Attackers Can Do Better: Over- and Understated Factors of Model Stealing Attacks

Daryna Oliynyk
CDL AsTra,
Faculty of Computer Science,
University of Vienna, Vienna, Austria
daryna.oliynyk@univie.ac.at

Rudolf Mayer
SBA Research
Vienna, Austria
rmayer@sba-research.org

Andreas Rauber
Institute of Information Systems Engineering,
Faculty of Informatics,
TU Wien, Vienna, Austria
andreas.rauber@tuwien.ac.at

Abstract—Machine learning (ML) models were shown to be vulnerable to model stealing attacks, which lead to intellectual property infringement. Among other attack methods, substitute model training is an all-encompassing attack applicable to any machine learning model whose behaviour can be approximated from input-output queries. Whereas previous works mainly focused on improving the performance of substitute models by, e.g. developing a new substitute training method, there have been only limited ablation studies that try to understand the impact the strength of an attacker has on the substitute model's performance. As a result, different authors came to diverse, sometimes contradicting, conclusions. In this work, we exhaustively examine the ambivalent influence of different factors resulting from varying the attacker's capabilities and knowledge on a substitute training attack.

Our findings suggest that some of the factors that have been considered important in the past are, in fact, not that influential; instead, we discover new correlations between attack conditions and success rate. In particular, we demonstrate that better-performing target models enable higher-fidelity attacks and explain the intuition behind this phenomenon. Further, we propose to shift the focus from the complexity of target models toward the complexity of their learning tasks. Therefore, for the substitute model, rather than aiming for a higher architecture complexity, we suggest focusing on getting data of higher complexity and an appropriate architecture. Finally, we demonstrate that even in the most limited data-free scenario, there is no need to overcompensate weak knowledge with unrealistic capabilities in the form of millions of queries. Our results often exceed or match the performance of previous attacks that assume a stronger attacker, suggesting that these stronger attacks are likely endangering a model owner's intellectual property to a significantly higher degree than shown until now.

Index Terms—Adversarial Machine Learning, Model Stealing, Model Extraction, Ablation Study

I. INTRODUCTION

The rapid evolution of Machine Learning (ML) has added significant value to ML-based solutions, especially to the models they are built upon. However, model owners might need to expose their models to third parties, for instance, through APIs, hence endangering their intellectual property to so-called model stealing (extraction) attacks [1]. As one form of this attack, a malicious end user can exploit the interaction channel with the model to collect labelled data and train a *substitute* model to behave similarly to the original *target* model [2]. As a result, malicious third parties can possess an illegitimate (approximate) copy of the original model, violating the intellectual property rights of the model owner and endangering their business model.

Substitute model attacks are suitable for targeting models that are vital in numerous fields, including image classification [2], image-to-image translation [3], natural language processing [4], and reinforcement learning [5]. In addition to being applicable to various task and data domains, substitute training attacks have been demonstrated to be effective under varied assumptions of the attacker's strength. These assumptions primarily relate to the attacker's knowledge of the target model and its training data, as well as the attacker's capabilities, primarily regarded as the number of queries needed to steal a model. However, the influence of these factors has been studied only sparsely, with most works addressing a limited subset tailored to their specific attack methods. As a consequence, a certain bias occurred in what factors are considered decisive when an attacker's strength is limited in a certain way. Moreover, as findings from different works sometimes disagree, the impact of some factors remains equivocal.

In this work, we address this gap and systematically examine the influence of different factors on the performance of the substitute model. We demonstrate that while some factors attracted a lot of attention, others have been under-explored or completely overlooked. We present new insights that not only can provide a more comprehensive explanation of observations from previous work but also make previous attacks more severe.

To this end, we comprehensively evaluate how the performance of a substitute model is influenced by (i) target model properties such as architecture and performance on the original classification task, (ii) substitute architecture choice, (iii) usage of transfer learning for training target and substitute models, (iv) attacker's data quality, and (v) attacker's capabilities represented by query budget and query optimisation strategies. Our conclusions are based on an analysis of 180 attack configurations, each being applied against three target models with different properties.

Our main contributions are:

- We are the first to demonstrate that the performance of the target model acts as a bottleneck, limiting the fidelity of substitute models. This stems from the ability of

substitute models to replicate the target model's behaviour significantly better on *correctly* predicted labels than on labels *incorrectly* predicted by the target model. Consequently, for stronger attackers, achieving high fidelity is easier when attacking better-performing models.

- We propose to shift focus from the target model complexity towards the complexity of its classification task. In particular, the complexity of the substitute model should be considered with respect to the data complexity rather than the complexity of the target model, as suggested in several prior works.
- We introduce novel insights into how the usage of transfer learning impacts the performance of substitute training attacks. In particular, if a substitute model is trained from scratch, it will perform better when stealing a target model trained from scratch than a target model with pre-trained weights.
- We show that attacks are more effective when the complexity of the attacker's dataset is higher than the original dataset; in contrast, attacks generally perform worse when the attacker's dataset has lower complexity than the original dataset. Moreover, attackers with simpler data tend to overestimate attack performance, whereas attackers with more complex data tend to underestimate it.
- Finally, we are the first to perform a data-free attack that is effective with a query budget smaller than the original training set of the target model.

Besides, our substitute models trained on original or problem-domain data outperform state-of-the-art, even when assuming weaker attacker knowledge. Our data-free attack achieves a performance comparable to both data-free attacks and attacks using non-problem domain data while requiring two orders of magnitude fewer queries.

The rest of the paper is organised as follows. Section II introduces relevant terminology and metrics used in this work for attack evaluation. Section III provides a comprehensive overview of related work on stealing image classifiers. In Section IV, we present our detailed analysis of factors influencing the attack success. Section V demonstrates a comparison of our results to the state-of-the-art, followed by a discussion in Section VI and a conclusion in Section VII.

II. BACKGROUND

The general goal of a model stealing (or model extraction) attack is to create an exact or approximate replica of a machine learning model to which the attacker has some sort of restricted access. We call the model under attack the *target model* and denote it as f. We further assume that we only have *black-box* access to the target model, which means that the only information an adversary can retrieve is the predictions (*outputs*) for given input samples (*inputs*). Moreover, in the context of the classification tasks considered in this work, we assume that the target models can only output top-1 predictions (*labels*) for input samples. In other words, the only available action for the adversary is to send a sample x to the target model and obtain a label $f(x) \in \{c_1, \ldots, c_k\}$. We call such

a single request to the model a *query*, where x is the *query input* and $f(x)$ is the *query output*. The adversary can create an *attacker's dataset*, exploiting the target model as an oracle for labelling data samples. Subsequently, the attacker's dataset can be used to train a so-called *substitute model*. The substitute model, which we denote as $\hat{f}$, can then further be used to launch an alternative service with lower fees, potentially leading to lower demand for the original service and profit loss for the target model owner.

A successfully performed attack should be both efficient and effective. In terms of efficiency, an attacker should spend a reasonable amount of resources to collect and label the data through the target model. The most crucial part of this process is the *number of queries* required to perform the attack. Previous work suggested reporting an *efficiency score*, which shows how many queries per parameter (weight) of the target model it takes to perform an attack [6]. In this work, we shift focus from the target model's complexity towards the complexity of its training data and, therefore, measure how many queries per target's training sample are required to perform the attack.

For evaluating effectiveness, the performance of a substitute model is compared with the performance of the target model. Three metrics are mainly used for that purpose: accuracy, fidelity, and transferability [6]. In the following, we briefly describe the metrics used in this work.

Accuracy shows how a substitute model performs on the classification task that the target model was trained to solve. It compares the outputs of the substitute model with the original labels of a dataset. The accuracy of the substitute model $\hat{f}$ on dataset X^{test} is measured as

$$\frac{1}{|X^{test}|} \sum_{i=1}^{|X^{test}|} \mathbb{1}_{(\hat{f}(x_i^{test})=y_i^{test})}.$$

Additionally, we introduce in this work a metric called *joint accuracy*, which shows how many samples are classified correctly by both target f and substitute $\hat{f}$ models, as follows

$$\frac{1}{|X^{test}|} \sum_{i=1}^{|X^{test}|} \mathbb{1}_{(f(x_i^{test})=\hat{f}(x_i^{test})=y_i^{test})}.$$

Fidelity measures the similarity of target and substitute predictions by comparing the labels that the target model f and the substitute model $\hat{f}$ output on the test set X^{test}. Therefore, fidelity counts both correct and incorrect equal predictions, as

$$\frac{1}{|X^{test}|} \sum_{i=1}^{|X^{test}|} \mathbb{1}_{(\hat{f}(x_i^{test})=f(x_i^{test}))}.$$

III. RELATED WORK

In this section, we discuss aspects and approaches relevant to the area of stealing image classifiers. Table I shows aggregated information from 44 relevant papers, which were selected based on the following criteria: (i) the paper introduces a new substitute training attack or extends a previous work; (ii) both target and substitute models are trained on image

TABLE I: Summary of current substitute training approaches.

Paper	Attacker's data	Data crafting technique	Target architecture	Query optimization	Target outputs	Metrics
[2]	NPD		✓✗	RL	Labels, Probab.	Acc
[7]	Original or NPD		✗	SSL	Probab.	Acc, Fid
[8]	Original or PD	Adver. augm.	✗		Labels	Acc
[9]	NPD or/and PD		N/A		Labels	Acc
[10]	NPD		✓✗	AL	Labels, Probab.	Fid
[11]	Original	Adver. augm.	✓✗		Labels, Probab.	Fid
[12]	NPD		✓	RL	Labels, Probab.	Acc
[13]	Original	Adver. augm.	✗	AL	N/A	Acc, Fid
[14]	NPD	Data composition	✗		Labels	Acc
[15]	Artificial	Generator	✓✗		Labels, Probab.	Acc
[16]	Original		✓✗		Gradients	Acc
[17]	Artificial	Generator	✗		Probab.	Acc
[18]	Artificial	Noise	✓		Probab.	Acc
[19]	NPD	Generator	✓✗	EA	Probab.	Acc
[20]	PD	Adver. augm.	✓✗		Probab.	Acc
[21]	NPD	Model inversion	✗		Labels	Acc, Fid
[22]	Artificial	Generator	✗		Probab.	Acc
[23]	Artificial	Generator	✗		Probab. + Expl.	Acc
[24]	Artificial	Generator	✗		Labels	Acc
[25]	Original	Adver. augm.	✓✗	RL	Probab.	Acc
[26]	NPD		N/A	AL	Labels	Acc
[27]	NPD		✓✗		Labels	Acc, Fid
[28]	Original		✗		Probab. + Expl.	Acc
[29]	(N)PD	Generator	✗	AL	Probab.	Acc, Fid
[30]	Original, PD or NPD		✓	AL, RL	Probab.	Acc, Fid
[31]	Original or NPD		✓	DR	Probab.	Acc
[32]	Original		✓		Probab.	Acc, Fid
[33]	Artificial	Generator	✓✗		Labels, Probab.	Acc
[34]	Original, PD or NPD		✓✗		Labels	Acc, Fid
[35]	Artificial	Generator	✗		Labels, Probab.	Acc
[36]	Artificial	Generator	✗		Labels + Expl.	Acc
[37]	Original		N/A		Labels + Expl.	Fid
[38]	Artificial	Generator	✗		Labels	Acc
[39]	Artificial	Adver. augm., Generator	✓✗	RL	Labels, Probab.	Acc
[40]	Original and PD		N/A	CBS	Probab.	Acc
[41]	Original		✓✗		Labels	Fid
[42]	Artificial	Generator	✗		Probab.	Acc, Fid
[43]	NPD		✗	RL	Probab.	Acc, Fid
[44]	Original, PD or NPD		✗	SSL	Probab.	Fid
[45]	NPD		✗	AL	Labels	Acc, Fid
[46]	NPD		✗	AL, SSL	Labels	Acc, Fid
[47]	Artificial	Generator	✗		Labels, Probab.	Acc
[48]	Artificial	Generator	✗		Labels, Probab.	Acc
[49]	NPD		✓	AL	Labels	Fid
Our work	Original, PD or Artificial	Adver. augm., Generator	✓✗	AL	Labels	Acc, Fid

data; (iii) the substitute model is trained with the intention of copying the behaviour (functionality) of the target model. In the following, we review each characteristic presented in Table I.

