Towards Data-Free Model Stealing in a Hard Label Setting

Sunandini Sanyal **Sravanti Addepalli** **R. Venkatesh Babu**
Video Analytics Lab, Department of Computational and Data Sciences
Indian Institute of Science, Bangalore

Abstract

Machine learning models deployed as a service (MLaaS) are susceptible to model stealing attacks, where an adversary attempts to steal the model within a restricted access framework. While existing attacks demonstrate near-perfect clone-model performance using softmax predictions of the classification network, most of the APIs allow access to only the top-1 labels. In this work, we show that it is indeed possible to steal Machine Learning models by accessing only top-1 predictions (Hard Label setting) as well, without access to model gradients (Black-Box setting) or even the training dataset (Data-Free setting) within a low query budget. We propose a novel GAN-based framework[1] that trains the student and generator in tandem to steal the model effectively while overcoming the challenge of the hard label setting by utilizing gradients of the clone network as a proxy to the victim's gradients. We propose to overcome the large query costs associated with a typical Data-Free setting by utilizing publicly available (potentially unrelated) datasets as a weak image prior. We additionally show that even in the absence of such data, it is possible to achieve state-of-the-art results within a low query budget using synthetically crafted samples. We are the first to demonstrate the scalability of Model Stealing in a restricted access setting on a 100 class dataset as well.

1. Introduction

Deep learning based systems have progressed leaps and bounds over the past few years, enabling their deployment in critical applications such as self-driving cars, surveillance systems and biomedical applications. Furthermore, organizations often provide pretrained machine learning models as a service (MLaaS) where the end user is allowed to query the model and get access to its predictions via APIs for use in various applications.

However, exposing the predictions of the models through queries makes the model susceptible to model stealing attacks, which attempt to clone the victim model even in a black-box setting that restricts access to its gradients. Protecting the privacy of an ML model is of paramount importance as organizations invest significant resources on cutting edge research and also on gathering and labelling large amounts of training data [14] for achieving competent performance on various tasks. In addition, recent works [29, 32, 35, 42] have shown that an adversary could train a substitute model via model stealing and use it for crafting adversarial examples [12] in a black-box setting, which poses a serious threat when the model is deployed in security critical applications. A stolen model could also compromise the privacy of users by leaking confidential data through a membership inference attack [31] or model inversion [40, 41]. Fig.-1 showcases some of the possible malicious outcomes of Model Stealing. In order to prevent model stealing attacks, some defenses attempt to perturb the softmax predictions of the model, while preserving its top-1 prediction [22]. In this work we consider the problem of model stealing in a more practical and challenging hard label setting, where only the top-1 prediction of the model

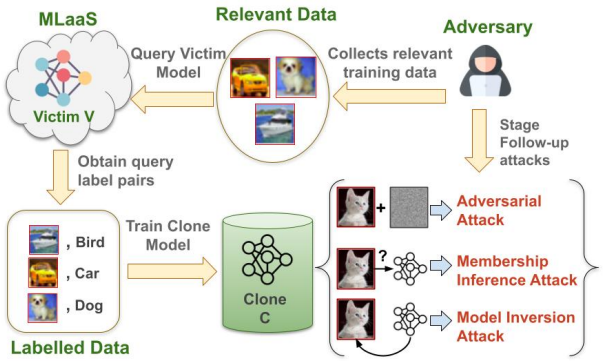

Figure 1. **Model Stealing Attack and its vulnerabilities**: An adversary queries the victim Model $\mathcal{V}$ with proxy data to obtain its labels. The labelled training data is used to train a clone Model $\mathcal{C}$ which can be further used by the adversary to stage membership inference [31], model inversion [10] or adversarial attacks [43].

[1]Project Page: https://sites.google.com/view/dfms-hl

is accessible, and is thus effective even in the presence of such defenses.

In a model stealing attack, an adversary first queries a black-box victim model $\mathcal{V}$ with input data and obtains a prediction for it as shown in Fig.1. This data along with victim model predictions is used to train a clone model $\mathcal{C}$. In a practical scenario, the attacker would not have access to the training data, and hence we consider the problem of Data-Free Model Stealing (DFMS) in this work. In such a data-free scenario, the attacker could use publicly available related datasets [26, 29], or synthetically generated samples [34] to query the model. While the use of publically available datasets assumes access to related data, the data-free generative approach could suffer from a large query budget, as the synthetic data can be far from the true training data distribution. In this work, we overcome both challenges by utilizing the available data that may be unrelated to the original training dataset, as a weak image prior. This enables the generation of representative samples under a low query budget, which is a crucial requirement in model stealing attacks, since MLaaS APIs work on a pay-per-query basis.

