

# StolenLoRA: Exploring LoRA Extraction Attacks via Synthetic Data

Yixu Wang<sup>1,2†</sup>, Yan Teng<sup>2\*</sup>, Yingchun Wang<sup>2</sup>, Xingjun Ma<sup>1\*</sup>

<sup>1</sup>Fudan University    <sup>2</sup>Shanghai Artificial Intelligence Laboratory

## Abstract

*Parameter-Efficient Fine-Tuning (PEFT) methods like LoRA have transformed vision model adaptation, enabling the rapid deployment of customized models. However, the compactness of LoRA adaptations introduces new safety concerns, particularly their vulnerability to model extraction attacks. This paper introduces a new focus of model extraction attacks named LoRA extraction that extracts LoRA-adaptive models based on a public pre-trained model. We then propose a novel extraction method called **StolenLoRA** which trains a substitute model to extract the functionality of a LoRA-adapted model using synthetic data. StolenLoRA leverages a Large Language Model to craft effective prompts for data generation, and it incorporates a Disagreement-based Semi-supervised Learning (DSL) strategy to maximize information gain from limited queries. Our experiments demonstrate the effectiveness of StolenLoRA, achieving up to a 96.60% attack success rate with only 10k queries, even in cross-backbone scenarios where the attacker and victim models utilize different pre-trained backbones. These findings reveal the specific vulnerability of LoRA-adapted models to this type of extraction and underscore the urgent need for robust defense mechanisms tailored to PEFT methods. We also explore a preliminary defense strategy based on diversified LoRA deployments, highlighting its potential to mitigate such attacks.*

## 1. Introduction

Adapting large-scale pre-trained foundation models via Parameter-Efficient Fine-Tuning (PEFT) such as Low-Rank Adaptation (LoRA) [11, 15] has become a popular approach for obtaining high-performance models for diverse downstream tasks. LoRA enables efficient adaptation of large models to a new task by fine-tuning only low-rank matrices added to specific layers, reducing computational cost and memory usage while preserving the pre-trained model’s

core knowledge. Despite the wide adoption of LoRA, the lightweight and compactness of LoRA parameters raise a new safety concern: *they might be more vulnerable to model extraction (ME) attacks*. An ME attack extracts the functionality [17, 22, 28, 39] of a victim model by training a substitute model based on the outputs of the victim model for a certain number of specially designed queries. While ME attacks have been extensively studied for traditional models [29, 40, 42, 43, 46, 47], the vulnerability of LoRA adaptations to ME attacks remains largely unexplored[25]. This gap is significant, as the public availability of a large number of pre-trained models, coupled with the compactness of LoRA parameters, makes it easier to replicate LoRA-adapted models and compromise intellectual property.

To address this gap, we introduce the concept of **LoRA Extraction**, a novel direction of ME attacks specifically targeting LoRA-adapted models (*i.e.*, ViTs [4, 12, 20]), based on a publicly available pre-trained model. Unlike traditional ME which focuses on replicating the entire model’s functionality, LoRA extraction centers on stealing the *efficient adaptations* encoded within the compact LoRA parameters. The adversary’s objective is to reconstruct the victim’s LoRA-adapted model by training their own LoRA-adapted substitute that achieves similar downstream performance using a pre-trained foundation model that is publicly available. This attack can manifest in two scenarios: the *identical-backbone scenario*, where the attacker uses the same pre-trained ViT model as the victim, and the more challenging *cross-backbone scenario*, where the attacker uses a different pre-trained ViT.

Existing traditional ME methods largely depend on sample selection from existing datasets or synthetic data generation. Sample selection methods [28, 29, 42, 46] typically require searching extensive datasets for in-distribution samples, making them computationally expensive and less practical for LoRA extraction. This impracticality arises because 1) inference in LoRA-adapted models is slower due to the large-scale parameters of LoRA-adapted models, and 2) finding appropriate samples is challenging, particularly for domain-specific data fine-tuned with LoRA. Alternatively, synthetic data generation approaches [18, 23, 40, 47] often rely on Generative Adversarial Networks (GANs) [8]

\*Corresponding authors:

< tengyan@pjlab.org.cn, xingjunma@fudan.edu.cn >

† Work done during internship at Shanghai Artificial Intelligence Laboratory.

to produce in-distribution data. However, GANs frequently face difficulties in generating high-quality, diverse samples, especially for high-dimensional data. As Zhao et al. [46] noted, creating high-dimensional samples (e.g.,  $224 \times 224$  images) for effective ME can require millions of queries and is prone to failure due to GAN mode collapse [1, 6, 21]. These limitations make GAN-based approaches less effective for LoRA extraction, where the tuning often involves high-dimensional image data.