Attacker's data corresponds to data categories used for querying the target model; they comprise original, problem-domain, non-problem domain, and artificial data [6]. Attacks that only rely on artificial data to train substitute models are also called *data-free* attacks. In Table I, we indicate in the corresponding column which data was used by related work. We use the connector "and" to mark cases when a substitute model was trained on a combination of data from different categories. If a paper assumed that a small amount of original or PD data is available (even if less than 5%), we still considered those attacks as ones requiring the corresponding type of data.

Data crafting techniques are commonly used for two goals: generating artificial data and creating more (high-quality) data from the one available. Below, we describe the methods from previous work, which are listed in Table I. *Adversarial augmentation* is the most common approach for improving the quality of the data [13], [20], [25]. The idea is to query a target model with adversarial examples crafted for a substitute model to "correct" the predictions of the substitute

model near its decision boundary. *Data composition* is an approach proposed for increasing the quality of NPD data by merging two images into one [14]. *Model inversion* is inspired by the model inversion attack [50], which aims to reconstruct training data from a model. In the context of model stealing, it can be applied to extract data from the target model, leading to more meaningful data when only NPD data is available [21]. *Generative models* (generators) are common for creating artificial data [22] and improving the quality or increasing the quantity of the attacker's data samples [19]. Finally, *noise* can also be used as a direct input for querying a model when no data or generative model is available [18].

If the **target architecture** is known, an adversary can use it as the architecture choice for a substitute model, simplifying the whole stealing process. In general, there are two possible scenarios: (i) the substitute architecture differs from the target, as the latest is assumed to be unknown (marked as ✗ in Table I), (ii) the substitute and target architectures are the same (marked as ✓ in Table I). We indicate papers with ✓✗ if both cases are reported. If it was unclear which strategy the authors chose, we marked such papers as N/A.

Query optimisation includes techniques that aim to increase the efficiency of an attack by reducing the number of queries. The most common technique is using *Active Learning* (AL) [10], [13], [26], [29]. Active learning was initially introduced as an optimisation for labelling data in supervised learning scenarios with a significant amount of unlabelled data. Since labelling data is also part of model stealing (with the API being the oracle), active learning can be applied for query optimisation [51]. A few other works trained *Reinforcement Learning* (RL) agents to pick samples with the highest impact on substitute model training [2], [25]. Besides, query optimisation can also be conducted by utilising *Evolutionary Algorithms* (EAs) [19], *Self-Supervised Learning* (SSL) [7], [44], [46], *Dataset Reduction* (DR) [30] or *Cluster-Based Selection* (CBD) of data samples [40].

Target outputs are the primary source of information an adversary can obtain from the target model. The most widespread assumptions are that the target model outputs either *labels* or *probabilities* (confidence scores). All papers that compared labels with probabilities concluded that the substitute performance is better when probabilities are used [2], [10]–[12], [15], [21]. In other scenarios, an API can reveal even more information though explicit model *gradients* [16] or model explanations, provided additionally to labels [36], [37] or probabilities [23], [28]. We marked in Table I papers as N/A in case it was unclear from the attack description which target outputs were utilised.

Finally, for each paper, we specified the **metrics** used for attack evaluation, namely, *accuracy* or *fidelity*.

Table I illustrates the following notable trends:

- PD data is the least explored in previous studies (18% of works), in contrast to artificial (27%), original (33%), and NPD (49%) data. However, as image data is nowadays widely accessible, gathering PD data for image classification tasks is a highly plausible scenario, making PD

TABLE II: Accuracy of target models trained on CIFAR-10.

Target model	Test accuracy	#parameters
SimpleNet	91.76%	∼5M
ResNet-34 (from scratch)	93.61%	∼21M
ResNet-34 (transfer learning)	97.14%	∼21M

attacks potentially the most threatening. As we demonstrate later in Section IV-E, PD data, even of noticeably higher complexity than the original training data, leads to highly effective model stealing attacks.

- Most of the methods were validated with mismatching substitute and target architectures, suggesting that knowledge about the target model architecture is not crucial. This observation aligns with our insight from Section IV-C that the substitute architecture choice should correspond to the complexity of the attacker's data rather than be compared with the complexity of the target model.

- None of the works that use artificial data incorporate a query optimisation strategy. However, as we demonstrate later in Section V, these attacks usually require millions of queries, which makes them significantly less efficient than attacks using non-artificial data.

IV. ATTACK FACTOR EVALUATION

In the following, we first present an overview of all attack configurations studied in this work. The overview is followed by a step-by-step analysis of five attack factors. For each factor, we first present relevant findings from prior work together with our observations, which were not necessarily highlighted or spotted by corresponding authors. Subsequently, we describe and analyse experiments conducted in this work that demonstrate the influence of the attack factor. We conclude the analysis of each attack factor with a summary of our insights.

A. Attack Setup

For our study, we first trained three target models and then exploited each of them to train 180 substitute models (see Appendix A for a detailed attack setup overview). The substitute models varied in terms of architecture, the data they were trained on, the amount of data, and the strategy of data collection. Below, we describe in detail the characteristics of each component of the attack setup.

Target model. The three target models are trained on the CIFAR-10 dataset [52], with their accuracy scores and complexities shown in Table II. Two target models have ResNet-34 architecture: one is trained from scratch, and the other is trained using transfer learning from the ImageNet dataset. The third model has the SimpleNet [53] architecture, which has significantly fewer parameters (around one-quarter of ResNet-34) and is trained without transfer learning.

Substitute model. The substitute models have SimpleNet, ResNet-18 and ResNet-34 architectures. The Simplenet model

is trained from scratch, whereas both residual networks are trained using transfer learning from ImageNet.

Attacker's Dataset. Three datasets are used as attacker's data, each corresponding to a different degree of knowledge about the original data, namely: CIFAR-10 (as original data), CINIC-10 (as problem-domain data), and an artificial dataset (for a data-free attack scenario). The artificial data is generated with a stable diffusion model[1] to approximate problem domain data (see Appendix E for more details). Each dataset is split into 45,000 training samples used for training substitute models and 5,000 validation samples. In all experiments, we assume that an attacker only has access to the output labels of the target model.

Query Budget. We carry out experiments with 5 query budgets: 1,000, 5,000, 10,000, 20,000, and 45,000 samples. We set the upper bound to be 45,000 for two reasons: (i) to avoid knowledge leakage, we keep validation sets unseen by the substitute models, and (ii) for attacks using data other than the original, we want to have a baseline trained on the original data with the same number of queries.

Attack Strategy. Alongside randomly selecting a subset of the inputs from the whole dataset for querying the target model (random sample selection), we applied three query optimisation strategies: active learning from [10], adversarial augmentation from [13], and their combination. These methods from previous work were slightly modified for better efficiency and additionally combined into a single method (see Appendix B for more details).

Evaluation. We evaluated each attack in terms of accuracy, joint accuracy (see Section II), and fidelity. The 10,000 test samples from the CIFAR-10 dataset were used for the final evaluation. All attackers' decisions about the optimal hyper-parameter choice were based on scores obtained on 5,000 samples from the attacker's validation set.

B. Factor 1: Target Model Properties

Prior Work Discussion. Jagielski et al. [7] demonstrated that high fidelity of substitute models can not be guaranteed due to the non-deterministic nature of learning-based approaches. We take this exploration further, showing that fidelity is actually limited by the accuracy of the target model, and the non-determinism occurs primarily while learning the mistakes (wrongly classified samples) of the target model. This insight can also be traced in other works, even though it remained unnoticed by the authors. For instance, Pape et al. [41] performed a substitute training attack with original data against three target models of different complexity and different performance. In their results, fidelity scores correlate with the performance of the target model, being the highest for the most complex and better-performing model. While one might connect such a trend with the complexity of the target model, we demonstrate that among two target models that have identical architectures but different accuracy scores, the better-performing one leads to higher fidelity.

[1]https://huggingface.co/stabilityai/stable-diffusion-2-1

Results. We present in Table III the results of attacks with the strongest attacker's knowledge assumption: original data and identical architectures for target and substitute models. The fidelity scores correlate with the performance of the target model—the higher the accuracy of the target model, the higher the fidelity score reached by a substitute model. Moreover, fidelity does not exceed the target accuracy, suggesting that the target accuracy acts as a limiting factor.

TABLE III: Performance of substitute models trained on the original data (CIFAR-10) with the same architectures as target models.

Target model	Query budget	Test Scores			Target accuracy
		Joint Acc	*Accuracy*	*Fidelity*	
SimpleNet	1k	48.07%	50.26%	50.39%	91.76%
	5k	69.84%	72.01%	73.01%	
	10k	76.39%	78.84%	79.93%	
	20k	82.70%	85.38%	86.33%	
	45k	**87.02%**	**90.27%**	**90.61%**	
ResNet-34 (from scratch)	1k	76.69%	79.35%	78.74%	93.61%
	5k	85.32%	88.54%	87.24%	
	10k	88.51%	91.91%	90.51%	
	20k	90.73%	94.73%	92.39%	
	45k	**92.15%**	**96.67%**	**93.50%**	
ResNet-34 (transfer learning)	1k	80.58%	81.61%	81.79%	97.14%
	5k	89.47%	90.29%	90.91%	
	10k	91.30%	92.26%	92.73%	
	20k	93.77%	94.79%	95.16%	
	45k	**95.37%**	**96.49%**	**96.84%**	

TABLE IV: Accuracy of substitute models on correct and incorrect predictions of the target model. Substitutes are trained with 45,000 original samples and have the same architectures as the target models.

Target model	Correct predictions	Incorrect predictions
SimpleNet	94.83%	43.57%
ResNet-34 (from scratch)	98.44%	21.13%
ResNet-34 (transfer learning)	98.18%	51.40%

To investigate this behaviour further, we measured how well the substitute model learned correct and incorrect predictions of the target model. Identical correct predictions are represented by joint accuracy. We measure the accuracy of the substitute model on correct target predictions as

$$\frac{\text{Joint accuracy}}{\text{Target accuracy}} \times 100\%.$$

Subsequently, for the incorrect predictions, the accuracy is

$$\frac{\text{Fidelity} - \text{Joint accuracy}}{100\% - \text{Target accuracy}} \times 100\%.$$

The results presented in Table IV clearly demonstrate that substitute models learn correct predictions significantly better than incorrect predictions. Therefore, if the target model makes fewer mistakes, its behaviour is easier to copy, which is represented by higher fidelity. Moreover, we speculate that higher incorrect prediction accuracy for SimpleNet and pre-trained ResNet-34 models in Table IV is the consequence of using the same training strategy (training from scratch

TABLE V: Performance of substitute models with different architectures trained on the original (CIFAR-10) data.

