the adversary controls $M = 32$ clients. In each round, the FL central server randomly selects $n = 16$ clients to participate in the global model update, in which $m = 4$ are malicious clients. We now investigate the impact of these four hyperparameters with Multi-krum aggregation rules using AlexNet on non-IID partitioned CIFAR-10 as illustrated in Figures 13, 14, 15 and 16. In each case, we investigate one parameter and keep other three parameters fixed as in the default setting. Across all settings, the importance of CLP awareness consistently exhibit, i.e., $\mathcal{A}$-CLP always outperforms its counterpart in all settings.

C.8 Results on Shakespeare Dataset

The attack impact of $\mathcal{A}$ and $\mathcal{A}$-CLP under various threat models using non-IID partitioned Shakespeare dataset when the gradients of benign clients are either unknown or known to the adversary is summarized in Table 9. Similar to the observations for image classification tasks, in this next-character prediction task, we still observe that for any attack $\mathcal{A}$, when augmented by CLP (i.e., the **CL** columns in Table 9, the attack impact is dramatically improved compared to attack $\mathcal{A}$ itself. $\mathcal{A}$-CLP attack is 1.2× to 3.8× more impactful than $\mathcal{A}$ attack itself. Likewise, the resilience against defenses is also improved as shown in Figure 17.



(a) FLTrust (b) Union

Figure 17: Attack impacts of $\mathcal{A}$ and $\mathcal{A}$-CLP when defended by FLTrust and Union with Trimmed-mean using LSTM on non-IID partitioned Shakespeare.

Benign Clients	Aggregation Rule	No Attack (Accuracy)	Fang		LIE		Min-Max		Min-Sum	
			Tradition	CL	Tradition	CL	Tradition	CL	Tradition	CL
Unknown	Multi-krum	47.14	9.65	**11.94**	2.65	**4.73**	8.8	**11.75**	8.08	**11.07**
	Bulyan	46.52	10.38	**13.71**	1.63	**3.48**	8.25	**12.14**	7.71	**11.5**
	Trimmed-mean	46.93	9.03	**12.18**	2.23	**3.98**	8.26	**11.12**	7.92	**10.76**
	Median	45.76	9.09	**11.53**	1.37	**3.16**	7.45	**10.38**	7.05	**9.96**
	AFA	47.41	7.19	**10.14**	4.09	**5.5**	8.58	**10.98**	8.47	**9.91**
Known	Multi-krum	47.14	12.6	**14.89**	3.73	**5.57**	12.53	**15.35**	11.98	**14.55**
	Bulyan	46.52	12.2	**14.83**	2.32	**4.27**	12.23	**15.58**	11.97	**14.89**
	Trimmed-mean	46.93	10.18	**13.82**	3.3	**5.85**	9.58	**12.57**	9.35	**12.27**
	Median	45.76	10.72	**13.87**	0.93	**3.56**	9.46	**12.52**	9.17	**12.17**
	AFA	47.41	9.06	**11.39**	6.3	**8.12**	10.1	**12.38**	9.25	**11.28**

Table 9: The attack impact for state-of-the-art model poisoning attack $\mathcal{A}$ and the corresponding CLP augmented attack $\mathcal{A}$-CLP under various threat models using non-IID partitioned Shakespeare dataset.

D Additional Discussions on CLP Augmented Similarity-based Attack

Background. In the general setting of model poisoning attacks to FL, there is one global optimization goal, which is to maximize the damage to the global model Fang et al. (2020); Shejwalkar & Houmansadr (2021). Specifically, let $s_j(t)$ be the changing direction of the j-th global model parameter in round t when there is no attack, where $s_j(t) = 1$ (resp. $s_j(t) = 1$) means that the j-th global model parameter increases (resp. decreases) upon the previous round. Denote $\mathbf{s}(t) = (s_j(t))_{j=1,\cdots,d}$. Suppose in round t, $\mathbf{w}_i(t)$ (resp. $\mathbf{g}_i(t)$) is the local model (resp. gradient) update that client i tends to send to the central server when there is no attack, and $\tilde{\mathbf{w}}_i(t)$ (resp. $\tilde{\mathbf{g}}_i(t)$) is the local model (resp. gradient) update if client i is compromised. Like most of the existing attacks, e.g., Fang et al. (2020); Shejwalkar & Houmansadr (2021), we restrict ourselves to