In this work, we propose a novel LoRA extraction method named *StolenLoRA*, which employs synthetic data to train the substitute model. To obtain effective in-distribution synthetic data, we leverage Large Language Models (LLMs) to generate diverse textual descriptions based on the target class names and prompt a pre-trained Stable Diffusion model [34, 36] to generate high-quality images. This allows us to tailor the generated data to the specific downstream task the victim’s LoRA model is fine-tuned for. Moreover, *StolenLoRA* introduces a distinctive strategy to improve attack efficiency, *i.e.*, the *Disagreement-based Semi-supervised Learning (DSL)*. DSL uses class information from the data synthesis process as pseudo-labels for part of the generated dataset and focuses on queries where the substitute model disagrees with these pseudo-labels. DSL effectively targets areas of uncertainty, guiding the substitute model’s learning process and refining its alignment with the victim’s behavior. To further enhance DSL, we iteratively refine the pseudo-labels based on evolving predictions from the substitute model, improving the quality of synthetic data labels and the effectiveness of the extraction.

In summary, our contributions are as follows:

- We explore the vulnerability of LoRA and introduce a novel focus of ME attacks called *LoRA Extraction* which aims to extract the functionality of LoRA-adapted models based on a publicly available pre-trained model.
- We present *StolenLoRA*, a novel LoRA extraction method that leverages LLM-driven Stable Diffusion to generate high-quality synthetic data, circumventing the need to search large-scale datasets or rely on less dependable GAN-based generation.
- Through comprehensive experiments on five widely used datasets, we demonstrate the effectiveness of *StolenLoRA* in both identical- and cross-backbone scenarios, highlighting critical vulnerabilities in LoRA-adapted models. We also explore a preliminary defense mechanism based on diversified LoRA deployments.

## 2. Related Work

**Low-Rank Adaptation (LoRA).** LoRA [15] is a type of popular Parameter-Efficient Fine-Tuning (PEFT) method [10, 14] for adapting large pre-trained models for downstream tasks. LoRA [15] introduces small rank-

decomposition matrices as trainable parameters into the pre-trained model. The number of trainable parameters is significantly reduced by freezing the original model weights and updating only these lightweight LoRA modules. LoRA inspires numerous variants and extensions, such as Mixture-of-LoRA Experts (X-LoRA)[3], Low-Rank Hadamard Product (LoHa)[16], and Low-Rank Kronecker Product (LoKr)[44], further enhancing its efficiency and flexibility for various applications. However, the compact nature of these LoRA updates raises concerns about their vulnerability to model extraction attacks. This paper investigates the potential for adversaries to extract the efficient adaptations encoded within these compact parameters, specifically focusing on the standard LoRA method [15].

**Model Extraction Attacks.** Model Extraction (ME) attacks [18, 22, 23, 28, 29, 40, 42, 46, 47] aim to construct a substitute model that mimics the functionality of a victim model by querying its API. Existing ME methods can be broadly categorized based on their query generation strategies: *Data-Driven ME* leverage existing datasets [28, 29, 42, 46] to select queries, often employing active learning[32] or other search strategies[28] to identify informative samples. However, this type can incur computational overhead as it necessitates traversing the large-scale dataset to find suitable in-distribution samples. *Data-Free ME* [18, 23, 40, 46, 47] utilize generative models (*e.g.*, GANs[8]) to synthesize data for querying the victim model. While promising, generating effective queries with high-dimensional inputs can require tens of millions of queries and is prone to failure due to GAN mode collapse [1, 6, 21].

**Learning from Synthetic Data.** The use of synthetic data for training models gains significant traction due to its potential to address data scarcity. While traditional synthetic data often lacked the complexity and representativeness of real data, recent advancements in generative models, particularly diffusion models like Stable Diffusion [34, 36], enable the synthesis of high-quality and diverse images. However, scaling the quantity of synthetic data does not necessarily translate to improved performance for training supervised image classifiers. As shown by Fan et al.[7], performance with synthetic data can lag significantly behind training with real data. This disparity stems from the limitations of current text-to-image models in accurately generating diverse representations of certain concepts at scale[30]. Efforts to mitigate this gap, such as the distribution-matching framework[45], focus on aligning the distributions of synthetic and real data. However, leveraging such high-quality synthetic data for model extraction attacks remains an unexplored area.