Substitute ↓	QB	SimpleNet			ResNet-34 (from scratch)			ResNet-34 (transfer learning)		
Target →		Joint Acc	Acc	Fid	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid
SimpleNet	1k	48.07%	50.26%	50.39%	48.95%	50.41%	51.01%	47.32%	48.08%	48.16%
	5k	69.84%	72.01%	73.01%	70.82%	72.28%	73.64%	70.50%	71.44%	71.49%
	10k	76.39%	78.84%	79.93%	77.88%	79.55%	81.05%	78.14%	79.13%	79.27%
	20k	82.70%	85.38%	86.33%	83.35%	85.15%	86.55%	84.32%	85.26%	85.55%
	45k	**87.02%**	**90.27%**	**90.61%**	**88.34%**	**90.45%**	**91.41%**	**89.08%**	**90.12%**	**90.33%**
ResNet-18	1k	78.54%	82.20%	81.31%	79.89%	82.31%	82.50%	81.54%	82.56%	82.68%
	5k	84.95%	89.33%	87.37%	86.25%	89.11%	88.54%	87.75%	88.73%	88.94%
	10k	87.51%	92.05%	89.95%	88.79%	92.01%	91.01%	91.07%	92.07%	92.31%
	20k	88.72%	93.46%	91.15%	90.09%	93.35%	92.40%	92.80%	93.87%	94.13%
	45k	**89.84%**	**95.28%**	**91.71%**	**91.49%**	**95.29%**	**93.40%**	**94.19%**	**95.39%**	**95.44%**
ResNet-34	1k	77.72%	81.70%	80.01%	76.69%	79.35%	78.74%	80.58%	81.61%	81.79%
	5k	83.93%	88.37%	86.26%	85.32%	88.54%	87.24%	89.47%	90.29%	90.91%
	10k	87.60%	92.60%	89.69%	88.51%	91.91%	90.51%	91.30%	92.26%	92.73%
	20k	89.36%	94.64%	91.32%	90.73%	94.73%	92.39%	93.77%	94.79%	95.16%
	45k	**90.35%**	**96.29%**	**91.95%**	**92.15%**	**96.67%**	**93.50%**	**95.37%**	**96.49%**	**96.84%**

vs transfer learning) for both target and substitute models. We elaborate more on the usage of transfer learning later in Section IV-D.

Conclusion. *The target model performance acts as a limiting factor for the fidelity of a substitute model. This behaviour occurs because learning the mistakes of the target model is significantly more challenging compared to the correct predictions. Therefore, for a strong attacker, targeting a better-performing model will lead to a better-performing attack.*

C. Factor 2: Substitute Architecture Choice

Prior Work Discussion. In more than 10 recent works, the impact of the substitute architecture was studied [2], [8], [10], [11], [15], [20], [21], [25], [30], [34], [41], [42], [45]. While comparing the architectures of substitute and target models, the majority of prior studies concluded that the substitute has to be of the same or higher complexity in order to achieve higher effectiveness. However, some studies encountered the opposite, with more complex models performing worse [25], [39], [42]. In our work, we obtain a similar effect: ResNet-18, in some cases, outperforms the more complex ResNet-34.

In general, the comparison of target and substitute architectures can be misleading. The target architecture can, for some reason, be selected to have a learning capacity significantly larger than is needed for the classification task. However, selecting a simple architecture that meets the task requirements can be enough for the adversary.

The substitute model has to learn the task from the data labelled by the target model. Therefore, data should be the key factor in deciding on a substitute architecture. Rather than comparing target and substitute architectures, we suggest measuring how applicable the substitute architecture is for the learning task. This can be done, for example, by evaluating the performance of the substitute model on the original classification task when trained on the original data with the original (truth) labels.

Results. Table V presents the performance of substitute models with different architectures trained on the original (CIFAR-10) data. As a substitute model, SimpleNet has the lowest scores for all target models and query budgets. However, as a target model, SimpleNet also has the lowest accuracy, suggesting that its learning capacity is likely too limited to learn CIFAR-10 better.

For the smallest query budgets of 1,000 samples, ResNet-18 outperforms ResNet-34. This trend sometimes persists for up to 20,000 queries. Therefore, having the same architecture as the target model does not imply the best attack performance. ResNet-34 performs the best for 45,000 queries for all target models. As the number of samples grows, the substitute model has to process more information. Consequently, in this scenario, a more complex architecture is beneficial.

We observe similar trends on PD data (CINIC-10): SimpleNet has the lowest performance, and ResNet-34 reaches the highest scores for the largest query budget (see Table XIV in Appendix D). However, unlike for original data, ResNet-18 mostly performs worse than ResNet-34. As we show later in Section IV-E, CINIC-10 is more complex than CIFAR-10. Hence, it is expected that it requires a more complex architecture such as ResNet-34.

For the artificial data, on the smaller datasets (1,000 and 5,000 queries), ResNet-18 outperforms ResNet-34 (see Table XV in Appendix D). As we demonstrate later, the artificial dataset is simpler than CIFAR-10 (Section IV-E). Hence, especially for small query budgets, a ResNet-18 architecture is complex enough to learn the artificial data.

Conclusion. *The complexity of the substitute architecture should correspond to the complexity of the attacker's data rather than be compared to the complexity of the target model. A more complex substitute model should be justified by data characteristics. In particular, selecting a more complex architecture can be beneficial if the adversary gathers more data or uses data of a higher complexity.*

TABLE VI: Performance of substitute models trained on 45,000 samples of the original (CIFAR-10) data.

Target →	SimpleNet			ResNet-34 (from scratch)			ResNet-34 (transfer learning)		
Substitute ↓	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid
SimpleNet	87.02%	90.27%	90.61%	88.34%	90.45%	91.41%	89.08%	90.12%	90.33%
ResNet-18	89.84%	95.28%	91.71%	91.49%	95.29%	93.40%	94.19%	95.39%	95.44%
ResNet-34	90.35%	96.29%	91.95%	92.15%	96.67%	93.50%	95.37%	96.49%	96.84%

TABLE VII: Fidelity scores of SimpleNet substitute model trained on the artificial dataset.

Target model	Fidelity					Target accuracy on validation set
	1k	5k	10k	20k	45k	
SimpleNet	23.10%	**40.77%**	**48.61%**	**58.12%**	**68.35%**	88.46%
ResNet-34 (from scratch)	**24.58%**	37.78%	43.49%	54.81%	68.34%	91.40%
ResNet-34 (transfer learning)	23.57%	35.04%	38.28%	44.95%	51.01%	96.78%

D. Factor 3: Usage of Transfer Learning

Prior Work Discussion. Atli et al. [12] demonstrated that stealing models trained from scratch results in lower effectiveness than stealing pre-trained models. In their experiments, all substitute models were trained with transfer learning. Moreover, Zhang et al. [25] showed that without pre-trained weights, the effectiveness of the substitute model attacks significantly degrades. However, in their experiments, the dataset used for transfer learning overlapped with the original training data of the target model. Therefore, the pre-trained weights were actually (at least partially) trained on the original data, providing even more advantage for the adversary. We take the previous observations further and demonstrate that substitutes trained from scratch learn better from target models trained from scratch compared to learning from pre-trained target models.

Results. In our experiments, two target models were trained from scratch (SimpleNet and ResNet-34), and one was trained with transfer learning from ImageNet (ResNet-34). Further, one substitute model is trained from scratch (SimpleNet), and two are trained from pre-trained on ImageNet weights (ResNet-18 and ResNet-34). Table VI illustrates that SimpleNet as a substitute reaches higher fidelity and accuracy scores when targeting models trained from scratch. However, joint accuracy is higher when targeting the pre-trained model, which also has the best performance. This observation, on the one hand, reinforces our findings from Section IV-B that it is easier to learn (i) from better-performing models and (2) correct predictions. On the other hand, higher joint accuracy together with lower fidelity means that SimpleNet can capture mistakes of the pre-trained ResNet-34 worse than mistakes of models trained from scratch. For the pre-trained substitute models, all metrics, especially joint accuracy and fidelity, increase with increasing performance of the target model. Even when the target model has a different architecture and is trained from scratch (SimpleNet), stealing it with a pre-trained substitute (ResNet models) yields high scores.

When using artificial data for training substitutes from scratch, the difference between attacks on pre-trained and trained-from-scratch models becomes even more significant. As Table VII illustrates, the SimpleNet substitute model trained with artificial data has 17% lower fidelity when targeting the pre-trained model. For reference, we also report the performance of the target models on the artificial validation set. The trend is the same as with the original data: SimpleNet has the lowest performance, and pre-trained ResNet-34 has the highest. Despite that, the pre-trained target leads to the lowest attack performance.

Conclusion. *Training target models from scratch does not make them less prone to be stolen. In fact, if a substitute model is trained from scratch, it reaches higher scores when attacking the target model trained from scratch compared to the model trained with transfer learning. Consequently, for the adversary, it is beneficial to follow the same training strategy as the one used for the target model.*

E. Factor 4: Attacker's Data Quality

Prior Work Discussion. As shown previously in Section III, PD data has been the least explored in prior works. Correia-Silva et al. [9] compared substitutes trained on PD and NPD data. In some settings, models trained on NPD data outperformed ones trained on PD data, leading to the conclusion that random real data is enough for an effective attack. However, in those experiments, the attack with NPD data used significantly more queries than the attack with PD data. In this work, we fix the query budget so that it does not exceed the original training set and explore how PD-like data of different quality and complexity impacts the attack performance.

Results. Table VIII demonstrates the performance of substitute models trained with ResNet-18 architecture on different types of attacker's data. The highest scores for all target models are obtained with the original (CIFAR-10) data, followed by PD (CINIC-10) data and PD-like artificial data. Surprisingly, the artificial data labelled by the SimpleNet target model was more beneficial for the attack than the artificial data labelled by ResNet target models, even though the SimpleNet model has the lowest performance on the artificial dataset (see Table VII).

TABLE VIII: Performance of ResNet-18 substitute models trained on different attacker's data.

