On the Importance of Diversity in Data-Free Model Stealing

Anonymous authors
Paper under double-blind review

Abstract

Machine learning as a Service (MLaaS) allows users to query the machine learning model in an API manner, which provides an opportunity for users to enjoy the benefits brought by the high-performance model trained on valuable data. This interface boosts the flourish of machine learning based applications, while on the other hand, introduces the attack surface for model stealing attacks. Existing model stealing attacks have relaxed their attack assumptions to the data-free setting, while keeping the effectiveness. However, these methods are complex and consist of several components, which obscure the core on which the attack really depends. In this paper, we revisit the model stealing problem from a diversity perspective and demonstrate that keeping the generated data samples more diverse across all the classes is the critical point for improving the attack performance. Based on this conjecture, we provide a simplified attack framework. We empirically signify our conjecture by evaluating the effectiveness of our attack, and experimental results show that our approach is able to achieve comparable or even better performance compared with the state-of-the-art method. Furthermore, benefiting from the absence of redundant components, our method demonstrates its advantages in attack efficiency and query budget.

1 Introduction

Machine Learning (ML) models have been deployed to perform a wide range of tasks with huge success. However, targeting a well-generalized machine learning model is difficult as it requires tremendous amounts of time and money invested in both dataset collection and model training, which has become an obstacle on the road to the popularization of ML techniques. To facilitate the use of ML techniques, companies make the trained model available as a service over the web (MLaaS), where the users can obtain the predictions with paid queries. However, this poses a new threat to the confidentiality of the machine learning models, since the information comprised in the output enables the adversary to conduct malicious activities, e.g., performing model stealing attacks.

Model stealing attacks (Orekondy et al., 2019) target to extract the functionality from the victim model and train a clone model locally. The stolen model can even be leveraged for further attacks (Shokri et al., 2017; Zhang et al., 2020; Yao et al., 2019). To mount a model stealing attack, an adversary first queries the victim model to label the inputs, and then trains a clone model using these input-label pairs in a supervised manner. The quality of the queried samples has a significant impact on the performance of the clone model, and experimental results show that using random noise as input often leads to a model with unacceptable performance. Previous attack methods (Papernot et al., 2017; Orekondy et al., 2019) utilize unlabeled samples from a similar distribution to query the model, which achieves nearly perfect clone model accuracy. Unfortunately, it is hard to get access to such a dataset in practice, which limits the feasibility of the attack.

Recent attacks take a step further to relax this dataset assumption, i.e., they explore the possibility of stealing the model without the knowledge of the victim's training dataset distribution. These attacks, which are known as data-free model stealing (Truong et al., 2021; Kariyappa et al., 2021; Sanyal et al., 2022), mainly based on the idea of leveraging generative models (Goodfellow et al., 2014; Arjovsky et al., 2017; Zhang et al., 2018; Karras et al., 2019) to construct data samples that satisfy certain properties. Specifically, Kariyappa et al. (2021); Truong et al. (2021) train a generator to synthesize difficult data samples by maximizing the disagreement between the victim model and clone model; Sanyal et al. (2022) force the generator to fit the distribution of a proxy dataset. Though these attacks are effective, there indeed exists some problems. First, the overall framework of such attacks is complicated, resulting in more computational costs. Second, the required query budget is much higher than the previous attacks with surrogate dataests, as it needs the prediction from victim model for every generated image. Such drawbacks confine the attack efficiency and obscure the core property that makes the attacks work.

In this work, we refine the existing attack strategies and point out that diversity is the critical factor for model stealing. Based on this conjecture, we provide a simplified attack framework from the angle of diversity, namely diversity-based data-free model stealing (DB-DFMS). Concretely, we take advantage of the generative models and force the generator to generate various images across all the classes, and our general hypothesis is that such images contain more information which can better represent the victim model's data distribution and thus enhance the attack performance. We conduct extensive experiments on three benchmark datasets, and the evaluation results demonstrate the effectiveness of our attack, which further confirms our conjecture. Additionally, as our attack remove other redundant components, the attack exhibits economic advantages like requiring less query budget and being computationally friendly. We further conduct our attack in more generalized settings, such as the attacker has no information about the clone model's architectures, which provides additional insights into understanding the success of our attack.

2 Related Work

Model Stealing. Model stealing attack aims to extract the information from the victim model and constructs a local surrogate model. This attack was first proposed by (Tramèr et al., 2016), after that, a sequence of works have been presented in recent years. Papernot et al. (2017) conduct the attack with partial dataset that is used to train the victim model, and more data samples are generated by utilizing the jacobian of the clone model's loss function. KnockoffNets (Orekondy et al., 2019) assumes the adversary has access to a suitable surrogate dateset, while the attack efficacy is closely related to how well the surrogate dataset can represent the dataset of the victim model.

Recent studies pay more attention to the most strict data-free setting, where no data is available for the adversary. Under this scenario, Kariyappa et al. (2021) propose MAZE, which uses a generative model to generate synthetic data samples for launching the attack. The generator is trained to maximize the disagreement between the victim model and the clone model, thus the gradients from victim model is required. They adopt zeroth order gradient estimation to approximate the gradients from victim model as only black-box access is assumed here. A similar work, named as DFME, is presented by Truong et al. (2021), while the key difference is to replace the loss function from Kullback-Leibler (KL) divergence to ℓ_1 norm loss for training the student model. Sanyal et al. (2022) go a step further to train a GAN with a synthetic dataset and utilize the gradients of the clone model as a proxy to the victim model's gradients, which we refer to as DFMS-SL. In this paper, we focus on the same threat model as DFME and DFMS-SL, that is, we assume the attacker has no knowledge of the training dataset.