While existing algorithms for Data-Free Knowledge Distillation [1, 9, 23, 25, 39] and Model Extraction [18, 34] achieve near perfect clone-model accuracy, there are additional challenges in a Model Stealing framework due to the lack of access to gradients and a hard-label setting. Therefore, we consider a practical setup of data-free hard-label model stealing and overcome the challenges by utilizing the clone model's gradients as a proxy to the gradients of the victim model. As the clone model starts training, it acts as a useful proxy for the victim model, and helps the generator learn to generate rich informative samples, which boosts the clone accuracy further. We explicitly enforce the generation of a class-balanced dataset from the generator that is also more aligned with the distribution of the training dataset. Additionally, we also utilize an adversarial loss in a GAN framework [11], by using publicly available potentially unrelated data, which we refer to as proxy data [1]. While this could be completely unrelated to the original training dataset, it still helps in enforcing a weak image prior in the generated data. This in turn reduces the number of victim model queries needed to perform Model Stealing. In fact, we show that it is possible to even use synthetic samples, such as multiple overlapping shapes with a planar background, to steal a model in a completely data-free setting.

Our method achieves a significant improvement over ZSDB3KD [37], a zero-shot data-free method in a similar hard label setting using only synthetic samples. In the upcoming sections, we describe our approach in detail and show results on various datasets.

Our **key contributions** are listed below:

- We propose DFMS-HL, a Data-Free Model Stealing (DFMS) attack in a Hard-Label (HL) setting, to train a clone model with the help of unrelated proxy data or manually crafted synthetic data. We show that DFMS-HL outperforms the existing baseline ZSDB3KD [37] and results in a significant reduction of around $500\times$ in the number of queries to the victim model.
- We demonstrate state-of-the-art results on the CIFAR-10 dataset using unrelated proxy samples, such as a given subset (containing 40 or 10 non-overlapping classes) from CIFAR-100, or synthetic data.
- We are the first to show noteworthy results of data-free model stealing on a dataset with a larger number of classes such as CIFAR-100. This demonstrates that our approach is both effective and scalable.
- We compare our method with the state-of-the-art model stealing attacks MAZE [17] and DFME [34], which additionally utilize softmax predictions of the victim model. Although we consider a more restrictive setting, we achieve a comparable accuracy using the DFMS-HL approach, and a significant boost of around 3% using a Soft-Label (SL) variant of the proposed method (DFMS-SL).

2. Related Work

In this section, we discuss existing Knowledge Distillation and Model Stealing works with varied levels of access to the victim model as shown in Table-1.

2.1. Knowledge distillation

Knowledge distillation [15] aims to transfer the knowledge of a large pretrained teacher model to a smaller student model without a significant impact on accuracy. This is primarily used to compress models for deployment, in order to reduce the memory requirements and inference time [2, 3, 13, 38]. In practical scenarios, training data is kept confidential due to privacy concerns. Hence, there has been a lot of focus on developing data-free approaches for knowledge-distillation. ZSKD [25], DAFL [7], DFKD [23] are popular knowledge distillation methods in a data-free setting. A data-free KD method DeGAN [1] demonstrated that it is possible to use publicly available unrelated data (proxy dataset) to distill the knowledge of a teacher model to a smaller student model. However, all these methods require access to the teacher model's gradients. Following this, Black-Box Ripper [4] was proposed to implement model stealing by querying a black-box teacher model with unrelated proxy data. A recent work ZSDB3KD [37] proposed knowledge distillation for a black box model with only hard-label outputs. However, this approach is highly computationally intensive due to the requirement of a very large number of queries (4000 million) to the teacher model. Our work considers the same setup of having access to only

Table 1. Taxonomy of prior works on Knowledge Distillation (KD) and model stealing attacks. Our approach DFMS-HL is a data-free model stealing attack on a black-box victim model with access to only hard labels.

Approach	White-Box Soft Label	Black-Box Soft Label	Black-Box Hard Label
Data free	ZSKD [25] DeGAN [1]	MAZE [18] DFME [34]	ZSDB3KD [37] DFMS-HL (Ours)
Data	KD with Data [15]	KnockoffNets [26] JBDA [29]	-

the top-1 labels, with a significantly lower query budget of 8 million.

2.2. Model Stealing

Tramer *et al*. [33] demonstrated that an attacker could use queries to steal a machine learning model with near perfect fidelity. Following this, model stealing has been implemented in various domains [8, 16, 21, 24, 28]. A partial data approach JBDA [29] assumed access to a small set of samples from the data distribution. On the other hand, surrogate data approaches such as KnockOffNets [26] and Black-Box dissector [36] consider that attackers could use images from a different data source to steal a model. These methods fail to perform well without a suitable surrogate dataset. This motivated the development of data-free approaches which work well without using surrogate data or seed samples from the training data. Recent data-free approaches such as MAZE [18] and DFME [34] attempt to extract models using GAN generated synthetic data. In these approaches, the generator is trained to produce images that maximize the dissimilarity score between the clone and victim models. The victim model's gradients are required to measure this dissimilarity score, and are estimated using zeroth-order gradient approximation. These approaches are computationally expensive as they require a lot of queries ($\sim$20 million) to the victim model for synthesizing data samples in a black-box setting. Moreover, these methods assume that the softmax vector from the teacher model is accessible. Contrary to this, we consider a practical setting that allows access to only hard labels from the victim model.