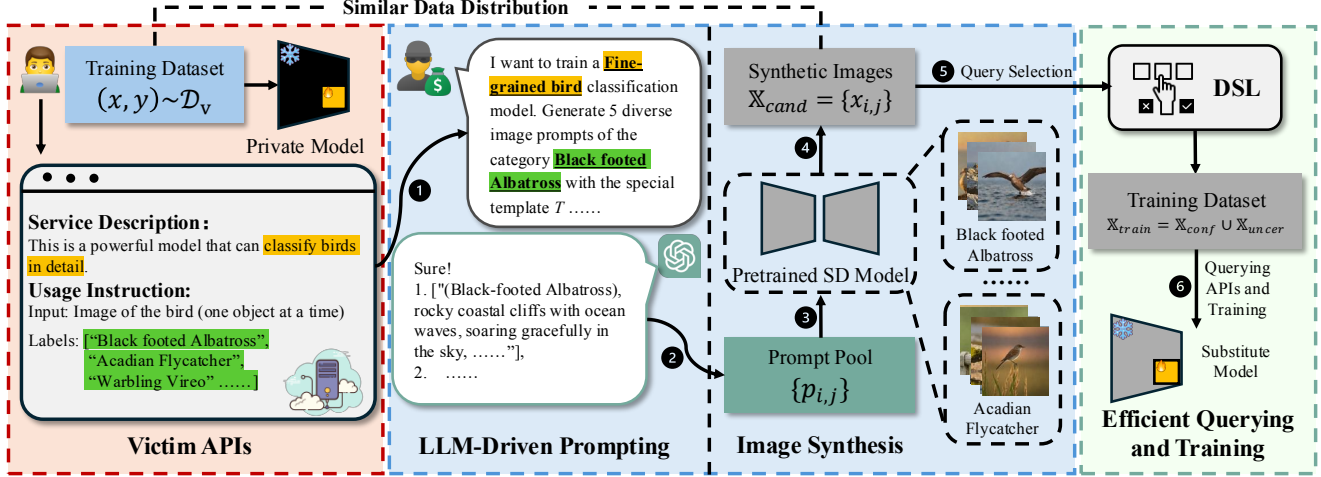


Figure 1. Overview of StolenLoRA. It collects target victim model information (dashed red box), synthesizes training data guided by an LLM-driven prompting (dashed blue box), and then efficiently queries the victim API to train a substitute model (dashed green box).

### 3. Methodology

#### 3.1. Problem Formulation

**LoRA Extraction.** Given a victim model  $F: [0, 1]^d \mapsto \mathbb{R}^N$ , which is a pre-trained model  $F_{base}$  (e.g., ViTs) adapted using LoRA. The parameters of  $F^1$  can be represented as  $\theta = \{\theta_{base}, \Delta\theta\}$  where  $\theta_{base}$  are the parameters of  $F_{base}$  and  $\Delta\theta$  represents the LoRA updates, which are typically low-rank matrices. The objective of a LoRA extraction attack is to train a substitute model  $F'$  with parameters  $\theta' = \{\theta'_{base}, \Delta\theta'\}$  that mimics the functionality of  $F$ . This is formulated as the following optimization problem:

$$\arg \min_{\Delta\theta'} \mathbb{E}_{\mathbf{x}} \mathcal{D}(F(\mathbf{x}), F'(\mathbf{x})), \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^d$  represents an input sample from the victim model task distribution,  $\mathcal{D}(\cdot, \cdot)$  represents a measure of functional similarity between the two models.

Depending on whether the attacker has knowledge of  $F_{base}$ , we distinguish between two scenarios: *identical-backbone* and *cross-backbone* LoRA extractions. The substitute model  $F'$  can be defined as follows:

$$F'(\mathbf{x}) = \begin{cases} F_{base}(\mathbf{x}) + \Delta F'(\mathbf{x}), & \text{identical-backbone} \\ G_{base}(\mathbf{x}) + \Delta F'(\mathbf{x}), & \text{cross-backbone} \end{cases} \quad (2)$$

where the former uses the same  $F_{base}$ , the latter uses different base models  $G_{base}$ , and the '+' symbol denotes the combination of the base model and the LoRA updates.