Dataset ↓	QB	SimpleNet			ResNet-34 (from scratch)			ResNet-34 (transfer learning)		
Target →		Joint Acc	Acc	Fid	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid
CIFAR-10	1k	78.54%	82.20%	81.31%	79.89%	82.31%	82.50%	81.54%	82.56%	82.68%
	5k	84.95%	89.33%	87.37%	86.25%	89.11%	88.54%	87.75%	88.73%	88.94%
	10k	87.51%	92.05%	89.95%	88.79%	92.01%	91.01%	91.07%	92.07%	92.31%
	20k	88.72%	93.46%	91.15%	90.09%	93.35%	92.40%	92.80%	93.87%	94.13%
	45k	**89.84%**	**95.28%**	**91.71%**	**91.49%**	**95.29%**	**93.40%**	**94.19%**	**95.39%**	**95.44%**
CINIC-10	1k	65.65%	68.60%	68.76%	69.14%	71.32%	71.49%	72.19%	73.20%	73.41%
	5k	76.33%	79.60%	79.41%	79.79%	82.19%	82.24%	82.78%	83.81%	84.06%
	10k	80.10%	83.48%	83.23%	82.25%	84.83%	84.69%	85.09%	86.13%	86.38%
	20k	81.91%	85.21%	85.40%	84.84%	87.32%	87.52%	87.90%	89.01%	89.12%
	45k	**84.28%**	**87.59%**	**87.83%**	**86.88%**	**89.43%**	**89.64%**	**90.42%**	**91.44%**	**91.76%**
Artificial	1k	63.28%	66.20%	65.86%	64.13%	66.06%	66.32%	66.05%	66.83%	67.10%
	5k	67.90%	71.11%	70.55	67.30%	69.44%	69.52%	71.12%	71.97%	72.14%
	10k	69.93%	73.04%	72.86%	71.38%	73.72%	73.40%	69.58%	70.46%	70.63%
	20k	73.43%	76.72%	76.42%	73.96%	76.31%	76.19%	68.08%	69.01%	69.16%
	45k	**75.69%**	**78.88%**	**78.78%**	**74.94%**	**77.10%**	**77.28%**	**72.96%**	**73.81%**	**74.00%**

TABLE IX: Comparison of validation and test scores of attacks using CINIC-10 dataset.

Target model	Query budget	Validation scores		Test Scores	
		Acc	Fid	Acc	Fid
SimpleNet	1k	61.22%	66.28%	68.60%	68.76%
	5k	65.51%	70.34%	79.60%	79.41%
	10k	66.99%	72.20%	83.48%	83.23%
	20k	66.49%	74.01%	85.21%	85.40%
	45k	**67.24%**	**76.02%**	**87.59%**	**87.83%**
ResNet-34 (from scratch)	1k	64.30%	67.70%	71.32%	71.49%
	5k	68.36%	72.16%	82.19%	82.24%
	10k	70.25%	74.13%	84.83%	84.69%
	20k	70.23%	75.89%	87.32%	87.52%
	45k	**71.02%**	**77.88%**	**89.43%**	**89.64%**
ResNet-34 (transfer learning)	1k	67.16%	72.18%	73.20%	73.41%
	5k	73.22%	78.65%	83.81%	84.06%
	10k	74.83%	80.73%	86.13%	86.38%
	20k	75.79%	81.83%	89.01%	89.12%
	45k	**77.50%**	**83.48%**	**91.44%**	**91.76%**

TABLE X: Comparison of validation and test scores of attacks using artificial dataset.

Target model	Query budget	Validation scores		Test Scores	
		Acc	Fid	Acc	Fid
SimpleNet	1k	93.18%	88.66%	66.20%	65.86%
	5k	**94.26%**	89.76%	71.11%	70.55%
	10k	93.46%	89.82%	73.04%	72.86%
	20k	93.58%	90.22%	76.72%	76.42%
	45k	93.34%	**90.80%**	**78.88%**	**78.78%**
ResNet-34 (from scratch)	1k	94.58%	91.22%	66.06%	66.32%
	5k	**95.74%**	92.24%	69.44%	69.52%
	10k	95.06%	92.38%	73.72%	73.40%
	20k	95.58%	92.84%	76.31%	76.19%
	45k	95.08%	**92.94%**	**77.10%**	**77.28%**
ResNet-34 (transfer learning)	1k	95.94%	95.50%	66.83%	67.10%
	5k	96.92%	96.46%	71.97%	72.14%
	10k	97.38%	97.10%	70.46%	70.63%
	20k	97.58%	97.10%	69.01%	69.16%
	45k	**97.74%**	**97.50%**	**73.81%**	**74.00%**

For further analysis, we compare scores obtained on the attacker's validation set with the CIFAR-10 test scores. This comparison shows the difference between the performance estimation of the attack by the adversary and the eventual real success of the attack. For the CIFAR-10 attacker's dataset, validation and test scores are very similar, as both datasets are drawn from the same distribution.

For CINIC-10 (shown in Table IX), the performance on the attacker's validation set is notably lower than on the CIFAR-10 test set, indicating that CINIC-10 is more complex. Hence, using CINIC-10, an adversary tends to underestimate the power of their attacks. In contrast, the scores measured on the artificial validation set are way higher than the test scores (see Table X). For example, with a query budget of 1,000, all substitute models reached an accuracy larger than 93% on the validation set. With the same query budget, substitutes trained on the CIFAR-10 data (Table III) reached at most 82% of accuracy on the validation set. This further indicates that the artificial data is not complex enough, and its quality has to be improved. Using artificial data, an adversary will likely overestimate the performance of the substitute model on the real data.

As Table VIII shows, attacks using the CINIC-10 dataset, which we concluded is more complex than the original data, in general follow the same trends as attacks using the original data and have reasonable performance. Therefore, we conclude that data that is more feasible for attacks should have either comparable complexity or even be more complex than the original data, which can be assessed by comparing validation and test scores. However, as we only consider data with the same context as the original data, this statement needs further investigation for non-problem domain attacks.

Conclusion. *PD data more complex than the original training data leads to better-performing substitute models than simpler PD data. Furthermore, based on substitute validation scores, attackers with more complex data tend to underestimate*

the performance of the substitute model. In turn, having data simpler than the original leads to an overestimation of substitute performance.

F. Factor 5: Attacker's Capabilities

Prior Work Discussion. As presented earlier in Table I, query optimisation techniques have been widely studied in related work. Prior results suggest that with optimisation, an attack can achieve higher scores with the same number of queries [10], [20]. Similarly, query optimisation can help to reach the same attack performance level while using fewer queries. However, as we observed earlier in Section III, query optimisation has not been applied for data-free attacks. In earlier work, these attacks were usually carried out using millions of queries and are the least efficient substitute training attacks (see Section V). While our data-free attack already has high efficiency, we demonstrate that it can be improved with active learning.

Results. Table XI shows the performance scores of the attacks optimised for the scenario of the weakest attacker in terms of knowledge about the target model and its data. All attacks in the table are performed with a ResNet-18 substitute model trained on the artificial dataset. For each score in Table XI, we also provide the difference to the scores from Table VIII, which contains attacks with the same settings but without any optimisation techniques. Therefore, a positive number means the optimisation technique improved the performance, while negative numbers indicate a degradation of the attack's performance. There is no reported active learning attack with 45,000 queries, as that is the size of the whole dataset, and applying active learning does not change the final attacker's set compared to the non-optimised attack.

In most of the configurations, active learning performs the best, although for the smaller query budgets (1,000 and 5,000), it does not always improve the baseline, and in a few cases, other optimisation techniques reach higher scores. However,

active learning improves the attack and holds the second-best score in all such cases. For small query budgets, the initial seed (set of samples on which the model is trained before the first optimisation round, see Appendix B) might be too small for the substitute model to learn how to rank the most valuable samples. In all configurations with a query budget of 10,000 or more, there is an optimisation technique that improves the baseline attack. Hence, applying those techniques, especially active learning, is most useful for mid-range query budgets.

With active learning, our data-free attack reaches more than 75% of fidelity and accuracy with a query budget of only 10,000 queries. The state-of-the-art results presented in Section V, are, while having similar effectiveness performance, three orders of magnitude less efficient. Therefore, we are the first to demonstrate that even with the weakest attacker's knowledge, substitute training attacks can be effectively performed with less data than the target model was trained on.

Conclusion. *Query optimisation techniques are the most beneficial for mid-range query budgets starting from 10,000 queries. Smaller query budgets (1,000 and 5,000) are not enough for a substitute model to learn how to rank the most valuable samples.*

Data-free attacks on image classifiers can be effective with query budgets smaller than the original training set. Weak attacker's knowledge does not require to be overcompensated with significantly increased querying capabilities.

V. COMPARISON WITH THE STATE-OF-THE-ART

We conclude our attack analysis by comparing the performance of our attacks with papers that performed model stealing attacks against CIFAR-10 classifiers. We summarise the performance scores in Table XII for query budgets below 40,000 and in Table XIII for query budgets from 45,000. Additionally, we provide information about the attacker's data, target model outputs, and the performance of the target model on the CIFAR-10 test set. For attack effectiveness evaluation,

TABLE XI: Performance of ResNet-18 substitute models trained on the artificial data with different query optimisation techniques. In parentheses, we report the difference with the non-optimised attack.

Query optimisation ↓	Target → QB	SimpleNet		ResNet-34 (from scratch)		ResNet-34 (transfer learning)	
		Acc	*Fid*	*Acc*	*Fid*	*Acc*	*Fid*
Active learning	1k	64.62% (-1.58%)	64.09% (-1.77%)	68.30% (+2.24%)	68.24% (+1.92%)	68.02% (+1.19%)	68.18% (+1.08%)
	5k	70.65% (-0.46%)	70.39% (-0.16%)	72.75% (+3.31%)	72.66% (+3.14%)	69.86% (-2.11%)	69.99% (-2.15%)
	10k	76.64% (+3.60%)	76.14% (+3.28%)	78.51% (+4.79%)	78.55% (+5.15%)	75.39% (+4.93%)	75.61% (+4.98%)
	20k	77.98% (+1.26%)	77.65% (+1.23%)	78.91% (+2.60%)	78.79% (+2.60%)	73.47% (+4.46%)	73.62% (+4.46%)
Adversarial augmentation	1k	63.60% (-2.60%)	63.03% (-2.83%)	65.69% (-0.37%)	65.83% (-0.49%)	64.45% (-2.38%)	64.78% (-2.32%)
	5k	68.16% (-2.95%)	67.53% (-3.02%)	70.75% (+1.31%)	70.46% (+0.94%)	68.84% (-3.13%)	68.92% (-3.22%)
	10k	69.47% (-3.57%)	69.07% (-3.79%)	76.16% (+2.44%)	76.24% (+2.84%)	70.28% (-0.18%)	70.44% (-0.19%)
	20k	75.39% (-1.33%)	75.26% (-1.16%)	76.29% (-0.02%)	76.48% (+0.29%)	72.04% (+3.03%)	72.14% (+2.98%)
	45k	78.05% (-0.83%)	77.73% (-1.05%)	77.96% (+0.86%)	78.18% (+0.90%)	74.39% (+0.58%)	74.54% (+0.54%)
Active adversarial augmentation	1k	63.53% (-2.67%)	63.03% (-2.83%)	65.71% (-0.35%)	65.44% (-0.88%)	64.23% (-2.60%)	64.49% (-2.61%)
	5k	68.58% (-2.53%)	67.73% (-2.82%)	74.34% (+4.90%)	74.20% (+4.68%)	69.39% (-2.58%)	69.45% (-2.69%)
	10k	73.84% (+0.80%)	73.71% (+0.85%)	75.70% (+1.98%)	75.72% (+2.32%)	74.57% (+4.11%)	74.86% (+4.23%)
	20k	74.84% (-1.88%)	74.62% (-1.80%)	77.90% (+1.59%)	77.88% (+1.69%)	74.41% (+5.40%)	74.74% (+5.58%)
	45k	79.42% (+0.54%)	79.16% (+0.38%)	77.41% (+0.31%)	77.58% (+0.30%)	75.50% (+1.69%)	75.71% (+1.71%)

TABLE XII: Comparison of attacks implemented in this work with the state-of-the-art for query budgets below 40,000.