Knowledge Distillation. Knowledge distillation (Hinton et al., 2015) is proposed to train a small student model efficiently with the knowledge from a large teacher model. It uses softened output from the final layer of the teacher model as the label for training the student model. In real scenarios, the training data of the teacher model is not available due to confidentiality, which motivates the concept of data-free knowledge distillation. Within this setting, the student has no information for the training data but access to the teacher model. Nayak et al. (2019) exploit to obtain the prior information about the data distribution from the teacher model to craft data samples for training the student model. Most current works utilize the generative model to perform the distillation process (Choi et al., 2020; Chen et al., 2019; Micaelli & Storkey, 2019). They train the generative model with different loss objectives, targeting to synthesize data samples that are more aligned with the distribution of the teacher model. However, all these distillation techniques require gradients from the teacher model, which is the major difference compared to data-free model stealing.

Generative Models. Image generation has drew huge attentions in computer vision domain as it boosts the development of image-related industries and provide more possibilities, and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) is one of major tools. Following the first

Table 1: The relationship between the entropy of query dataset and the clone model accuracy for different model stealing attacks. The victim model is ResNet-34-8x trained on CIFAR-10 with testing accuracy as 0.930, the clone model is ResNet-18-8x.

Attack	Surrogate Datasets				Data-free	
Scenarios	CelebA	SVHN	CIFAR-100	CIFAR-10	Random Noise	DB-DFMS (ours)
Entropy (nats)	1.05	1.10	2.16	2.30	1.20	1.95
Accuracy	0.184	0.369	0.888	0.925	0.328	0.885

study in (Goodfellow et al., 2014), there are a large amount of variants have been proposed, including using auxiliary information to control the generated images (Mirza & Osindero, 2014; Perarnau et al., 2016), utilizing a substitute loss derived from Wasserstein distance to prevent from vanishing gradient problem (Arjovsky et al., 2017), and redesigning the generator architecture to generate style-specific images (Karras et al., 2020). Specifically, the training of GANs is based on a minimax game where a generator and a discriminator compete against each other, while in our work, we adopt the single generator and view the clone model as a discriminator to support generating more diverse images.

3 Diversity-based data-free model stealing (DB-DFMS)

3.1 Problem Statement

This paper we focus on the problem of model stealing attack in the data-free setting. In a nutshell, model stealing aims to train a local clone model $\mathcal{C}$ which is similar to the victim model $\mathcal{V}$. The general attack workflow is as follows: the adversary has black-box access to the victim model, they sample unlabeled data x from certain distribution. For every unlabeled data x, the adversary queries the victim model $\mathcal{V}$ to obtain the prediction $\mathcal{V}(x)$. With the prediction $\mathcal{V}(x)$ and the corresponding input x, we can use (x, $\mathcal{V}(x)$)-pairs to form the surrogate dataset, which is used to train the clone model in a supervised way.

The distribution of the queried data has a significant influence on the performance of the clone model. Orekondy et al. (2019) conduct model stealing attack by leveraging a surrogate dataset, but they fail to perform well if the surrogate dataset is not suitable to represent the distribution of the victim model. Later, Roberts et al. (2019) even consider using random noise to launch the attack, however, results show that this attack cannot be generalized to sophisticated tasks like CIFAR-10. In this paper, we consider the most challenging case where the adversary has no knowledge of the training dataset.

The problem studied in this paper has been explored in (Kariyappa et al., 2021; Truong et al., 2021; Sanyal et al., 2022), their works propose data-free model stealing that can steal the model with high accuracy. However, it is unclear what factor is the critical point that influences the quality of the clone model. In this paper, we first reveal that diversity is the key to achieve good performance. Based on this conjecture, we further propose a data-free model stealing attack which has comparable performance with lower query budget and computational costs.

3.2 Diversity is All You Need

To have a better understanding of how surrogate dataset influences the attack performance, we conduct model stealing attack on a ResNet-34-8x model trained on CIFAR-10 using different surrogate datasets. As shown in Table 1, we observe that CelebA has even worse performance compared with Random Noise, which rules out the hypothesis that more realistic images contribute more to the model performance. There is still a widely accepted speculation that the more similar the data distribution is, the better the attack performance can be. This speculation is partially proved in the table, as we can see that CIFAR-100 is the most similar one to the training dataset CIFAR-10, which also has the best accuracy among the CelebA, SVHN, and CIFAR-100 dataset.

```
Algorithm 1: DB-DFMS
Input: Query budget $\mathcal{Q}$, generator iterations $n_\mathcal{G}$, clone iterations $n_\mathcal{C}$, learning rate $\eta$.
Output: Trained $\mathcal{C}$ and $\mathcal{G}$.
while $\mathcal{Q} > 0$ do
    for $i = 1 \cdots n_\mathcal{G}$ do
        $x = \mathcal{G}(z; \theta_\mathcal{G}), \quad \text{with} \quad z \sim \mathcal{N}(0,1)$
        $\alpha_k = \frac{1}{N} \sum_{j=1}^{N} \text{softmax}_k(\mathcal{C}(x_j))$
        $\mathcal{L}_{div} = \sum_{k=1}^{K} \alpha_k \log \alpha_k$
        $\theta_\mathcal{G} = \theta_\mathcal{G} - \eta \nabla_{\theta_\mathcal{G}} \mathcal{L}_{div}$
    end
    for $i = 1 \cdots n_\mathcal{C}$ do
        $x = \mathcal{G}(z; \theta_\mathcal{G}), \quad \text{with} \quad z \sim \mathcal{N}(0,1)$
        $\mathcal{L}_{l_1} = \frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{K} |\mathcal{V}_k(x_j) - \mathcal{C}_k(x_j)|$
        $\theta_\mathcal{C} = \theta_\mathcal{C} - \eta \nabla_{\theta_\mathcal{C}} \mathcal{L}_{l_1}$
    end
    update query budget Q
end
```

