EMGAN: Early-Mix-GAN on Extracting Server-Side Model in Split Federated Learning

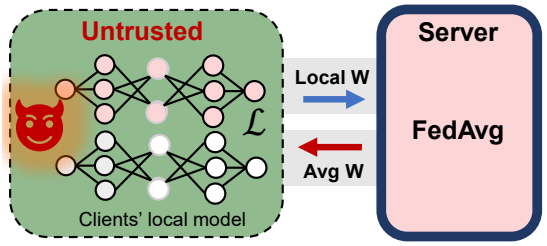
Jingtao Li[1]*, Xing Chen[2], Li Yang[3], Adnan Siraj Rakin[4], Deliang Fan[5], Chaitali Chakrabarti[2]

[1]Sony AI
[2]Arizona State University
[3]University of North Carolina at Charlotte
[4]Binghamton University (SUNY)
[5]Johns Hopkins University

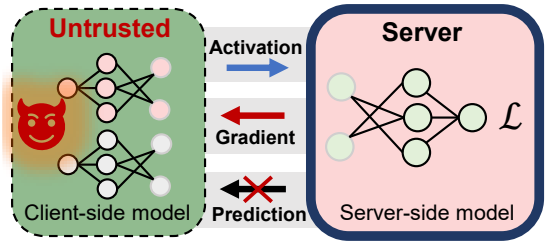
jingtao.li@sony.com, xchen382@asu.edu, lyang50@uncc.edu, arakin@binghamton.edu, dfan10@jhu.edu, chaitali@asu.edu

Abstract

Split Federated Learning (SFL) is an emerging edge-friendly version of Federated Learning (FL), where clients process a small portion of the entire model. While SFL was considered to be resistant to Model Extraction Attack (MEA) by design, a recent work (Li et al. 2023b) shows it is not necessarily the case. In general, gradient-based MEAs are not effective on a target model that is changing, as is the case in training-from-scratch applications. In this work, we propose a strong MEA during the SFL training phase. The proposed Early-Mix-GAN (EMGAN) attack effectively exploits gradient queries regardless of data assumptions. EMGAN adopts three key components to address the problem of inconsistent gradients. Specifically, it employs (i) Early-learner approach for better adaptability, (ii) Multi-GAN approach to introduce randomness in generator training to mitigate mode collapse, and (iii) ProperMix to effectively augment the limited amount of synthetic data for a better approximation of the target domain data distribution. EMGAN achieves excellent results in extracting server-side models. With only 50 training samples, EMGAN successfully extracts a 5-layer server-side model of VGG-11 on CIFAR-10, with 7% less accuracy than the target model. With zero training data, the extracted model achieves 81.3% accuracy, which is significantly better than the 45.5% accuracy of the model extracted by the SoTA method. The code is available at https://github.com/zlijingtao/SFL-MEA.



(a) IP threat in standard Federated Learning (FL)



(b) IP threat in Split Federated Learning (SFL)

Figure 1: IP threats in Federated Learning. (a) FL suffers from direct model leakage. (b) SFL prevents direct model leakage and is immune to prediction-based MEAs by blocking prediction queries.

Introduction

Federated Learning (FL) has become increasingly popular thanks to its ability to protect users' data and comply with the General Data Protection Regulation policy. In FedAvg (McMahan et al. 2017), which is the *standard FL* scheme, clients locally update their model copies and send them to the server, which then aggregates the model parameters and sends the aggregated model back to the clients. Such a setting only allows model parameters to be shared with the server, and direct data sharing is avoided. Standard FL is clearly vulnerable to Intellectual Property (IP) threat as a malicious client can acquire the entire model for free (Fig. 1 (a)). Also, the stolen model can be used to perform much more effective adversarial attacks (Goodfellow, Shlens, and Szegedy 2014), making model IP security in FL absolutely essential.

Split Federated Learning (SFL) scheme (Thapa et al. 2020) is a variant of FL for training with resource-constrained clients, where the neural network is split into a client-side model and a server-side model. Each client only computes the forward/backward propagation of the smaller client-side model while the server, which has more compute resources, computes the forward/backward propagation of the larger server-side model. SFL follows the same model averaging mechanism as standard FL to synchronize clients' models. Unlike standard FL which passes on the entire model over to the clients, SFL preserves the server-side model and prevents the model from direct leakage. More-

*Work was done when Jingtao Li was a PhD student at Arizona State University.

over, since the attacker does not necessarily have access to prediction logits as illustrated in Fig. 1 (b), all the prediction-based MEAs (Correia-Silva et al. 2018; Orekondy, Schiele, and Fritz 2019; Truong et al. 2021) fail to succeed.

However, SFL also suffers from model leakage due to (1) availability of client-side models, and (2) access to gradients sent by the server to clients. It is demonstrated in (Li et al. 2023b) that the client-side model along with a small amount of training data is enough to get an accurate model using Train-ME, a method that simply trains the server-side model from scratch. (Li et al. 2023b) also mentions four other gradient-based attacks, which can succeed when the model parameters are fixed. However, these attacks perform poorly, when the target model is changing such as in the training-from-scratch case. We attribute the failure of these methods to inconsistent gradient statistics caused by the changes in the target model during training.