**Distinction from Traditional Extraction.** Traditional model extraction trains a substitute from scratch, while LoRA extraction leverages a pre-trained base model, inheriting its knowledge and biases. Consequently, we hypothesize that a perfectly extracted LoRA substitute might

<sup>1</sup>We omit the parameter for simplicity when there is no ambiguity, e.g., using  $F(\mathbf{x})$  instead of  $F(\mathbf{x}; \theta)$ .

achieve functional equivalence on In-Distribution (ID) data but not on Out-of-Distribution (OOD) data, particularly in the cross-backbone setting, and we experimentally verify this in the experimental section. This motivates the use of ID data for effective LoRA extraction.

#### 3.2. StolenLoRA

**Overview.** An overview of our proposed LLM-driven generative attack for extracting LoRA, *StolenLoRA*, is illustrated in Fig. 1. It comprises two key stages: 1) *Data Synthesis*, where an LLM guides the generation of synthetic data tailored to the victim's task, and 2) *Efficient Querying and Training*, which strategically selects queries and refines labels to maximize information gain from the victim model.

##### 3.2.1. Data Synthesis

The above discussion motivates us to use ID data for extraction. However, acquiring such datasets in real-world scenarios is often challenging because the victim model's training data is proprietary. While the attacker might not have access to the original training data, they often have access to the model's deployment context, which usually includes a description of the model's functionality and the names of the target classes. This information can be leveraged to semantically expand the search space for potential training data. Inspired by this, we propose to utilize LLM-driven Stable Diffusion to generate synthetic data based on the class names, effectively bridging the gap between semantic information and visual representations.

**LLM-Driven Prompting.** Given a set of target class names  $\mathbb{C} = \{c_1, c_2, \dots, c_n\}$ , we employ an LLM to generate detailed prompts for image synthesis. We structure the prompts using a predefined template  $T$  comprising key visual elements:  $T = [\text{Subject, Background, Angle/Pose, Lighting, Style}]$ . For each class  $c_i$ , the LLM generates  $m$  variations of the

prompt, denoted as  $p_{i,j}$ , where  $j \in \{1, \dots, m\}$ . This diversification ensures a richer representation of the target class. The prompt generation process is formalized as follows:

$$p_{i,j} = \text{LLM}(c_i, T, \omega_j), \quad (3)$$

where  $\omega_j$  represents a random seed or instruction to control the variation generated by the LLM.

**Image Synthesis.** We employ a public pre-trained Stable Diffusion model, denoted as  $\text{SD}(\cdot)$ , to generate images corresponding to the crafted prompts. For each prompt  $p_{i,j}$ , we synthesize one image, resulting in a set of synthetic images  $\mathbb{X}_i = \{\mathbf{x}_{i,j}\}$ , where  $\mathbf{x}_{i,j} = \text{SD}(p_{i,j})$ . The complete synthetic dataset is then  $\mathbb{X} = \bigcup_i \mathbb{X}_i$ .

### 3.2.2. Efficient Querying and Training

This stage focuses on strategically querying the victim model to refine the substitute model’s LoRA parameters while minimizing the number of queries. We propose two attack strategies: random learning and DSL.

**Random Learning (StolenLoRA-Rand).** This straightforward approach directly uses the synthesized dataset  $\mathbb{X}$  to query the victim model  $F$ . Then, the substitute model’s LoRA parameters  $\Delta\theta'$  are trained by minimizing the following objective function:

$$\mathcal{L}'(\Delta\theta') = \mathbb{E}_{\mathbf{x} \in \mathbb{X}} \mathcal{L}(F'(\mathbf{x}), F(\mathbf{x})), \quad (4)$$

where  $\mathcal{L}$  denotes a loss function (*e.g.*, cross-entropy loss). This provides a baseline for evaluating the benefits of more sophisticated attack strategies.

**Disagreement-based Semi-supervised Learning (StolenLoRA-DSL).** While directly using synthesized data for querying and training is feasible, we propose DSL to significantly enhance both the effectiveness and efficiency of the attack. DSL improves query efficiency by selectively querying the victim model based on the substitute model’s uncertainty. Furthermore, it iteratively refines both the pseudo-labels of the synthetic data and the labels obtained from querying the victim, mitigating the impact of noisy pseudo-labels and bridging the gap between the substitute and victim model, especially in cross-backbone scenarios.