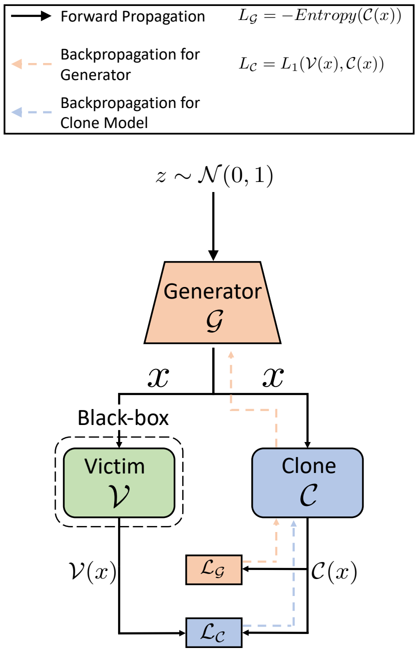

Forward Propagation
$L_\mathcal{G} = -Entropy(\mathcal{C}(x))$
Backpropagation for Generator
$L_\mathcal{C} = L_1(\mathcal{V}(x), \mathcal{C}(x))$
Backpropagation for Clone Model
$z \sim \mathcal{N}(0,1)$
Generator $\mathcal{G}$
x
x
Black-box
Victim $\mathcal{V}$
Clone $\mathcal{C}$
$\mathcal{V}(x)$
$\mathcal{L}_\mathcal{G}$
$\mathcal{C}(x)$
$\mathcal{L}_\mathcal{C}$

Figure 1: Workflow of DB-DFMS.

However, our experiments point out that this speculation is not entirely true, as the last column shows, our method could generate images (see Figure 5) that have low visual similarity to the training dataset, but leads to comparable performance as using CIFAR-100. In other words, there is something more intrinsic behind it.

In this paper, we first point out that, the diversity of the query datasets (surrogate datasets or synthetic datasets in data-free setting), defined as the entropy of the prediction probabilities from the victim model, is the key point that influences the clone model performance, no matter in model stealing attacks with surrogate datasets or in data-free setting. Concretely, we calculate the diversity of each datasets in Table 1 and the results demonstrate a clear positive correlation. Based on this observation, we design a diversity-based data-free model stealing attack in the following.

3.3 Attack Pipeline

The attack workflow is shown in Figure 1, and it can be divided into two entangled parts, i.e., the clone model training and the generator training. In the following, we illustrate each part separately, then summarize them together.

Clone Model Training. The training of the clone model follows the traditional one. Concretely, the attack starts by taking a vector of random noise z sampling from a normal distribution $\mathcal{N}(0, 1)$ as input to the generator $\mathcal{G}$ and obtains a generated image x. Then the prediction $\mathcal{V}(x)$ can be acquired by querying the victim model $\mathcal{V}$, and the same to the prediction $\mathcal{C}(x)$ from the clone model $\mathcal{C}$. The clone model is trained to minimize the disagreement between $\mathcal{V}(x)$ and $\mathcal{C}(x)$. In this paper, we adopt l_1 distance to measure the agreement, since l_1 norm loss can prevent gradient vanishing, which has an advantage over KL divergence as shown in (Truong et al., 2021). Formally, the loss for training the clone model is as follows:

$$\mathcal{L}_{l_1} = \sum_{i=1}^{K} |\mathcal{V}_i(x) - \mathcal{C}_i(x)| \tag{1}$$

where K is the number of classes. Note that l_1 norm loss requires the logits (i.e., the values before the $\mathrm{softmax}$ function), while we can only get the probability posteriors from victim model. To address this issue, we follow the method proposed in (Truong et al., 2021) to approximate the logits from the probabilities, where we first calculate the logarithm of the probability vector and then subtract the log-probabilities with its mean value.

Generator Training. Now we focus on the generator training part, which is vital as the generator determines the quality of generated samples. As we discuss in the previous section, that diversity is the most important factor that influences the performance of model stealing, we want the generator to generate highly diverse images. To achieve this goal, we use the negative entropy as the diversity loss to force the generation of more diverse images.

$$\mathcal{L}_{div} = \sum_{i=1}^{K} \alpha_i \log \alpha_i, \quad \text{with} \quad \alpha_i = \frac{1}{N} \sum_{j=1}^{N} \mathrm{softmax}_i(\mathcal{C}(x_j)) \tag{2}$$

where N is the batch size. The diversity loss is calculated with the prediction from the clone model, as the victim model can only be accessed in black-box.