2.3. Defenses against model stealing

Lee *et al*. [22] propose to defend against model stealing attacks by perturbing the model predictions while preserving its top-1 label, to maintain similar classification accuracy. On similar lines, Prediction Poisoning [27] perturbs model predictions by poisoning the output distribution at the cost of model accuracy. However, such defenses fail in a scenario where an attacker has access to only hard labels from the model. A more sophisticated approach EDM [17] introduces randomness into the predictions by using an ensemble of diverse models to produce dissimilar outputs for Out-of-Distribution (OOD) samples, that are likely to be used for querying the victim model in a model stealing attack. Similarly, Adaptive Misinformation [19] perturbs the predictions for OOD inputs only. However, these approaches have been shown to cause utility degradation [27], or can be made ineffective using an adaptive query synthesis strategy [5]. Further, Chandrasekaran *et al*. [5,6] provide theoretical insights to demonstrate that "model extraction is inevitable", even in a realistic setting with only hard labels, and even when models use randomised defenses. Hence, a model with a reasonably good accuracy would always leak information that could lead to model extraction. In this work we demonstrate that it is indeed possible to perform model stealing in a severely restricted setting as well, and further achieve competent clone accuracy. This paves way to the development of better defenses for preserving model privacy in future.

3. Proposed Approach

For model stealing, the goal of an adversary is to learn the parameters of the clone model $\mathcal{C}$ so as to match the predictions of the victim model $\mathcal{V}$. Towards this end, we propose a data-free model stealing approach **DFMS-HL** that requires only hard-label access. In the following sections, we describe the proposed model stealing attack algorithm.

3.1. Overview

We use a GAN based architecture to train the clone model. We first train a DCGAN [30] by imposing an image prior using synthetic data or unrelated proxy data, and use this as an initialization for the generator $\mathcal{G}$. Further, the clone model and generator are trained alternately. The data flow of the proposed model stealing attack is shown in Fig. 2, wherein the generator $\mathcal{G}$ generates data $x = \mathcal{G}(z)$ from a random vector z. The victim model takes input x and generates input-label pairs $(x, \hat{y}(x))$. Since, the victim model is black-box, we do not backpropagate the gradients through it. We use the input-label pairs to train the clone model. Further, the generated data x is used to train the generator using the adversarial loss [11] and a diversity loss [1]. The discriminator learns to differentiate between fake and real proxy data using the adversarial loss. In the subsequent sections, we describe the loss functions for training the generator and clone model in further detail.

3.2. Clone model Training

The clone model $\mathcal{C}$ is trained using the data samples generated from the generator $\mathcal{G}$. In every iteration, we sample an m-dimensional random vector z, whose elements are sampled from m *i.i.d.* Standard Normal distributions. This vector is forward propagated through $\mathcal{G}$ to generate images x. These images are then passed to the victim model to

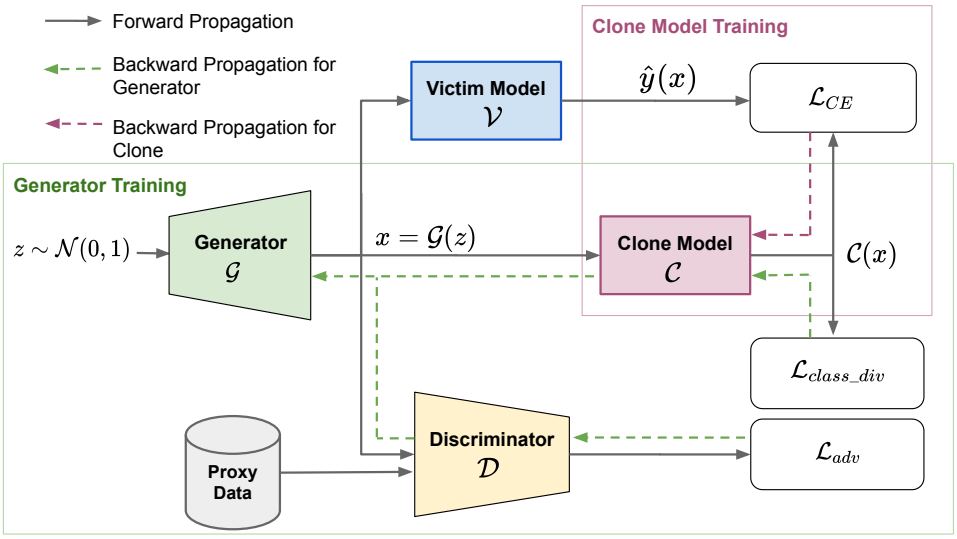

Figure 2. **Architecture of DFMS-HL**: Generator $\mathcal{G}$ generates data x with a proxy image prior. The clone model $\mathcal{C}$ is trained using the predictions from the victim model $\mathcal{V}$ with a cross-entropy loss objective $\mathcal{L}_{CE}$. The discriminator $\mathcal{D}$ learns to discriminate between proxy data and the samples generated from $\mathcal{G}$. The generator $\mathcal{G}$ is trained using the adversarial loss $\mathcal{L}_{adv}$ along with the class-diversity loss $\mathcal{L}_{class\_div}$. The generator and clone model are trained alternately in every iteration of the algorithm.

obtain its hard-labels. The clone model is trained with the cross-entropy loss objective using the victim predictions as ground truth, as shown below:

$$\mathcal{L}_C = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I)} \left[\mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))\right], \; x = \mathcal{G}(z) \quad (1)$$

where $\hat{y}(x) = \underset{i}{\text{argmax}}\, \mathcal{V}_i(x)$ is the class label corresponding to the maximum probability class, I is an m dimensional identity matrix, and $\mathcal{C}(x)$ is the pre-softmax output from the clone model.