The process begins by generating an initial synthetic dataset  $\mathbb{X}^0 = \bigcup_i \mathbb{X}_i^0$  and assigning pseudo-labels based on their generating prompts *without querying the victim model*. These pseudo-labels, denoted as  $c(\mathbf{x})$ , provide a starting point for training the initial substitute model  $F'_0$ . Then, DSL proceeds iteratively, refining both the synthetic data for querying and the substitute model. In each iteration  $t$ , a new set  $\mathbb{X}_{cand}^t = \bigcup_i \mathbb{X}_{cand,i}^t$  with  $\beta * b_t$  samples is generated using the LLM-driven Stable Diffusion process, where  $\beta$  is a scaling factor. These candidates are then subjected to a *disagreement-based filtering* based on the current substitute model’s predictions. For each candidate  $x \in \mathbb{X}_{cand}^t$ ,

---

#### Algorithm 1 StolenLoRA-DSL

---

**Input:** Victim model  $F$ , Class names  $\mathbb{C}$ , Query budget  $b_t$ , Scaling factor  $\beta$ , Threshold  $\tau$ , Initial sample size  $N$ .

**Output:** Substitute LoRA model  $F'$ .

```

1:  $\mathbb{X}_0 \leftarrow \bigcup_i \text{SynIMG}(\mathbb{C}, N)$ 
2:  $F'_0 \leftarrow \text{Update}(\mathbb{X}_0, \mathcal{L}_{LR})$   $\triangleright$  Initial substitute model
3:  $t \leftarrow 0, B_t \leftarrow B$ 
4: while  $B_t > 0$  do
5:    $\mathbb{X}_{cand}^t \leftarrow \text{SynIMG}(\mathbb{C}, \beta b_t)$   $\triangleright$  Generate candidate
6:    $\mathbb{X}_{conf}^t \leftarrow \emptyset, \mathbb{X}_{uncer}^t \leftarrow \emptyset$ 
7:   for  $\mathbf{x} \in \mathbb{X}_{cand}^t$  do
8:      $\hat{c}, \hat{p} \leftarrow \text{Predict}(F'_t(\mathbf{x}))$ 
9:     if  $\hat{c} = c(\mathbf{x})$  and  $\hat{p} \geq \tau$  then  $\triangleright$  Disagreement filtering
10:       $\mathbb{X}_{conf}^t \leftarrow \mathbb{X}_{conf}^t \cup \{(\mathbf{x}, c(\mathbf{x}))\}$ 
11:    else
12:       $\mathbb{X}_{uncer}^t \leftarrow \mathbb{X}_{uncer}^t \cup \{\mathbf{x}\}$ 
13:    end if
14:  end for
15:   $\mathbb{X}_{query}^t \leftarrow \text{Query}(F, \mathbb{X}_{uncer}^t, b_t)$   $\triangleright$  Selective Query
16:   $\mathbb{X}_{train}^t \leftarrow \mathbb{X}_{conf}^t \cup \mathbb{X}_{query}^t$ 
17:   $F'_{t+1} \leftarrow \text{Update}(\mathbb{X}_{train}^t, \mathcal{L}_{LR})$ 
18:   $B_t \leftarrow B_t - |\mathbb{X}_{query}^t|, t \leftarrow t + 1$ 
19: end while
20: return  $F'_t$ 

```

---

the substitute model  $F'_t$  predicts a class  $\hat{c}$  and associated confidence  $\hat{p}$ . If the predicted class  $\hat{c}$  matches the prompt-based pseudo-label  $c(\mathbf{x})$  and the confidence  $\hat{p}$  exceeds a predefined threshold  $\tau$ , the pseudo-label is considered reliable, and the sample is added to a confidently labeled set  $\mathbb{X}_{conf}^t = \{(\mathbf{x}, c(\mathbf{x}))\}$ . Conversely, if the prediction disagrees with the pseudo-label or the confidence is low, the sample is added to an uncertain set  $\mathbb{X}_{uncer}^t$ .

From the uncertain set  $\mathbb{X}_{uncer}^t$ , a subset of  $b_t$  samples with the lowest confidence scores are selected for querying the victim model  $F$ . These queried samples, along with their true labels obtained from  $F$ , form the query set  $\mathbb{X}_{query}^t = \{(\mathbf{x}, F(\mathbf{x}))\}$ . Combine the confidently pseudo-labeled samples  $\mathbb{X}_{conf}^t$  and the queried samples  $\mathbb{X}_{query}^t$  to form the training set for this iteration:  $\mathbb{X}_{train}^t = \mathbb{X}_{conf}^t \cup \mathbb{X}_{query}^t$ . Crucially, this combined set undergoes *label refining* during training to further address potential inaccuracies in pseudo-labels and mitigate the distribution shift.