Collaborative Training. As the diversity loss is calculated with the prediction from the clone model, therefore, the clone model is involved in the training of the generator. Meanwhile, the training of clone model also requires the contribution of the generator. To solve this problem, we train the generator and clone model alternatively. In order to better balance the training between the generator and clone model, for each iteration, the generator and clone model will be trained $n_{\mathcal{G}}$ and $n_{\mathcal{C}}$ times respectively.

4 Evaluation

In this section, we empirically evaluate the effectiveness of our diversity-based data-free model stealing (DB-DFMS). All experiments are performed on NVIDIA DGX A100 with Debian GNU/Linux 11[1].

4.1 Experiment Setup

Datasets. We conduct the experiments on three commonly used datasets: CIFAR-10 (CIF), SVHN and CelebA (Liu et al., 2015). Among them, CIFAR-10 and SVHN both have 10 classes and each class of CIFAR-10 has balanced number of data samples. While for CelebA, each data sample has 40 binary attributes, and we choose three of the most balanced attributes to form 8 classes. Detailed partitioning process can be found in Section 6.

Model Training. We choose ResNet-34-8x (He et al., 2016) as the victim model architecture for all three tasks. For each dataset, we train the model for 50 epochs on SVHN and 200 epochs on CIFAR-10 and CelebA. The optimizer is SGD with initial learning rate as 0.1, decayed by a cosine scheduler. We use ResNet-18-8x as the architecture of our clone model, and we explore the influence of model architectures in Section 4.4. For the generator, we adopt the one used in (Truong et al., 2021), which comprises three convolutional layers, together with linear up-sampling, batch normalization and ReLU layers. To make the output lies in the range [-1,1] (the predefined image domain), we add a hyperbolic tangent function to the last layer. The clone model and generator are trained with SGD at 0.1 initial learning rate and Adam at 10^{-4} initial learning rate respectively, and both have an batch size of 256 and an scheduler that multiplies the learning rate with 0.3 at 10%, 30% and 50% of the total training epochs.

Attack Settings. We choose 2M query budget for SVHN and 20M for CIFAR-10 and CelebA to launch the attack. And for training iterations of generator and clone model $n_{\mathcal{G}}$ and $n_{\mathcal{C}}$, we have tried different ratios, the general find is setting $n_{\mathcal{C}}$ a little bit higher than $n_{\mathcal{G}}$ can make the generator trained smoothly and let the clone model see enough diverse data samples at the same time, thus we choose 1 and 5 as a suitable pairs.

Evaluation Metrics and Baselines. We choose accuracy and the agreement to measure the quality of the clone model as they can directly demonstrate the similarity between the victim model and the clone model. The training time of the clone model is also considered to reflect the attack efficiency. We compare our attack to two state-of-the-art methods, which we refer to as DFME (Truong et al., 2021) and DFMS-SL (with synthetic dataset) (Sanyal et al., 2022), and view the attack with random noise as a baseline.

[1] Our code is available at `https://anonymous.4open.science/r/DB-DFMS-72B4/`.

Table 2: Performance of data-free model stealing against ResNet-34-8x trained on CIFAR-10, SVHN and CelebA, "Acc" and "Agr" represent for clone model accuracy and agreement between victim model and clone model respectively. The clone model is ResNet-18-8x.

Datasets	Victim	Random Noise		DFME		DFMS-SL		DB-DFMS (Ours)	
(budget)	accuracy	Acc	Agr	Acc	Agr	Acc	Agr	Acc	Agr
CIFAR-10 (20M)	0.930	0.328	0.314	0.869	0.893	0.896	0.926	0.885	0.921
SVHN (2M)	0.962	0.808	0.814	0.952	0.971	0.955	0.977	0.955	0.975
CelebA (20M)	0.769	0.706	0.759	0.750	0.865	0.743	0.813	0.746	0.853

Table 3: Train time (s) of data-free model stealing against ResNet-34-8x trained on CIFAR-10, SVHN and CelebA. The Clone model is ResNet-18-8x.

Datasets (budget)	Random Noise	DFME	DFMS-SL	DB-DFMS (Ours)
CIFAR-10 (20M)	2749	3825	> 10000	3596
SVHN (2M)	323	380	> 1000	374
CelebA (20M)	8322	11759	> 30000	11147

4.2 Effectiveness of DB-DFMS

We conduct our experiments on three benchmark datasets and compare the attack performance of different data-free model stealing in Table 2. According to the clone model accuracy and the agreemnet between the victim model and clone model, our attack can obtain a comparable attack result as the other state-of-the-art methods on all the three datasets. These results demonstrate that with diversity loss only, the generated images are capable of contributing to a well-performed attack.

We further measure the computational costs consumed in the attack process with the training time, which is exhibited in Table 3. For Random Noise, it has the least training time as it doesn't need to train the generator, however, its attack performance is bad as showed in Table 2. Among the rest methods, our attack saves the most computational time. Here the reason why DFMS-SL requires such high computation is it uses the proxy dataset to initialize the generator and clone model with hundreds of epochs. However, when taking both the clone model performance and the training time into consideration, it is unnecessary to include such a proxy dataset in the attack process as it benefits so less and even has worse attack result in some datasets like CelebA.