$$\tilde{\mathbf{w}}_i(t) := \mathbf{w}_i(t) - \eta\lambda_i\mathbf{s}_t, \tag{3}$$

which models the deviation between the crafted local model $\tilde{\mathbf{w}}_i(t)$ and the before-attack local model $\mathbf{w}_i(t)$, with $\lambda_i > 0$. Since $\mathbf{w}_i(t) = \mathbf{w}_i(t-1) - \eta\mathbf{g}_i(t)$ and $\tilde{\mathbf{w}}_i(t) = \mathbf{w}_i(t-1) - \eta\tilde{\mathbf{g}}_i(t)$,where $\mathbf{w}_i(t-1)$ is the received latest global model at the beginning of round t, from (3), we have $\tilde{\mathbf{g}}_i(t) = \mathbf{g}_i(t) + \lambda_i\mathbf{s}_t$. The adversary's goal is then to derivate the global model parameter *the most* towards the *inverse of the direction* along which the global model parameter would change without attacks at each round t, i.e.,

$$\max_{\tilde{\mathbf{g}}_1(t),\cdots,\tilde{\mathbf{g}}_m(t)} \mathbf{s}_t^\intercal(\mathbf{g}_t - \tilde{\mathbf{g}}_t), \tag{4}$$

$$\text{s.t.} \quad \mathbf{g}(t) = \mathcal{H}(\mathbf{g}_1(t), \cdots, \mathbf{g}_m(t), \mathbf{g}_{m+1}(t), \cdots, \mathbf{g}_n(t)), \tag{5}$$

$$\tilde{\mathbf{g}}(t) = \mathcal{H}(\tilde{\mathbf{g}}_1(t), \cdots, \tilde{\mathbf{g}}_m(t), \mathbf{g}_{m+1}(t), \cdots, \mathbf{g}_n(t)), \tag{6}$$

where $\mathcal{H}(\cdot)$ is an aggregation rule (see Section B.2). Given (3), Fang et al. Fang et al. (2020) showed that the above optimization problem can be transformed to one with the objective function of $(\lambda_i)_{i=1,\cdots,m}$. However, optimizing such a global objective for $(\lambda_i)_{i=1,\cdots,m}$ becomes difficult due to highly non-linear constraints, large state space of local models and non-IID local data distributions at each client Li et al. (2020). Below, we first provide intuition behind our attack and then propose a CLP-aware similarity-based poisoning attack, `SimAttack-CLP` to compromise malicious clients in FL.

`SimAttack-CLP`. Therefore, for a given attack threshold τ, *the goal of the adversary* is to find the changing directions via $\lambda_i, \forall i$ to craft gradients of each of the m^{CLP} malicious clients by solving (7), i.e.

$$\mathcal{F}(\mathbf{g}(t), \tilde{\mathbf{g}}(t)) = \tau, \tag{7}$$

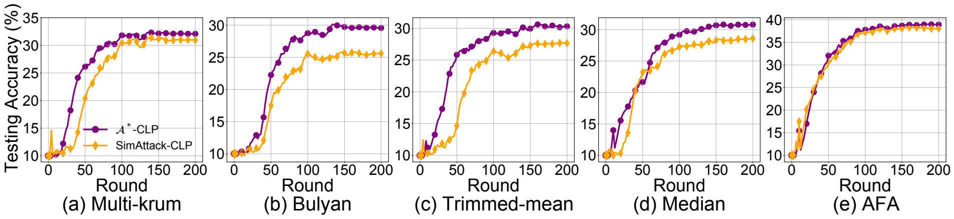
where $\mathbf{g}(t)$ and $\tilde{\mathbf{g}}(t)$ are given in (5) and (6), respectively, and $\tilde{\mathbf{g}}_i(t) = \mathbf{g}_i(t) + \lambda_i\mathbf{s}_t, \forall i = 1, \cdots, m^{\text{CLP}}$. The key challenge in solving (7) is that the adversary does not know the aggregation rule $\mathcal{H}(\cdot)$. To address this challenge, we make one approximation. Note that the attack threshold provides a "flexibility" to the adversary so that it does not need to attack towards the most inverse direction by solving a complex optimization problem, and hence our approximation here can be treated as part of such a flexibility. As we will demonstrate in our experiments, our `SimAttack-CLP` attack using such an approximation for all Byzantine-robust aggregation rules discussed in Section B.2 can already substantially increase the attack impact compared to the strongest state of the arts.