3.3. Generator Training

For imposing an image prior, we initially train a DCGAN generator using proxy data or synthetic images. However, we find that this is not sufficient as the generator could potentially suffer from mode collapse and lack of diversity. Moreover, lack of class diversity can severely impact the learning of tail classes in a hard-label setting. Hence, it crucial for the generator to generate a class-balanced set of images for learning the information across all classes. Therefore, we use a class-diversity loss formulation [1] to generate diverse samples from the generator $\mathcal{G}$ while remaining close to the manifold of the proxy/synthetic images.

The generator loss has two components. The first component is the adversarial loss [11] which causes the generator to generate data close to the proxy data distribution. The second component is a class balancing loss [1], to enforce a diversity constraint. The two loss formulations for the generator are described in more detail below.

Adversarial Loss [11]: The adversarial loss ensures that the distribution of images is close to the images in the proxy or synthetic dataset.

$$\mathcal{L}_{adv,real} = \mathop{\mathbb{E}}_{x\sim p_{data}(x)} [log\mathcal{D}(x)] \quad (2)$$

$$\mathcal{L}_{adv,fake} = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I)} [log(1 - \mathcal{D}(\mathcal{G}(z)))] \quad (3)$$

The discriminator $\mathcal{D}$ and generator $\mathcal{G}$ play a min-max game [11] as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake} \quad (4)$$

Class Diversity Loss [1]: The class diversity loss encourages the generation of diverse images across all classes. In a batch of N samples, we consider the expected confidence value over the batch as α_j for every class j, and obtain the entropy over all K classes. The negative entropy, denoted as $\mathcal{L}_{class\_div}$ is computed as shown below:

$$\mathcal{L}_{class\_div} = \sum_{j=0}^{K} \alpha_j \log \alpha_j \quad (5)$$

$$\alpha_j = \frac{1}{N} \sum_{i=1}^{N} \text{softmax}(\mathcal{C}(x_i))_j \quad (6)$$

Using Clone Model as a Proxy for Victim: Since, the victim model is black-box, backpropagation through $\mathcal{V}$ is not permitted. Hence, for imposing diversity we use the clone model parameters to compute the loss. Over the training process, the clone learns to imitate the gradients of the victim, making it a suitable proxy for enforcing diversity in the generated images.

Algorithm 1 DFMS-HL : Algorithm for Model Stealing

Require: $N_Q, \mathcal{G}, \mathcal{D}, n_G, n_C$
// Initialize a Generator $\mathcal{G}$ with DCGAN parameters
// Train the clone model $\mathcal{C}$ with DCGAN and proxy images using n_C queries for initialization.
while $n_G \neq 0$ **do**
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_G \leftarrow \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div}$
$\mathcal{L}_D \leftarrow \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$
$\theta_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}} - \epsilon_{\mathcal{G}}\nabla_{\theta_G}\mathcal{L}_G$
$\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} - \epsilon_{\mathcal{D}}\nabla_{\theta_D}\mathcal{L}_D$
$n_G \leftarrow n_G - 1$
end while
// Train clone model $\mathcal{C}$
while $n_C \neq 0$ **do**
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_C \leftarrow \mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))$
$\theta_{\mathcal{C}} \leftarrow \theta_{\mathcal{C}} - \epsilon_{\mathcal{C}}\nabla_{\theta_C}\mathcal{L}_C$
$n_C \leftarrow n_C - 1$
end while
// Start alternate training between $\mathcal{G}$ and $\mathcal{C}$
while $N_Q \neq 0$ **do**
// Train $\mathcal{G}$ and $\mathcal{D}$ with $\mathcal{C}$ as fixed
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_G \leftarrow \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div}$
$\mathcal{L}_D \leftarrow \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$
$\theta_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}} - \epsilon_{\mathcal{G}}\nabla_{\theta_G}\mathcal{L}_G$
$\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} - \epsilon_{\mathcal{D}}\nabla_{\theta_D}\mathcal{L}_D$
// Train $\mathcal{C}$ with $\mathcal{G}$ and $\mathcal{D}$ as fixed
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_C \leftarrow \mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))$
$\theta_{\mathcal{C}} \leftarrow \theta_{\mathcal{C}} - \epsilon_{\mathcal{C}}\nabla_{\theta_C}\mathcal{L}_C$
end while

The equations given below describe the overall generator and discriminator losses.

$$\mathcal{L}_G = \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div} \tag{7}$$

$$\mathcal{L}_D = \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake} \tag{8}$$

3.4. Algorithm

The overall training algorithm is outlined in Algorithm-1. We first train a DCGAN to initialize the generator model with an image prior. Following this, we train the clone model using a mix of images from the DCGAN and the proxy dataset to obtain a good initialization for the clone model. Using this clone model, we further fine-tune the generator for n_G epochs using the two proposed losses; adversarial loss and class-diversity loss. We then train a clone model from scratch for n_C epochs using the images from the diverse generator $\mathcal{G}$. Following this, we start the alternate training process for the generator and clone model. We train the generator for one iteration by freezing weights of the clone model and subsequently train the clone model for one iteration using labels from the victim model. This procedure is repeated until the query budget N_Q is exhausted.

