# CaBaGE: Data-Free Model Extraction using ClAss BAlanced Generator Ensemble

Jonathan Rosenthal, Shanchao Liang, Kevin Zhang, & Lin Tan

Purdue University, West Lafayette, IN 47906, USA {rosenth0,liang422,zhan4196,lintan}@purdue.edu

**Abstract.** Machine Learning as a Service (MLaaS) is often provided as a pay-per-query, black-box system to clients. Such a black-box approach not only hinders open replication, validation, and interpretation of model results, but also makes it harder for white-hat researchers to identify vulnerabilities in the MLaaS systems. Model extraction is a promising technique to address these challenges by reverse-engineering black-box models. Since training data is typically unavailable for MLaaS models, this paper focuses on the realistic version of it: data-free model extraction. We propose a data-free model extraction approach, CaBaGE, to achieve higher model extraction accuracy with a small number of queries. Our innovations include (1) a novel experience replay for focusing on difficult training samples; (2) an ensemble of generators for steadily producing diverse synthetic data; and (3) a selective filtering process for querying the victim model with harder, more balanced samples. In addition, we create a more realistic setting, for the first time, where the attacker has no knowledge of the number of classes in the victim training data, and create a solution to learn the number of classes on the fly. Our evaluation shows that CaBaGE outperforms existing techniques on seven datasets-MNIST, FMNIST, SVHN, CIFAR-10, CIFAR-100, ImageNet-subset, and Tiny ImageNet—with an accuracy improvement of the extracted models by up to 43.13%. Furthermore, the number of gueries required to extract a clone model matching the final accuracy of prior work is reduced by up to 75.7%.

# 1 Introduction

MLaaS [25] has seen rapid growth, where a provider offers limited access, e.g. via Application Programming Interfaces (API), to a machine learning system at a cost. This is known as a pay-per-query system [12]. Many MLaaS systems are *black-boxes* to the clients. For example, the clients have no knowledge of the model architecture, training method, or the data used to train the model.

Thus, research results built on top of black-box MLaaS models could be difficult to reproduce, validate, or interpret, which harms scientific development. In addition, it is hard for white hat researchers to identify vulnerabilities and issues in such deployed black-box models. These problems faced by MLaaS clients incentivize the development of model extraction techniques [12, 26, 27, 31], i.e.,

techniques that steal the MLaaS models. Such models can be used for reconnaissance to launch further attacks [24, 28].

Existing research focuses on learning-based model extraction [26, 27, 31, 36], the process of using only information gained by querying a black-box model, the *victim*, to train a machine learning system, the *clone*, for application on the same task. However, the victim training data is often inaccessible, and constructing a surrogate dataset for training is a difficult and expensive task [31]. This prompts researchers to borrow ideas from Generative Adversarial Networks (GAN), and use generative models to generate synthetic data for querying the victim model. In this paper, we follow this generative approach and assume the following constraints to make the extraction process "data-free". First, the attacker knows only the victim's input data format and has no further information pertaining to the training data or the target system. The second assumption is that the attacker has no access to any data that can be used in a comparable format for evaluation or training purposes.

Since data-free model extraction is a difficult problem, much of the prior work has focused on more accessible variants of the problem. These often involve either high ( $\geq 8$  million) query budgets or the assumption that the target model offers exact model confidence values, known as soft-label (SL) extraction [14,26,27,31]. Yet, in a pay-per-query system, a high query budget means the technique is prohibitively expensive or impractical. Additionally, any attack relying on exact model confidence values can be countered by the trivial defence of giving only the top-1 or top-k label predictions. To address the countermeasure of providing only the top-k label predictions, previous studies [26, 27, 36] have adopted hard-label (HL) extraction, where the victim model returns only the top-1 label prediction. Prior papers make the assumption that the total number of classes is known.

To make the extraction as realistic as possible, our extractor for HL extraction is assumed to have no knowledge of the number of classes, and must learn the number of classes in the target domain. This *class-agnostic* setting is a stricter, i.e., more realistic, assumption than the class-aware setting used by prior work. To the best of our knowledge, this is the first work where the attacker knows only the input data format (i.e., images and their dimensions) and the general goal of the task, in this case, image classification.

We propose a novel, data-free model extraction approach—CaBaGE—that combines three key techniques: class-balanced difficulty-weighted replay, generator ensemble, and selective query. Class-balanced difficulty-weighted replay balances the class distribution of the replayed samples and leverages priority sampling to keep the most difficult samples when the memory is full. Generator ensemble uses an ensemble of generators with increased generator training iterations to generate more diverse data in data-free model extraction. Selective query is a filtering process to select balanced samples to query the victim.

This paper answers three key questions:

- 1. Does CaBaGE achieve a higher accuracy than existing approaches?
- 2. Does CaBaGE achieve a higher query efficiency than existing approaches?
- 3. How do each of CaBaGE's novel components contribute to the results?

Our work makes the following contributions:

- We propose a novel data-free model extraction approach CaBaGE that combines three key techniques: class-balanced difficulty-weighted replay, generator ensemble, and selective query. CaBaGE generates and selects more diverse, balanced, and higher-quality data for data-free model extraction to achieve higher extracted accuracy with fewer number of queries.
- We create a realistic class-agnostic setting, for the first time, where the attacker has no knowledge of the number of classes in the victim training data, and must instead learn the number of classes on the fly. CaBaGE adaptively modifies the prediction head of the clone models based on the labels obtained from querying the victim model. For all HL evaluations in this paper, CaBaGE uses the more challenging, realistic class-agnostic setting, while existing work uses the class-aware setting.
- Our evaluation shows that in limited-budget settings, CaBaGE outperforms the State-of-The-Art (SoTA) techniques, DisGUIDE and IDEAL, on all seven datasets—MNIST, FMNIST, SVHN, CIFAR-10, CIFAR-100, ImageNetsubset, and Tiny ImageNet. On simpler datasets such as MNIST, FMNIST, and SVHN, CaBaGE improves the final accuracy by up to 43.13%, 37.09%, and 9.04% respectively. On the more complex datasets, CIFAR-100 and ImageNet subset, CaBaGE achieves 11.10% and 26.23% gains in final accuracy.
- For a fair comparison with DisGUIDE, we also evaluate CaBaGE following DisGUIDE's extraction configuration, which assumes a higher query budget. In this setting, CaBaGE's HL extraction performance outperforms the best final accuracy of DisGUIDE on CIFAR-10 and CIFAR-100 by 1.35% and 5.73%. In the SL setting, we observe similar gains, improving final accuracy by 0.34% and 6.49% on CIFAR-10 and CIFAR-100 models respectively. Most significantly, CaBaGE achieves a leap in accuracy to reach 75.96% out of a 77.52% victim, on CIFAR-100 in the SL setting.

### 2 Related Work

### 2.1 Model Extraction

In the context of machine learning, model extraction is a class of attacks whereby an adversary with black-box access to a machine learning system seeks to obtain valuable information of the model, which includes: training hyperparameters, learned parameters, or an approximation of the model with a high agreement over relevant input spaces [23, 30].

Data-Free Model Extraction: In a more challenging scenario of model extraction, the authors of DFME [31] assume that adversaries have no access to initial training data. Instead, DFME trains a substitute (clone) model on synthetic data generated by a GAN-like mechanism, and queries the target model for class predictions to serve as proxy labels. Since the victims are black-box models, DFME employs a forward differences method to approximate the necessary gradients

for clone training. However, this approach requires many queries to estimate gradients, and suffers in HL extractions. Subsequent studies including DFMS [27], DisGUIDE [26], and IDEAL [36] have expanded on DFME's work. DFMS proposes training a GAN to emulate synthetic or real data while maximizing clone label confidence entropy. DisGUIDE introduces the use of replay methods and utilizes a generator training loss that calculates the difference in clone models' prediction to make queries more efficient. IDEAL pushes model extraction further towards low query budget settings by querying generated samples with the highest clone confidence.