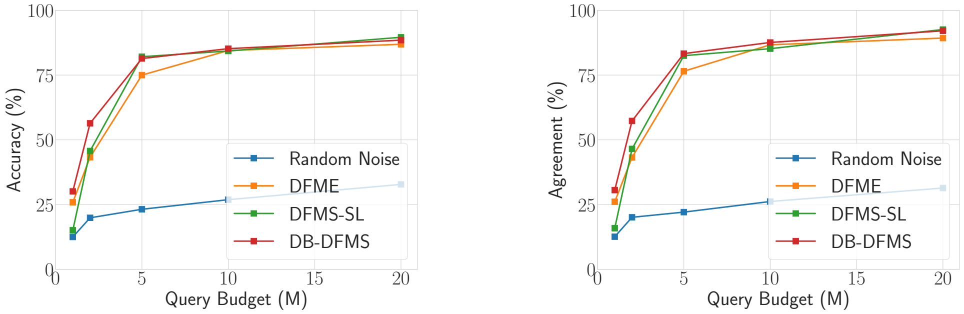
4.3 Influence of the Query Budget

In previous experiments, we set the query budget to 20M for CIFAR-10, which is a common setting in previous works (Kariyappa et al., 2021; Truong et al., 2021). In this section, we aim to explore how the query budget influences the attack performance. A thorough understanding of the influence of query budget is beneficial as a lower query budget requires less computational power and lower money payment to the MLaaS platform.

The results are given in Figure 2. As we can see, all the attack performance has the positive correlation to the query budget, both holding for the clone model accuracy and the agreement between the victim model and the clone model, while the margin increase above 10M is not that large. However, our attack still consistently performs well and even shows the superiority over other methods in some cases of query budget.

4.4 Influence of the Clone Model Architecture

As investigated in the previous work about knowledge distillation (Cho & Hariharan, 2019; Micaelli & Storkey, 2019), a smaller student model is sufficient to distill the knowledge from the teacher model, as long as it has enough capability. Thus we choose Resnet-18-8x as the clone model though the victim model is ResNet-34-8x. However, we are still interested in whether the clone model performance will be improved with higher capability. Except Resnet-18-8x and ResNet-34-8x, we test other 4 commonly used model architectures, including MobileNetV2 (Sandler et al., 2018),



(a) Accuracy

(b) Agreement

Figure 2: The attack performance of DB-DFMS against ResNet-34-8x trained on CIFAR-10 with different query budget. The clone model is ResNet-18-8x.

Table 4: Data-free model stealing with different clone model architectures against ResNet-34-8x trained on CIFAR-10, “Acc” and “Agr” represent for clone model accuracy and agreement between victim model and clone model respectively.

Architectures	Victim	Random Noise		DFME		DFMS-SL		DB-DFMS (Ours)	
(parameters)	accuracy	Acc	Agr	Acc	Agr	Acc	Agr	Acc	Agr
MobileNetV2 (2.3M)	0.930	0.264	0.252	0.819	0.815	0.870	0.894	0.853	0.881
DenseNet-121 (7.0M)	0.930	0.309	0.311	0.863	0.876	0.885	0.898	0.875	0.892
WideResNet-32 (7.4M)	0.930	0.245	0.239	0.777	0.770	0.832	0.825	0.829	0.835
ResNet-18-8x (11.2M)	0.930	0.328	0.314	0.869	0.893	0.896	0.926	0.885	0.921
ResNet-34-8x (21.3M)	0.930	0.308	0.297	0.883	0.900	0.905	0.929	0.891	0.922
VGG-16BN (134.3M)	0.930	0.191	0.194	0.699	0.677	0.793	0.770	0.789	0.772

DenseNet-121 (Huang et al., 2017), WideResNet-32 (Zagoruyko & Komodakis, 2016) and VGG-16BN (Simonyan & Zisserman, 2015).

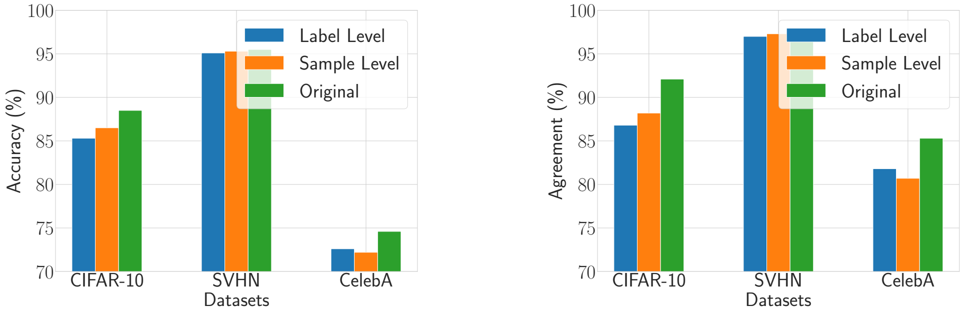
In general. the attack performance is increased as the clone model has more number of parameters (See Table 4). However, there are two exceptions. The performance of DenseNet-121 is awesome regarding its capability, which can be explained by its specific designing. That is, any two layers in the model are connected together to strengthen feature propagation. The other case is VGG-16BN, though it has hundreds of millions of parameters, it cannot obtain a satisfying attack result, which may due to the more obvious differences in architectures to the other models. Compared to DenseNet-121, WideResNet-32 is more similar to the victim model ResNet-34-8x, however, the gap between the performance of these two models rules out that more similar network achieves higher attack performance.