Specifically, we assume that the adversary adopts an "average rule" to approximate the aggregation rule of the server, i.e., $\mathbf{g}(t) \approx \frac{1}{n}\sum_{i=1}^{n}\mathbf{g}_i(t)$, $\tilde{\mathbf{g}}(t) \approx \frac{1}{n}\sum_{i=1}^{n}\tilde{\mathbf{g}}_i(t)$, and $\lambda \triangleq \sum_{i=1}^{m^{\text{CLP}}}\lambda_i$. Then we have $\tilde{\mathbf{g}}(t) \approx \mathbf{g}(t) + \lambda\mathbf{s}(t)$. Combing this approximation with (7), we can easily solve a so-called "global" λ that is common for all m^{CLP} malicious clients. Formally, we have the following proposition.

Proposition 1. *Suppose that λ is the changing direction to craft gradients of m^{CLP} malicious clients based on the cosine similarity (7). Then for any given attack threshold τ, the value of λ satisfies*

$$\lambda = \frac{-z - \sqrt{z^2 - 4xy}}{2x}, \tag{8}$$

where $x = (\boldsymbol{g}(t)^\intercal\boldsymbol{s}(t))^2 - \tau^2\|\boldsymbol{g}(t)\|^2\|\boldsymbol{s}(t)\|^2$, $y = (1-\tau^2)\|\boldsymbol{g}(t)\|^4$, and $z = 2(\tau^2 - 1)\|\boldsymbol{g}(t)\|^2\boldsymbol{g}(t)^\intercal\boldsymbol{s}(t)$.



(a) Multi-krum (b) Bulyan (c) Trimmed-mean (d) Median (e) AFA

Figure 18: Comparisons between testing accuracy of `SimAttack-CLP` and $\mathcal{A}^*$`-CLP`, which is the best among `LIE-CLP`, `Min-Sum-CLP` and `Min-Max-CLP`. All attacks *do not know the gradients on benign clients.*

D.1 Additional Numerical Results on SimAttack-CLP

Complementary to attack impact results presented in Table 2, we present the testing accuracy of `SimAttack-CLP` and $\mathcal{A}^*$`-CLP` using AlexNet on non-IID partitioned CIFAR-10 in Figure 18. Similar observations can be made in other cases and hence are omitted here. Again, we observe that `SimAttack-CLP` outperforms $\mathcal{A}^*$`-CLP`.

E Feasibility guarantee for the $\mathcal{A}$-CLP framework

As aforementioned, the central sever randomly selects a subset of n out of N clients to participate in the global model update in each round. In our $\mathcal{A}$`-CLP` framework, m out of n clients should be malicious clients. Then a natural question is that *how many clients in total (denoted as M) should the adversary $\mathcal{A}$ control so that our $\mathcal{A}$-CLP framework is feasible?* In the following, we provide a theoretical performance guarantee on the feasibility of $\mathcal{A}$`-CLP`. In other words, we determine the so-called control rate M of attack $\mathcal{A}$ such that the event that at least m malicious clients are selected in each round t and hence contribute to the global model update, occurs with a probability p_0, i.e.,

$$\frac{1}{\binom{N}{n}} \sum_{i=m}^{\min(M,n)} \binom{N-M}{n-i}\binom{M}{i} \geq p_0. \tag{9}$$

Unfortunately, (9) is hard to be solved directly due to the computational complexity, especially when N is large. Our key insight is that this problem can be equivalently transformed into a *hypergeometric distribution problem* Harkness (1965); Guenther (1978). Specifically, denote X as a random variable indicating the number of malicious clients selected by the central server at each round, which follows the hypergeometric distribution, i.e., $X \sim H(n, M, N)$, with its mean $\tilde{\mu} = \frac{nM}{N}$ and variance $\tilde{\sigma}^2 = \frac{nM}{N}(1-\frac{M}{N})\frac{N-n}{N-1}$. When the total number of clients N is large, the hypergeometric distribution can be approximated by the binomial distribution and hence X approximately follows the normal distribution $\psi(\tilde{\mu}, \tilde{\sigma}^2)$ due to the central limit theorem. As a result, the number of selected malicious clients X satisfies