In this paper, we focus on enhancing MEA on SFL for the training-from-scratch case. We propose EMGAN, which stands for Early-Mix-GAN, a new MEA that can effectively utilize the gradient queries and is suitable for different attacker data assumptions. To address the inconsistent gradient problem, we adopt three key strategies. First, we employ the Early-learner approach to train a surrogate model that adapts better to the target model during SFL training. Second, we adopt multiGAN to allow randomness in generator training to mitigate mode collapse. Finally, we propose ProperMix to augment the limited amount of synthetic data so that it approximates the target data distribution better.

EMGAN works for both with-data and data-free scenarios and achieves excellent results in extracting models. For attackers with training data, EMGAN succeeds in extracting a VGG-11 model with only 50 samples, with only a 7% drop in accuracy. And for the data-free case, where the attacker has no training data, EMGAN succeeds with an 11% accuracy drop. Worth noting, the data-free EMGAN excels under strong Non-IID client data distribution, delivering better MEA performance than with-data EMGAN. In summary, we make the following contributions:

- We propose a strong MEA on SFL that utilizes a model-based attack methodology and Early-learner approach for better adaptability to a changing target model.
- To further improve attack performance, we adopt MultiGAN and ProperMix methods. Our analysis shows that MultiGAN mitigates mode collapse by increasing the output variance. ProperMix utilizes gradients to optimize the mixture so that the resulting data distribution better approximates the target data distribution.
- Our empirical experiments show the effectiveness of EMGAN. Using the VGG-11 architecture on the CIFAR-10 classification task, with a client-side model consisting of 6 layers, our results demonstrate significant improvements over previous methods. Without training data, EMGAN achieves 81.3% accuracy, which is much better than the 45.5% accuracy achieved by GAN-ME. With only 50 training samples, EMGAN achieves around 83.7% accuracy, outperforming the 71.7% accuracy obtained by Train-ME, the current SoTA method.

Related Work

Split Federated Learning

Scheme Detail The idea of splitting the model into a client-side model and a server-side model was first proposed in (Kang et al. 2017; Teerapittayanon, McDanel, and Kung 2017; Liu, Qi, and Banerjee 2018) for inference tasks and extended by (Thapa et al. 2020) into Split Federated Learning (SFL), a collaborative learning scheme suitable for resource-constrained devices. In SFL, clients perform forward propagation locally till the last layer of client-side model, sending the intermediate activation with label information to the server. The server processes the activation on the server-side model, calculates the loss, performs backpropagation, and returns the gradients back to clients to update their local copies of the client-side model. After each epoch, periodic synchronization is performed as in FedAvg (McMahan et al. 2017). Among the two variants of SFL proposed in (Thapa et al. 2020), we adopt the SFL-V1 scheme since it is a more favorable option for its better scalability to a large number of clients and Non-IID performance (Li et al. 2023a).

Data Security in SFL Similar to FL, SFL scheme avoids sending clients' data directly to the server. However, data protection in SFL can be compromised by attacks such as Model Inversion Attack (MIA). In model-based MIA (Fredrikson, Jha, and Ristenpart 2015), the attacker can directly reconstruct raw inputs from the intermediate activation. According to (Vepakomma et al. 2020; Singh et al. 2021), MIAs in SFL are highly successful, especially in SFL with shallow client-side models. Recently, MI-resistant SFL schemes ResSFL have been proposed (Li et al. 2022, 2023a). They require the model owner to have an auxiliary dataset from a similar domain to perform the pretraining.

Model Extraction Attack

In SFL, the model is split and so basic IP threat due to directly downloading the model is non-existent. However, advanced IP threats such as Model Extraction Attack (MEA) can exist. Here we elaborate on the potential of prediction-based MEA and gradient-based MEA.

Prediction-based MEA MEA is first demonstrated for model prediction API in (Tramèr et al. 2016). More recent works include CopyCat CNN (Correia-Silva et al. 2018), Knockoff-random (Orekondy, Schiele, and Fritz 2019) showcase this attack on deep neural network. The follow-up work (Jagielski et al. 2020) shows that a high-fidelity and accurate model can be obtained with very few model prediction queries. Moreover, (Truong et al. 2021) proposed an MEA that can succeed without any authentic training data, where the attacker only has noise or data from a totally different distribution. These *prediction-based* MEAs all rely on the assumption that the attacker has query access to the prediction logits or labels of the target model. However, in an SFL scheme, the server does not have to send back the model outputs during the training process and so such an assumption does not hold.

Gradient-based MEA More recently, (Li et al. 2023b) investigated *gradient-based* MEAs and proposed five ME attacks (Craft, GAN, GM, Train, and SoftTrain MEs) tailored for SFL. However, we found that these attacks are suitable for a model that has frozen model parameters and fails to work well when the attack occurs during the training of the target model. So in the rest of the paper, we focus on improving gradient-based MEA performance for training-from-scratch SFL applications.

Threat Model

Attack Assumption The attacker is a client who participates in the SFL training to extract the server-side model. Thus, similar to normal clients, the attacker holds *white-box assumption* on the client-side model, that is, it knows the exact model architecture and parameters of those layers. The attacker holds a *grey-box assumption* on the server-side model, that is, it knows its architecture and loss function but does not know the model parameters. The goal of the attacker is to (i) obtain a surrogate model that maximizes the prediction correctness, and (ii) derive a surrogate model with a similar decision boundary as the target model, and use it to launch adversarial attacks (Biggio et al. 2013).