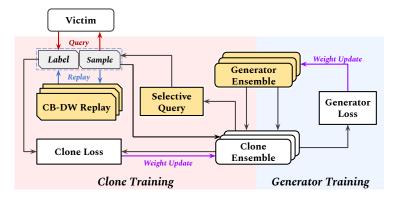
#### 2.2 Model Distillation

Model distillation, also known as knowledge distillation, refers to the transfer the knowledge from a teacher model to a smaller student model [7,9]. Different from model extraction, there is typically white-box access to the teacher model and the emphasis is often on the performance relative to parameter size of the resulting trained model or the amount of arithmetic operations required in the training process. This differs from model extraction, where the attacker only has black-box access to the victim and aims to reduce the number of victim queries to reach high fidelity or accuracy in the targeted domain [12].

Data-Free Model Distillation: In the assumption that the teacher's training data is not accessible, i.e., data-free, some existing works aim to capture the distribution of teacher training data by using information stored in teacher model's layers [20,34]. Other newer approaches borrow ideas from GANs—using a generator to produce training data where the generator's goal is to maximize the disagreement between the teacher and student [4,21]. These approaches attempt to explore and map out the decision boundaries of the teacher to more effectively train the student by querying on the decision boundary. Data-Free Model Extraction borrows these newer ideas, using synthetically generated data to transfer the knowledge of the victims to clones.

### 2.3 Ensemble Learning

Ensemble methods can improve the generalization of neural networks and reduce the high variance properties of the models [5]. The general simplicity in implementation and overall improvement of Ensemble Learning methods has resulted in their use across many different machine learning and deep learning fields [35]. **Ensemble Learning in Adversarial-Learning:** Prior work shows that training a GAN with an ensemble of generators improves performance and the diversity of the generated outputs [6,10]. Existing work in data-free model extraction also utilizes the concept of ensemble learning. Rosenthal et al. leverage an ensemble of clones to improve the stability of clones prediction [26]. Others deploy two generators as an ensemble and optimize the disagreement of the generator outputs, trying to boost the diversity of the generated outputs [33]. In contrast, CaBaGE does not directly compare generator outputs, but relies on the joint



**Fig. 1:** Overview of CaBaGE. Our three novel components are colored in yellow. CB-DW Replay is the Class-Balanced Difficulty-Weighted Replay.

optimization process to incentivize ensemble diversity implicitly. In addition, our approach has a negligible increase in computational cost (details in Sec. 3.1).

# 3 Approach

Fig. 1 is an overview of CaBaGE. CaBaGE' three novel components are highlighted in yellow: Generator Ensemble (Sec. 3.1), Selective Query (Sec. 3.2), and Class-Balanced Difficulty-Weighted Replay (Sec. 3.3). We build CaBaGE upon the foundational method introduced by DisGUIDE [26]. In DisGUIDE, an attacker trains a generator and two clones in an adversarial-like setting and the final extracted model is derived from the ensemble of the clones.

Similar to DisGUIDE, CaBaGE's extraction process is composed of two phases, shown in Fig. 1: (1) Clone Training, and (2) Generator Training. Within the given query budget, CaBaGE cycles between these phases. Before entering any phases, CaBaGE's extraction process starts by initializing the generator ensemble and the clone ensemble from random weights. Afterwards, CaBaGE optimizes the generator ensemble's weights in the Generator Training phase, while keeping the clone ensemble's parameters frozen. Conversely, the Clone Training phase updates the clone ensemble while keeping the generator ensemble fixed.

The Clone Training phase has two stages. First, we apply Selective Query, our proposed filtering technique, to generate samples and select a final batch to query the victim with. The clones then use these samples and the returned labels for training. The samples and labels are then stored in the Class-Balanced Difficulty-Weighted Replay. In the second stage of Clone Training, we sample from the Class-Balanced Difficulty-Weighted Replay and train the clones without querying the victim. We introduce three key improvements in CaBaGE:

1. Generator Ensemble: We propose to train an ensemble of generators instead of a single one. Ensembling the generator prevents mistakes in a single generator from poisoning the sole source of data for the extraction. Compared

to the single generator approach used in DisGUIDE, our proposed solution requires neither additional queries to the victim nor extra computation.

- 2. Selective Query: During Clone Training, we propose a strategy for choosing harder, class-balanced samples to query the victim with. This helps to improve the stability of the extraction without increasing the query budget.
- 3. Class-Balanced Difficulty-Weighted Replay: Prior work generally fails to efficiently utilize the replay method, which is crucial in the DFME context where our only data sources are those we have queried. We thus propose an improvement to the replay memory of prior work [26], by both balancing the class distribution of returned samples and keeping more difficult samples for the clones as memory fills up.

#### 3.1 Generator Ensemble

To prompt diversity in the generated data and a more stable extraction performance in the DFME-HL setting, we propose the use of generator ensembles. Traditionally, only a single generator G is used to capture the entire data distribution  $p_{\text{data}}$ . Instead, we propose the use of  $G_E$ , an ensemble of n generators  $\{G_1, G_2, \ldots, G_n\}$ , to capture the data distribution. In this approach, for a batch of data samples B to be generated, each sample within the batch will be associated with an index and is denoted by  $x_i$ . The batch is partitioned into n sub-batches such that each generator  $G_j$  in the generator ensemble  $G_E$  is responsible for generating samples corresponding to the indices in  $S_j$ , its assigned sub-batch. The batch size remains fixed irrespective of generator ensemble count. Formally, if  $S_j$  represents the set of indices assigned to  $G_j$ , then  $x_i$ , the image output of  $G_j$  for a corresponding input noise vector  $z_i$  is:

$$x_i \leftarrow G_j(z) \text{ for } i \in S_j$$
 (1)

Each time CaBaGE enters the generator training phase, the whole generator ensemble is trained by one batch. Within this batch, each generator model  $G_j$  is trained to maximize both the diversity and clone ensemble disagreement on its own generated image data,  $\{x_i\}_{i\in S_i}$ .

We note that the diversity of the generated data is a global objective shared between ensemble members. If one generator creates too many instances of a given class, other ensemble members can compensate by creating fewer instances of that class. On the other hand, the clone ensemble disagreement on a given sample is independent of other samples in a given batch.

During Clone Training, all generator outputs are combined to produce a full image batch of size |B|, where |B|/n images are produced by each generator in the ensemble. Due to this design, our ensembling approach incurs no extra computational cost in the forward and backward propagation.

This approach benefits from two advantages: Specialization: Each generator specializes in a subset of the data distribution, reducing the complexity of what individual generators must learn. *Diversity*: Multiple generators may enhance the ability to cover the full targeted data distribution and help to mitigate the risk of mode collapse, a common problem faced by GAN-like methods.

### 3.2 Selective Query

To query the victim with more challenging images in *Clone Training*, we introduce a selection process for newly generated data, *Selective Query*. This method is based on the belief that querying the victim with diverse and difficult samples benefits clone training. *Selective Query* is a three-step process:

- 1. Oversampling: Generators produce multiple batches of images.
- 2. Evaluation: The clone ensemble evaluates every sample in the over-sampled data pool. Two primary metrics are computed for each input based on these outputs: The ensemble disagreement loss and the class label.
- 3. Selection: Select an equal number of images from each class, determined by its predicted class label using the averaged class probabilities of the clones, prioritizing those with the highest disagreement loss. This ensures the final selected data is balanced and challenging.

