DisGUIDE: Disagreement-Guided Data-Free Model Extraction

Jonathan Rosenthal[1], **Eric Enouen**[2*], **Hung Viet Pham**[3†], **Lin Tan**[1]

[1] Purdue University [2] The Ohio State University [3] York University
rosenth0@purdue.edu, enouen.9@osu.edu, hvpham@yorku.ca, lintan@purdue.edu

Abstract

Recent model-extraction attacks on Machine Learning as a Service (MLaaS) systems have moved towards data-free approaches, showing the feasibility of stealing models trained with difficult-to-access data. However, these attacks are ineffective or limited due to the low accuracy of extracted models and the high number of queries to the models under attack. The high query cost makes such techniques infeasible for online MLaaS systems that charge per query.

We create a novel approach to get higher accuracy and query efficiency than prior data-free model extraction techniques. Specifically, we introduce a novel generator training scheme that maximizes the disagreement loss between two clone models that attempt to copy the model under attack. This loss, combined with diversity loss and experience replay, enables the generator to produce better instances to train the clone models. Our evaluation on popular datasets CIFAR-10 and CIFAR-100 shows that our approach improves the final model accuracy by up to 3.42% and 18.48% respectively. The average number of queries required to achieve the accuracy of the prior state of the art is reduced by up to 64.95%. We hope this will promote future work on feasible data-free model extraction and defenses against such attacks.

1 Introduction

Many deployed machine learning models are accessible via a pay-per-query system (Tramèr et al. 2016). It is profitable for an adversary to steal these models for either theft or reconnaissance (Jagielski et al. 2020). For theft, the goal is to avoid continued payment to the model owner by copying the original model under attack, also known as the *victim model*, into another model, referred to as a *clone model*. For reconnaissance, the goal is to set up a base for further attacks on a proprietary model or data. White-box attacks apply to the clone model as a proxy to uncover the training data or parameters of the victim model (Papernot et al. 2017; Shumailov et al. 2021). Recent techniques propose model extraction attacks, which attempt to copy victim model functionality into a clone model via black-box queries to the victim (Tramèr et al. 2016). These techniques expose possible attacks so one can deploy defenses to protect the models.

Traditional model extraction techniques require some related data to query the victim model (Orekondy, Schiele, and Fritz 2019). Recent techniques (Truong et al. 2021; Kariyappa, Prakash, and Qureshi 2021) introduce *data-free model extraction*, where no related training data is required to steal models. Data-free approaches are more general and practical because valuable models are often trained on private data to which attackers have no access. Otherwise, attackers could train their own model if the data was publicly available. Data-free approaches address this challenge by creating synthetic samples to query the victim model.

Data-Free Model Extraction (Truong et al. 2021), henceforth DFME, and MAZE (Kariyappa, Prakash, and Qureshi 2021) adapt techniques used in knowledge distillation (Fang et al. 2019; Micaelli and Storkey 2019) to generate synthetic data to train clone models for model extraction. They train a generator to learn what samples maximize the difference between the victim and the clone to create better samples to learn from and use approximation techniques to estimate the victim model's hidden gradient. This gradient is then passed to the generator to update its weights, so it will create better samples for the clone model to copy the victim model. These techniques are used in the *soft-label setting*, where the victim model returns the confidence values of all class labels (i.e., softmax outputs).

The recent paper Data-Free Model Stealing (Sanyal, Addepalli, and Babu 2022), henceforth DFMS, extends data-free model extraction to the more practical and complex *hard-label* setting, where the victim returns only the predicted class label instead of confidence values of all labels. This paper achieves state-of-the-art performance in both the hard-label and soft-label settings. They introduce a novel GAN framework combined with pretraining on a generated synthetic dataset to improve the clone's final accuracy.

Despite the promises of data-free model extraction, existing approaches suffer from poor model accuracy. For example, on the CIFAR-100 (Krizhevsky, Hinton et al. 2009) dataset with 100 classes, the clone models extracted by existing approaches only achieve an accuracy of 43.56% (Sanyal, Addepalli, and Babu 2022) (Section 5.1).

In addition, these data-free model extraction techniques require a large query budget because the generated sam-

[*] The work was done when Eric was at Purdue University.

[†] The work was done when Hung was at the University of Waterloo.

ples do not provide as much learning signal to the clone model compared to the actual training data. For example, MAZE, DFME, and DFMS report their soft-label accuracies on CIFAR-10 (Krizhevsky, Hinton et al. 2009) after *20 million queries*. Querying the victim model 20 million times could cost up to $100,000 (Tramèr et al. 2016), which defeats the purpose of model stealing.

To get better accuracy and query efficiency than current model-extraction approaches, we propose a new method—*Disagreement-Guided Data-Free Model Extraction (DisGUIDE*[1] *)*, utilizing a novel generator training scheme. Specifically, we simultaneously train *two clone models* with the same samples and introduce a *disagreement loss* between the two clone models to force the generator to create samples where one or both of the clone models are wrong. If the clones disagree with each other, both cannot match the victim's prediction, so at least one of the clone models can learn to better match the victim from the sample produced. In addition, we add the class diversity loss (Sanyal, Addepalli, and Babu 2022) on top of our disagreement loss to further promote the generator to create samples from all classes. We also add an experience replay (Kariyappa, Prakash, and Qureshi 2021) to reuse samples to improve the effectiveness of generated samples. By using two clones with this new joint-loss and experience replay, *our approach achieves a higher final accuracy of the extracted models with better query efficiency*.

This paper makes the following main contributions.