We assume that the attackers' data assumption falls into two categories: (1) data-free, where the attacker has no access to training data and (2) with-data, where the attacker has a very limited amount ($< 2\%$) of training data. In the data-free case, the attacker may use randomly generated noise as input to feed into the SFL scheme. This case corresponds to a popular assumption adopted in Truong et al. (2021) and in data-free knowledge distillation (Fang et al. 2022). In a collaborative learning scenario, this can happen when the attacker participates as a "free-rider" without contributing any data, or when the attacker does not have a *similar enough* dataset.

SFL Assumption We denote N as the total number of layers (or layer-like blocks, i.e. BasicBlock in ResNet) of the target neural network model, and L as the number of layers of the client-side model. Specifically, the scheme owner (victim) chooses L as the minimum number of layers at the client-side model which results in the reconstructed images having an MSE greater than a threshold compared to the ground-truth. We refer to L as the *secure* cut-layer setting. In practice, to find the minimum number of layers, we can use another dataset as a proxy and sweep the cut-layers to estimate L. Such an example is demonstrated in Fig. 2, where choosing L to be 6 meets the threshold requirement for CIFAR-10 and CIFAR-100.

We assume that the server *allows gradient queries* so that the clients can update their client-side models. Based on a client's activation $\boldsymbol{A} = C(\boldsymbol{x})$ and its label $\boldsymbol{y}$, gradient information $\nabla_{\boldsymbol{A}}\mathcal{L}$ is computed and sent back to clients. We also assume neither logits nor prediction labels are accessible by the attacker since they are not necessary during training.

Proposed EMGAN Attack

The proposed EMGAN consists of three components as illustrated in Fig. 3. It requires the attacker to (i) Apply Early-learner by performing mini-batch SGD on training the surrogate model from the start of SFL training, (ii) use MultiGAN to sample a random generator from a pool of generators, and (iii) perform ProperMix, send the mixture together with real data to the server, and utilize the gradients sent back from the server to optimize the generator.

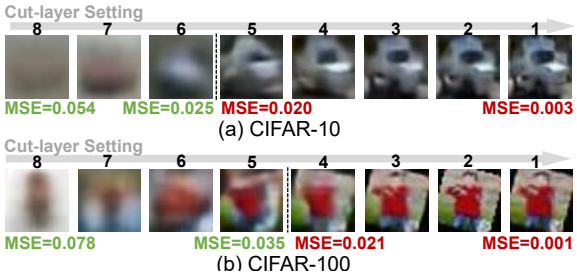


Figure 2: Choice of secure cut-layer L for MSE threshold of 0.025, L=6 for CIFAR-10 and L=5 for CIFAR-100.

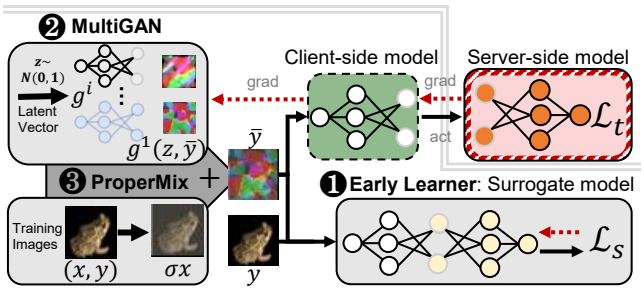


Figure 3: Proposed EMGAN. It is equipped with three components: (1) Early Learner: embedded surrogate model during SFL training, (2) MultiGAN: use of multiple generator models, and (3) ProperMix: a trainable mixing method to augment the training data.

Inconsistent Gradient Problem

We have identified that gradient consistency plays a crucial role in MEAs. In finetuning case when the model is mostly static, attackers can obtain consistent gradient information through gradient queries. It is the consistent gradients that contribute to the success of MEAs such as Craft-ME and GAN-ME in (Li et al. 2023b).

However, when it comes to training-from-scratch scenarios, queries to the SFL model yield inconsistent gradient information due to significant changes in the server-side model during the training process. As shown in Fig. 4, for the same query input, the gradient statistics (mean and standard deviation) differ drastically in different epochs.

These inconsistent gradients present a challenge for gradient-based MEAs to effectively utilize the gradient information. Prior works (Li et al. 2023b) demonstrate that for training-from-scratch case, gradient-based MEAs result in a large accuracy drop of over 40% in the surrogate model.

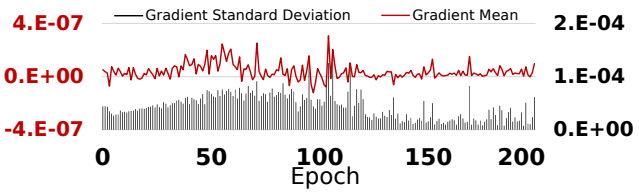


Figure 4: Inconsistent gradient problem in training-from-scratch SFL.

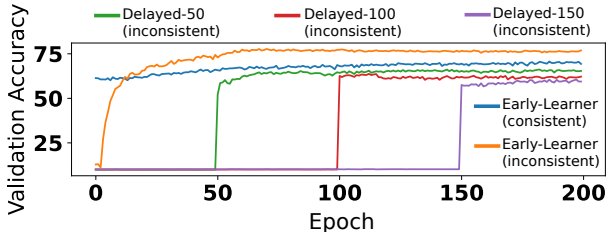


Figure 5: Validation accuracy for different starting epochs of surrogate model training.