Problems arise when limited or no data generated is predicted to be of certain classes. To address this situation, missing samples are replaced in two steps. First, half the missing samples are chosen by selecting from the remaining images with the highest disagreement loss, regardless of their classes. The other half of the missing samples are drawn uniformly from the residual image samples. Selective Query aligns with the intuition of [13] and [11], in which the generated data is constrained with specific rules at both training and inference time. A detailed algorithm is provided in supplementary materials.

#### 3.3 Class-Balanced Difficulty-Weighted Replay

We draw inspiration from work that is shown to be promising in the Continual Learning domain [3], and create a simple yet effective memory replay. This replay has the following primary features.

- 1. The replay yields class-balanced samples.
- 2. When the allocated space for a particular class is saturated, replacement skews towards easier samples: those with a lower clone training loss.

For each respective class k the attacker discovers, we initialize a separate class memory bank, denoted as  $M_k$ . Samples are stored to  $M_k$  if and only if the victim classifies it as belonging to that class. The maximum capacity of each memory bank is equal to a fixed total memory size divided by the number of classes the attacker has discovered. The class memory banks are managed by a container M which ensures samples are stored correctly and sampled evenly. During training from replay, an equal number of samples from each respective

class known to the attacker are sampled from the class memory banks. This simple improvement reduces class imbalance.

As the memory bank fills up, samples eventually need to be removed to make space for new ones. We follow inspiration from [3]: using a weighted random sampling to select samples for replacement. This aims to keep the most valuable data in storage. We compute the weighting based on the most recent inverse clone training loss value for each respective sample. For each batch of samples we update, we apply a transformation where we subtract loss from the maximum loss value within the batch. In this way, samples that are harder for the clones to learn are more likely to be retained for longer in the memory bank.

To the best of our knowledge, previous works on data-free model extraction that employ a memory bank for experience replay rely on simpler methods, such as a circular buffer for memory storage or random sampling strategies [2, 26], which do not consider the importance of class balance or the difficulty of samples.

#### 3.4 Loss Functions

The equation below describes the clone training loss  $L_C$  for clone  $c_i$  in the HL setting. K represents the classes we have discovered up to this point of CaBaGE's extraction process, and  $c_i(s_n)_k$  represents the logit of clone  $c_i$  on class k for generated sample  $s_n$ . Finally,  $p_V$  is label assigned by querying the victim.

$$HL: L_{C} = -\frac{1}{N} \sum_{n=1}^{N} \log \left( \frac{\exp(c_{i}(s_{n})_{p_{V}})}{\sum_{k \in K} \exp(c_{i}(s_{n})_{k})} \right)$$
(2)

For SL extraction, the clone training loss is the MSE loss computed between the pseudo logits of the victim and the clones' raw logits. Here, we follow DFME's approach to obtain the pseudo logits  $V(s_n)$  of the victim [31].

$$SL: L_C = -\frac{1}{N} \sum_{n=1}^{N} (c_i(s_n) - V(s_n))^2$$
 (3)

CaBaGE's generator training's optimization goal follows previous work, in which the generator jointly optimizes the disagreement loss  $L_D$ , which aims to maximize the disagreement between clone models, and the class diversity loss  $L_{div}$ , aiming for a balanced data distribution in the generated data.  $\lambda$  is a hyperparameter used as a weighting coefficient for the class diversity loss.

$$L_G = L_D + \lambda L_{div} \tag{4}$$

Following previous work [26] and [1], the disagreement loss  $L_D$  is the standard deviation of the fixed clone models' prediction over all previously discovered classes, and  $L_{div}$  is the information entropy of the clones' prediction.

### 4 Experimental Setup

To compare with prior work in data-free model extraction, we follow their experimental configurations. Specifically, we examine two settings, the first from

IDEAL [36], and the other from DisGUIDE [26]. We denote the settings from IDEAL, where query budgets are much lower, as the *limited-budget setting*  $(\leq 2M \text{ queries})$ , and the setting from DisGUIDE as the relaxed-budget setting (> 8M queries). We perform extraction from the following seven datasets: MNIST [18], FMNIST [32], SVHN [22], CIFAR-10 [15], CIFAR-100 [15], Tiny ImageNet [17] and an ImageNet-subset [19]. For each dataset, we extract 2 or 3 victim architectures, dependent on the dataset. The list of model architectures is: MLP, LeNet [18], AlexNet [16], VGG-16 [29], ResNet-18 [8], and ResNet-34 [8]. We use IDEAL's published implementations of MLP, LeNet, AlexNet, ResNet-18, and ResNet-34 architectures [36], and DFME's published implementation for VGG-16 [31]. To eliminate any potential biases, we run all extraction techniques on the same victim models for each setting in which we make comparisons. The victims and their training details are specified in the supplementary materials. Evaluation Metric: To compare with prior work in DFME [26, 36], we focus our evaluation on accuracy. Since researchers are also interested in fidelity results [12], we refer readers to the supplementals for CaBaGE's fidelity results.

#### 4.1 Limited-Budget and Relaxed-Budget Setting

In the limited-budget setting, we compare three techniques: IDEAL, DisGUIDE, and CaBaGE. Following IDEAL's settings, we use 25K queries for extractions on MNIST, 100K for FMNIST and SVHN, 250K for CIFAR-10 and ImageNetsubset, and 2M for CIFAR-100 and Tiny ImageNet. Tab. 1 reports the average final accuracies obtained by each respective method. On CIFAR-100 and Tiny ImageNet the clone model architecture is ResNet18, while in all other cases the clone architecture is congruent to the victim model architecture.

For reproducibility, we use the publicly accessible repositories of IDEAL¹ and DisGUIDE². However, there is a discrepancy between the query budget definitions in the IDEAL paper [36] and its code base. IDEAL's code base queries the victim model with multiple different images, that are augmented versions of each other, but the IDEAL paper only counts these multiple queries as a single query. Due to this discrepancy, IDEAL's extraction experiments query the victim model multiple times more than the reported query budget in their paper. These additional queries represent a budget increased by a factor of 2 times for non-CIFAR or 162 times for CIFAR datasets. To ensure an equitable comparison in the limited-budget setting, we report reproduced results of IDEAL without mutating the stored images to make the query budget constraint consistent with what is described in the IDEAL paper. We denote this query budget adjusted version of IDEAL as IDEAL\*. The details of this discrepancy and IDEAL\* are provided in the supplementary materials.

In the relaxed-budget setting, we compare CaBaGE with DisGUIDE in both SL and HL extraction on CIFAR-10 and CIFAR-100 datasets. The experiments follow DisGUIDE's exact experimental settings, details can be found in [26].

<sup>1</sup> https://github.com/SonyResearch/IDEAL/tree/main

<sup>&</sup>lt;sup>2</sup> https://github.com/lin-tan/disguide

### 4.2 Hyper-Parameters

In a realistic setting for DFME, the attacker is only aware of the model architecture they have selected to replicate the target system, as well as the query budget. Thus, most hyper-parameters should be identical across all experiments. Consequently, we choose to fix the learning rate for clone models with the same architecture and with the same query budget. We set the learning rate for AlexNet clones to be fixed at 0.004, VGG-16 clones at 0.01, ResNet18 clones at 0.03, and ResNet34 clones at 0.1, respectively, for all query budgets. For MLP clones, learning rates are 0.01 (25K queries) and 0.0125 (100K queries). For LeNet clones, they are 0.1 (25K queries) and 0.01 (100K queries).