- We propose a new generator training approach that uses two clone models trained from identical samples.
- We introduce a novel disagreement loss for generator training, which enables data-free model extraction to generate and learn from better instances on which the clone models are likely to disagree with the victim.
- We create a new data-free model extraction approach (DisGUIDE), which combines the new generator training, the disagreement loss with a diversity loss, and experience replay.
- Our evaluation shows that compared to state-of-the-art, DisGUIDE improves the final model accuracy by 2.78% and 3.42% on CIFAR-10 in the soft-label and hard-label settings, respectively. On the more challenging CIFAR-100 dataset, DisGUIDE improves the accuracy in the hard-label setting by 18.48%. In addition, DisGUIDE reduces the number of queries required to match the accuracy of existing techniques by 64.95% and 49.75% on CIFAR-10 soft- and hard-label and 61.30% on CIFAR-100 for hard-label.

2 Related Work

2.1 Data Free Model Extraction

Model extraction (Tramèr et al. 2016) is the task of stealing a model's functionality or other value-able information such as the architecture and or learned parameters (Oliynyk, Mayer, and Rauber 2022). Some model extraction techniques assume the victim model returns the confidence value for a given input instance (Tramèr et al. 2016). Two methods build upon the techniques introduced in the knowledge distillation domain by utilizing a generator to create training samples for model extraction (Truong et al. 2021; Kariyappa, Prakash, and Qureshi 2021). The key contribution of these papers is that they estimate the gradients from the victim model using black-box gradient estimation methods (Ghadimi and Lan 2013; Duchi et al. 2012). However, DFME and MAZE both require a large number of queries (20M) to reach acceptable performance. DisGUIDE aims to reduce this number of queries.

Unlike these two methods, DisGUIDE requires no queries to the victim model to train the generator. DisGUIDE is the first model extraction technique to train two different clone models with identical training samples and the first to utilize the standard deviation in the generator loss.

Hard-Label Setting: A simple defense against these model extraction techniques is only returning the top-1 prediction for each query. This results in no change in prediction accuracy for an MLaaS provider while preventing attack methods relying on small changes in confidence values for close input samples. To overcome this, DFMS proposed a method where the generator training is entirely independent of the victim. Instead, the generator is trained with the help of an additional discriminator model to output data that is similar to some proxy distribution. The generator is further trained to maximize the distribution of labels, as judged by the clone model. Contrary to DFMS, DisGUIDE requires no proxy data and is completely data-free. In addition, we rely on a novel loss and second clone model as opposed to a discriminator model.

Model Extraction for Specific Scenarios: A few methods have been proposed for model extraction attacks under specific circumstances. In the explainable-AI setting, an attacker may use gradients from the explanation in order to train a model with high accuracy (Miura, Hasegawa, and Shibahara 2021). Another recent work (Li 2021) proposed utilizing side channel information, should it be available. Finally, tabular data has been explicitly targeted with the help of publicly known statistics (Tasumi et al. 2021).

2.2 Knowledge Distillation

Knowledge distillation (KD) is the task of distilling the functionality of a larger model or ensemble of models into a smaller one (Hinton et al. 2015). It assumes white-box knowledge of the teacher model, while model extraction treats the victim model as a black-box to which the clone model has no access (Tramèr et al. 2016). Many techniques have been developed for performing KD from a teacher model with high accuracy using the dataset it was trained on (Gou et al. 2021). However, the original training data is not always available, so others have introduced data-free KD techniques. Some of these techniques use the metadata or intrinsic information from the teacher to create synthetic samples to train on (Nayak et al. 2019; Lopes, Fenu, and Starner 2017; Yin et al. 2020; Mopuri, Uppala, and Babu

[1] DisGUIDE codebase: https://github.com/lin-tan/disguide

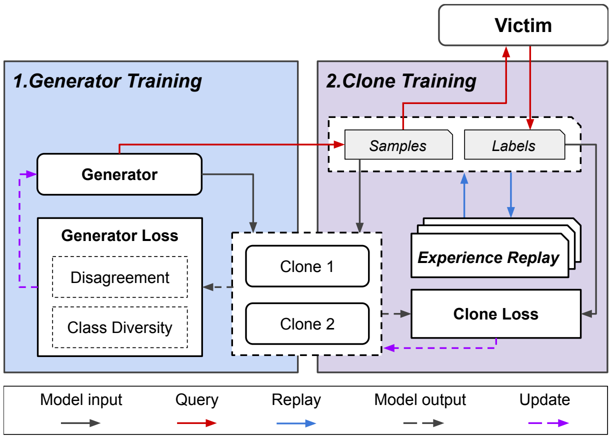

Figure 1: Complete overview of DisGUIDE training

2018). Other techniques utilize a GAN for training (Micaelli and Storkey 2019; Fang et al. 2019; Chen et al. 2019).

A few recent KD papers have methods relying on an ensemble of student networks. (Walawalkar, Shen, and Savvides 2020) proposed a method for online model distillation, where an ensemble of students learned from an ensemble of teacher models. (Chang et al. 2022) distilled a single teacher model into multiple students based on differing views for the purpose of sentiment classification.

3 Approach

Figure 1 presents an overview of our new data-free model extraction approach DisGUIDE, which consists of two phases, (1) *Generator Training* and (2) *Clone Training*. Given a black-box *Victim* model, an iterative process between the two phases produces two clone models, i.e., *Clone 1* and *Clone 2*, in the middle. Our final extracted model could be clone 1, clone 2, or some combination of these two models. In this paper, without loss of generality, DisGUIDE's output model is the ensemble of the pair of clone models, where the model output is an equal weight soft-vote (i.e., element-wise sum of the softmax output) of the two clones.