Specifically, we employ a label refining strategy. Let  $\mathbf{z}$  be the logits produced by the substitute model for a given input  $\mathbf{x}$ . The label refining training loss is calculated as:

$$\mathcal{L}_{LR}(\mathbf{z}, \mathbf{q}) = - \sum_{i=1}^C q_i \log \left( \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)} \right), \quad (5)$$

where  $\mathbf{q}$  are the soft labels. They are updated using an ex-

Scenario	Method	CUBS200		Caltech256		Indoor67		Food101		Flowers102	
		Acc	ASR	Acc	ASR	Acc	ASR	Acc	ASR	Acc	ASR
Identical Backbone	KnockoffNets	70.95	80.54	85.69	90.71	79.18	92.59	<b>82.53</b>	<b>90.62</b>	76.24	77.28
	ActiveThief	72.33	82.11	86.92	92.01	77.46	90.57	80.34	88.22	77.51	78.56
	DFME	0.56	0.64	0.48	0.51	2.71	3.17	1.62	1.78	2.81	2.85
	$E^3$	71.94	81.67	82.36	87.18	81.43	95.22	79.65	87.46	80.67	81.77
	StolenLoRA-Rand	<b>75.35</b>	<b>85.54</b>	<u>87.62</u>	<u>92.75</u>	<u>82.38</u>	<u>96.33</u>	79.00	86.75	<b>93.74</b>	<b>95.01</b>
	StolenLoRA-DSL	<u>73.23</u>	<u>83.13</u>	<b>89.30</b>	<b>94.53</b>	<b>82.61</b>	<b>96.60</b>	<u>80.57</u>	<u>88.47</u>	<u>87.46</u>	<u>88.65</u>
Cross Backbone	KnockoffNets	6.77	7.69	41.34	43.76	41.79	48.87	14.47	15.89	14.16	14.35
	ActiveThief	15.42	17.50	42.89	45.40	36.04	42.14	15.11	16.59	29.09	29.49
	DFME	0.51	0.58	0.50	0.53	1.33	1.56	0.89	0.98	1.46	1.48
	$E^3$	17.55	19.92	48.17	50.99	55.85	65.31	40.58	44.56	48.28	48.94
	StolenLoRA-Rand	<u>45.70</u>	<u>51.88</u>	<u>51.75</u>	<u>54.78</u>	<u>59.18</u>	<u>69.20</u>	<u>43.16</u>	<u>47.39</u>	<u>59.29</u>	<u>60.10</u>
	StolenLoRA-DSL	<b>50.14</b>	<b>56.92</b>	<b>65.01</b>	<b>68.82</b>	<b>65.07</b>	<b>76.09</b>	<b>44.53</b>	<b>48.90</b>	<b>61.16</b>	<b>61.99</b>

Table 1. Effectiveness of StolenLoRA: the Acc (%) and ASR (%) of extracted substitute model by different model extraction attacks under 10k queries. (**Boldface**: the best value, Underline: the second-best value.)

ponential moving average of the predicted probabilities  $\mathbf{p}$ :

$$\mathbf{q}^{(i+1)} = \mu \mathbf{q}^{(i)} + (1 - \mu) \mathbf{p}^{(i+1)}, \quad (6)$$

where  $\mu$  is the momentum parameter. This iterative refinement allows the model to learn from both the initial labels and its own evolving predictions, progressively improving the quality of the training data. The updated substitute model  $F'_{t+1}$  is thus trained on  $\mathbb{X}_{train}^t$  by minimizing:

$$\mathcal{L}'(\Delta\theta'_{t+1}) = \mathbb{E}_{\mathbf{x} \in \mathbb{X}_{train}^t} \mathcal{L}_{LR}(F'_{t+1}(\mathbf{x}), \mathbf{q}(\mathbf{x})). \quad (7)$$

This iterative process continues until the query budget is exhausted, resulting in a final substitute model  $F'$  trained on a combination of high-confidence pseudo-labeled data and a smaller set of strategically queried data. DSL allows for efficient use of the query budget by focusing on the most informative samples, leading to a more accurate substitute LoRA. We present the algorithm detail of it in Alg. 1.