3.5. Computing the Query Cost

In this section, we compute the total number of queries to the victim model. The number of samples in the proxy data is denoted as N_P. Initially, we require n_C queries to obtain a clone model to initialize the generator and an additional n_C queries to initialize the Classifier $\mathcal{C}$. For our experiments, we set n_C as 50,000. The alternate training of the clone and generator continues for E epochs and in each epoch, the victim model is queried N_P times. So the total query cost is computed as follows,

$$N_Q = E \cdot N_P \tag{9}$$

$$\text{Total Queries} = 2 \cdot n_C + N_Q \tag{10}$$

We set the query limit N_Q to 8 million for our proxy and synthetic data experiments on CIFAR-10.

3.6. Insights on Query Budget

Chandrasekaran *et al*. [5] formulated the model extraction task as a query synthesis active learning problem where an adversary learns a hypothesis function with a query complexity $q_A(\epsilon, \delta)$. The authors observe that it is possible for an adversary to implement an ϵ-extraction attack with query complexity $q_A(\epsilon, \delta)$ and confidence $1-\delta$ (described in Section 2 of the Supplementary). The authors [5] further prove that model stealing is inevitable and there exists a query bound within which a model could be stolen. We empirically find the query budget needed for the proposed approach in the Query ablation (Section 5).

4. Experiments

We perform experiments to evaluate the effectiveness of the proposed algorithm DFMS-HL in a hard-label data-free setting. We primarily compare our approach to the existing method ZSDB3KD [37], which is a zero-shot hard-label Knowledge distillation method. We present evaluations of DFMS-HL by using various proxy datasets as well as with synthetically crafted data. Our attack not only outperforms ZSDB3KD by a large margin, but also achieves clone-model accuracy comparable to the soft-label methods by using only hard-labels from the victim model. Additionally, we perform ablations to highlight the number of queries required to successfully steal a model, and also to understand the impact of the class-diversity loss. Our analysis reveals that the proposed attack is computationally more efficient when compared to existing approaches since it requires significantly lesser queries.

Table 2. **Comparison of DFMS-HL with state-of-the-art KD methods (Top) and ZSDB3KD (Bottom) on CIFAR-10**: Clone model accuracy (%) reported using Proxy data as unrelated (40 or 10) CIFAR-100 classes and synthetic data. Victim and clone model architectures used are Alexnet and AlexNet-half respectively.

Method	Hard Label	Black-Box	Data-Free	Victim Accuracy	Synthetic/ Data-Free	CIFAR-100 (40C)	CIFAR-100 (10C)
Victim Accuracy = 82.5%							
ZSKD [25]	×	×	✓	82.50	**69.50**	-	-
DeGAN [1]	×	×	✓	82.50	-	76.30	72.60
KnockoffNets [26]	×	✓	✓	82.50	-	65.70	46.60
Black-Box Ripper [4]	×	✓	✓	82.50	-	**76.50**	**77.90**
DFMS-HL (Ours)	✓	✓	✓	82.52	65.70	76.02	71.36
Victim Accuracy $\sim$ 80%							
ZSDB3KD [37]	✓	✓	✓	79.30	59.46	59.46	59.46
DFMS-HL (Ours)	✓	✓	✓	80.18	**67.03**	**74.27**	**70.57**

Table 3. **Comparison of DFMS-HL with data-free model stealing methods MAZE, DFME (Top) and ZSDB3KD (Bottom) on CIFAR-10**: Clone Accuracy (%) is reported using proxy data from unrelated classes (40 or 10) of CIFAR-100 and synthetic data, with victim models as ResNet34 and ResNet-18 for the top and bottom sections respectively. ResNet18 architecture is used for the clone model.

Method	Hard Label	Black-Box	Data-Free	Victim Accuracy	Synthetic/ Data-Free	CIFAR-100 (40C)	CIFAR-100 (10C)
Victim Accuracy $\sim$ 95.5%, Victim Model: ResNet-34							
MAZE [17]	×	✓	✓	95.50	45.60	-	-
DFME [34]	×	✓	✓	95.50	88.10	-	-
DFMS-HL (Ours)	✓	✓	✓	95.59	84.51	**92.06**	**85.53**
DFMS-SL (Ours)	×	✓	✓	95.59	**91.24**	**93.96**	**90.88**
Victim Accuracy $\sim$ 93.7%, Victim Model: ResNet-18							
ZSDB3KD [37]	✓	✓	✓	93.65	50.18	-	-
DFMS-HL (Ours)	✓	✓	✓	93.83	**85.92**	**90.51**	**83.37**

4.1. Experimental Setup

We evaluate DFMS-HL on two datasets, CIFAR-10 and CIFAR-100. For evaluation, we first train a victim model with the same teacher accuracy as ZSDB3KD [37] for a fair comparison. The victim models are trained until the accuracy reaches the expected value. We evaluate our approach using the following two (Victim, Clone) pairs: (ResNet18, ResNet-18) and (AlexNet, AlexNet-half).