4.5 Influence of the Diversity Loss

The core idea of our attack is to train a generator by simply using a diversity loss, and the definition of diversity could be interpreted in different ways. In this section, we form two different types of diversity loss and evaluate their impact on the attack performance.

Sample Level. As showed in Equation 2, after the $\mathrm{softmax}$ function, the original loss first calculates the mean value over the batch of data samples, and then gets the negative entropy. Our first thought is to change the order of these two calculations and compute the entropy over each data sample first then average the entropy on all samples, thus we call it “Sample Level” diversity loss:

$$\mathcal{L}_{sl\_div} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\alpha_{ij}\log\alpha_{ij}, \quad \text{with} \quad \alpha_{ij} = \mathrm{softmax}_j(\mathcal{C}(x_i)) \tag{3}$$



(a) Accuracy

(b) Agreement

Figure 3: The attack performance of DB-DFMS with different diversity loss against ResNet-34-8x trained on CIFAR-10. The clone model is ResNet-18-8x.

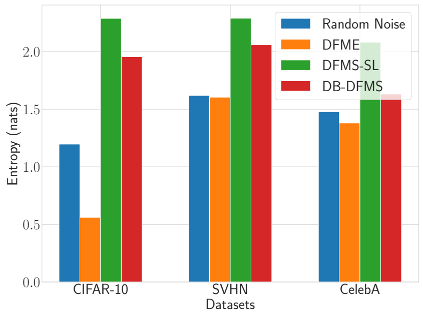


Figure 4: Entropy of generated data samples with different data-free model stealing methods according to the prediction from victim model. The victim model is ResNet-34-8x and the clone model is ResNet-18-8x.

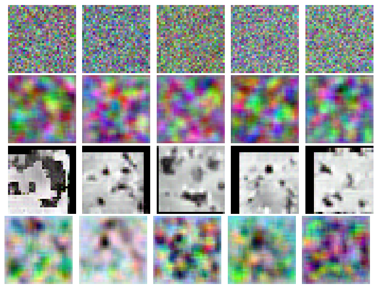


Figure 5: Data Samples generated by Random Noise, DFME, DFMS-SL and DB-DFMS (from top to bottom). The victim model is ResNet-34-8x trained on CIFAR-10 and the clone model is ResNet-18-8x.

Label Level. As the goal of diversity loss is to help the generation of more diverse data samples across all the classes, thus we try to calculate the diversity in a more direct way, i.e., by utilizing the hard labels:

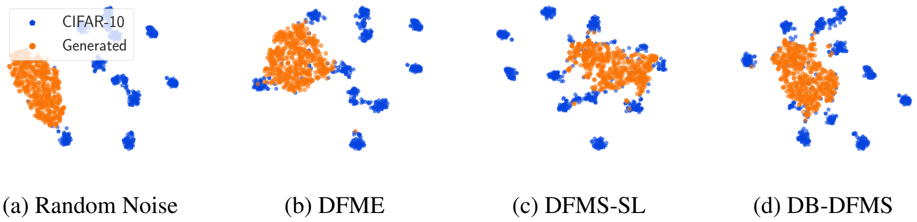
$$\mathcal{L}_{hl\_div} = \sum_{i}^{K} \alpha_i \log \alpha_i, \quad \text{with} \quad \alpha_i = \frac{1}{N} \sum_{j=1}^{N} \mathcal{F}_{one\_hot}(\arg\max(\text{softmax}(\mathcal{C}(x_j))), K)_i \tag{4}$$

where the $\arg\max$ is to obtain the class index with the highest prediction probability, and $\mathcal{F}_{one\_hot}$ is a function to form a one-hot vector according to the class index and the total number of classes. The difference between this loss and the original one is this loss does not take sample-wise diversity into consideration, but only tries to generate more samples belonging to different classes.

We put the results in Figure 3, results show that all three losses achieve good attack performance. And we find that the best is still the one with the original diversity loss for all the three datasets and the two evaluation metrics. Its advantage is derived from more information it utilizes as it considers all data samples in a mini-batch together and uses posteriors from the clone model. We leave the finding of more elegant diversity loss as a future work.

5 Exploration

Our simplified attack has comparable clone model performance and needs less training time, here we provide deep insights to show why diversity of the generated data samples is the key point for enhancing the attack.



(a) Random Noise (b) DFME (c) DFMS-SL (d) DB-DFMS

Figure 6: t-SNE representations for the embedding of 512 randomly generated data samples with different data-free model stealing methods. The Victim model is ResNet-34-8x trained on CIFAR-10 and the clone model is ResNet-18-8x.

5.1 Entropy of Generated Dataset

As showed in Table 1, the attack performance has a positive correlation to the diversity of the generated images, and here we further prove this finding. Figure 4 reports that DFMS-SL and our DB-DFMS have impressive clone model performance as both of them enable the generation of high entropy data samples. We also admit that diversity of the query datasets is not the only factor that influences the attack results. We can find DFME also performs well though the entropy of its generated dataset is comparatively low, which means the output of more difficult query samples from the victim model can lead to clone model with high performance as well.