$$\mathbb{P}(X \geq m) = \mathbb{P}\left(\frac{X-\tilde{\mu}}{\tilde{\sigma}} \geq \frac{m-\tilde{\mu}}{\tilde{\sigma}}\right) \geq p_0, \tag{10}$$

where $\frac{X-\tilde{\mu}}{\tilde{\sigma}} \sim \psi(0,1)$. Therefore, we can obtain M by solving (10), which satisfies

$$M \geq \frac{-b+\sqrt{b^2-4ac}}{2a}, \tag{11}$$

where $a = \frac{2n}{N} + (Q(p_0))^2\frac{(N-n)n}{N^2(N-1)}$, $b = -2\frac{nm}{N} - (Q(p_0))^2\frac{(N-n)n}{N(N-1)}$, $c = m^2$, and $Q(p_0) = \frac{m-\tilde{\mu}}{\tilde{\sigma}}$ is the quantile of normal distribution. We remark that (11) can be easily solved for any given n, m, N, p_0.

We now numerically evaluate the performance of our proposed lightweight approximated method in Equation (11) to determine the control rate, i.e., M of attack $\mathcal{A}$ for any given n, m, N, p_0. We compare it with the exact results computed from Equation (9), which is order of magnitude complex than our method in Equation (11). For ease of complexity (mainly for computing Equation (9)), we consider two cases: (i) $N = 128$, and the FL central server selects $n = 16$, 24 or 32 clients for global model update in each round; and (ii) $N = 256$, and the FL central server selects $n = 32$, 48 or 64

$p_0 = 0.55$	Method	$n = 16$	$n = 24$	$n = 32$
$m = 0.125n$	Equation (9)	15	15	16
	Equation (11)	17	17	17
$m = 0.25n$	Equation (9)	31	31	32
	Equation (11)	33	33	33
$m = 0.375n$	Equation (9)	47	47	48
	Equation (11)	50	49	49

Table 10: Comparison of the control rate M computed by Equation (9) and Equation (11) when $N = 128$ and $p_0 = 0.55$.

$p_0 = 0.55$	Method	$n = 32$	$n = 48$	$n = 64$
$m = 0.125n$	Equation (9)	31	32	32
	Equation (11)	34	33	33
$m = 0.25n$	Equation (9)	63	64	64
	Equation (11)	66	66	65
$m = 0.375n$	Equation (9)	95	96	96
	Equation (11)	98	98	98

Table 11: Comparison of the control rate M computed by Equation (9) and Equation (11) when $N = 256$ and $p_0 = 0.55$.

clients for global model update in each round. The adversary needs to guarantee m malicious clients are selected with probability p_0. Specifically, we consider the following cases with $m = 0.125n$ to $m = 0.375n$ and $p_0 = 0.55, 0.9$. The number of malicious clients M that the adversary need to control computed by Equation (9) and Equation (11) for the above cases are presented in Tables 10 and 12, and Tables 11 and 13, respectively. It is clear that the results computed by these two methods are quite close to each other, especially when n and m become larger. Hence we use our lightweight method in Equation (11) to determine the control rate M for the adversary in our experiments.

$p_0 = 0.9$	Method	$n = 16$	$n = 24$	$n = 32$
$m = 0.125n$	Equation (9)	28	26	24
	Equation (11)	31	27	25
$m = 0.25n$	Equation (9)	47	44	42
	Equation (11)	49	45	43
$m = 0.375n$	Equation (9)	64	61	59
	Equation (11)	65	62	59

Table 12: Comparison of the control rate M computed by Equation (9) and Equation (11) when $N = 128$ and $p_0 = 0.9$.

$p_0 = 0.9$	Method	$n = 32$	$n = 48$	$n = 64$
$m = 0.125n$	Equation (9)	49	46	44
	Equation (11)	52	47	44
$m = 0.25n$	Equation (9)	86	82	79
	Equation (11)	87	82	79
$m = 0.375n$	Equation (9)	119	115	112
	Equation (11)	120	115	112

Table 13: Comparison of the control rate M computed by Equation (9) and Equation (11) when $N = 256$ and $p_0 = 0.9$.