Taking Advantage of Inconsistent Gradient

A pitfall of prior work (Li et al. 2023b) is that the surrogate model training only starts after the SFL training is completed. Thus, the surrogate model is only being trained using the latest synthesized samples (small variance). We hypothesize that more variant synthesized samples during the training can be used to launch a stronger attack. So, we develop a new model-based attack method that utilizes gradients to train a generator to create synthetic samples. At the core of the EMGAN attack is the *Early-Learner* approach, where we begin training the surrogate model right from the beginning of the SFL process.

We demonstrate the effectiveness in Fig. 5, where we plot the surrogate model accuracy for five cases: Early-learner with consistent and inconsistent gradients, where the surrogate model training starts from epoch 0, and inconsistent gradients with starting delay of 50, 100 and 150 epochs. First, we find that Early-learner using inconsistent gradients has a slower convergence rate but achieves better accuracy performance than consistent gradients. As expected, inconsistent gradients help produce more varied synthetic samples across training epochs, resulting in better surrogate model accuracy. Second, as we vary the starting point of the surrogate model training, we found that cases that start late (Delayed-50, 100, 150) have an initial steep increase in accuracy but saturate at a lower accuracy compared to the one that starts early. This indicates that generated outputs in early epochs serve as an easy curriculum that lowers the learning difficulty for the surrogate model. This is very similar to progressive curriculum learning (Bengio et al. 2009).

Mitigating Model Collapse Problem

Using the Early-Learner approach, the surrogate model is able to train with samples that have increased *inter-epoch variety*. However, we notice the *intra-epoch variety* is still limited. We observe a serious mode collapse problem (Goodfellow 2016) since the synthetic samples generated in each epoch are almost identical (see Fig. 6).

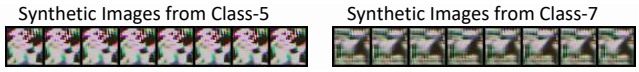


Figure 6: Visualization of the mode collapse.

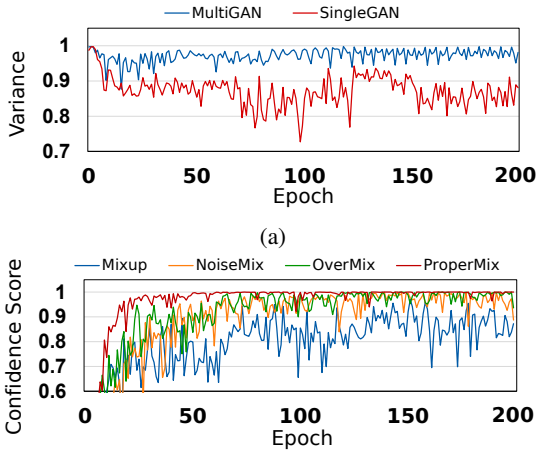


Figure 7: Analysis of EMGAN components. (a) Randomness induced by Multi-GAN improves the variance of synthesized images. (b) ProperMix creates in-distribution synthesized samples with high confidence score.

To address the performance limitation caused by mode collapse, we propose the use of MultiGAN (Hoang et al. 2017). MultiGAN uses a number of N_G identical conditional-generator models and uses them one at a time in a round-robin fashion. The randomness in initialization and training dynamics helps mitigate the mode collapse problem. Fig. 7 (a) shows how MultiGAN technique generates synthesized images with a higher variance in images from the same class throughout the training.

Augmentation with ProperMix

We denote the training data for training the surrogate model as $\mathbf{X}_S$. To ensure a high accuracy on the surrogate model, it is crucial that $\mathbf{X}_S$ well represent the target training data distribution $\mathbf{X}_t$. However, if the attacker has limited training data, $\mathbf{X}_S$ cannot represent $\mathbf{X}_t$ well, even with the help of MultiGAN.

Applying data augmentation is a straightforward way to improve surrogate accuracy. For attackers with training data, we can apply Mixup (Zhang et al. 2017), a method that has been successfully used in knowledge distillation. In Mixup, the mixture $\mathbf{X}_{mixup}$ is composed as follows:

$$\mathbf{X}_{mixup} = (1-\alpha)\mathbf{X}_{syn} + \alpha\mathbf{X}_{real} \quad (1)$$

While Mixup mixes data from the training dataset, we mix training data with synthetic data, or synthetic data with synthetic data, since training data may not be easily available.

Unfortunately, direct application of the Mixup method reduces the surrogate model accuracy because of the distribution conflicts between training data and synthetic data. Without proper regularization, the mixed images can move out of distribution and cause the decision boundary of the surrogate model to deviate from that of the target model. In Fig. 7 (b), we observe that the target model has a low confidence score on the mixture for the mixup method, suggesting that our hypothesis about the mixture falling outside the training data distribution of the target model, is correct.

To address the discrepancy between the distribution of training and synthetic data, we propose ProperMix, which creates mixtures during SFL training and sends them to the server. The generator is then forced to shift the distribution of mixtures towards the desired data distribution so that the cross-entropy loss can be minimized. Hence, the mixture stays in distribution and the target model no longer gets confused by them. This is illustrated in Fig. 7 (b), which shows that the target model has a higher confidence score on the ProperMix mixture than on the original Mixup method.