For the clone model training, we use an SGD optimizer with momentum of 0.9, maximum learning rate of 0.1 and a weight decay of 5×10^{-4}. We use a cosine annealed scheduler to decay the learning rate across epochs. For initialization, the clone model is trained for 200 epochs. For the main approach, the clone model is further trained with the images generated from the generator within the query budget or until the accuracy saturates.

For the generator, we use a DCGAN [30] with upto five transpose convolution layers followed by batch-normalization and ReLU units. We use Tanh activation units after the last convolution layer to generate images in the normalised range of $[-1, 1]$. The discriminator contains a stack of five convolution layers followed by batch normalization and Leaky ReLU units. The last layer of the discriminator uses Sigmoid activation. The GAN is trained with an Adam optimizer [20] and a learning rate of 2×10^{-4} with (β_1, β_2) set to $(0.5, 0.999)$.

4.2. Results

Comparison with Knowledge distillation (KD) methods: We compare the proposed approach with existing KD methods on CIFAR-10 in Table 2. DeGAN [1] and ZSKD [25] are data-free KD methods with white-box teacher access, while KnockoffNets [26] and Black-Box Ripper [4] are data-free KD methods in a black-box setting. Similar to the experimental setting of prior works [1, 4], we use images from 40 unrelated classes of CIFAR-100 as the proxy dataset for CIFAR-10 model stealing. We also show results using images from 10 classes randomly sampled from these 40 unrelated classes. We achieve results comparable to the data-free KD methods despite having more restrictions on access to the victim model.

We also show results by using synthetically crafted data for imposing image priors using the discriminator. For this, we generate a synthetic dataset of 50k samples by including random shapes (triangle, rectangle, ellipse or circles) of randomly sampled sizes at random locations on a plain

Table 4. **Performance of DFMS-HL on CIFAR-100**: Clone Accuracy (%) achieved on CIFAR-100 with different proxy data. Victim and Clone architectures are ResNet18.

Method	Proxy Data	Victim Accuracy	Clone Accuracy
DeGAN [1]	CIFAR-10	78.52	75.62
DFMS-HL (Ours)	CIFAR-10	78.52	72.83
DFMS-HL (Ours)	Synthetic	78.52	43.56

Table 5. **Clone model accuracy (%) using DFMS-HL with different proxy datasets**. ResNet-18 architecture is used for both victim and clone models.

Victim training Data:	CIFAR-10					Fashion MNIST
Proxy Data:	SVHN	Data Free	CelebA	Tiny imagenet	Imagenette	CIFAR-10
ZSDB3KD	-	50.18	-	-	-	-
DFMS-HL (Ours)	84.83	**84.51**	85.82	92.26	90.06	81.98

background of random color (details in Section 1.1 of the Supplementary). We also generate textured images by increasing the maximum number of shapes to 100 and reducing the maximum region occupied by the shapes in the image. These images are converted to grey-scale as shown in Fig.3, and further used as proxy data to train the generator. For comparing our results with ZSDB3KD, we train a victim model with a comparable accuracy of 80.18%. From Table 2, it can be observed that our approach not only outperforms ZSDB3KD by a large margin, but also achieves a comparable accuracy with respect to DeGAN and Black-Box Ripper using 40 unrelated classes from CIFAR-100 as the proxy data. We use a significantly lower query budget of 8M when compared to ZSDB3KD which requires 4000M queries. We report the clone model accuracy with other proxy datasets in Table 5. When synthetic data is used, we report our numbers under the "Data-Free" column across all tables since we do not use any additional data in this case. We obtain significant gains when compared to ZSDB3KD across different proxy datasets.

Comparison with Model Stealing methods. We compare our approach with the state-of-the-art data-free Model Stealing approaches [18, 34] on CIFAR-10 in Table 3. We obtain an accuracy of 84.51% by merely using synthetic samples in a completely data-free hard-label setting. We use a lower query budget of 8M, as compared to that of DFME and MAZE that require 20M queries for CIFAR-10. We further extend our attack to the soft-label black-box scenario (denoted as DFMS-SL in Table 3) where the softmax predictions of the victim model are available. In order to utilize the soft labels in a KD setting, we use the L1 loss formulation of DFME [34], which computes the L1 distance between the victim and clone model's logits. Victim model logits are estimated by first taking log of the softmax output, followed by a mean correction. We use the same query budget of 20M and get a boost of almost 3% using synthetic data and proxy data of 10 CIFAR-100 classes.