5.2 Visualization of Generated Dataset

We then visualize the generated data samples from different perspectives to see the differences from each attack. The original generated images are showed in Figure 5. None of them are close to real images, but we can still find the differences between them. For Random Noise, each pixel is generated randomly, which means the neighboring pixels do not have correlation neither, thus resulting in a grainy image. While the rest images generated from other methods are comparably more smoothed.

We further take the generated data samples as the input to the victim model and visualize the embeddings output from the penultimate layer by using t-Distributed Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008), which is depicted in Figure 6. The general trend is the better the distribution of generated datasets matches the distribution of the victim datasets, the greater the attack performance is. Specifically, “Random Noise” seems outliers to the dataset used in the victim model, while our attack is able to produce data samples more close to the victim’s distribution. Though these two distributions cannot match to each other ideally, the generated distribution is already enough to extract the information for achieving impressive attack performance.

6 Conclusion

In this paper, we revisit the generator-based data-free model stealing attack from a diversity perspective, and investigate the possibility of simplifying the existing approaches. We find that the diversity of the generated data samples used for querying the victim model is one of the key points related to the attack performance. We conduct extensive experiments to show that simply using a diversity loss to train a generator can force the generation of data samples across all the classes and enable the attack to achieve comparable results as the state-of-the-art methods while with much less computational costs. We further conduct our attack in more realistic scenarios, e.g., the query budget is limited, or the target model architecture is not available. Results evince our attack consistently performs well, which demonstrates the practicality of our attack. Moreover, entropy and visualization of the generated data samples are provided for explaining the success of the our attack.

References

https://www.cs.toronto.edu/~kriz/cifar.html.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*, pp. 214–223. PMLR, 2017.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-Free Learning of Student Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3513–3521. IEEE, 2019.

Jang Hyun Cho and Bharath Hariharan. On the Efficacy of Knowledge Distillation. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

Yoojin Choi, Jihwan P. Choi, Mostafa El-Khamy, and Jungwon Lee. Data-Free Network Quantization With Adversarial Knowledge Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3047–3057. IEEE, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2672–2680. NIPS, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, 2016.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *CoRR abs/1503.02531*, 2015.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. IEEE, 2017.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K. Qureshi. MAZE: Data-Free Model Stealing Attack Using Zeroth-Order Gradient Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13814–13823. IEEE, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410. IEEE, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116. IEEE, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738. IEEE, 2015.

Paul Micaelli and Amos J. Storkey. Zero-shot Knowledge Transfer via Adversarial Belief Matching. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 9547–9557. NeurIPS, 2019.

Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *CoRR abs/1411.1784*, 2014.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-Shot Knowledge Distillation in Deep Networks. In *International Conference on Machine Learning (ICML)*, pp. 4743–4751. PMLR, 2019.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff Nets: Stealing Functionality of Black-Box Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4954–4963. IEEE, 2019.

Nicolas Papernot, Patrick D. McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks Against Machine Learning. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, pp. 506–519. ACM, 2017.

Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Alvarez. Invertible Conditional GANs for Image Editing. *CoRR abs/1611.06355*, 2016.

Nicholas Roberts, Vinay Uday Prabhu, and Matthew McAteer. Model Weight Theft With Just Noise Inputs: The Curious Case of the Petulant Attacker. *CoRR abs/1912.08987*, 2019.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520. IEEE, 2018.

Sunandini Sanyal, Sravanti Addepalli, and R. Venkatesh Babu. Towards Data-Free Model Stealing in a Hard Label Setting. *CoRR abs/2204.11022*, 2022.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 3–18. IEEE, 2017.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *USENIX Security Symposium (USENIX Security)*, pp. 601–618. USENIX, 2016.

Jean-Baptiste Truong, Pratyush Maini, Robert J. Walls, and Nicolas Papernot. Data-Free Model Extraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4771–4780. IEEE, 2021.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent Backdoor Attacks on Deep Neural Networks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2041–2055. ACM, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2016.

Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. *CoRR abs/1805.08318*, 2018.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 250–258. IEEE, 2020.

A DATASETS DESCRIPTION

CIFAR-10. CIFAR-10 is a benchmark dataset for image classification task. It has 10 classes where each class has 5000 and 1000 data samples for training and testing respectively. The size of each image is 32×32×3.

SVHN. SVHN is a image dataset for digits in real scenarios, which has 10 classes for numbers from "0" to "9". There are in total 73257 data samples for training and 26032 for testing. It also consists of additional 531131 difficult images as extra dataset, while in our experiments we don't consider it.

CelebA. CelebA is a large-scale dataset for face recognition. It contains 202599 number of images and each of them has 40 binary attributes. We choose "Male", "Mouth_Slightly_Open" and "Smiling" as the target attributes, and it splits the whole dataset into 8 classes and each class at least has 8561 number of images. Thus we randomly select 8000 images from each class to form a balanced dataset, and use 60000 and 4000 among for training and testing respectively. We resize each image to 64×64 as they don't have a fixed size.

B HYPER-PARAMETERS TUNING

Training Times between Generator and Clone Model. In our experiments, we set $n_\mathcal{G}$ and $n_\mathcal{C}$ to balance the training between the generator and clone model. Here we show the effect of the ratio between these two hyper-parameters on the attack performance in Table 5.

Table 5: The attack performance of DB-DFMS against ResNet-34-8x trained on CIFAR-10 with different ratio of $n_\mathcal{G}$ and $n_\mathcal{C}$. The accuracy of victim model is 0.930, and the clone model is ResNet-18-8x.