The generator training uses the joint loss (the *Generator Loss* box) from the disagreement and class diversity losses to update the generator model to create better *Samples* (e.g., images for an image classification model) to query the victim model. These samples improve DisGUIDE's query efficiency and better train the two clone models for higher accuracy. Then the clone training uses samples created by the updated generator to query the victim model. It then uses the samples and the victim's output (e.g., *Labels*) along with an *Experience Replay* (Kariyappa, Prakash, and Qureshi 2021) (Section 3.2) to simultaneously train both clone models to mimic the victim's predictions. Experience replay reuses samples to further train the clones without incurring new queries to the victim. The red arrows in Figure 1 show this query flow, while the blue arrows show the experience replay flow. The clone training uses *Clone Loss* to update the clone models to improve the clones' prediction accuracy. These updated clones are used in the next step of generator training. The process stops when the query budget is exhausted.

Intuition and Novelty: Different from existing model-extraction techniques, which only use one clone model, DisGUIDE uses two clone models. This design enables us to create a new *Disagreement* Generator Loss (Section 3.1) by leveraging differences between a pair of clone models (clone 1 and clone 2) to generate better samples, i.e., samples on which the two clone models disagree. As explained in the Introduction, these samples are also samples on which at least one clone model disagrees with the victim model, as it is impossible for both clones to match the victim while disagreeing with each other. Training the clone models with such samples should help the clones to copy the victim. We combine this loss with a *Class Diversity* loss (Sanyal, Addepalli, and Babu 2022) to further improve the samples produced by the generator. Since the two clone models are trained using the exact same training samples, using two models does not increase the query budget, and we instead leverage the variance of model training given the same training samples (Pham et al. 2020; Qian et al. 2021). The variance comes from both algorithms (such as random seeds) and software implementations (such as parallelism and floating point imprecision).

3.1 Generator Training

DisGUIDE's *generator* is a generative deep learning model that takes in random noise and produces new data samples from a distribution. The generator loss (a joint disagreement and class diversity loss) guides the generator to produce samples that induce disagreement between the clone models and samples belonging to every class equally. Training with these samples helps the clone models match the victim model quicker. Specifically, in the forward pass, a batch of random noise vectors (drawn from the latent space) is fed into the generator to create new data samples. The prediction output of the clone models is collected using these generated samples. The generator's joint disagreement and class diversity loss are then computed and used to update the generator's weights in the backward pass.

Generator Training Loss: DisGUIDE utilizes a joint loss of the disagreement loss and class diversity loss during generator training to guide the generator to create query samples that induce disagreement between clone models and class diversity in their predictions. This, in turn, boosts the clone training phase's effectiveness and efficiency, enabling DisGUIDE to extract models with state-of-the-art accuracy while requiring a smaller query budget. Equation 1 shows our joint loss function, which is the sum of our novel disagreement loss L_D and the class diversity loss L_{div}, weighted by λ.

$$L_G = L_D + \lambda L_{div} \tag{1}$$

Disagreement Generator Loss: We propose a novel generator loss, the disagreement loss, that computes the difference between the predictions of a pair of clone models. This loss can guide the generator to create query samples that induce more significant differences in confidence values of the clone models. Since the loss is computed based on the clone models, no additional query budget is spent on the generator training. Intuitively, when the two clones disagree on a sample, two cases can happen: 1) either one of the clones matches the victim, and the sample will help the other clone also learn on that sample, 2) or both of the clones fail to match the victim (when both clones generate different wrong predictions), and they can both learn from the sample. Our proposed disagreement loss estimates this difference mathematically by using the standard deviation of the clones outputs for each respective class and sample. In this work, we focus on the two clone case, where the standard deviation could be equivalently replaced by any l_n norm. We formulate and implement the loss so it can be used for more than two clones in Equation 2.

$$L_D = -\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\left[\mathrm{S}\left(A_{1ik}, A_{2ik}, \ldots, A_{cik}\right)\right] \qquad (2)$$

where S represents the standard deviation across clones. A_{cik} is the confidence value (i.e. softmax activation), that clone $c \leq C$ produces, of sample instance $i \leq N$ belonging to class $k \leq K$. In this paper, C is two for two clones.

Class Diversity Loss: Following a prior paper (Sanyal, Addepalli, and Babu 2022), DisGUIDE also uses the class diversity component in the loss to guide the generator to create samples from a diverse set of classes. This was first introduced in (Addepalli et al. 2020), where they used the loss to prevent generated samples from being biased towards any one class. Ideally, the goal of this class diversity loss is to promote a diversity of classes predicted by the victim model. However, this is not possible due to the victim's black-box nature. For this reason, the clone model is used as a proxy for the class diversity loss. The diversity loss may require a few training iterations to become meaningful. Equation 3 shows the definition of the diversity loss.

$$L_{div} = \sum_{k=1}^{K}\omega_k log(\omega_k) \qquad (3)$$

where ω_k is the kth component of the mean confidence value defined in Equation 4 with A_{cik} explained above.

$$\omega_k = \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}(A_{cik}) \qquad (4)$$

3.2 Clone Training

DisGUIDE trains the clone models iteratively alongside the generator in the extraction process. The clones are initialized with different random weights sampled from an identical distribution. This difference in initialization is the only source of randomness between the two models, aside from randomness introduced by the low-level libraries (Pham et al. 2020; Qian et al. 2021).

To start the clone training step, DisGUIDE collects a batch of training data in the following two steps. First, the generator creates a new batch of samples. Second, the victim model is queried to obtain the labels for these query samples. The labels are then used as the ground truth to train the clone models. The red arrows in Figure 1 show the query flow.

For each training batch, we compute the clones' predictions and update their weights using a loss function appropriate for the tasks. This paper uses the cross-entropy loss for the CIFAR-10 and CIFAR-100 classification tasks.