For all experiments, CaBaGE uses a batch size of 250. We use a clone ensemble size of 2, and a generator ensemble size of 8. Within each Generator Training phase, we train the generator for 3 batches. In the Clone Training phase, Selective Query selects a batch from 1000 samples to query the victim with, and the clone models are trained for 1 iteration using the newly queried batch. The clone model is trained with 12 batches of data sampled from the knowledge replay. Learning rates are scheduled to drop by 0.3 at 40% and 80% of the total query budget under the relaxed-budget setting. Learning rate drops were not used in the limited-query budget setting. Additionally, as the diversity loss weighting should increase based on the number of discovered classes [26], we dynamically scale  $\lambda$  during training, varying it approximately inversely with the number of discovered classes via a simple relation:  $\lambda = \frac{4}{(10+K)}$ . For all experiments, we follow DisGUIDE [26], and use a replay size of 1 Million.

# 5 Results

#### 5.1 Extraction Accuracy

We compare the performance of our method, CaBaGE, with two SOTA data-free model extraction techniques: DisGUIDE [26] and IDEAL [36]. IDEAL emphasizes extraction under stringent query budget constraints, while DisGUIDE aims at high-performance extraction with a more generous budget. We report the accuracy in extraction settings matching both of these prior papers respectively.

Limited-Budget Setting Tab. 1 shows the extraction results of IDEAL\*, DisGUIDE, and CaBaGE in terms of final extracted accuracy and 95% confidence interval under the limited-budget setting, described in Sec. 4.1. IDEAL\* is the query budget adjusted version of IDEAL, for a fair comparison as described in Sec. 4.1. For example, when extracting the MLP victim trained on MNIST, CaBaGE reaches 57.13% accuracy on the test set, outperforms the prior best extraction result of 14.00% by 43.13%. Consistently, CaBaGE outperforms prior work on all victims, on any target dataset.

For elementary victims trained on less intricate datasets, such as MNIST and FMNIST, CaBaGE demonstrates significant performance improvements, improving the final test-set accuracy by an average of 23.00% for AlexNet, 6.27%

**Table 1:** Accuracy (%) and 95% confidence interval of clone in various limited-budget settings. Experiments result for DisGUIDE and CaBaGE are computed over 3 runs. IDEAL\*'s result are run by a fixed random seed based on the published code. IDEAL\* is the query budget adjusted version of IDEAL for a fair comparison (Sec. 4.1).

Dataset	Model	Victim	$IDEAL^*$	DisGUIDE	CaBaGE
MNIST	MLP LeNet AlexNet	98.25 99.27 99.35	14.00 86.40 66.30	$11.35 \pm 0.00$ $91.23 \pm 2.01$ $18.11 \pm 5.23$	$egin{array}{c} {\bf 57.13} \pm 4.15 \ {\bf 94.49} \pm 0.81 \ {\bf 85.31} \pm 0.85 \end{array}$
FMNIST	MLP LeNet AlexNet	84.54 90.23 92.66	19.00 27.80 35.20	$37.53 \pm 5.75$ $59.43 \pm 2.86$ $54.47 \pm 5.29$	$74.62 \pm 3.15$ $68.72 \pm 2.39$ $81.45 \pm 0.64$
SVHN	VGG-16 ResNet-18 AlexNet	94.41 95.28 89.82	68.35 72.60 67.00	$79.96 \pm 3.21$ $75.83 \pm 0.77$ $19.39 \pm 6.70$	$84.10 \pm 0.36$ $78.09 \pm 0.84$ $76.04 \pm 2.42$
CIFAR-10	AlexNet ResNet-34	84.76 93.85	25.50 20.40	$24.73 \pm 3.23$ $18.05 \pm 5.97$	$33.35 \pm 1.63$ $26.03 \pm 2.58$
CIFAR-100	AlexNet ResNet-34	63.38 $77.52$	$6.17 \\ 7.94$	$24.45 \pm 0.38$ $32.65 \pm 0.64$	$33.09 \pm 0.66$ $43.75 \pm 1.75$
ImageNet-subset	AlexNet VGG-16	72.96 $78.53$	$20.60 \\ 20.50$	$18.92 \pm 0.91$ $31.01 \pm 2.09$	$46.83 \pm 3.06$ $37.04 \pm 7.60$
Tiny Imagenet	ResNet-34 VGG-16	$59.28 \\ 42.04$	4.93 4.15	$11.87 \pm 2.73$ $7.87 \pm 1.50$	$15.36 \pm 0.64$ $11.98 \pm 0.64$

**Table 2:** Final clone accuracy comparison with 95% confidence intervals in the relaxed-budget setting. DisGUIDE [26] results are the paper reported accuracies.

Setting	Technique	CI	FAR-10	CI	CIFAR-100		
<b>6</b>			Clone (%	) Victim	Clone (%)		
Soft Label	DisGUIDE CaBaGE	00.0-	$94.02 \pm 0.2$ $94.36 \pm 0.0$		$69.47 \pm 0.88$ <b>75.96</b> $\pm 0.25$		
Hard Label	DisGUIDE CaBaGE		$87.93 \pm 1.7$ $89.28 \pm 0.6$		$58.72 \pm 2.42$ $64.45 \pm 0.87$		

for LeNet, and 40.11% for MLP. More specifically, CaBaGE extracts a LeNet model trained on MNIST with an averaged final accuracy of 94.49%, which is very close to the victim model's accuracy of 99.27%. On slightly more complex datasets including SVHN and CIFAR-10, CaBaGE also exhibits improvements over previous SOTA. Extraction on SVHN datasets achieves on average a 5.15% increase, reaching 79.41%, and extraction on CIFAR-10 achieves on average a 6.74% increase, reaching 29.69%. Similar improvements are also shown in extracting victim models trained on more challenging datasets. Namely, on the ImageNet-subset, CIFAR-100, and Tiny ImageNet datasets, we observe mean improvements of 16.13%, 9.87%, and 3.80% respectively.

**Summary**: In the limited-budget setting, CaBaGE outperforms both Dis-GUIDE and IDEAL\* across all 17 settings, increasing accuracy up to 43.13%.

**Table 3:** Mean number of queries (in millions) to reach prior work—DisGUIDE's reported final accuracies with respective 95% confidence intervals. Lower is better.

Setting	CII	FAR-10	CIFAR-100		
<b>-</b>	DisGUIDE	CaBaGE	DisGUIDE	CaBaGE	
Soft Label	20M	$\mathbf{16.04M}\pm0.01\mathrm{M}$	10M	$\mathbf{2.43M}\pm0.15\mathrm{M}$	
Hard Label	8M	$6.32M \pm 0.13M$	10M	$6.07M \pm 1.81M$	

Relaxed-Budget Setting Tab. 2 compares the final accuracy of CaBaGE on CIFAR-10 and CIFAR-100 following DisGUIDE configurations in the relaxed-budget setting for both SL and HL extractions. For SL extractions, when extracting from the ResNet-34 victim trained on CIFAR-100, CaBaGE outperforming DisGUIDE by 6.49%, achieving a final accuracy of 75.96% on the test set, which is 97.99 percent of the victim model's accuracy (77.52%). This underscores that with a higher query budget, CaBaGE offers highly accurate SL model extraction, even on more intricate models trained with complex datasets. CaBaGE also increases the final extracted accuracy on CIFAR-10 by 0.34%. Similarly, in the HL setting, the test-set accuracies are increased by 1.35% and 5.73% for extraction processes on CIFAR-10 and CIFAR-100, reaching 89.28% and 64.45% accuracy respectively. Besides the gains of final accuracy of the extracted models, CaBaGE also makes the extraction process more stable, i.e., reduces the fluctuation of results, when the query budget is relaxed.