$n_\mathcal{G} : n_\mathcal{C}$	1:1	1:2	1:3	1:4	1:5	1:6	1:7	1:8	1:9	1:10
Accuracy	0.829	0.860	0.880	0.881	0.885	0.886	0.887	0.885	0.880	0.875
Agreement	0.841	0.884	0.913	0.914	0.921	0.915	0.918	0.917	0.907	0.897

Clone Model Loss Functions. As recommended in (Truong et al., 2021), we adopt l_1 norm loss for training the clone model. Here we consider different loss functions and see the impact on the attack performance.

Table 6: The attack performance of DB-DFMS against ResNet-34-8x trained on CIFAR-10 with different loss functions for clone model. The accuracy of victim model is 0.930, and the clone model is ResNet-18-8x. "Acc" and "Agr" represent clone model accuracy and agreement between victim model and clone model respectively.

Datasets	KL Divergence		L_2 Loss		L_1 Loss	
(budget)	Acc	Agr	Acc	Agr	Acc	Agr
CIFAR-10 (20M)	0.758	0.783	0.851	0.880	0.885	0.921

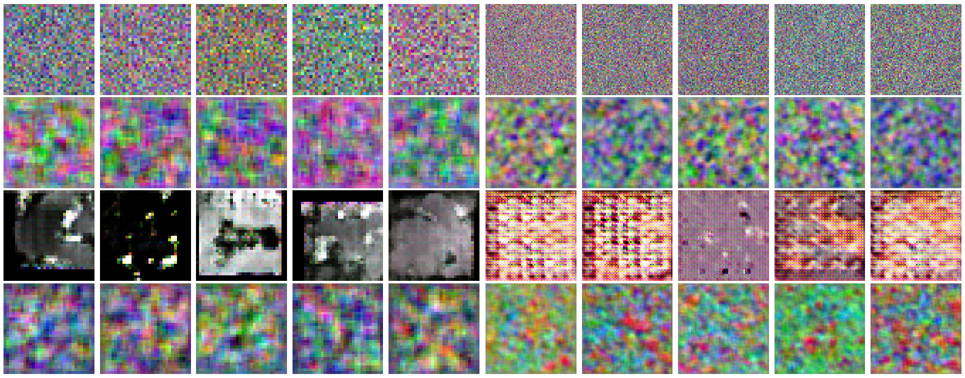
Batch Size. The core idea of our attack is to train a generator by leveraging a diversity loss, and such a diversity loss is calculated as the negative entropy of the predictions from a mini-batch. Thus we consider the effect of batch size as it will influence the information used in the calculated loss.

Table 7: The attack performance of DB-DFMS against ResNet-34-8x trained on CIFAR-10 with different batch size. The accuracy of victim model is 0.930, and the clone model is ResNet-18-8x.

Batch Size	16	32	64	128	200	256	300	400	512	1024
Accuracy	0.790	0.849	0.874	0.875	0.890	0.885	0.887	0.867	0.860	0.829
Agreement	0.815	0.876	0.904	0.909	0.928	0.921	0.923	0.895	0.888	0.849

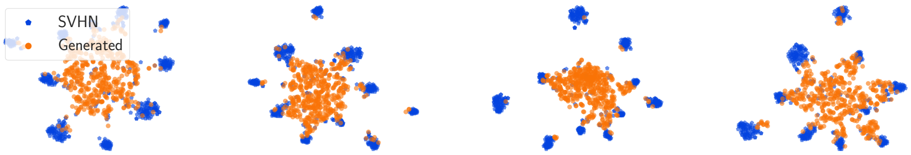
C Additional Visualization Results

Here we provide the visualization for SVHN and CelebA, and the results follow the patterns we claim in Section 5.2. For SVHN, all methods perform well as it is a simple task, thus the embedding of the generated data samples is more aligned to what from the victim model training data, including "Random Noise", as showed in Figure 8.



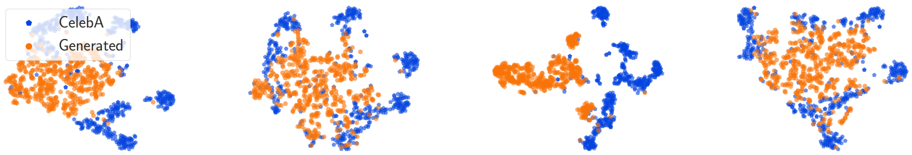
(a) SVHN (b) CelebA

Figure 7: Data samples generated by different data-free model stealing methods. The victim model is ResNet-34-8x trained on SVHN and CelebA and the clone model is ResNet-18-8x. The methods from top to bottom are Random Noise, DFME, DFMS-SL and DB-DFMS.



(a) Random Noise (b) DFME (c) DFMS-SL (d) DB-DFMS

Figure 8: t-SNE representations for the embedding of 512 randomly generated data samples with different data-free model stealing methods. The Victim model is ResNet-34-8x trained on SVHN and the clone model is ResNet-18-8x.



(a) Random Noise (b) DFME (c) DFMS-SL (d) DB-DFMS

Figure 9: t-SNE representations for the embedding of 512 randomly generated data samples with different data-free model stealing methods. The Victim model is ResNet-34-8x trained on CelebA and the clone model is ResNet-18-8x.