In the soft-label setting, the cross-entropy loss is computed between the softmax outputs of both the clones and the victim model to enable faster learning. However, in the more challenging hard-label setting, the loss is computed between the softmax outputs of the clones and the victim model's one-hot outputs (i.e., without confidence values), which reduces the effectiveness of each training step. This clone training phase updates the clones to more closely match the victim on each batch of query samples with the goal to eventually update the clone to match the output of the victim model as much as possible for the entire input space.

Experience Replay: Since querying new instances is costly (with pay-per-query systems), we leverage experience replay (Lin 1992) to reuse existing instances to improve clone model accuracy further. Researchers in the reinforcement learning domain found that experience replay helps avoid catastrophic forgetting (Kirkpatrick et al. 2017) and helps use samples more efficiently (Fedus et al. 2020). To supplement the clone training, we follow the experience replay algorithm from existing work (Kariyappa, Prakash, and Qureshi 2021) with a similar performance boost. A circular buffer of length s stores training samples as they are returned from the queries to the victim model. Then, after each clone training step, the clone models are trained with b batches randomly sampled from the experience replay module. Figure 1 shows this replay procedure, where we store query samples in the experience replay module and later sample from this storage to perform additional clone training steps.

4 Experimental Setup

In this section, we describe the datasets, the model architecture choices, the extraction scenario assumptions, and the settings to evaluate DisGUIDE in two different extraction scenarios: the soft-label setting (where the victim model returns the softmax predictions) and the hard-label setting (where only the predicted class is returned).

Datasets, Victim, and Clone Architectures: Following prior papers (Truong et al. 2021; Sanyal, Addepalli, and Babu 2022), we evaluate DisGUIDE on the two widely-used image classification datasets—CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009).

Following prior work (Fang et al. 2019), we evaluate the effectiveness of DisGUIDE at extracting the functionality of a ResNet-34 victim model into ResNet-18 clone models. We use the provided ResNet-34 victim models with test accuracies of 95.54% and 77.52% on CIFAR-10 and CIFAR-100, respectively. An exception is for the hard-label setting on CIFAR-100, where we evaluate DisGUIDE with ResNet-18 as both the victim and the clone models for an apple-to-apple comparison, as that is the exact setting of the only prior paper (Sanyal, Addepalli, and Babu 2022) of hard-label model extraction on CIFAR-100. Since the DFMS paper did not release the used ResNet-18 model, we had to train the ResNet-18 victim model ourselves. Our training should be a faithful

Setting	Technique	CIFAR-10			CIFAR-100		
		Query Budget	Victim (%)	Clone (%)	Query Budget	Victim (%)	Clone (%)
Soft-label	DFME	20M	95.54	88.10	/	/	/
	DFMS	20M	95.59	91.24	/	/	/
	DisGUIDE	20M	95.54	**94.02** ± 0.25	10M	77.52	**69.47** ± 0.88
Hard-label	DFMS	8M	95.59	84.51	10M	78.52	43.56
	DisGUIDE	8M	95.54	**87.93** ± 1.74	10M	77.70	**62.04** ± 1.03

Table 1: Final clone accuracy comparison. Numbers from DFME and DFMS are reported accuracy from their papers. '/' indicates that such numbers are not reported in the prior papers.

reproduction since the victim model has a test accuracy of 77.70% on CIFAR-100, which matches the reported accuracy in the paper very closely. We use the same generator as DFME, which has three convolutional layers.

DisGUIDE Hyperparameters and Settings: We use the same generator training hyperparameters as DFME: a batch size of 256, Adam optimizer with an initial learning rate of 1×10^{-4} and weight decay of 5×10^{-4}. Similarly, we use DFME's hyperparameters for clone training: batch size of 256, SGD with an initial learning rate of 0.1, and the same weight decay as above.

We start the extraction process with generator training first. Then the mentioned iterative training process starts by alternating between the clone training and the generator training. Within the iterative training process, there are many options for the ratio of generator to clone training. We choose the simplest setting of 1:1 by training the generator with one batch of samples, then training the clone from one generated batch of samples. We then empirically select the number of replay batches b to train the clone. We select $b = 3$ as well as a replay buffer size $s = 1M$ based on our stability results in Section 5.3.

DFME utilizes a learning rate scheduler that multiplies the learning rates by a factor of 0.3 at intervals specified by fractions of the query budget: [10%, 30%, 50%]. In other words, the learning rate is 1 initially and it is multiplied by 0.3 at 10% of the budget query, again at the 30% of the query budget, and so on. Since DisGUIDE is more query efficient, we set a lower initial learning rate of 0.3 and change it, also by a factor of 0.3, at intervals specified by fractions [40%, 80%] of the query budget.

We empirically set the class diversity loss weight λ = 0.2 and λ = 0.04 for CIFAR-10 and CIFAR-100 experiments respectively. We use the same values in both hard and soft label settings. The reduction by a factor of 5 from CIFAR-10 to CIFAR-100 follows a prior paper (Sanyal, Addepalli, and Babu 2022). Since a larger number of classes implies a higher diversity loss, the lambda value should is reduced.

Finally, specific to the image domain, we follow findings from a prior paper (Sanyal, Addepalli, and Babu 2022) to transform a fraction of the images created by the generator to grayscale as they found their synthetic dataset had better class diversity when converted to grayscale. We empirically select $\frac{1}{8}$ of the generated samples to be set to grayscale. We explore this further in Section 5.3.

Hardware and Software: We conduct our experiments on a server with 48 CPU cores with 504 GB of RAM and 2080Ti GPUs. Our code uses Pytorch 1.11 and CUDA 10.2.

5 Results

We evaluate DisGUIDE compared to the state-of-the-art data-free model extraction approaches. In the soft-label setting, we compare DisGUIDE to DFME and DFMS-SL. In the hard-label setting, we only compare DisGUIDE to DFMS-HL since the DFME paper did not evaluate DFME in the hard-label setting.