**Summary**: Under the relaxed-budget setting, CaBaGE outperforms prior work on CIFAR-10 and CIFAR-100. In the HL setting, CaBaGE improves the final test-set accuracy by 1.35% and 5.73%. Our biggest gain is in the SL setting, where CaBaGE achieves an accuracy improvement of 6.49%, reaching 97.99% of the victim model's accuracy on CIFAR-100.

#### 5.2 CaBaGE's Query Efficiency

We compare the number of queries required for CaBaGE to reach the final accuracy of the prior work DisGUIDE against its query budgets, and report the result in Tab. 3. In all extraction settings, CaBaGE shows better query efficiency compared to DisGUIDE. For SL extraction on CIFAR-100 CaBaGE needs only 2.43 million queries on average to achieve similar accuracy to DisGUIDE's final result, with a 95% confidence interval of 0.15 million, reducing the prior standard by 75.7%. For SL extraction on CIFAR-10 the number of queries needed to reach prior work's best is reduced by 3.96 million on average, 19.8% of the total queries.

In the HL settings, CaBaGE reaches the prior SOTA accuracy in 6.32 million queries on CIFAR-10 and with 6.07 million queries on CIFAR-100. This constitutes a reduction in the needed queries by 21% and 39.3%, respectively.

**Summary**: CaBaGE consistently improves query efficiency in both the CIFAR-10 and CIFAR-100 datasets under SL and HL settings. Extractions on the CIFAR-100 dataset, in particular, show a reduction in the number of required queries in the SL setting by **75.7**%.

#### 5.3 Ablation Study

**Table 4:** Final Accuracy of CaBaGE and ablation settings shown with 95% confidence interval. Independent t-tests are used to determine improvement over the baseline DisGUIDE\*, where statistically significant improvements are highlighted in Blue.

Dataset	Model	$\mathbf{DisGUIDE}^*$	$\mathbf{DisGUIDE}^* + \mathbf{CB-DW}$	$\mathbf{DisGUIDE}^*$ +CB-DW+GE	CaBaGE
MNIST	MLP	$11.35 \pm 0.00$	$48.58 \pm 5.73$	$54.32 \pm 4.04$	54.94 ±3.58
	LeNet	$94.20 \pm 0.55$	$94.02 \pm 0.88$	$94.84 \pm 1.07$	95.32 ±0.63
	AlexNet	$74.10 \pm 2.11$	$71.64 \pm 4.31$	$82.54 \pm 3.30$	86.08 ±0.94
FMNIST	MLP	$42.28 \pm 4.40$	$72.80 \pm 2.85$	$76.61 \pm 2.35$	$74.45 \pm 3.28$
	LeNet	$69.12 \pm 1.12$	$67.10 \pm 1.83$	$70.29 \pm 1.58$	$68.08 \pm 1.50$
	AlexNet	$74.66 \pm 2.09$	$75.76 \pm 2.19$	$78.86 \pm 1.32$	$79.63 \pm 1.48$
SVHN	VGG-16	$88.37 \pm 0.87$	$86.67 \pm 2.12$	$83.63 \pm 1.32$	84.14 ±2.09
	ResNet-18	$76.05 \pm 1.81$	$77.87 \pm 1.52$	$79.49 \pm 1.18$	78.70 ±0.98
	AlexNet	$64.01 \pm 2.40$	$72.27 \pm 2.89$	$72.35 \pm 4.06$	74.46 ±2.19
CIFAR-10	AlexNet	$26.25 \pm 3.45$	$30.09 \pm 1.82$	$35.66 \pm 1.44$	$33.23 \pm 1.04$
	ResNet-34	$22.04 \pm 2.64$	$28.78 \pm 4.10$	$28.25 \pm 2.98$	$26.34 \pm 2.54$
ImageNet-Subset	AlexNet	$43.11 \pm 3.17$	$46.67 \pm 2.07$	$46.54 \pm 2.88$	$48.54 \pm 2.68$
	VGG-16	$38.94 \pm 5.39$	$27.13 \pm 4.37$	$39.57 \pm 8.13$	$35.53 \pm 5.97$

We evaluate the individual contribution of Class-Balanced Difficulty-Weighted Replay (CB-DW), Generator Ensemble (GE), and Selective Query by progressively adding them to the baseline, DisGUIDE\*, which only differs from DisGUIDE by incrementing the replay iteration from 3 to 12 to match CaBaGE' for a fair comparison. We report the averaged extraction accuracy with a 95% confidence interval in Tab. 4. Column  $DisGUIDE^*_{+CB-DW}$  represents DisGUIDE\* with CB-DW. Other columns define similar ablations of our technique or our full approach. We run each experiment 9 times and evaluate the statistical significance using independent t-tests against DisGUIDE\*.

With the inclusion of more components over the baseline method, there is an increase in the number of settings showing statistically significant improvements over the baseline. Column  $DisGUIDE^*_{+CB-DW}$  shows that adding Class-Balanced Difficulty-Weighted Replay results in four statistically significant improvements over the baseline while only adversely affecting one extraction result. Likewise, Column DisGUIDE $^*_{+CB-DW+GE}$  shows that Generator Ensembles with Class-Balanced Difficulty-Weighted Replay further increase the improved cases to 8 statistically significant improvements. Finally, our comprehensive approach, CaBaGE, achieves statistically significant improvement in 10 out of 13 extraction settings, with only one setting where the performance is reduced, demonstrating the effectiveness of Selective Query.

**Summary**: Each of the three CaBaGE components, i.e., Class-Balanced Difficulty-Weighted Replay, Generator Ensemble, and Selective Query, improves model extraction accuracy.

**Table 5:** Evaluation of the impact Generator Ensemble Size on computation cost. Extraction result performed by  $DisGUIDE^*_{+GE}$  following limited-budget setting on the AlexNet victim trained on CIFAR-10. We run each experiment setting 10 times.

Metric	Gener	Generator Ensemble Size				
	1	4 8				
Accuracy (%) Time (seconds)		$28.56 \pm 2.79$ $240.71 \pm 1.10$				

#### 5.4 GE's Impact on Computational Cost

We show in Tab. 5 that under a fixed generator training iteration, the increase in Generator Ensemble size does not lead to a noticeable increase of our implementation's runtime. Increasing the generator size from 1 to 4 results in a 6.76% increase in final accuracy, yet the time required for extraction remains approximately the same. In fact, the time is reduced by 6 seconds, which we believe is due to system randomness. This evaluation was performed on an NVIDIA GeForce RTX 2080 Ti with 11 GB of memory and an Intel(R) Xeon(R) Gold 5120 CPU.

### 6 Conclusion, Limitations, and Future Work

We propose a data-free model extraction approach, CaBaGE, which utilizes a combination of generator ensemble, class-balanced difficulty-weighted replay, and selective query. This approach enhances the accuracy and efficiency of model extraction results. In addition, CaBaGE works in the more strict class-agnostic setting across seven different datasets and six victim model architectures.

One limitation is that there is no easy way to tune hyperparameters in a DFME environment. CaBaGE tries to use a robust set of parameters between experiment settings, varying only model learning rate between clone architectures and query budgets. However, the chosen settings may not generalize to untested datasets and architectures. Coming up with ways for the attacker to dynamically verify chosen hyperparameters is a challenging problem that needs to be solved for real world data-free model extraction to work.