We also notice that if the mixing is too much (with a large $alpha$), the mixture cannot stay in distribution. This is illustrated in the lower confidence score of "overMix" in Fig. 7 (b). We observe that using the original formula in Eq. (1), the mixing tends to lean towards overmixing. To prevent the mixing from being excessively strong, we incorporate a clipping function to limit its impact. Furthermore, to ensure a robust generator, we eliminate the scaling factor applied to the synthesized images. So the mixing formula in ProperMix is defined as follows:

$$\mathbf{X}_{propermix} = clip(\mathbf{X}_{syn} + \alpha \mathbf{X}_{real}, -1, 1) \qquad (2)$$

where α is randomly sampled in every training step to encourage variation. The sampling of α follows a uniform distribution between $[\alpha_{min}, \alpha_{max}]$. The final $\mathbf{X}_S$ is a concatenation of $\mathbf{X}_{propermix}$ and $\mathbf{X}_{real}$, with each contributing equally. A detailed procedure for ProperMix is given in Algorithm 1.

Algorithm 1: ProperMix Method

function PROPERMIX-WITH-DATA(G)
  $\boldsymbol{z} \leftarrow randn([B/2,]), \alpha \leftarrow rand(\alpha_{min}, \alpha_{max})$
  Draw Data $(\boldsymbol{x}_s, \boldsymbol{y}_s) \leftarrow (\mathbf{X}_0, \mathbf{Y}_0)$
  Synthesize $\boldsymbol{x}_{syn} \leftarrow \mathrm{G}(\boldsymbol{z}, \boldsymbol{y}_s[B/2:])$
  Mixture $\boldsymbol{x}_m \leftarrow$ **clip**($\boldsymbol{x}_{syn}$ + $\alpha \boldsymbol{x}_s$[:B/2])
  Image $\boldsymbol{x}_0 \leftarrow$ **concat**($\boldsymbol{x}_m, \boldsymbol{x}_s[B/2:]$)
  Label $\boldsymbol{y}_0 \leftarrow$ **concat**($\boldsymbol{y}_s[B/2:], \boldsymbol{y}_s[B/2:]$)
  return $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
end function
function PROPERMIX-DATA-FREE(G)
  $\boldsymbol{z} \leftarrow randn([B,]), \alpha \leftarrow rand(\alpha_{min}, \alpha_{max})$
  Label $\boldsymbol{y}_0 \leftarrow randint(0, N_{class}, [B])$
  Synthesize $\boldsymbol{x}_{syn} \leftarrow \mathrm{G}(\boldsymbol{z}, \boldsymbol{y}_0)$
  Mixture $\boldsymbol{x}_m \leftarrow$ **clip**($\boldsymbol{x}_{syn}$[B/2:] + $\alpha \boldsymbol{x}_{syn}$[:B/2])
  Image $\boldsymbol{x}_0 \leftarrow$ **concat**($\boldsymbol{x}_m, \boldsymbol{x}_{syn}[B/2:]$)
  return $(\boldsymbol{x}_0, \boldsymbol{y}_0)$
end function

Algorithm 2: EMGAN during SFL Training

Require: For M clients, instantiate private training data $(\mathbf{X}_i, \mathbf{Y}_i)$ for $1, 2, ..., M-1$, and training data $(\mathbf{X}_0, \mathbf{Y}_0)$ for the attacker client-0. We initialize client-side model C^i for every participant. We specify a proper range $(\alpha_{min}, \alpha_{max})$ for the mixing parameter, and intialize a pool of N_G conditional generators. B denotes the batch size.

1: initialize C^i, G, S
2: **for** epoch $t \leftarrow 1$ to num_epochs **do**
3: $C^* = \frac{1}{M}\sum_{i=1}^{M} C^i$
4: **for** client $i \leftarrow 1$ to M **in Parallel do**
5: $C^i \leftarrow C^*$
6: **for** step $s \leftarrow 1$ to num_batches **do**
7: **if** client i is attacker **then**
8: $\mathrm{G} \leftarrow G_{pool}[s\%N_G]$
9: $(\boldsymbol{x}_0, \boldsymbol{y}_0) \leftarrow$ **PROPERMIX**(G)
10: **TRAIN-SURROGATE**(C^0, S^*, $(\boldsymbol{x}_0, \boldsymbol{y}_0)$)
11: **else**
12: $(\boldsymbol{x}_i, \boldsymbol{y}_i) \leftarrow$ Draw from $(\mathbf{X}_i, \mathbf{Y}_i)$
13: **end if**
14: $\mathbf{A}^i = C^i(\boldsymbol{x}_i)$
15: **end for**
16: **end for**
17: $loss = \mathcal{L}_{CE}(S(\boldsymbol{W}_S; [\mathbf{A}^{0,1,...,M}], [\boldsymbol{y}_{0,1,...,M}])$
18: $\min_{\boldsymbol{W}^i_C, \boldsymbol{W}_S, \boldsymbol{W}_G}(loss)$
19: **end for**
20: **function TRAIN-SURROGATE**(C^0, S^*, $(\boldsymbol{x}_0, \boldsymbol{y}_0)$)
21: $\mathbf{A}^0 = C^0(\boldsymbol{x}_0)$
22: $loss = \mathcal{L}_{CE}(S^*(\boldsymbol{W}_S; \mathbf{A}^0), \boldsymbol{y}_0)$
23: $\min_{\boldsymbol{W}_{S^*}}(loss)$
24: **end function**

Implementation of EMGAN

The EMGAN algorithm is described in Algorithm 2, where three key components are marked in blue. For attackers with training data, we use the "propermix-with-data" function in Algorithm 1 to replace line 8 in Algorithm 2. For attackers without training data, we use the "propermix-data-free" function.