5.1 Accuracy Comparison

We compare DisGUIDE' final accuracy with the state-of-the-art approaches (Sanyal, Addepalli, and Babu 2022; Truong et al. 2021) on CIFAR-10 and CIFAR-100 datasets in both the hard-label and soft-label settings. To have a fair comparison, we use the same query budget as the DFME and DFMS papers. Specifically, we use 20M and 8M for CIFAR-10 in soft-label and hard-label settings, respectively, and for CIFAR-100, we use 10M for all experiments. We measure the accuracy of our clone models on the held-out test sets following standard practice. After the query budget is exhausted, we report the final soft-vote test accuracy of the two clone ensemble (Section 3). The victim and clone models are ResNet-34 and ResNet-18, respectively, for all experiments except for CIFAR-100 hard label with ResNet-18 for victim and clone models, all following prior papers (Details in Section 4).

Table 1 shows the final clone accuracy (Col. *Clone (%)*) achieved by each technique given the query budget (Col. *Query Budget*) and the victim accuracy (Col. Victim (%)) on the CIFAR-10 and CIFAR-100 datasets for the soft-label and hard-label settings (Col. *Setting*). We repeat the experiment for DisGUIDE five times and compare the *mean* ± *95% confidence interval* of the final test accuracy with the reported accuracies from the DFME and DFMS papers.

DisGUIDE outperforms by achieving accuracies of 94.02±0.25% and 87.93±1.74% in the soft- and hard-label settings, respectively. These are improvements of 2.78±0.25% and 3.42±1.74% over the current best approach, DFMS, using the same query budgets.

We also evaluate DisGUIDE on a more complex dataset, CIFAR-100, to understand how it expands to harder problems. Table 1 shows this result under the CIFAR-100 column. Since we are the first to evaluate a data-free model

Setting	Technique	CIFAR-10			CIFAR-100		
		Clone (%)	Reported	DisGUIDE	Clone (%)	Reported	DisGUIDE
Soft-label	DFME	88.10	20M	**3.13M** ± 0.57M	/	/	/
	DFMS	91.24	20M	**7.01M** ± 1.30M	/	/	/
Hard-label	DFMS	84.51	8M	**4.02M** ± 1.43M	43.56	10M	**3.87M** ± 0.37M

Table 2: Mean number of victim queries to reach prior papers' reported final accuracies. Lower is better. '/' indicates that the final accuracies are not reported in the prior papers, so it is not applicable for us to reach those accuracies.

extraction technique on CIFAR-100 in the soft-label setting, we do not have a direct state-of-the-art technique to compare. However, DisGUIDE achieves a promising final accuracy of 69.47±0.88% (89.62% of the victim's accuracy).

In the hard-label setting on CIFAR-100, DisGUIDE achieves a final accuracy of 62.04±1.03% and outperforms DFMS by 18.48±1.03%. As discussed in Section 4, we use the exact settings as the only prior paper (Sanyal, Addepalli, and Babu 2022) of hard-label model extraction on CIFAR-100. This means we use ResNet-18 as both the victim and the clone models here. Since using the same clone architecture as the victim is less realistic, we also use DisGUIDE to extract from a ResNet-34 victim model to ResNet-18 clones. In this less favorable setting, DisGUIDE still achieves a final accuracy of 58.72 ± 2.42%.

Summary: On the CIFAR-10 dataset, DisGUIDE outperforms the state-of-the-art data-free model extraction techniques by 2.78±0.25% and 3.42±1.74% in accuracy in the soft-label and hard-label settings respectively. On the CIFAR-100 dataset, DisGUIDE achieves a promising clone accuracy on the hard-label setting and outperforms the state-of-the-art DFMS technique by 18.48±1.03%.

5.2 Query Efficiency

We compare the number of queries required to match the final accuracies reported in prior papers (Truong et al. 2021; Sanyal, Addepalli, and Babu 2022) on CIFAR-10 in the soft-label and hard-label settings.

To study the query efficiency of our approach, we measure the minimum number of queries required to reach prior papers' reported accuracy. Table 2 shows the mean number of victim queries that DisGUIDE requires (Col. *DisGUIDE*) to reach prior papers' clone accuracies (Col. *Clone (%)*) with the number of queries reported in prior papers (Col. *Reported*) on the CIFAR-10 and CIFAR-100 datasets. We repeat the experiment for DisGUIDE five times and report the *average ± 95% confidence interval* of the number of required victim queries.

On CIFAR-10, DisGUIDE is significantly more efficient than prior work. Specifically, in the soft-label setting, DisGUIDE requires only 3.13±0.57M and 7.01±1.30M (versus a 20M budget) to match DFME's and DFMS's reported accuracies, respectively, with a 20M query budget (84.35±7.13% and 64.95±16.25% query reduction respectively). Similarly, in the hard-label setting, our approach requires a smaller query budget than DFMS: DisGUIDE requires 49.75±17.88% fewer queries (from 8M to 4.02±1.43M) to reach the final reported accuracy of DFMS.

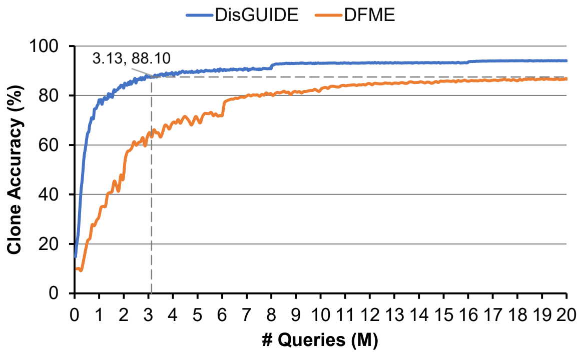

Figure 2: Visualizes the growth of the clone accuracy of DisGUIDE (the blue line) and DFME (the orange line) on the CIFAR-10 dataset in the soft-label setting.