Machine learning is a very broad field and many types of problems exist. Both prior work and CaBaGE have naturally focused on a set of image data tasks, in order to be able to compare with one another. It remains to be seen how well current DFME SOTAs generalize to extracting models trained on class imbalanced data and non-image data.

In addition, the attackers may have some domain knowledge about the training data. One promising future work direction is to utilize such domain knowledge effectively to improve mode extraction accuracy and efficiency.

### 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., Radhakrishnan, V.B.: Degan: Dataenriching gan for retrieving representative samples from a trained classifier. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3130– 3137 (2020) 8
- Binici, K., Pham, N.T., Mitra, T., Leman, K.: Preventing catastrophic forgetting and distribution mismatch in knowledge distillation via synthetic data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 663–671 (January 2022) 8
- 3. Buzzega, P., Boschini, M., Porrello, A., Calderara, S.: Rethinking experience replay: a bag of tricks for continual learning. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 2180–2187. IEEE Computer Society, Los Alamitos, CA, USA (jan 2021). https://doi.org/10.1109/ICPR48806.2021.9412614, https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412614, 7, 8
- Fang, G., Song, J., Shen, C., Wang, X., Chen, D., Song, M.: Data-free adversarial distillation. arXiv preprint arXiv:1912.11006 (2019) 4
- Ganaie, M., Hu, M., Malik, A., Tanveer, M., Suganthan, P.: Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence 115, 105151 (2022). https://doi.org/https://doi.org/10.1016/j.engappai.2022.105151, https://www.sciencedirect.com/science/article/pii/S095219762200269X 4
- Ghosh, A., Kulharia, V., Namboodiri, V., Torr, P.S., Dokania, P.K.: Multi-agent diverse generative adversarial networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8513-8521. IEEE Computer Society, Los Alamitos, CA, USA (jun 2018). https://doi.org/10.1109/CVPR. 2018.00888, https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00888 4
- Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision 129(6), 1789–1819 (2021) 4
- 8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 9
- 9. Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015) **4**
- Hoang, Q., Nguyen, T.D., Le, T., Phung, D.: MGAN: Training generative adversarial nets with multiple generators. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rkmu5b0a-4
- 11. Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.: Harnessing deep neural networks with logic rules. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2410–2420. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1228, https://aclanthology.org/P16-1228 7
- Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., Papernot, N.: High accuracy and high fidelity extraction of neural networks. In: 29th USENIX security symposium (USENIX Security 20). pp. 1345–1362 (2020) 1, 4, 9
- Jiang, N., Lutellier, T., Lou, Y., Tan, L., Goldwasser, D., Zhang, X.: Knod: Domain knowledge distilled tree decoder for automated program repair. In: Proceedings of the 45th International Conference on Software Engineering. p. 1251–1263. ICSE '23, IEEE Press (2023). https://doi.org/10.1109/ICSE48619.2023.00111, https://doi.org/10.1109/ICSE48619.2023.00111

- Kariyappa, S., Prakash, A., Qureshi, M.K.: 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). pp. 13814–13823 (June 2021)
- 15. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. Rep. 0, University of Toronto, Toronto, Ontario (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf 9
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012) 9
- 17. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015) 9
- Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791 9
- 19. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: Training clean models on poisoned data. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 14900-14912. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper\_files/paper/2021/file/7d38b1e9bd793d3f45e0e212a729a93c-Paper.pdf 9
- Lopes, R.G., Fenu, S., Starner, T.: Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535 (2017) 4
- 21. Micaelli, P., Storkey, A.J.: Zero-shot knowledge transfer via adversarial belief matching. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 9551–9561. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/9151-zero-shot-knowledge-transfer-via-adversarial-belief-matching.pdf 4
- 22. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011), http://ufldl.stanford.edu/housenumbers/nips2011\_housenumbers.pdf 9
- 23. Oliynyk, D., Mayer, R., Rauber, A.: I know what you trained last summer: A survey on stealing machine learning models and defences. arXiv preprint arXiv:2206.08451 (2022) 3
- 24. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. pp. 506–519 (2017)
- 25. Ribeiro, M., Grolinger, K., Capretz, M.A.: Mlaas: Machine learning as a service. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). pp. 896–902. IEEE (2015) 1
- 26. Rosenthal, J., Enouen, E., Pham, H.V., Tan, L.: Disguide: Disagreement-guided data-free model extraction. Proceedings of the AAAI Conference on Artificial Intelligence 37(8), 9614-9622 (Jun 2023). https://doi.org/10.1609/aaai.v37i8.26150, https://ojs.aaai.org/index.php/AAAI/article/view/26150 1, 2, 4, 5, 6, 8, 9, 10, 11, 19, 24
- 27. Sanyal, S., Addepalli, S., Babu, R.V.: Towards data-free model stealing in a hard label setting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15284–15293 (2022) 1, 2, 4

- 28. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 3–18 (2017). https://doi.org/10.1109/SP.2017.41 2
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
- 30. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction APIs. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 601-618. USENIX Association, Austin, TX (Aug 2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer 3
- Truong, J.B., Maini, P., Walls, R.J., Papernot, N.: Data-free model extraction.
   In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021) 1, 2, 3, 8, 9
- 32. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017) 9
- 33. Yang, E., Wang, Z., Shen, L., Yin, N., Liu, T., Guo, G., Wang, X., Tao, D.: Continual learning from a stream of apis (2023) 4
- Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8715–8724 (2020) 4
- 35. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE Transactions on Neural Networks and Learning Systems 23, 1177–1193 (2012). https://doi.org/10.1109/TNNLS.2012.2200299 4
- 36. Zhang, J., Chen, C., Lyu, L.: IDEAL: Query-efficient data-free learning from black-box models. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=ConT6H7MWL 2, 4, 9, 10

# A Fidelity Results

In the Data-Free setting, prior works generally focus on replicating model performance. In this section we show the performance of CaBaGE in terms of fidelity, i.e. how well the extracted model matches the victim model, as opposed to the true test label.

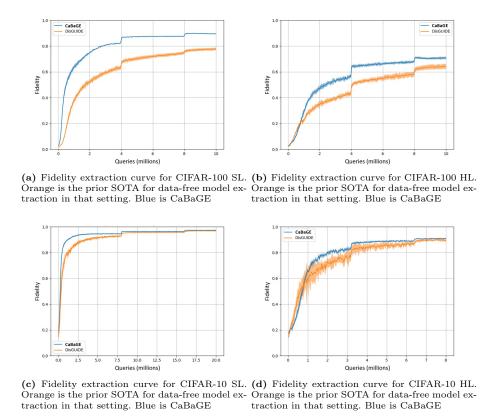


Fig. 2: Fidelity extraction curves for CIFAR-10 and CIFAR-100

Fig. 2 compares the fidelity extraction curves of the prior SOTA with CaBaGE. In the next section, we present the accuracy curves of the same runs in Fig. 3. The final fidelity values for these runs can be found in Tab. 6. Subjectively the accuracy and fidelity results look to be inline with one another, with fidelity values being higher than accuracy.