Experiments

Experiment Setting

In this section, we demonstrate the performance of the proposed MEAs on SFL schemes. All experiments are conducted on a single RTX-3090 GPU. We use VGG-11 as the default model architecture.

Training Setting For model training, we set the total number of epochs to be 200. To perform MEAs, the attacker uses an SGD optimizer with a learning rate of 0.02 with decay (multiply learning rate by 0.2 at epochs 60, 120 and 160) to train the surrogate model and the generator. If not specified, we use a 5-client setting, where one of the clients is an attacker, and the other four clients are benign.

EMGAN Setting The surrogate model training uses the same setting as the target model, where only the server-side model is trainable. For MultiGAN, we set N_G to be 10. And for ProperMix, we empirically set $\alpha_{min}, \alpha_{max}$ to be 0.4 and 0.6 for EMGAN with-data, and set $\alpha_{min}, \alpha_{max}$ to be 0.6 and 0.8 for data-free EMGAN.

Dataset Setting We primarily use CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), SVHN (Netzer et al. 2011), and ImageNet-12 datasets (a subset of (Deng et al. 2009) used in (Li et al. 2023a), as they are used extensively in AI research. We use a fixed amount of training data for the attacker and distribute the remaining training data evenly among the benign clients, in an IID fashion, if not specified.

Cut-layer Setting As discussed in the threat model, we assume the SFL scheme cut-layer setting for each dataset-architecture combination strictly follows the privacy standard, which is client-side model must have enough layers to ensure the MSE is greater than 0.025. For example, based on Fig. 2, for VGG-11 on CIFAR-10, $L = 6$.

Non-IID Definition To describe the Non-IID data distribution among clients, we consider the pathological (aka. class-wise) non-IID distribution as in (McMahan et al. 2017; Zhuang, Wen, and Zhang 2022). For a non-IID degree of 0.2, we assign 2 classes randomly to each client if the dataset is CIFAR-10, SVHN or ImageNet-12, and assign 20 classes randomly to each client if the dataset is CIFAR-100.

Performance Metrics Since the attacker does not have access to the validation dataset, we use the final surrogate model accuracy as the reported extraction accuracy (instead of using the peak validation accuracy during the surrogate training). For the fidelity test, we use "label agreement" defined as the percentage of samples that the surrogate and target models agree with over the entire validation dataset, as in (Jagielski et al. 2020).

Performance of EMGAN

Comparison with Competing Methods – With-data Case For an attacker with 0.1% of the training samples (50 of the 50,000 CIFAR-10 training data), EMGAN achieves much better surrogate model accuracy than other competing methods, as shown in Fig. 8. The target model accuracy is 90.5%. Train-ME method in (Li et al. 2023b) achieves the lowest accuracy of 71.71%. Use of the "Early Learner" immediately improves the accuracy by about 5%. The effect of using a generator shown by "+ GAN" gets 3% better accuracy on top of Early-learner. After mitigating the mode collapse using "MultiGAN", the accuracy is improved by another 1%, showing that randomness in GAN training dynamics helps. Finally, the proposed ProperMix augmentation that takes a random α between 0.4 and 0.6, shows a boost in accuracy performance. All these techniques help EMGAN to achieve an accuracy of 83.65%, which is about 12% higher than Train-ME (Li et al. 2023b).

Comparison with Competing Methods – Data-free Case For an attacker without training data, EMGAN achieves significantly better surrogate model accuracy, as shown in Fig. 9. The GAN-ME method that performs the best in (Li et al. 2023b) achieves an accuracy of 45.45%. With "Early Learner", the accuracy immediately improves by over 28%. Using "MultiGAN", the accuracy improves by 1.5%. Finally, the proposed ProperMix augmentation boosts the MultiGAN accuracy performance by around 6%, resulting in EMGAN achieving an accuracy of 81.25%.

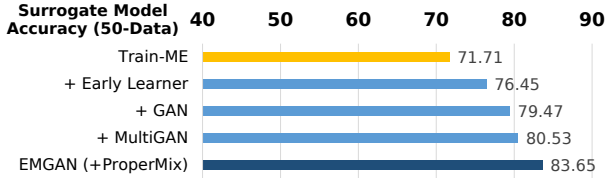


Figure 8: MEA performance comparison for attacker with 0.1% training data.

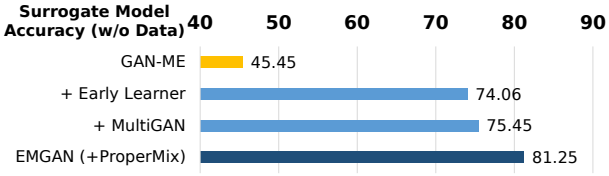


Figure 9: MEA performance comparison for the data-free attacker.