On the more complex CIFAR-100 dataset, in the hard-label setting, DisGUIDE is much more query efficient than DFMS, i.e., DisGUIDE uses 61.30±3.70% fewer queries (from 10M to 3.87±0.37M) to reach the final reported accuracy of DFMS. In the soft-label setting, since the final accuracies are not reported in the prior papers for the soft-label setting, it is not applicable for us to analyze how efficient DisGUIDE is in reaching those accuracies.

As shown in Figure 2, DisGUIDE improves the clone accuracy very quickly, with a steep curve in the first 2 million queries. In contrast, the DFME curve is much more shallow and takes 6 million queries to reach a reasonable accuracy. Because of this, DisGUIDE is able to reach DFME's final accuracy at the 3.13 million queries mark as the dashed lines show, and DisGUIDE continues to improve and reaches a final accuracy of 94.02%.

Summary: DisGUIDE is much more query efficient than the state-of-the-art technique DFMS in both soft-label and hard-label settings. Specifically, on CIFAR-10, DisGUIDE requires 64.95% fewer queries (12.99 million query reduction) in the soft-label setting and 49.75% fewer queries in the hard-label setting. On the harder CIFAR-100, DisGUIDE requires 61.30% fewer queries in the hard-label setting.

5.3 DisGUIDE Stability

In this section, we study the stability of DisGUIDE with regards to the hyperparameter settings of experience replay

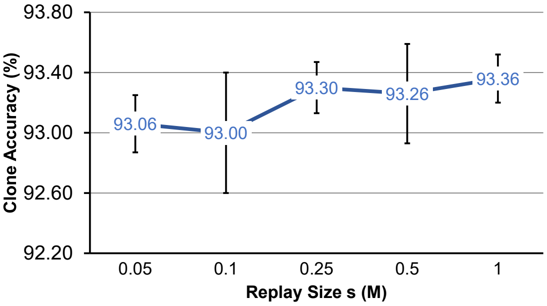

Figure 3: Impact of different replay sizes (s) on DisGUIDE on the CIFAR-10 dataset in the soft-label setting

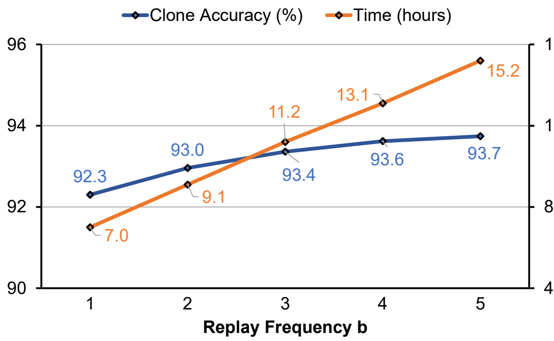

Figure 4: Impact of different replay frequency (b) on DisGUIDE on CIFAR-10 dataset in the soft-label setting

size, sampling frequency, and the use of grayscaling on our final accuracy. We use a query budget of 8M to save time.

Experience Replay Size: We study the impact of the replay size s on the final accuracy. We measure the average final accuracy over four runs for different replay sizes: [0.05M, 0.1M, 0.25M, 0.5M, 1M]. A replay size of 1M means that we store a maximum of one million samples to be sampled for further clone model training.

Figure 3 shows the final *clone accuracy (%)* with respect to the replay size s for DisGUIDE on CIFAR-10 dataset in the soft-label setting. The error bars show the 95% confidence intervals. The results show that DisGUIDE is insensitive to the replay size and a small replay size is sufficient for DisGUIDE to perform well. However, given more storage budget, DisGUIDE can achieve slightly better results. Thus, we select a replay size of s = 1M as our default setting.

Frequency of Experience Replay Sampling: The frequency of experience replay sampling affects both the extracted clones' accuracy and the running time of DisGUIDE; thus, choosing a good replay sampling frequency is essential. A sampling frequency of three means that we randomly sample three batches from the replay per training loop (Section 3.2). The higher the frequency, the more samples we use from the experience replay, thus potentially higher clone accuracy and longer training time. We study the impact of replay sampling frequency by experimenting with different b values (i.e., the number of batches sampled from the experience replay module in each training loop). We perform the study with values of b: from 1 to 5. Figure 4 shows the final test accuracy of the clone (blue line) and the runtime (orange line in hours) with respect to the *Replay Frequency* b for DisGUIDE on the CIFAR-10 dataset with soft-labels.

Setting	Grayscale	Clone (%)
Soft-label	✗	93.25 ± 0.15%
	✓	93.36 ± 0.16%
Hard-label	✗	85.54 ± 2.47%
	✓	87.93 ± 1.74%

Table 3: Impact of grayscaling on DisGUIDE on CIFAR-10

The results show that higher replay sampling frequency results in more accurate clones at the cost of higher compute time. Specifically, when $b = 5$, DisGUIDE achieves the best accuracy of 93.74% at the cost of 15.2 hours. However, compared to $b = 3$ this improvement of 0.38% is at the cost of almost 4 hours, or a 36% increase in the run time. This result prompts us to select $b = 3$ with the overall best trade-off between running time and final accuracy.

Grayscale: We investigate the impact of our design choice to convert a fraction of the generated images to grayscale. Table 3 shows the final clone accuracy (Col. *Clone (%)*) of DisGUIDE with (✓) or without (✗) grayscaling (Col. *Grayscale*) in both soft-label and hard-label settings (Col. *Setting*) on the CIFAR-10 dataset. Results show that DisGUIDE is not too sensitive to grayscaling in the soft-label setting. However, our hard-label results see an accuracy boost of 2.39% from grayscaling. This matches the findings from prior paper (Sanyal, Addepalli, and Babu 2022) as they found their synthetic dataset had better class diversity when converted to grayscale in the hard-label settings.