**Table 6:** Final clone fidelity along with 95% CI. Results from DisGUIDE are from reproduced runs with their codebase.

Setting	Technique	CIFAR-10			CIFAR-100		
		Victim	Clone	(%)	Victim	Clone (%)	
Soft Label	DisGUIDE CaBaGE		96.82 ± <b>97.32</b> ±			$77.64 \pm 1.04$ <b>89.56</b> ± 0.19	
Hard Label	DisGUIDE CaBaGE		89.38 ± <b>90.98</b> ±			$63.91 \pm 1.94$ <b>70.54</b> $\pm 1.02$	

# **B** Accuracy Training Curves

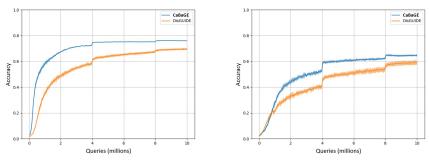
In section 5.2 we follow prior work in terms of quantifying the queries needed to reach the prior SOTA accuracy in different settings. Not limited in terms of space, we offer the reader accuracy training plots here.

The plots in Fig. 3 are in the relaxed budget settings with the exact runs used to generate the CaBaGE main results compared with reproduced runs of DisGUIDE with results inline with the numbers reported in the paper.

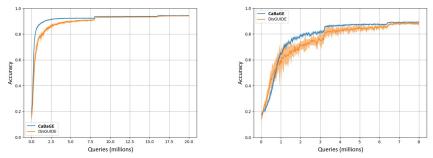
# C Effect of Replay Iteration

DisGUIDE incorporated a circular buffer mechanism for retaining previously queried samples, in order to optimize the query budget utilization. The paper suggested an enhancement in performance with increased replay frequencies, a concept analogous to the frequency of memory bank updates in related literature [26]. Our analysis presents a more nuanced perspective on this claim. We evaluated the performance dynamics of DisGUIDE's replay against our proposed replay strategy across varying settings in two distinct model extraction scenarios: a multi-layer perceptron on the MNIST dataset and VGG-16 on the SVHN dataset. Both methods employed the same extraction technique derived from DisGUIDE, with the replay strategy being the sole variable.

Figure 4 demonstrates the performance trajectories of the two replays as a function of replay iterations, with the error bars indicating a 95% confidence interval. The upper chart demonstrates that the performance of CaBaGE's replay consistently exceeds that of DisGUIDE and exhibits a positive correlation with replay iterations when extracting the MLP victim on MNIST. Conversely, DisGUIDE's performance first increases then declines with additional iterations. In the lower chart, Nemesis replay marginally trails behind DisGUIDE's replay, and a uniform decrement in performance for both methods is observed as the replay iteration count escalates. These findings indicate that the influence of replay strategies is not uniform across datasets. It is imperative for future research in model extraction to leverage experience replay to scrutinize the differential impacts of replay modalities relative to the victim model and dataset characteristics.



(a) Accuracy training curve for CIFAR-100 SL (b) Accuracy training curve for CIFAR-100 HL extraction. Orange is the prior SOTA for data-free model extraction in that setting. Blue is free model extraction in that setting. Blue is CaBaGE CaBaGE



(c) Accuracy training curve for CIFAR-10 SL ex- (d) Accuracy training curve for CIFAR-10 HL traction. Orange is the prior SOTA for data-free extraction. Orange is the prior SoTA for data-free model extraction in that setting. Blue is CaBaGE model extraction in that setting. Blue is CaBaGE

Fig. 3: Accuracy extraction curves for CIFAR-10 and CIFAR-100

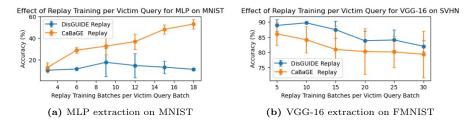


Fig. 4: Performance comparison of DisGUIDE's replay and CaBaGE, with different replay iterations used

# D Victim Training Details

CIFAR-10 and CIFAR-100 ResNet-34 models were taken from the DFAD paper. Due to the difficulty in acquiring the exact trained victim models from the IDEAL paper, new victim models were trained for the purpose of running the CaBaGE experiments. Aside from the DFAD models, all other victims were trained with the script provided in the codebase in run train teacher.sh

Models were trained via 240 epochs of stochastic gradient descent with a batch size of 250, momentum of 0.9, and an initial learning rate of 0.1. If the training run failed (model weights did not converge) the training was repeated with a smaller initial learning rate, until no such issues occurred. After every 40 epochs the learning rate was divided by 5. From within the training runs, models were selected to be close to the accuracies reported in the IDEAL paper, wherever possible.

# E Reproducing IDEAL's result

As described in section 4.1, there is a discrepancy between the IDEAL codebase and paper. Specifically, IDEAL's extraction technique queries the victim many times with augmented versions of generated images. The transformation function from the codebase is given in the following code snippet:

For non-CIFAR datasets, the horizontally flipped versions of the inputs are used to query the victim, but only 1 query is counted instead of 2. For CIFAR-10 and CIFAR-100, this is pushed further by also performing a random crop on generated images with a padding of 4. The random crop function adds padding to each side of an image and then randomly selects an section to match the dimensions specified by the first parameter. This means that vertically and horizontally there are 9 different possible outcomes (prepend 1-4 zeros, no change, or append 1-4 zeros). Combined with the horizontal flipping, this gives us  $9 \times 9 \times 2 = 162$  different versions of each generated input, each of which the victim model may be queried with.

To remedy this issue, the transform is changed to the simply remove the transform, which is not mentioned in the paper. The code section in question after the fix is as follows:

# F Selective Query Specifics

The Selective Query pseudocode can be found in **Algorithm 1** Selective Query.

# Algorithm 1 Selective Query

```
Input: S = \{S_{i \in m}\}, m \text{ batches of generated images; } C(S), Clone ensemble's pre-
dictions on S; K, discovered classes; N, batch size.
Output: B, selected data
B \leftarrow [\ ]
N_k \leftarrow \lfloor \frac{N}{K} \rfloorR \leftarrow N - K \cdot N_k
                                                          Expected number of samples per class
                                                                        Number of missing samples
for k in K do
     {S}_k \leftarrow \text{select images in } S \text{ with prediction } k \text{ in } C(S)
     \{S_k\}^{sorted} \leftarrow sort(\{S\}_k): Sort in descending order by corresponding value in
\sigma(C(S))
    if |\{S\}_k| \geq N_k then
         add \{S_{kj}\}_{j\leq N_k}^{sorted} to B
    else
         add \{S_k\}^{sorted} to B
         R \leftarrow R + N_k - |\{S\}_k|
    end if
end for
if R > 0 then
    S^{*sorted} = sort(\{S \setminus B\}) : Sort by decreasing value in \sigma(C(S))
    add \{S_i^{*sorted}\}_{i \leq \lfloor \frac{R}{2} \rfloor} to B
    add uniformly sampled samples from remaining to B
end if
return B
```

# G Stability Against Hyper-parameters

We study the impact of hyper-parameters, including the generator ensemble size  $|G_e|$ , and the generator training iterations per victim query  $g_{iter}$ .

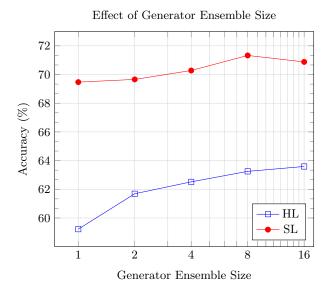


Fig. 5: Final accuracy comparison for ResNet34 extraction on CIFAR-100 under the relaxed-budget setting. Method tested is DisGUIDE with only the addition of a generator ensemble. Effect of varying the ensemble size.

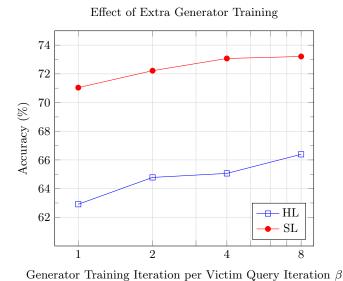


Fig. 6: Final accuracy comparison for ResNet34 extraction on CIFAR-100 under the relaxed-query setting. Generator ensembles' size is fixed to 16.