## 4. Experiments

### 4.1. Experimental Setup

**Victim Models.** We use a ViT-Base model pre-trained on ImageNet-21k [33] with additional augmentations and regularization [38]. This base model is then fine-tuned with LoRA ( $r = 4$ ) on five commonly used datasets: CUBS200 [41], Caltech256 [9], Indoor67 [31], Food101 [2], and Flowers102 [26]. The resulting victim models achieve test accuracies of 88.09%, 94.47%, 85.52%, 91.07%, and 98.66%, respectively.

**Attack Settings.** We evaluate LoRA extraction attacks in Identical-Backbone (IB) and Cross-Backbone (XB) scenarios. In the IB setting, we use the same base model as the substitute base model. In the XB setting, we use

the ViT-base model pre-trained on ImageNet-1k[35] with self-supervised masked autoencoder (MAE) method[12] as the substitute base model. Both substitute models utilize LoRA with  $r = 4$  and are trained for 20 epochs using the Adam optimizer[19] with a cosine annealing learning rate schedule[24] and a base learning rate of 0.01.

We use GPT-4o mini[27] as the LLM in StolenLoRA as it has similar performance at a lower price and can effectively reduce the attack cost. For the SD model, we use an open-source SDXL-Turbo model[36]. It only takes 4 sampling steps (compared to SDXL 1.0 which usually requires 40 steps) to synthesize high-quality images, significantly reducing the computational cost. For the StolenLoRA-DSL method, initial per-category sample size  $N$  is set to 10, the scaling factor  $\beta$  is 1.5 to not increase too much generation overhead, and the confidence threshold  $\tau$  is 0.95. Hyperparameter analysis is provided in later sections.

**Baselines and Evaluation Metric.** We compare StolenLoRA against KnockoffNets [28], ActiveThief [29], DFME [40], and  $E^3$  [48] using a query budget of 10k. For methods such as KnockoffNets, ActiveThief, and  $E^3$  that require real data, we use 3M images in CC3M[37] as the attack dataset. For  $E^3$ , which originally selects samples based on semantic similarity between dataset category names and target class names, we improve its effectiveness by calculating semantic similarity between image captions (available in CC3M) and target class names for a more fine-grained sample selection. Following prior work [28, 40], we report Test Accuracy (Acc) and Attack Success Rate (ASR), defined as the ratio of the substitute model’s accuracy to the victim model’s accuracy.

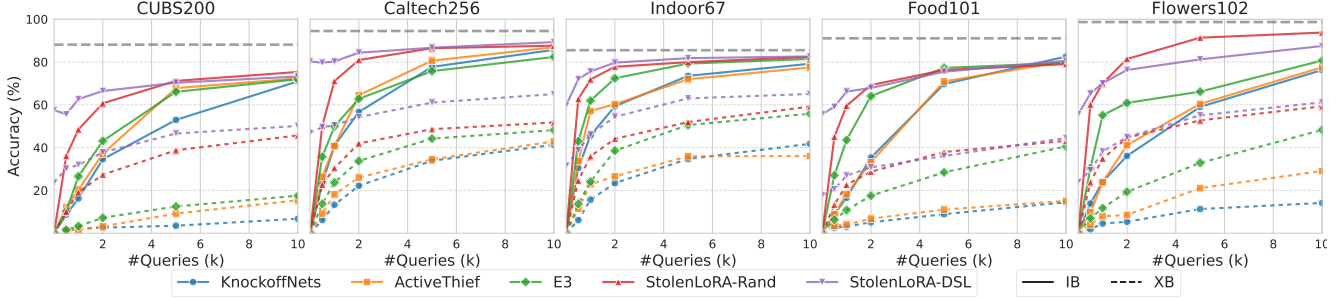


Figure 2. Curves of the test accuracy versus the number of queries in IB (solid line) and XB (dashed line) scenarios respectively. The gray dashed line represents the victim models’ test accuracy.