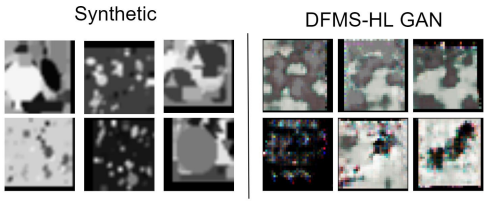

Figure 3. Samples of grey-scale synthetic images shown on the left, along with images generated from the DFMS-HL generator shown on the right.

Results on CIFAR-100. We perform experiments on CIFAR-100 (Table 4) with CIFAR-10 [1, 4] and synthetic data as proxy datasets using a ResNet-18 victim model of accuracy 78.52%. DeGAN attains an accuracy of 75.62% in a soft-label setting with access to the teacher gradients. DFMS-HL reaches a comparably close accuracy of 72.83% using CIFAR-10 as the proxy and 43.56% using synthetic data without any access to the victim model's gradients and only using hard labels. This shows that gradients from the clone model effectively act as a proxy to the victim model's gradients, for training the generator to generate diverse samples across all classes.

5. Ablation Study

Query Budget: We analyze the impact of query budget on the accuracy of the clone model. Our approach achieves a good accuracy within a query budget of 7.6 million using synthetic data as proxy, with AlexNet as the victim model and AlexNet-half as the clone model on CIFAR-10. From Fig. 4 we observe that even with a small query budget of 1.26M, our method performs well and the accuracy saturates within 8M. We report the saturating accuracies in Tables 2 and 3. We use a query budget of 10M for the CIFAR-100 experiments (Table 4) and 8M for the CIFAR-10 experiments (Tables 2 and 3). The class-diversity loss has a huge impact on the clone model accuracy as we observe a significant boost of 6% using this.

Class Diversity Loss: We perform an ablation study by varying the coefficient of the diversity loss from 0 to 1000 in Fig. 5. We use synthetic data as proxy with CIFAR-10 as the original training dataset of the Victim model. We run the ablations for 150 epochs of training, which limits the queries to 7.6M. We find that the clone model accuracy is stable across a wide range of loss coefficients. We set λ_{div} to 500 for CIFAR-10 and 100 for CIFAR-100.

Alternate training of Clone model and generator: The generator and clone model are trained once in every iteration. We check the impact of training each model after every t iterations in Fig. 6. We use synthetic dataset as proxy data and CIFAR-10 as the Victim training dataset, with 85 epochs of training for this ablation. We vary the iteration

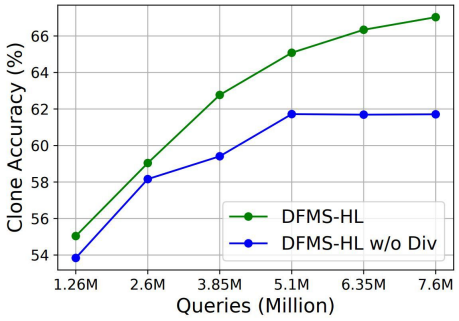

Figure 4. **Query Ablation on CIFAR-10 using synthetic images as proxy data:** Plot of clone model accuracy (%) w.r.t. the number of queries. We achieve a significant boost of 6% by using the class-diversity loss.

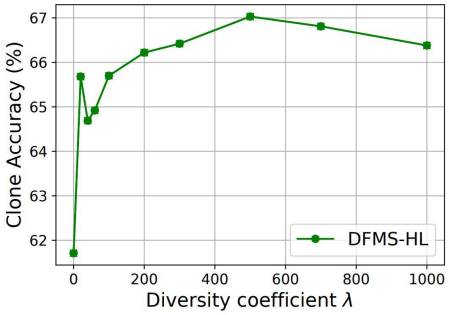

Figure 5. **Sensitivity Plot for Class-diversity Loss:** Clone model accuracy is stable across a wide range of loss coefficients λ_{div}.

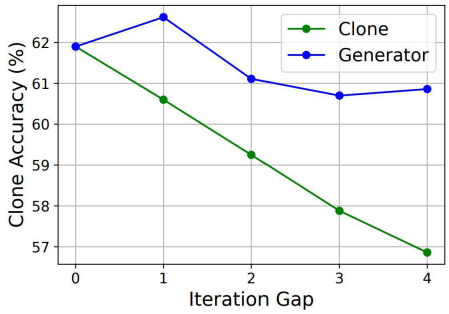

Figure 6. **Iteration Gap ablation:** Clone model accuracy plotted against iteration gap for training the clone and generator.

gap of training each model from 0 to 4. A gap of 0 indicates that the respective model is trained every iteration. The results show that increasing the iteration gap impacts the clone accuracy. We obtain a marginally better accuracy when the generator is trained in alternate iterations. We report our final results with iteration gap set to 0 for both clone model and generator.

Generation of Diverse Images: The DFMS-HL generator is initialized with a DCGAN generator at the start of the training process. As the training progresses, the generator learns to generate images distributed evenly across different classes of the victim model as shown in Fig. 7. We use synthetic images as the proxy data and CIFAR-10 as the Victim model training dataset, with AlexNet/ AlexNet-half as victim/ clone model architectures. The initial distribution of images generated using DCGAN is skewed, with very few samples in classes 1, 4 and 6. The distribution of images without the diversity loss is also skewed. Based on the plots, we note that the class-diversity loss has a huge impact in making the class distribution uniform.