### G.1 Effect of Generator Ensemble Size and Training Iterations

We explored the impact of both **generator ensemble size** and **generator training iterations** on the CIFAR-100 dataset, and our experiments results are encapsulated in Fig. 5 and Fig. 6 respectively. To ensure a fair comparison, our experimental setup solely applies the generator ensemble technique to DisGUIDE, excluding the integration of Selective Query and class-balanced difficulty-weighted replay. The configurations mirror the DisGUIDE setup for CIFAR-100 as detailed in Sec. 5.1 [26].

In experiments for evaluating the generator ensemble size, shown in Fig. 5), for both SL Setting and HL Setting, the extracted model's final accuracy generally improves with an increase in ensemble size, peaking at a size of 8, which results in 3.48% improvement over base DisGUIDE with a final accuracy of 66.39%. However, expanding the ensemble from 8 to 16 members results in a marginal decline in performance.

The impact of generator training iterations per victim query iteration are shown in Fig. 6. In both the *HL Setting* and the *SL Setting*, A clear upward trend is observed with increased training iterations. The increments in accuracy for each increasing generator training iteration ( $\beta$ ) are as follows. For **HL**: an increase of 1.87% from  $\beta=1$  to  $\beta=2$ , 0.28% from  $\beta=2$  to  $\beta=4$ , and 1.33% from  $\beta=4$  to  $\beta=8$ . For **SL**: an increase of 1.18% from  $\beta=1$  to  $\beta=2$ , 0.85% from  $\beta=2$  to  $\beta=4$ , and 0.14% from  $\beta=4$  to  $\beta=8$ . The result suggests that more generator training leads to enhanced performance in both HL and SL settings. However, similar to the generator size, we can observe the diminishing returns and it is possible that the increased generator training iterations' positive effect on model performance will be reversed beyond a specific threshold.

#### G.2 Comparison Against Improved Baseline

In Tab. 7, we present a performance comparison between DisGUIDE and CaBaGE method while eliminating the effect of replay iterations. The experimental configurations remain consistent with those in Tab. 1, except that we increased DisGUIDE's replay iteration from 3 to 12, and denote this method as DisGUIDE\*. Based on our findings in Appendix C, we have observed that increasing the replay iterations can lead to a boost in the final accuracy of the extracted model, although the gains diminish over time. To provide a fair comparison between the two methods, we chose to set DisGUIDE's replay iterations equal to CaBaGE's, and we run all experiments 9 times. The results are shown as mean values of extracted models' accuracies, along with standard deviations. Statistical significance is determined using independent t-tests, and the p value for each test is recorded (Column p val) with three decimal places.

CaBaGE demonstrates superior performance in most of the configurations. Setting a significance level at  $\alpha=0.05$ , most results demonstrate statistical significance. Specifically, CaBaGE outperforms DisGUIDE\* in 10 out of the 13 configurations. CaBaGE is only outperformed by DisGUIDE\* while extracting a VGG-16 model on the SVHN dataset.

**Table 7:** Independent t-test comparing CaBaGE (using class-balanced difficulty-weighted replay) and DisGUIDE (using original replay) under 12 replay iterations in the setting of various configurations. There are 9 runs for all experiments, accuracy is shown along with 95% confidence interval.

Dataset	Model	$\mathbf{DisGUIDE}^*$	CaBaGE	p val	Better
MNIST	MLP	$11.35\pm0.00$	$54.94\pm3.58$	p=0.000	CaBaGE
	LeNet	$94.20\pm0.55$	$95.32\pm0.63$	p=0.011	CaBaGE
	AlexNet	$74.10\pm2.11$	$86.08\pm0.94$	p=0.000	CaBaGE
FMNIST	MLP LeNet AlexNet	$42.28\pm4.40$ $69.12\pm1.12$ $74.66\pm2.09$	$74.45\pm3.28$ $68.08\pm1.50$ $79.63\pm1.48$	$p{=}0.000$ $p{=}0.246$ $p{=}0.001$	$\begin{array}{c} {\rm CaBaGE} \\ {\rm N/A} \\ {\rm CaBaGE} \end{array}$
SVHN	VGG-16	$88.37\pm0.87$	$84.14\pm2.09$	p=0.002	DisGUIDE
	ResNet-18	$76.05\pm1.81$	$78.70\pm0.98$	p=0.016	CaBaGE
	AlexNet	$64.01\pm2.40$	$74.46\pm2.19$	p=0.000	CaBaGE
CIFAR-10	AlexNet	$26.25\pm3.45$	$33.23\pm1.04$	$p{=}0.002$	CaBaGE
	ResNet-34	$22.04\pm2.64$	$26.34\pm2.54$	$p{=}0.021$	CaBaGE
$ImageNet_{12}$	AlexNet VGG-16	$43.11\pm3.17$ $38.94\pm5.39$	$48.54\pm2.68$ $35.53\pm5.97$	$p{=}0.012$ $p{=}0.370$	$\begin{array}{c} {\rm CaBaGE} \\ {\rm N/A} \end{array}$

# H Remaining Ablation on Larger Datasets

**Table 8:** Final Accuracy of CaBaGE and ablation settings shown with 95% confidence interval. Independent t-tests are used to determine improvement over the baseline DisGUIDE\*, where statistically significant improvements are highlighted in Blue.

Dataset	Model	$\mathbf{DisGUIDE}^*$	DisGUIDE* +CB-DW	DisGUIDE* +CB-DW+GE	CaBaGE
CIFAR-100	AlexNet ResNet-34	$25.22 \pm 1.98$ $34.71 \pm 1.28$	$25.74 \pm 2.04$ $38.99 \pm 1.69$	$34.68 \pm 1.30$ $43.00 \pm 0.69$	$33.09 \pm 0.54$ $43.75 \pm 1.75$
Tiny ImageNet	ResNet-34 VGG-16	$13.70 \pm 0.97$ $9.68 \pm 1.19$	$12.15 \pm 0.60$ $9.98 \pm 0.73$	$18.02 \pm 1.13$ $13.43 \pm 1.10$	$15.36 \pm 0.64$ $11.98 \pm 0.64$

We show the additional experiments on CIFAR-100 and Tiny-Imagenet for our ablation on the individual contribution of our novel components in Tab. 8. All settings are the same as Sec. 5.2 in our paper, except all experiments are run 3 times instead of 9 times on this table due to time constraints. We progressively add our novel components to the baseline, DisGUIDE\*, and present the ablations for DisGUIDE\*,  $DisGUIDE^*_{+CB-DW}$ ,  $DisGUIDE^*_{+CB-DW+GE}$  and CaBaGE.

When including class-balanced difficulty-weighted replay, the performance does not differ from the baseline much, which indicates that class-balanced difficulty-weighted replay has a similar performance compares to the circular buffer replay used by DisGUIDE\*. Conversely, the integration of GE consistently demonstrates statistically superior results compared to the baseline on both CIFAR-100 and Tiny-ImageNet, highlighting the robust benefits of GE. Regarding our comprehensive method, CaBaGE, which also includes SQ, there

are two settings where it outperforms the baseline. While this may appear less effective compared to  $DisGUIDE^*_{+CB-DW+GE}$ , it is important to note that the inclusion of SQ generally leads to a decrease in uncertainty levels, suggesting that SQ enhances the stability of the extraction results.