Query budget	Data	Outputs	Paper	Target Acc.	Sub. Acc.	Sub. Fid.	Queries	Query score
< 2.5k	Original	N/A	[13]	>91%		77.47%	1.6k	0.032
	Original	N/A	[44]	N/A		76.40%	2k	0.04
	Original	Gradients	[16]	est. 90%	est. 88%		1k	0.02
	Original	Probabilities	[7]	95.75%	90.63%	91.39%	1k	0.02
	Original	Probabilities	[31]	93.90%	82.28%		1k	0.02
	Original	Labels	this work	93.61%	84.51%	84.23%	1k	0.02
	PD	Labels	this work	93.61%	71.58%	71.99%	1k	0.02
	Artificial	Labels	this work	93.61%	68.30%	68.24%	1k	0.02
< 10k	Original	N/A	[13]	>91%		77.96%	6.4k	0.128
	Original	N/A	[44]	N/A		80.25%	4k	0.08
	Original	Probabilities+XAI	[28]	92.03%	est. 77%		8k	0.16
	Original	Probabilities	[7]	95.75%	93.29%	93.99%	4k	0.08
	Original	Probabilities	[31]	93.90%	89.74%		4k	0.08
	Original	Labels+XAI	[37]	92.03%		72.50%	8k	0.16
	Original	Labels	[34]	90.45%	58.95%	59.90%	8k	0.16
	Original	Labels	this work	93.61%	90.22%	89.20%	5k	0.1
	PD	Labels	this work	93.61%	82.96%	83.03%	5k	0.1
	NPD	Labels	[34]	90.45%	31.21%	32.20%	8k	0.16
	NPD	Labels	[45]	91.82%	71.65%		8.4k	0.168
	NPD	Labels	[49]	N/A		71.45%	8k	0.16
	Artificial	Labels	this work	93.61%	74.34%	74.20%	5k	0.1
< 20k	Original	N/A	[13]	>91%		83.61%	12.8k	0.256
	Original	Probabilities	[31]	93.90%	92.50%		10k	0.2
	Original	Labels	this work	93.61%	93.05%	91.93%	10k	0.2
	PD	Labels	this work	93.61%	85.85%	85.95%	10k	0.2
	NPD	Probabilities	[10]	N/A		77.29%	10k	0.2
	NPD	Probabilities	[21]	N/A		77.80%	10k	0.2
	NPD	Probabilities	[43]	N/A	est. 69%	est. 68%	15k	0.3
	NPD	Labels	[10]	N/A		64.23%	10k	0.2
	NPD	Labels	[21]	N/A		75.40%	10k	0.2
	NPD	Labels	[49]	N/A		74.06%	12k	0.24
	Artificial	Labels	this work	93.61%	78.51%	78.55%	10k	0.2
< 40k	Original	N/A	[13]	>91%		84.12%	25.6k	0.512
	Original	Labels	[41]	93.70%		94.09%	est. 25k	0.5
	Original	Labels	this work	93.61%	94.61%	92.89%	20k	0.4
	PD	Labels	this work	93.61%	88.21%	88.69%	20k	0.4
	NPD	Probabilities	[43]	N/A	est. 71%	est. 70%	30k	0.6
	NPD	Labels	[10]	N/A		78.36%	30k	0.6
	NPD	Labels	[26]	N/A	80.90%		30k	0.6
	NPD	Labels	[27]	91.56%	80.47%	82.14%	30k	0.6
	NPD	Labels	[46]	92.18%	83.06%	84.12%	30k	0.6
	NPD	Labels	[49]	N/A		80.67%	20k	0.4
	Artificial	Labels	this work	93.61%	78.91%	78.79%	20k	0.4

we report accuracy and fidelity of substitute models. Efficiency is represented by the exact number of queries used by an attack and *query score*. The latter shows how many queries are needed per a target model's training sample, i.e. it is a relation between the attacker's dataset and the CIFAR-10 training dataset (50,000 samples). If some information is not reported by a paper, we mark it as N/A. Some values are estimated based on the information provided in the papers (see Appendix C for more details). For our work, we report scores for attacks using query optimisation, ResNet-18 as a substitute model, and ResNet-34 trained from scratch as the target model.

We group attacks with similar query budgets in Table XII (shown in the first column), the same data type (second column), and target outputs (third column). In general, as attacks from the same group have the most similar attacker's strength, a performance comparison should only be launched within

a group. However, since often there is no work matching ours, we also compare our attacks with ones having stronger attacker profiles. Below, we summarise our main findings from Table XII for each query budget.

Query budget <2.5k. All prior attacks with this query budget relied on availability of original data. Therefore, we are the first one to demonstrate feasibility of attacks with this query budget for PD and artificial data. Our attack with original data is outperformed by two prior works, both having a stronger assumption about the target outputs, namely probabilities [7] and gradients [16].

Query budget <10k. From a total of seven prior attacks using original data, only one achieves higher performance than ours [7]. However, it relies on probabilities whereas we only use labels. Further, even our PD attack outperforms five of prior works using original data. Moreover, our data-free attack outperforms all previous NPD attacks, while using less queries

TABLE XIII: Comparison of attacks implemented in this work with the state-of-the-art for query budgets from 45,000.

Data	Outputs	Paper	Target Acc.	Sub. Acc.	Sub. Fid.	Queries	Query score
Original	Labels	this work	93.61%	95.29%	93.40%	45k	0.9
PD	Probabilities	[9]	95.30%	90.00%		269k	5.38
	Labels	this work	93.61%	89.71%	90.03%	45k	0.9
NPD	Probabilities	[43]	N/A	est. 76%	est. 75%	50k	1
	Probabilities	[9]	95.30%	94.00%		3.4m	68
	Probabilities	[12]	94.60%	88.20%		100k	2
	Probabilities	[19]	82.50%	79.00%		est. >50k	est. >1
	Labels	[14]	90.48%	89.59%		est. >1m	est. >20
	Labels	[10]	N/A		81.57%	100k	2
	Labels	[10]	N/A		84.99%	120k	2.4
	Labels	[12]	94.60%	53.60%		100k	2
Artificial	N/A	[35]	N/A	61.90%		60k	1.2
	Probabilities+XAI	[23]	95.50%	72.10%		1m	20
	Probabilities+XAI	[23]	95.50%	90.40%		10m	200
	Probabilities+XAI	[23]	95.50%	92.30%		20m	400
	Probabilities	[17]	92.26%	89.85%		30m	600
	Probabilities	[22]	95.50%	88.10%		20m	400
	Probabilities	[22]	95.50%	89.90%		30m	600
	Probabilities	[24]	95.50%	91.24%		8m	160
	Probabilities	[15]	91.93%	80.79%		est. >1m	est. >20
	Probabilities	[42]	93.00%	88.50%	92.10%	20m	400
	Probabilities	[47]	95.50%	91.34%		20m	400
	Probabilities	[33]	95.54%	94.02%		20m	400
	Probabilities	[48]	90.71%	84.70%		13m	260
	Probabilities	[39]	81.74%	70.30%		100k	2
	Labels+XAI	[36]	95.54%	est. 90%		20m	400
	Labels	[24]	95.50%	84.51%		8m	160
	Labels	[15]	91.93%	69.64%		est. >1m	est. >20
	Labels	[47]	95.50%	78.72%		20m	400
	Labels	[48]	90.71%	29.58%		6m	120
	Labels	[33]	95.54%	87.93%		8m	160
	Labels	[38]	82.50%	73.10%		1.6m	32
	Labels	[38]	82.50%	73.81%		17.6m	352
	Labels	this work	93.61%	78.52%	79.06%	45k	0.9

(5,000 compared to 8,000). As earlier, we are the first to use PD or artificial data with such query budget.

Query budget <20k. Among attacks using original data, our has the highest accuracy and fidelity scores. As was the case before, none of the prior studies used PD or artificial data. However, using artificial data, we achieve higher accuracy and fidelity than all six previous NPD attacks.

Query budget <40k. One of the prior attacks using original data has slightly better fidelity than ours [41]. Besides, the limitations artificial data become more visible compared to NPD data—most of the prior works outperform our attack, although the difference in performance remains rather small (up to 5.5%). There are still no other PD or data-free attacks within this query budget to compare our work with.

Notably, we are the first to demonstrate feasibility of data-free attacks with query budgets below 40,000. Besides, for the same query budgets, none of the prior works launched attacks using PD data. Meanwhile, our PD attacks outperform some prior works that use original data, more queries, and more revealing target outputs. Furthermore, our data-free attack has better performance than any NPD attack for query budgets below 20,000.

We compare the last group of attacks that used at least 45,000 queries in Table XIII. No prior works used that many queries for attacks based on the original data. For PD data, the only prior work [9] marginally outperforms our attack, while using significantly more queries and probabilities as target outputs.

The most significant group of papers with large query budgets uses artificial data for training a substitute model. Among six data-free attack that use labels as target outputs, our attack is marginally (by 0.2%) outperformed by one prior work [47] and more significantly (by 6-9.5%) by two other works [24], [33]. Both of them used 8 million of queries, which is 160 times more than the size of the target model training set. Meanwhile, we only utilised 45,000 queries, proposing the most efficient data-free attack as of now.

Overall, our data-free attack shows promising results for future work. As optimising artificial data quality was not the prime goal of this work, we expect to reach higher scores with further research.

VI. Discussion

In this section, we discuss two additional aspects important for attack evaluation, namely the transferability of a substitute model and defences against model stealing.

A. Transferability

In this work, we focused on accuracy and fidelity as performance metrics. However, transferability can be very insightful in estimating how similar two models behave close to the decision boundary. The main difficulty with this metric is that there is no unified way to measure it. Related work uses different adversarial example crafting methods and various hyperparameters defining the strength of the perturbation. For this reason, we omitted to report transferability in Section V, as the scores would simply be incomparable.

However, as two of our query optimisation techniques utilise adversarial examples, we additionally measured transferability to see if these optimisations could improve the scores. For this purpose, we used exactly the same Deepfool algorithm as in adversarial augmentation [13]. However, the results were poor, varying from 0% to 16%. We speculate that the reason is that the perturbations Deepfool makes are too minute to impact the decision boundary and improve the transferability significantly. Using a method with stronger perturbations would likely result in a better score. However, it has to be considered that while it is likely easier to achieve a high transferability score if the adversarial examples are created with stronger perturbation, a commonly agreed upper bound of acceptable perturbation should be established to enable the above-mentioned comparability.

B. Defences against Model Stealing

Defences against model stealing can be categorised into proactive and reactive [6]. Most common proactive defences are based on adding noise to model inputs or outputs to perturb the information obtained from the target model and mislead the substitute training process. In this work, we only assume labels as outputs, perturbing which will directly lead to target model performance degradation on its original classification task (unlike confidence score perturbations). In safety-critical settings and when the quality of service is crucial, this approach is not feasible. For this reason, we evaluated attacks against defences that incur an acceptable loss of correctness and perturb only a small fraction of predictions. While an acceptable value depends on the specific scenario, we set approximately 1% of the predicted labels to be modified. As a result, outputs of unprotected and protected models were identical for approximately 99% of data. We tested input perturbation defence by Grana [54] and a region-based classifier by Cao and Gong [55] as an output perturbation (output perturbation defences designed against model stealing assume confidence scores as outputs [56]–[58]). These defences have shown to be ineffective—in most of the cases, the performance of the substitute model did not decrease by more than 1%.