Generator Sample Quality: Samples generated by our method do not visually represent anything close to the victim's training distribution. Some example images are in the appendix in Figure 1. The generated samples need only provide useful training signals for the clone models to match the victim model.

6 Conclusion, Limitations, and Future Work

We introduce a new data-free model extraction algorithm DisGUIDE that significantly improves the accuracy of model extraction and reduces the number of queries required to reach the accuracies of prior techniques.

While we evaluate our approach on two tasks, CIFAR-10 and CIFAR-100, future work is needed to extract models for more complex problem domains. While it did not happen during our experiments, DisGUIDE could theoretically get stuck in local minima, as models could agree on incorrect predictions completely. A possible solution is to retrain, given the training non-determinism.

We expect our approach to be generally applicable and that it can be extended to ensembles of more than two models. Most importantly, we hope this research into data-free model extraction techniques highlights the vulnerabilities of current systems and will promote future work into safeguards against these possible attacks.

A Appendix

A.1 Generator Produced Images

The DisGUIDE generator is trained to maximize the disagreement between student models as well as the class diversity of generated samples. Figure 5 shows images generated by a DisGUIDE generator at the end of a CIFAR-10 soft-label training run. The images are shown to satisfy the reader's curiosity.

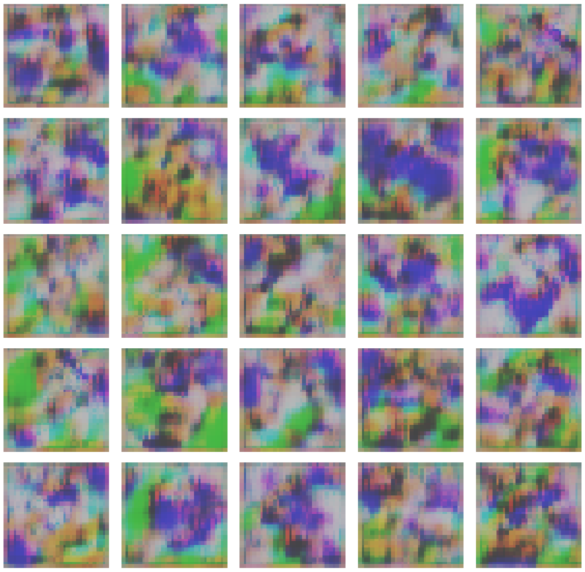

Figure 5: Images generated by a DisGUIDE generator at the end of training. Model was taken from a CIFAR-10 soft-label run after 20M queries.

A.2 Transform

When training a machine learning model, it is common to do some preprocessing of the training dataset, such as normalizing pixel values based on the training distribution. In a MLaaS setting, a client sends data to the provider who then does the same preprocessing on the input features before using the result as input for their trained model. The preprocessing step is not known to the client and should thus be handled with care.

Table 4 compares the results of the method with and without knowledge of the transformation on CIFAR-10 in the soft-label setting with a query budget of 8M queries. In the rest of the paper, the reported results of DisGUIDE all assume the attacker does not have access to the image normalization transform of the service provider.

Attacker Knowledge	Victim (%)	Clone(%)
None	95.54	93.36 ± 0.16
Input Transform	95.54	93.45 ± 0.23

Table 4: Effect of knowing input transform on DisGUIDE. Results on CIFAR-10 in soft-label setting with a query budget of 8M.

A.3 Process Overview

Algorithm 1 contains pseudocode for a possible implementation of the DisGUIDE algorithm. It is given here to supplement the readers understanding of the method. The full codebase will be open sourced on acceptance and is available to the reviewer.

A.4 Final vs Maximum Achieved Accuracy

In the data-free model extraction setting, there is no concept of a validation or test set for the attacker. Future work may find ways to select at what point in training the student model is best. Until then, the reported metrics should be mean final values.

In this work, reported values are the mean final model accuracies over multiple runs. Table 5 displays the same models as table 1 in the main portion of the paper. There is an extra column (*Max Clone(%)*) which gives the mean maximum accuracy reached across the runs for both CIFAR-10 and CIFAR-100.

For all experiments, the mean maximum achieved accuracy was within the 95% confidence interval of the final ac-

Algorithm 1: DisGUIDE: Full Algorithm

```
Input: Victim $V$
Parameter: g-iter, d-iter, rep-iter
Output: Clone models
  Initialize 2 clone models
  Initialize empty experience replay
  for i=1 to g-iter do
    Generate features
    Get clone models predictions
    Compute disagreement and diversity loss
    Update generator weights based on loss
  end for
  for i=1 to d-iter do
    Generate features
    Query victim for labels
    Update experience replay with features and labels
    for j=1 to 2 do
      Query clone j for predictions
      Compute clone j loss based on predictions and labels
      Update clone j weights based on loss
    end for
  end for
  for i=1 to rep-iter do
    Randomly select a batch of (feature, label) pairs
    for j=1 to 2 do
      Query clone j for predictions
      Compute clone j loss based on predictions and labels
      Update clone j weights based on loss
    end for
  end for
  return clone models
```

Setting	CIFAR-10				CIFAR-100			
	Budget	Victim (%)	Clone (%)	Max Clone (%)	Budget	Victim (%)	Clone (%)	Max Clone (%)
Soft-label	20M	95.54	94.02± 0.25	94.26 ± 0.20	10M	77.52	69.47± 0.88	69.83 ± 1.01
Hard-label	8M	95.54	87.93± 1.74	88.94 ± 0.92	10M	77.70	62.04± 1.03	62.52 ± 0.91

Table 5: Comparison of final accuracies at the end of training and the maximum achieved accuracies for DisGUIDE on CIFAR-10 and CIFAR-100 in soft- and hard-label settings.