Non-IID Performance Under Non-IID data distribution, MEAs using limited training data have much lower accuracy performance due to the biased data distribution. Surprisingly, we found that data-free EMGAN outperforms EMGAN with training data, as shown in Table 1. This finding showcases the worth of data-free EMGAN.

Scalability We increase number of clients from 5 to 10, 20 and 50, and compare the performance with the baseline Train-ME method that does not rely on gradients. For 20 and 50 clients, we use client sampling ratios of 0.25 and 0.10 and increase the number of epochs to 800 and 2000 respectively to compensate for the sampling ratio. Table 2 shows that the MEA performance improves dramatically with more clients. We hypothesize that increasing the interval of attacks (with same number of attacking epochs) improves the MEA per-

Non-IID Degree	**EMGAN with 0.1% Data** 0.2	0.4	0.6	0.8	1.0 (IID)
Target	70.66	84.77	86.22	87.63	90.06
Surrogate	19.49	36.17	51.92	69.64	83.65
Non-IID Degree	**Data-free EMGAN** 0.2	0.4	0.6	0.8	1.0 (IID)
Target	63.25	77.11	80.54	85.31	89.18
Surrogate	65.47	68.89	73.03	78.14	81.25

Table 1: Proposed EMGAN under Non-IID client data distribution.

	EMGAN with 0.1% Data				
No. of Client	5	10	20	50	10
Sampling Ratio	1.0	0.5	0.25	0.125	1.0
Target acc.	90.06	90.51	90.64	91.25	89.11
Surrogate acc.	83.65	87.93	87.76	87.99	78.28
	Data-free EMGAN				
No. of Client	5	10	20	50	10
Sampling Ratio	1.0	0.5	0.25	0.125	1.0
Target acc.	89.18	89.58	91.13	91.30	88.33
Surrogate acc.	81.25	79.37	85.03	87.66	76.92

Table 2: Proposed EMGAN with a larger number of clients.

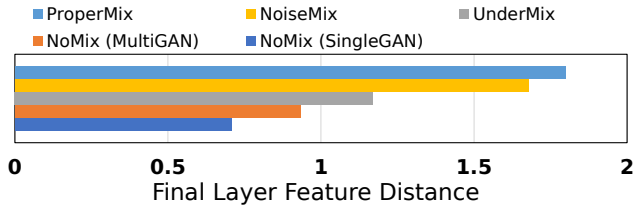


Figure 10: Feature embedding similarity between samples in mixtures obtained by using different mixing strategies.

formance. Since the more frequent model updating produces a less consistent gradient and adds to the variance. We further test this by reducing the number of epochs to 200 for the 10-client case and find that the extraction performance is 3% lower than the 400 epoch case.

Ablation Study

Attack Analysis We calculate the feature embedding similarity between samples in mixtures obtained by using different mixing strategies. The feature embedding corresponds to the final layer outputs of a sample image, and we use Euclidean distance as the similarity measure. Fig. 10 describes the corresponding findings. MultiGAN and ProperMix both help reduce the similarity between the synthetic images. The resulting higher variance contributes to a better MEA performance.

Attack Performance with More Training Data As shown in Fig. 11, for the attacker with more training data, the proposed EMGAN is still better than the Train-ME with Early Learner but the improvement is comparatively smaller. This is reasonable since the effect of augmentation diminishes when the training data is abundant.

Effect of Cut-layer Settings Previously, we had used a cut-layer setting of 6 to ensure data security. Prior works on improving resistance to MIA (Li et al. 2022) show that it is possible to have fewer layers at the client-side model to achieve the same level of data security. So we further test the accuracy and fidelity performance of the proposed method for cut-layer settings of 5, 4 and 3 (corresponding to 6, 7, 8 layers at the server-side) for 0.1% data and data-free cases. As shown in Fig. 12, We find that reducing the client-side

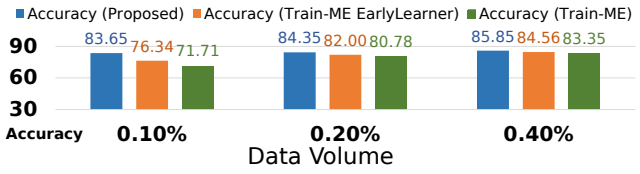


Figure 11: Proposed EMGAN for an attacker with different amounts of training data (in percentage).

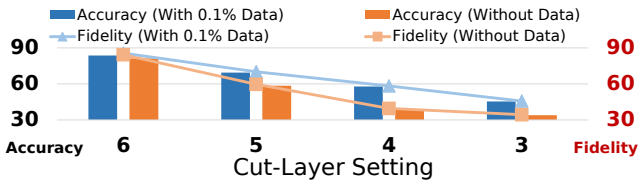


Figure 12: Proposed EMGAN for different cut-layers.

model can thwart MEA effectively. This suggests that improving resistance to MIA can also improve the resistance to MEA. Thus, future works should address the co-design of data security and IP security of SFL.

Defend against MEAs To defend against MEAs, (Li et al. 2023b) proposes L1 regularization to limit the capabilities of the client-side model, thereby reducing the potential leakage in the white-box client-side model. However, this regularization-based defense sacrifices model accuracy significantly.