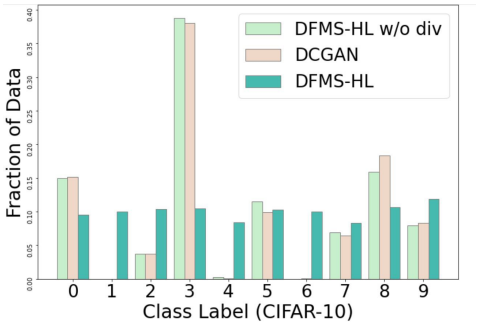

Figure 7. **Distribution of images across classes:** The images generated by DFMS-HL are distributed evenly across all classes.

6. Conclusions

In this paper, we propose an effective model stealing attack in a practical setting of having access to only hard-labels of a black-box victim model. Extensive experiments show that our method DFMS-HL performs better than the state-of-the art model stealing method ZSDB3KD at a $500\times$ lower query budget. We further show that our attack is effective in a completely data-free setting as well, that uses synthetically generated images to impose an image prior. We demonstrate the scalability of the proposed model stealing attack to CIFAR-100 within a low query budget, which has not been attempted in prior works. Our ablations reveal that the class-diversity loss plays a major role in achieving diversity in the generated images, boosting the clone model accuracy evenly across all classes.

Although our work describes methods of attacking the privacy of models through model stealing, the goal is indeed to create better awareness and understanding of the vulnerabilities of Machine Leaning models. This would in turn promote research towards the development of novel defenses against such attacks, leading to a more robust ecosystem with increased security and privacy.

7. Acknowledgements

This work was supported by a project grant from MeitY (No.4(16) /2019-ITEA), Govt. of India and a grant from Uchhatar Avishkar Yojana (UAY, IISC_010), MHRD, Govt. of India. Sunandini Sanyal is supported by Prime Minister's Research Fellowship, and Sravanti Addepalli is supported by Google PhD Fellowship. We are thankful for the support.

References

[1] Sravanti Addepalli, Gaurav Kumar Nayak, Anirban Chakraborty, and Venkatesh Babu Radhakrishnan. DeGAN: Data-enriching gan for retrieving representative samples from a trained classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2, 3, 4, 6, 7

[2] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and B Yoshua. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2015. 2

[3] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing GANs using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019. 2

[4] Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. Black-Box Ripper: Copying black-box models using generative evolutionary algorithms. *arXiv preprint arXiv:2010.11158*, 2020. 2, 6, 7

[5] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020. 3, 5

[6] Varun Chandrasekaran, Hengrui Jia, Anvith Thudi, Adelin Travers, Mohammad Yaghini, and Nicolas Papernot. Sok: Machine learning governance. *arXiv preprint arXiv:2109.10870*, 2021. 3

[7] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2

[8] Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F de Souza, and Thiago Oliveira-Santos. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *International Joint Conference on Neural Networks (IJCNN)*, 2018. 3

[9] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*, 2019. 2

[10] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2, 3, 4

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 2

[14] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. 1

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3

[16] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020. 3

[17] Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Protecting dnns from theft using an ensemble of diverse models. In *International Conference on Learning Representations*, 2020. 2, 3, 6

[18] Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 7

[19] Sanjay Kariyappa and Moinuddin K Qureshi. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2020. 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019. 3

[22] Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. Defending against model stealing attacks using deceptive perturbations. *arXiv preprint arXiv:1806.00054*, 2018. 1, 3

[23] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 2

[24] Smitha Milli, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9, 2019. 3

[25] 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*. PMLR, 2019. 2, 3, 6

[26] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 6

[27] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against dnn model stealing attacks. *arXiv preprint arXiv:1906.10908*, 2019. 3

[28] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. A framework for the extraction of deep neural networks by leveraging public data. *arXiv preprint arXiv:1905.09165*, 2019. 3

[29] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 1, 2, 3

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3, 6

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. 1

[32] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 1

[33] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016. 3

[34] Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4771–4780, 2021. 2, 3, 6, 7

[35] Wenxuan Wang, Bangjie Yin, Taiping Yao, Li Zhang, Yanwei Fu, Shouhong Ding, Jilin Li, Feiyue Huang, and Xiangyang Xue. Delving into data: Effectively substitute training for black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[36] Yixu Wang, Jie Li, Hong Liu, Yan Wang, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Black-box dissector: Towards erasing-based hard-label model stealing attack. *arXiv preprint arXiv:2105.00623*, 2021. 3

[37] Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. *arXiv preprint arXiv:2106.03310*, 2021. 2, 3, 5, 6

[38] Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 690–698, 2020. 2

[39] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 2

[40] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[41] Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Y Lim. Exploiting explanations for model inversion attacks. *arXiv preprint arXiv:2104.12669*, 2021. 1

[42] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[43] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1