Setting	CIFAR-10				CIFAR-100			
	Budget	Victim (%)	Soft Vote (%)	Individual (%)	Budget	Victim (%)	Soft Vote (%)	Individual (%)
Soft-label	20M	95.54	94.02± 0.25	93.95 ± 0.15	10M	77.52	69.47± 0.88	68.84 ± 0.54
Hard-label	8M	95.54	87.93± 1.74	87.33 ± 0.98	10M	77.70	62.04± 1.03	60.55 ± 0.54

Table 6: Comparison of model soft vote accuracies with individual model accuracies.

curacy reached. The only set of runs where the maximum was more than 0.5% above the final accuracy was CIFAR-10 in the hard-label setting.

A.5 Soft Vote vs Individual Models

The DisGUIDE method outputs two separate clone models. In some instances, an attacker may prefer an individual clone model as output. Table 6 illustrates the difference between the soft vote accuracies and individual model accuracies under differing conditions. The soft-vote accuracy is generally slightly higher, as to be expected. The mean individual accuracy is within the 95% confidence interval of the soft vote accuracy for all conditions, with exception of CIFAR100 in the hard-label setting.

Acknowledgements

This work has been partially supported by NSF 2006688 and a J.P. Morgan AI Faculty Research Award.

References

Addepalli, S.; Nayak, G. K.; Chakraborty, A.; and Radhakrishnan, V. B. 2020. Degan: Data-enriching gan for retrieving representative samples from a trained classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3130–3137.

Chang, X.; Lee, S. Y. M.; Zhu, S.; Li, S.; and Zhou, G. 2022. One-Teacher and Multiple-Student Knowledge Distillation on Sentiment Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, 7042–7052.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3514–3522.

Duchi, J. C.; Jordan, M. I.; Wainwright, M. J.; and Wibisono, A. 2012. Finite Sample Convergence Rates of Zero-Order Stochastic Optimization Methods. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 1439–1447. Red Hook, NY, USA: Curran Associates Inc.

Fang, G.; Song, J.; Shen, C.; Wang, X.; Chen, D.; and Song, M. 2019. Data-Free Adversarial Distillation. *arXiv preprint arXiv:1912.11006*.

Fedus, W.; Ramachandran, P.; Agarwal, R.; Bengio, Y.; Larochelle, H.; Rowland, M.; and Dabney, W. 2020. Revisiting fundamentals of experience replay. In *International Conference on Machine Learning*, 3061–3071. PMLR.

Ghadimi, S.; and Lan, G. 2013. Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. *SIAM Journal on Optimization*, 23(4): 2341–2368.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6): 1789–1819.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

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.

Kariyappa, S.; Prakash, A.; and Qureshi, M. K. 2021. MAZE: Data-Free Model Stealing Attack Using Zeroth-Order Gradient Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13814–13823.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

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

Li, T. 2021. *Model Extraction and Adversarial Attacks on Neural Networks Using Side-Channel Information*. Rochester Institute of Technology.

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3): 293–321.

Lopes, R. G.; Fenu, S.; and Starner, T. 2017. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot Knowledge Transfer via Adversarial Belief Matching. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 9551–9561. Curran Associates, Inc.

Miura, T.; Hasegawa, S.; and Shibahara, T. 2021. MEGEX: Data-Free Model Extraction Attack against Gradient-Based Explainable AI. *arXiv preprint arXiv:2107.08909*.

Mopuri, K. R.; Uppala, P. K.; and Babu, R. V. 2018. Ask, acquire, and attack: Data-free uap generation using class impressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 19–34.

Nayak, G. K.; Mopuri, K. R.; Shaj, V.; Radhakrishnan, V. B.; and Chakraborty, A. 2019. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, 4743–4751. PMLR.

Oliynyk, D.; Mayer, R.; and Rauber, A. 2022. I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. *arXiv preprint arXiv:2206.08451*.

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.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Pham, H. V.; Qian, S.; Wang, J.; Lutellier, T.; Rosenthal, J.; Tan, L.; Yu, Y.; and Nagappan, N. 2020. Problems and Opportunities in Training Deep Learning Software Systems: An Analysis of Variance. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 771–783.

Qian, S.; Pham, V. H.; Lutellier, T.; Hu, Z.; Kim, J.; Tan, L.; Yu, Y.; Chen, J.; and Shah, S. 2021. Are my deep learning systems fair? An empirical study of fixed-seed training. *Advances in Neural Information Processing Systems*, 34: 30211–30227.

Sanyal, S.; Addepalli, S.; and Babu, R. V. 2022. Towards Data-Free Model Stealing in a Hard Label Setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15284–15293.

Shumailov, I.; Zhao, Y.; Bates, D.; Papernot, N.; Mullins, R.; and Anderson, R. 2021. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 212–231. IEEE.

Tasumi, M.; Iwahana, K.; Yanai, N.; Shishido, K.; Shimizu, T.; Higuchi, Y.; Morikawa, I.; and Yajima, J. 2021. First to Possess His Statistics: Data-Free Model Extraction Attack on Tabular Data. *arXiv preprint arXiv:2109.14857*.

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. Austin, TX: USENIX Association. ISBN 978-1-931971-32-4.

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 (CVPR)*.

Walawalkar, D.; Shen, Z.; and Savvides, M. 2020. Online ensemble model compression using knowledge distillation. In *European Conference on Computer Vision*, 18–35. Springer.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.