Another type of proactive defence is (re-)training the target model from scratch [6] instead of building on a pre-trained model. However, as we showed in Section IV-D, training from scratch also does not protect the target models. Therefore, we conclude that none of the current proactive defences can defend against attacks on image classifiers that use only labels.

Reactive defences include ownership verification methods (watermarking, fingerprinting) and monitors. While ownership verification methods might be effective against our attacks, they can not protect a model from being stolen and can only serve as evidence. Monitors, which aim to detect an ongoing attack by tracking suspicious queries, should, in most settings, not mark problem-domain-like data as suspicious, as it can lead to false positives, harming API clients. In our work, we use original, problem domain, and problem-domain-like data generated by a diffusion model. Therefore, none of these data distributions should be marked as malicious.

VII. Conclusion

In this paper, we presented new insights into the influence different factors have on the success rate of substitute training attacks. In particular, we demonstrated that attacks benefit from (i) targeting better-performing models, (ii) adopting an architecture that fits the quality and quantity of the attacker's data, (iii) using the target model's training strategy, (iv) having more complex data, and (v) optimising queries, in particular, for data-free attacks. While most of these findings suggest that attacks can be improved by some modifications, our result about the target model performance implies that previous attacks could have been underestimated and might produce even higher scores when targeting better models.

A key takeaway from our work is that attackers can do better—even with less effort. We adopted a simple diffusion-based data generation approach, performed fair hyperparameter tuning on every step of the attack and boosted the performance with query optimisation techniques. As a result, we attained a data-free attack with a query budget of 10,000 queries and performance comparable with those using millions of queries. With original or problem-domain data, we outperform the state-of-the-art, even having weaker assumptions about model outputs while achieving a better efficiency score. *Finally, if attackers indeed can do better—can defenders do better as well?*

REFERENCES

[1] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 601–618. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer

[2] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff Nets: Stealing Functionality of Black-Box Models," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 4949–4958. [Online]. Available: https://ieeexplore.ieee.org/document/8953839/

[3] S. Szyller, V. Duddu, T. Gröndahl, and N. Asokan, "Good Artists Copy, Great Artists Steal: Model Extraction Attacks Against Image Translation Models," 2021, version Number: 2. [Online]. Available: https://arxiv.org/abs/2104.12623

[4] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer, "Thieves on Sesame Street! Model Extraction of BERT-based APIs," in *International Conference on Learning Representations*, 2020.

[5] K. Chen, S. Guo, T. Zhang, X. Xie, and Y. Liu, "Stealing Deep Reinforcement Learning Models for Fun and Profit," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. Virtual Event Hong Kong: ACM, May 2021, pp. 307–319. [Online]. Available: https://dl.acm.org/doi/10.1145/3433210.3453090

[6] D. Oliynyk, R. Mayer, and A. Rauber, "I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–41, Dec. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3595292

[7] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High Accuracy and High Fidelity Extraction of Neural Networks," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1345–1362. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski

[8] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical Black-Box Attacks against Machine Learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. Abu Dhabi United Arab Emirates: ACM, Apr. 2017, pp. 506–519. [Online]. Available: https://dl.acm.org/doi/10.1145/3052973.3053009

[9] J. R. Correia-Silva, R. F. Berriel, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data," in *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro: IEEE, Jul. 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8489592/

[10] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy, "ActiveThief: Model Extraction Using Active Learning and Unannotated Public Data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 865–872, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5432

[11] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "PRADA: Protecting Against DNN Model Stealing Attacks," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. Stockholm, Sweden: IEEE, Jun. 2019, pp. 512–527. [Online]. Available: https://ieeexplore.ieee.org/document/8806737/

[12] B. G. Atli, S. Szyller, M. Juuti, S. Marchal, and N. Asokan, "Extraction of Complex DNN Models: Real Threat or Boogeyman?" in *Engineering Dependable and Secure Machine Learning Systems*, O. Shehory, E. Farchi, and G. Barash, Eds. Cham: Springer International Publishing, 2020, pp. 42–57.

[13] L. Pengcheng, J. Yi, and L. Zhang, "Query-Efficient Black-Box Attack by Active Learning," in *2018 IEEE International Conference on Data Mining (ICDM)*. Singapore: IEEE, Nov. 2018, pp. 1200–1205. [Online]. Available: https://ieeexplore.ieee.org/document/8594968/

[14] I. Mosafi, E. O. David, and N. S. Netanyahu, "Stealing Knowledge from Protected Deep Neural Networks Using Composite Unlabeled Data," in *2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8851798/

[15] X. Yuan, L. Ding, L. Zhang, X. Li, and D. O. Wu, "ES Attack: Model Stealing Against Deep Neural Networks Without Data Hurdles," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 5, pp. 1258–1270, Oct. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9726514/

[16] S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt, "Model Reconstruction from Model Explanations," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*. Atlanta GA USA: ACM, Jan. 2019, pp. 1–9. [Online]. Available: https://dl.acm.org/doi/10.1145/3287560.3287562

[17] S. Kariyappa, A. Prakash, and M. 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 (CVPR)*, Jun. 2021, pp. 13 814–13 823.

[18] N. Roberts, V. U. Prabhu, and M. McAteer, "Model Weight Theft With Just Noise Inputs: The Curious Case of the Petulant Attacker," in *ICML Workshop on the Security and Privacy of Machine Learning*, Long Beach, CA, Jun. 2019. [Online]. Available: https://icml2019workshop.github.io/

[19] A. Bărbălău, A. Cosma, R. T. Ionescu, and M. Popescu, "Black-Box ripper: copying black-box models using generative evolutionary algorithms," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020, event-place: Vancouver, BC, Canada.

[20] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples," in *Proceedings 2020 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2020. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2020/02/24178.pdf

[21] X. Gong, Y. Chen, W. Yang, G. Mei, and Q. Wang, "InverseNet: Augmenting Model Extraction Attacks with Training Data Inversion," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 2439–2447. [Online]. Available: https://www.ijcai.org/proceedings/2021/336

[22] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-Free Model Extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 4771–4780.

[23] T. Miura, T. Shibahara, and N. Yanai, "MEGEX: Data-Free Model Extraction Attack Against Gradient-Based Explainable AI," in *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*. Singapore Singapore: ACM, Jul. 2024, pp. 56–66. [Online]. Available: https://dl.acm.org/doi/10.1145/3665451.3665533

[24] S. Sanyal, S. Addepalli, and R. V. Babu, "Towards Data-Free Model Stealing in a Hard Label Setting," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 15 263–15 272. [Online]. Available: https://ieeexplore.ieee.org/document/9880326/

[25] X. Zhang, C. Fang, and J. Shi, "Thief, Beware of What Get You There: Towards Understanding Model Extraction Attack," 2021, version Number: 1. [Online]. Available: https://arxiv.org/abs/2104.05921

[26] Y. Wang and X. Lin, "Enhance Model Stealing Attack via Label Refining," in *2022 7th International Conference on*

Intelligent Computing and Signal Processing (ICSP). Xi'an, China: IEEE, Apr. 2022, pp. 1040–1043. [Online]. Available: https://ieeexplore.ieee.org/document/9778562/

[27] Y. Wang, J. Li, H. Liu, Y. Wang, Y. Wu, F. Huang, and R. Ji, "Black-Box Dissector: Towards Erasing-Based Hard-Label Model Stealing Attack," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Berlin, Heidelberg: Springer-Verlag, 2022, pp. 192–208, event-place: Tel Aviv, Israel. [Online]. Available: https://doi.org/10.1007/978-3-031-20065-6_12

[28] A. Yan, R. Hou, X. Liu, H. Yan, T. Huang, and X. Wang, "Towards explainable model extraction attacks," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 9936–9956, Nov. 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/int.23022

[29] Y. Xie, M. Huang, X. Zhang, C. Dong, W. Susilo, and X. Chen, "GAME: Generative-Based Adaptive Model Extraction Attack," in *Computer Security – ESORICS 2022*, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds. Cham: Springer International Publishing, 2022, pp. 570–588.

[30] Y. Chen, R. Guan, X. Gong, J. Dong, and M. Xue, "D-DAE: Defense-Penetrating Model Extraction Attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2023, pp. 382–399. [Online]. Available: https://ieeexplore.ieee.org/document/10179406/

[31] Y. He, G. Meng, K. Chen, X. Hu, and J. He, "DRMI: A Dataset Reduction Technology based on Mutual Information for Black-box Attacks," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 1901–1918. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/he-yingzhe

[32] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. D. Cristofaro, M. Fritz, and Y. Zhang, "ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 4525–4542. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/liu-yugeng

[33] J. Rosenthal, E. Enouen, H. V. Pham, and L. Tan, "DisGUIDE: Disagreement-Guided Data-Free Model Extraction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, pp. 9614–9622, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26150

[34] A. Yan, H. Yan, L. Hu, X. Liu, and T. Huang, "Holistic Implicit Factor Evaluation of Model Extraction Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4678–4689, Nov. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9999271/

[35] J. Zhang, B. Li, J. Xu, S. Wu, S. Ding, L. Zhang, and C. Wu, "Towards Efficient Data Free Blackbox Adversarial Attack," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 15 094–15 104. [Online]. Available: https://ieeexplore.ieee.org/document/9880352/

[36] A. Yan, R. Hou, H. Yan, and X. Liu, "Explanation-based data-free model extraction attacks," *World Wide Web*, vol. 26, no. 5, pp. 3081–3092, Sep. 2023. [Online]. Available: https://link.springer.com/10.1007/s11280-023-01150-6

[37] A. Yan, T. Huang, L. Ke, X. Liu, Q. Chen, and C. Dong, "Explanation leaks: Explanation-guided model extraction attacks," *Information Sciences*, vol. 632, pp. 269–284, Jun. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002002552300316X

[38] P. Yang, Q. Wu, and X. Zhang, "Efficient Model Extraction by Data Set Stealing, Balancing, and Filtering," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 717–22 725, Dec. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10214537/

[39] Z. Lin, K. Xu, C. Fang, H. Zheng, A. Ahmed Jaheezuddin, and J. Shi, "QUDA: Query-Limited Data-Free Model Extraction," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*. Melbourne VIC Australia: ACM, Jul. 2023, pp. 913–924. [Online]. Available: https://dl.acm.org/doi/10.1145/3579856.3590336

[40] Y. Liu, J. Luo, Y. Yang, X. Wang, M. Gheisari, and F. Luo, "ShrewdAttack: Low Cost High Accuracy Model Extraction," *Entropy*, vol. 25, no. 2, p. 282, Feb. 2023. [Online]. Available: https://www.mdpi.com/1099-4300/25/2/282

[41] D. Pape, S. Däubener, T. Eisenhofer, A. E. Cinà, and L. Schönherr, "On the Limitations of Model Stealing with Uncertainty Quantification Models," in *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=ONRFHoUzNk