We consider an alternative defense strategy based on reducing the frequency of the server sending gradients to clients which is based on the much lower training loss of the attacker compared to benign clients. Based on this, we propose loss-based gradient dispatching, where we use β times the average training loss as the threshold to send gradients back to clients. This prevents a potential attacker from getting gradients since its training loss will usually be lower than the threshold. The proposed loss-based gradient dispatching defense effectively filters out most of the gradient queries from the attacker without affecting benign clients and hence has better performance compared to the L1-regularization defense. Under the proposed defense, EMGAN with 1% data can only achieve 75% (8% less) accuracy, and EMGAN data-free achieves 38% (43% less) accuracy. However, loss-based gradient dispatching defense only works when there is a significant gap between the number of available training data of the attacker and benign clients. We find if the number of data at a benign client is also small (i.e. in a cross-device FL application), it is no longer effective as there is no clear difference in the training loss statistics.

Conclusion

In this work, we propose a new attack method, EMGAN, to effectively extract the server-side model during SFL training. EMGAN utilizes conditional-GAN with Early Learner, MultiGAN, and ProperMix methods. We show that EMGAN significantly outperforms all previous MEAs.

Ethics Statement

Why Play the Role of "Bad guys"?

Adversary research plays the role of "Red Team", which is a group of cybersecurity experts who identify new vulnerabilities to prevent actual harm. The red team is thus essential in improving security. Recently, OpenAI incorporated a dedicated red team across its products including ChatGPT [1] to work on adversarial testing of new systems. Moreover, President Biden has signed an executive order [2] specifically addressing the important role of a red team in AI, quoting: "The National Institute of Standards and Technology will set the rigorous standards for extensive red-team testing to ensure safety before public release." Attack research is widely investigated by researchers (Goodfellow, Shlens, and Szegedy 2014) to push the boundary of Safe AI through the cat-and-mouse game of attacks and defenses.

To summarize, the goal of our playing the "bad guys" is to make AI safer by identifying the vulnerabilities of SFL and motivating future works to address them. In this paper, we play the role of a red team and gently discuss about defense. Before the *Conclusion* section, we propose a basic defense based on loss statistics to detect the attacker. However, it has some limitations and we leave more effective defenses for future work.

References:

[1] OpenAI Red Teaming Network: https://openai.com/blog/red-teaming-network

[2] FACT SHEET: President Biden Issues Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence: https://www.whitehouse.gov/briefing-room/statements-releases/2023/10/30/

Potential Societal Harm

SFL is still considered an emerging technique and has not been deployed in practice. Thus, our proposed new attack will not cause any immediate harm to society. We hope that our work will prompt more defense work and also delay the wide adoption of SFL until its vulnerability gets addressed.

Limitations

The limitations of this method are weaker attack performance when the number of classes increases or when the number of layers in the server side is large. Our future work will address the attack scalability issues and more importantly develop effective defense schemes.

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 387–402. Springer.

Correia-Silva, J. R.; Berriel, R. F.; Badue, C.; de Souza, A. F.; and Oliveira-Santos, T. 2018. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Fang, G.; Mo, K.; Wang, X.; Song, J.; Bei, S.; Zhang, H.; and Song, M. 2022. Up to 100x faster data-free knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6597–6604.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333.

Goodfellow, I. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hoang, Q.; Nguyen, T. D.; Le, T.; and Phung, D. 2017. Multi-generator generative adversarial nets. *arXiv preprint arXiv:1708.02556*.

Jagielski, M.; Carlini, N.; Berthelot, D.; Kurakin, A.; and Papernot, N. 2020. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, 1345–1362.

Kang, Y.; Hauswald, J.; Gao, C.; Rovinski, A.; Mudge, T.; Mars, J.; and Tang, L. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1): 615–629.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, J.; Lyu, L.; Iso, D.; Chakrabarti, C.; and Spranger, M. 2023a. MocoSFL: enabling cross-client collaborative self-supervised learning. In *The Eleventh International Conference on Learning Representations*.

Li, J.; Rakin, A. S.; Chen, X.; He, Z.; Fan, D.; and Chakrabarti, C. 2022. ResSFL: A Resistance Transfer Framework for Defending Model Inversion Attack in Split Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10194–10202.

Li, J.; Rakin, A. S.; Chen, X.; Yang, L.; He, Z.; Fan, D.; and Chakrabarti, C. 2023b. Model Extraction Attacks on Split Federated Learning. *arXiv preprint arXiv:2303.08581*.

Liu, P.; Qi, B.; and Banerjee, S. 2018. Edgeeye: An edge service framework for real-time intelligent video analytics. In *Proceedings of the 1st international workshop on edge systems, analytics and networking*, 1–6.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4954–4963.

Singh, A.; Chopra, A.; Garza, E.; Zhang, E.; Vepakomma, P.; Sharma, V.; and Raskar, R. 2021. Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12125–12135.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2017. Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, 328–339. IEEE.

Thapa, C.; Chamikara, M. A. P.; Camtepe, S.; and Sun, L. 2020. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088*.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 601–618.

Truong, J.-B.; Maini, P.; Walls, R. J.; and Papernot, N. 2021. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4771–4780.

Vepakomma, P.; Singh, A.; Gupta, O.; and Raskar, R. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. In *2020 International Conference on Data Mining Workshops (ICDMW)*, 933–942. IEEE.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhuang, W.; Wen, Y.; and Zhang, S. 2022. Divergence-aware Federated Self-Supervised Learning. *arXiv preprint arXiv:2204.04385*.