[42] Y. Liu, R. Wen, M. Backes, and Y. Zhang, "Efficient Data-Free Model Stealing with Label Diversity," 2024, version Number: 1. [Online]. Available: https://arxiv.org/abs/2404.00108

[43] K. Khaled, G. Nicolescu, and F. G. De Magalhaes, "Careful What You Wish For: on the Extraction of Adversarially Trained Models," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*. Fredericton, NB, Canada: IEEE, Aug. 2022, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/document/9851981/

[44] S. Zhao, K. Chen, M. Hao, J. Zhang, G. Xu, H. Li, and T. Zhang, "Extracting Cloud-based Model with Prior Knowledge," 2023, version Number: 4. [Online]. Available: https://arxiv.org/abs/2306.04192

[45] P. Karmakar and D. Basu, "Marich: A Query-efficient Distributionally Equivalent Model Extraction Attack," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 72 412–72 445. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/e5440ffceaf4831b5f98652b8a2

[46] A. Jindal, V. Goyal, S. Anand, and C. Arora, "Army of Thieves: Enhancing Black-Box Model Extraction via Ensemble based sample selection," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 3811–3820. [Online]. Available: https://ieeexplore.ieee.org/document/10484430/

[47] J. Beetham, N. Kardan, A. S. Mian, and M. Shah, "Dual Student Networks for Data-Free Model Stealing," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=VE1s3e5xriA

[48] C. Hong, J. Huang, R. Birke, and L. Y. Chen, "Exploring and Exploiting Data-Free Model Stealing," in *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part V*. Berlin, Heidelberg: Springer-Verlag, 2023, pp. 20–35, event-place: Turin, Italy. [Online]. Available: https://doi.org/10.1007/978-3-031-43424-2_2

[49] W. Yang, X. Gong, Y. Chen, Q. Wang, and J. Dong, "SwiftTheft: A Time-Efficient Model Extraction Attack Framework Against Cloud-Based Deep Neural Networks," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 90–100, Jan. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10410588/

[50] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Denver Colorado USA: ACM, Oct. 2015, pp. 1322–1333. [Online]. Available: https://dl.acm.org/doi/10.1145/2810103.2813677

[51] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, "Exploring Connections Between Active Learning and Model Extraction," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1309–1326. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/chandrasekaran

[52] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, Tech. Rep., 2009.

[53] S. H. Hasanpour, M. Rouhani, M. Fayyaz, and M. Sabokrou, "Lets keep it simple, Using simple architectures to outperform deeper and more complex architectures," 2016, version Number: 8. [Online]. Available: https://arxiv.org/abs/1608.06037

[54] J. Grana, "Perturbing Inputs to Prevent Model Stealing," in *2020 IEEE Conference on Communications and Network Security (CNS)*. Avignon, France: IEEE, Jun. 2020, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9162336/

[55] X. Cao and N. Z. Gong, "Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. Orlando FL USA: ACM, Dec. 2017, pp. 278–287. [Online]. Available: https://dl.acm.org/doi/10.1145/3134600.3134606

[56] T. Orekondy, B. Schiele, and M. Fritz, "Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks," in *International Conference on Learning Representations*, ser. ICLR, Virtual Event, Apr. 2020. [Online]. Available: https://iclr.cc/virtual_2020/poster_SyevYxHtDB.html

[57] J. Chen, C. Wu, S. Shen, X. Zhang, and J. Chen, "DAS-AST: Defending Against Model Stealing Attacks Based on Adaptive Softmax Transfor-

mation," in *Information Security and Cryptology*, Y. Wu and M. Yung, Eds. Cham: Springer International Publishing, 2021, pp. 21–36.

[58] M. Mazeika, B. Li, and D. Forsyth, "How to Steer Your Adversary: Targeted and Efficient Model Stealing Defenses with Gradient Redirection," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, Jul. 2022, pp. 15 241–15 254. [Online]. Available: https://proceedings.mlr.press/v162/mazeika22a.html

[59] M. Ducoffe and F. Precioso, "Adversarial Active Learning for Deep Networks: a Margin Based Approach," 2018, version Number: 1. [Online]. Available: https://arxiv.org/abs/1802.09841

[60] O. Sener and S. Savarese, "Active Learning for Convolutional Neural Networks: A Core-Set Approach," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1aIuk-RW

[61] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2574–2582. [Online]. Available: http://ieeexplore.ieee.org/document/7780651/

A. Attack Setup Overview

Figure 1 illustrates an overview of all conducted attacks. Overall, we trained 180 substitute models against each target model.

B. Query Optimisation Methods

1) Active Learning: In active learning, there is a model that has to be trained on data labelled by an oracle. For model stealing attacks, a substitute model $\hat{f}$ corresponds to that trainable model, and the target model f corresponds to the oracle. Further, both seed (labelled data) and pool (unlabelled) data belong to the attacker's data. Hence, at each active learning round, the target model labels a certain amount of the attacker's data.

The active learning optimisation strategy used in this work is a slightly modified version of an approach from previous work by Pal et al. [10]. The authors compared several active learning strategies for image and text classification. We selected the strategy with the best performance rate on image classification tasks. It combines two active learning algorithms, namely the Deepfool Active Learning (DFAL) [59], and the κ-center algorithm [60]. Pal et al. [10] combined the algorithms in the following way. In each active learning round, they applied DFAL to select q samples, where q corresponds to the total query budget. Then they applied κ-center to select k most suitable samples out of q. In this work, in order to decrease the computation time and make the attack more efficient, we made the following changes: (i) reducing the number of samples selected by DFAL to $2k$, and (ii) splitting the pool into several sub-pools, so that at each round data is selected from a single sub-pool. The latter modification is applied to all optimisation techniques.

We summarise the stealing process with active learning in Algorithm 1. As for any model stealing attack, we need target and substitute models, the attacker's data (pool), and the query budget. Additionally, unlike in the previous work,

we have two hyperparameters specific to attacks with query optimisation: seed size and the number of rounds, which are tuned on a validation dataset of the attacker's dataset. We assume that the seed is randomly selected from the pool, and the substitute model is trained on it. The value of k (number of samples to select per round) is calculated based on the query budget, seed size, and the number of rounds. At each round, we consequently apply DFAL and κ-center, add selected samples to the seed, and, as suggested in previous work, train the substitute model from scratch on the augmented dataset.

Algorithm 1: Active learning attack

Input: target model f, substitute model $\hat{f}$, data pool D, query budget q, seed size s, number of rounds r

Output: $\hat{f}$

1   $S_0 \leftarrow$ select randomly s samples from D ;
2   $y_0 \leftarrow f(S_0)$;
3   $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_0, y_0)$;
4   $k \leftarrow \dfrac{q - |S_0|}{r}$;
5   **for** $i = 1$ *to* r **do**
6      $x_1 \ldots x_{2k} \leftarrow \texttt{DFAL}(\hat{f}, D, 2k)$;
7      $x'_1 \ldots x'_k \leftarrow \kappa\texttt{-center}(\hat{f}, S_{i-1}, \{x_1 \ldots x_{2k}\}, k)$;
8      $D \leftarrow D \setminus \{x'_1, \ldots, x'_k\}$;
9      $S_i \leftarrow S_{i-1} \cup \{x'_1, \ldots, x'_k\}$;
10     $y_i \leftarrow y_{i-1} \cup \{f(x'_1), \ldots, f(x'_k)\}$;
11     $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_i, y_i)$;
12 **end**

---

2) Adversarial Augmentation: The second query optimisation technique used in this work is adversarial augmentation. Instead of selecting the most promising samples as active learning, adversarial augmentation modifies samples to make them more information-revealing. Since adversarial examples lay close to the decision boundary, it is assumed that they should help better approximate the target model's

Target Model (3)
1. SimpleNet
2. ResNet-34 (from scratch)
3. ResNet-34 (transfer learning)

3 target models
vs
180 substitute models

Substitute Model (3)		Attacker's Dataset (3)		Query Budget (5)		Attack Strategy (4)
1. SimpleNet	X	1. CIFAR10	X	1. 1,000 2. 5,000 3. 10,000	X	1. Random
2. ResNet-18		2. CINIC10		4. 20,000		2. Active learning
						3. Adversarial augmentation
3. ResNet-34		3. Artificial		5. 45,000		4. Active adversarial augmentation

Fig. 1: Overview of attack setups examined in this work.

decision boundary. Similarly to active learning, we picked an adversarial example crafting strategy based on previous work. Pengcheng et al. [13] compared different adversarial crafting techniques for model stealing targeting image classifiers. As Deepfool's [61] performance was the most promising, we selected it for our adversarial augmentation attack.

We summarise our implemented attack in Algorithm 2. The input parameters are the same as for the active learning attack. At each augmentation round, we randomly select samples from the pool, which are then augmented with their Deepfool adversarial examples. Both clean and adversarial samples are then labelled by the target model and added to the seed. The substitute model is trained from scratch after each round.

Algorithm 2: Adversarial augmentation attack

Input: target model f, substitute model $\hat{f}$, data pool D, query budget q, seed size s, number of rounds r

Output: $\hat{f}$

1   $S_0 \leftarrow$ select randomly s samples from D ;
2   $y_0 \leftarrow f(S_0)$;
3   $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_0, y_0)$;
4   $k \leftarrow \dfrac{q - |S_0|}{r}$;
5   **for** $i = 1$ *to* r **do**
6     $x'_1 \ldots x'_{\frac{k}{2}} \leftarrow$ select randomly $\frac{k}{2}$ samples from D ;
7     $x'_{\frac{k}{2}+1} \ldots x'_k \leftarrow$
     $\texttt{Deepfool}(\hat{f}, x'_1), \ldots, \texttt{Deepfool}(\hat{f}, x'_{\frac{k}{2}})$;
8     $D \leftarrow D \setminus \{x'_1, \ldots, x'_k\}$;
9     $S_i \leftarrow S_{i-1} \cup \{x'_1, \ldots, x'_k\}$;
10    $y_i \leftarrow y_{i-1} \cup \{f(x'_1), \ldots, f(x'_k)\}$;
11    $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_i, y_i)$;
12 **end**

---

3) Active Adversarial Augmentation: The combined attack is shown in Algorithm 3. The algorithm repeats the behaviour of the adversarial augmentation attack (Algorithm 2) with one difference: line 6 in Algorithm 2 is replaced with lines 6 and 7 in Algorithm 3. Instead of randomly selecting $\frac{k}{2}$ samples out of the pool, they are now selected using active learning algorithms.

C. Value Estimation for the State-of-the-Art Comparison

As mentioned in Section V, some of the values in Table XII and Table XIII were estimated, as the exact values were not provided by the authors. Below, we explain how each value was configured and which information was used for the estimation.

- The target and substitute accuracy for [16], substitute accuracy for [28], [36], as well as substitute accuracy and fidelity for [43] were estimated from plots, as no actual scores were reported.
- The number of queries used by [41] is estimated as 25,000 queries as the authors mention using half of the

---

Algorithm 3: Active adversarial augmentation attack

Input: target model f, substitute model $\hat{f}$, data pool D, query budget q, seed size s, number of rounds r

Output: $\hat{f}$

1   $S_0 \leftarrow$ select randomly s samples from D ;
2   $y_0 \leftarrow f(S_0)$;
3   $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_0, y_0)$;
4   $k \leftarrow \dfrac{q - |S_0|}{r}$;
5   **for** $i = 1$ *to* r **do**
6     $x_1 \ldots x_k \leftarrow \texttt{DFAL}(\hat{f}, D, k)$;
7     $x'_1 \ldots x'_{\frac{k}{2}} \leftarrow \kappa\texttt{-center}(\hat{f}, S_{i-1}, \{x_1 \ldots x_k\}, \frac{k}{2})$;
8     $x'_{\frac{k}{2}+1} \ldots x'_k \leftarrow$
     $\texttt{Deepfool}(\hat{f}, x'_1), \ldots, \texttt{Deepfool}(\hat{f}, x'_{\frac{k}{2}})$;
9     $D \leftarrow D \setminus \{x'_1, \ldots, x'_k\}$;
10    $S_i \leftarrow S_{i-1} \cup \{x'_1, \ldots, x'_k\}$;
11    $y_i \leftarrow y_{i-1} \cup \{f(x'_1), \ldots, f(x'_k)\}$;
12    $\hat{f} \leftarrow \texttt{Train}(\hat{f}, S_i, y_i)$;
13 **end**

(CIFAR-10) training set for an attack, but explicitly the number is not confirmed.

- The number of queries used by [19] was not reported. However, the authors used CIFAR-100, which contains 50,000 samples, as a *starting point* for their evolutionary algorithm that creates new query images. They claimed that optimising the query budget was not a priority, so we can assume that the number of queries could be significantly larger than 50,000. However, as that is only a speculation, we conservatively estimate that they used "more than 50,000" samples.
- The authors of [14] also do not provide information about the total number of queries. However, they have an iterative algorithm that generates 1,000,000 samples at each round. Assuming that there should be at least one round, we estimated the number of queries as "larger than 1,000,000".
- The query budget was also not given by [15]. However, the authors estimated the price of their attack on Amazon Web Services[2] to be \$360,000. Pricing from 2023 suggests that the price per query for the first million queries is usually around \$0.001, getting cheaper for the subsequent millions. That means that a million queries cost \$1000, and with this price, the authors could have made more than 300 million queries. However, since the prices could differ back then, we lowered our estimate to 1 million.
- Some of the papers did not report the architecture of the target model or used a custom architecture without reporting the number of trainable parameters. In these

---

[2]https://aws.amazon.com/rekognition/pricing/

TABLE XIV: Performance of substitute models with different architectures trained on PD (CINIC-10) data.

Substitute ↓	QB	SimpleNet			ResNet-34 (from scratch)			ResNet-34 (transfer learning)		
Target →		Joint Acc	Acc	Fid	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid
SimpleNet	1k	36.66%	38.62%	38.88%	39.83%	41.28%	41.62%	38.51%	39.25%	39.30%
	5k	59.13%	61.10%	62.25%	61.30%	62.87%	63.87%	60.80%	61.70%	61.76%
	10k	67.48%	69.70%	70.98%	69.24%	71.02%	71.91%	68.04%	68.99%	69.10%
	20k	75.07%	77.46%	79.01%	76.81%	78.40%	79.94%	74.41%	75.36%	75.67%
	45k	**82.18%**	**84.67%**	**86.38%**	**82.72%**	**84.44%**	**85.98%**	**80.48%**	**81.46%**	**81.73%**
ResNet-18	1k	65.65%	68.60%	68.76%	69.14%	71.32%	71.49%	72.19%	73.20%	73.41%
	5k	76.33%	79.60%	79.41%	79.79%	82.19%	82.24%	82.78%	83.81%	84.06%
	10k	80.10%	83.48%	83.23%	82.25%	84.83%	84.69%	85.09%	86.13%	86.38%
	20k	81.91%	85.21%	85.40%	84.84%	87.32%	87.52%	87.90%	89.01%	89.12%
	45k	**84.28%**	**87.59%**	**87.83%**	**86.88%**	**89.43%**	**89.64%**	**90.42%**	**91.44%**	**91.76%**
ResNet-34	1k	63.60%	66.77%	66.55%	71.10%	73.50%	73.18%	73.74%	74.73%	74.99%
	5k	74.59%	78.03%	77.57%	79.31%	81.94%	81.61%	85.22%	86.11%	86.60%
	10k	79.63%	83.05%	82.65%	83.17%	85.95%	85.46%	87.92%	88.84%	89.29%
	20k	83.46%	86.94%	86.56%	86.36%	89.14%	88.79%	89.96%	90.97%	91.30%
	45k	**85.17%**	**88.73%**	**88.49%**	**87.56%**	**90.15%**	**90.26%**	**92.04%**	**93.28%**	**93.25%**

TABLE XV: Performance of substitute models with different architectures trained on the artificial data.

Substitute ↓	QB	SimpleNet			ResNet-34 (from scratch)			ResNet-34 (transfer learning)		
Target →		Joint Acc	Acc	Fid	Joint Acc	Acc	Fid	Joint Acc	Acc	Fid
SimpleNet	1k	21.27%	22.65%	23.10%	23.34%	24.39%	24.58%	22.93%	23.48%	23.57%
	5k	38.51%	40.31%	40.77%	36.13%	37.5%	37.78%	34.43%	35.11%	35.04%
	10k	45.87%	47.75%	48.61%	41.56%	43.02%	43.49%	37.66%	38.28%	38.28%
	20k	55.06%	57.03%	58.12%	52.68%	54.33%	54.81%	44.12%	44.87%	44.95%
	45k	**64.88%**	**67.10%**	**68.35%**	**65.83%**	**67.57%**	**68.34%**	**50.19%**	**50.99%**	**51.01%**
ResNet-18	1k	63.28%	66.20%	65.86%	64.13%	66.06%	66.32%	66.05%	66.83%	67.10%
	5k	67.90%	71.11%	70.55%	67.30%	69.44%	69.52%	71.12%	71.97%	72.14%
	10k	69.93%	73.04%	72.86%	71.38%	73.72%	73.40%	69.58%	70.46%	70.63%
	20k	73.43%	76.72%	76.42%	73.96%	76.31%	76.19%	68.08%	69.01%	69.16%
	45k	**75.69%**	**78.88%**	**78.78%**	**74.94%**	**77.10%**	**77.28%**	**72.96%**	**73.81%**	**74.00%**
ResNet-34	1k	62.11%	65.16%	64.37%	63.38%	65.69%	65.44%	65.39%	66.20%	66.48%
	5k	66.46%	69.58%	69.02%	67.03%	69.33%	69.16%	66.09%	66.86%	67.09%
	10k	71.83%	75.15%	74.46%	72.62%	74.89%	74.86%	68.83%	69.73%	69.82%
	20k	75.39%	78.75%	78.17%	73.59%	76.05%	75.81%	70.75%	71.65%	71.84%
	45k	**77.83%**	**81.03%**	**80.88%**	**74.94%**	**77.17%**	**77.44%**	**75.12%**	**75.97%**	**76.38%**

cases, we could not estimate the number of parameters of the target model and, consequently, could not report the efficiency score of an attack. In cases when we estimated the number of queries, we also marked the efficiency score as estimated. In one paper [13], the authors only mention that they use a ResNet model, referring to the paper where ResNet models were introduced. As this original paper mainly focuses on ResNet-34 architecture, we assumed that ResNet-34 was used as a target architecture by [13] and calculated efficiency scores based on this assumption.

D. Additional Results

Table XIV and Table XV show the performance of substitute models with different architectures trained on PD (CINIC-10) and artificial data.

E. Artificial Dataset Generation

The artificial dataset for the data-free attack was generated using a pre-trained stable diffusion model[3]. For each class of

[3] https://huggingface.co/stabilityai/stable-diffusion-2-1

the original dataset (CIFAR-10), we create text prompts to generate images. Each prompt consists of two parts: positive and negative. The positive prompt corresponds to inclusion criteria, and the negative prompt corresponds to exclusion criteria. Positive prompts usually include either a class label ("bird") or its subcategory ("swan"). Using subcategories makes the artificial data more diverse and helps to approximate the original data better. However, if no information is provided on the subcategories appearing in the original dataset, using them can be even misleading for a substitute model. Negative prompts aim to correct mistakes a diffusion model may make. For instance, they can prevent occurrences of bad anatomy or visual artefacts.

Table XVI shows positive and negative prompts used to generate images of each class. Sometimes, the model was biased towards generating very similar images for the same positive prompt. For instance, the prompt "bird photo" rendered birds of the same size and colour. Hence, we used a list of different bird families to generate representatives for them, making the dataset more diverse. We also replaced "airplane" with "plane"

TABLE XVI: Prompts used to generate the artificial dataset.

Class	Positive prompt	Negative prompt
airplane	plane photo	3d, grid, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple images, illustration, cropped, partial view, jpeg artifacts, grayscale
automobile	car photo — automobile photo	3d, grid, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple images, illustration, cropped, partial view, jpeg artifacts, grayscale
bird	cassowary photo — ostrich photo — emu photo — kiwi bird photo — owl photo — hawk photo — grebe photo — loon photo — duck photo — pheasant photo — tern photo — hummingbird photo — hen photo — rooster photo — swan photo — goose photo — parrot photo — bustard photo — tit photo — sparrow photo — woodpecker photo — pigeon photo — cuckoo photo — raven photo — oriole photo — warbler photo — chickadee photo — starling photo — dove photo — finch photo — nuthatch photo — bird photo	3d, bad anatomy, duplicated eyes, no eyes, extra eyes, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple birds, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs, black and white, two heads
cat	cat photo	3d, bad anatomy, duplicated eyes, no eyes, extra eyes, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple cats, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs, black and white
deer	deer photo	3d, bad anatomy, duplicated head, missing head, extra head, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple deers, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs, black and white, grayscale, painting, watermark, signature, two heads
dog	dog photo	3d, bad anatomy, duplicated eyes, no eyes, extra eyes, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple dogs, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs, black and white
frog	brown frog photo — green frog photo	3d, bad anatomy, duplicated eyes, no eyes, extra eyes, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple frogs, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs
horse	black horse photo — gray horse photo — chestnut horse photo — bay horse photo — dun horse photo	3d, bad anatomy, duplicated head, missing head, extra head, grid, extra limbs, close up, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple horses, multiple images, illustration, cropped, partial view, duplicated limbs, jpeg artifacts, missing limb, floating limbs, disconnected limbs, black and white, grayscale, painting, watermark, signature, two heads
ship	watercraft photo — ship photo — sailboat photo	3d, grid, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple images, illustration, cropped, partial view, jpeg artifacts, grayscale
truck	truck photo	3d, grid, deformed, ugly, mutation, mutated, blurry background, bokeh, multiple images, illustration, cropped, partial view, jpeg artifacts, grayscale

in the positive prompt because the model always generated the same type of aircraft, up in the air, for the "airplane" prompt. In contrast, "plane" images were more diverse and contained both flying and still vehicles. We observed similar behaviour for the "ship" class and addressed it by adding two additional prompts. "Frog" and "horse" classes lacked diversity in colours, so we asked the model explicitly to make them more diverse.