## 4.2. Experimental Results

**Effectiveness of StolenLoRA.** Tab. 1 presents the effectiveness of StolenLoRA, comparing both the random strategy (StolenLoRA-Rand) and DSL-based (StolenLoRA-DSL) variants against existing extraction attacks. Across both identical- and cross-backbone scenarios, StolenLoRA consistently achieves state-of-the-art performance. In the IB setting, StolenLoRA-Rand shows superior performance on CUBS200 (75.35%) and Flowers102 (93.74%), while StolenLoRA-DSL excels on Caltech256 (89.30%) and Indoor67 (82.61%), maintaining competitive results on other datasets. The XB setting, more challenging due to base model differences, highlights StolenLoRA’s significant advantage over baselines, with StolenLoRA-DSL achieving the highest accuracy across all five datasets. This demonstrates DSL’s effectiveness in refining synthetic data and improving attack efficiency. Fig. 2 further analyzes query efficiency, showing StolenLoRA-DSL’s rapid convergence in the IB setting and consistent progress in the XB setting, outperforming other methods in both scenarios. This efficiency stems from DSL’s ability to prioritize informative queries, focusing on uncertain samples, thereby maximizing information gain from each query.

**Effectiveness under Hard-Label Scenario.** We also evaluate StolenLoRA’s performance when only hard labels (one-hot encoded predictions) are available, as is the case with some real-world APIs. Fig. 3 presents the results for both StolenLoRA-Rand and StolenLoRA-DSL in the IB and XB settings. In the IB setting, using hard labels leads to a minor performance decrease. For example, StolenLoRA-Rand’s accuracy on CUBS200 drops from 75.35% with soft labels to 67.00% with hard labels. A similar trend is observed for other datasets and StolenLoRA-DSL. This suggests that the additional information provided by soft labels is beneficial in the IB setting. In the XB setting, the impact of hard labels is inconsistent. While a slight decrease is observed on some datasets, others, like Caltech256 and Indoor67, show a marginal increase in accuracy (from 51.75% to 56.69%). This unexpected improvement might be due to the hard labels acting as a regularizer, preventing the substi-

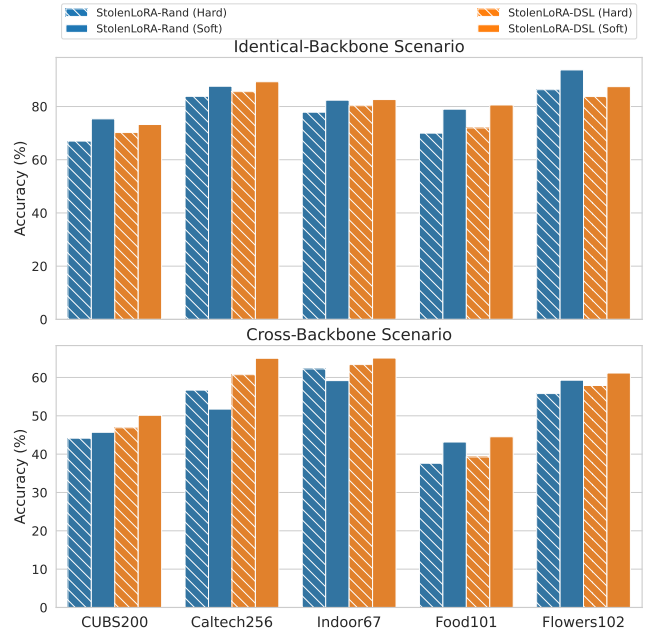


Figure 3. Comparison of StolenLoRA performance using hard labels (one-hot) versus soft labels (probabilities) from the victim model. Results are shown for both Random and DSL in identical- and cross-backbone scenarios across five datasets.

tute from overfitting to the synthetic data, which is more prone to distributional shifts in the XB setting. Overall, StolenLoRA exhibits robustness to hard labels in both IB and XB scenarios, demonstrating its practical applicability.

**Ablation Study.** We conduct an ablation study to analyze the contributions of the key components of StolenLoRA: the LLM-driven prompting and the DSL strategy, including the label refining loss. Tab. 2 presents the results on CUBS200 and Indoor67 datasets under both IB and XB settings with a 10k query budget. Removing the structured prompt template leads to a performance drop in both IB and XB settings, demonstrating the importance of providing structured visual information to the LLM. Removing the LLM entirely results in a more significant performance decrease, confirming the crucial role of the LLM in generating effective prompts for diverse and representative synthetic data. Furthermore, we evaluate different LLMs, observing that using

Method	IB		XB	
	CUBS200	Indoor67	CUBS200	Indoor67
Random	75.35	82.38	45.70	59.18
- Template	72.33	80.94	42.21	54.49
- LLM	70.66	74.70	39.20	47.09
+ Llama-3.1-8B	73.96	81.04	46.74	58.43
+ GPT-4o	76.30	83.13	49.19	63.58
DSL	73.23	82.61	50.14	65.07
- $\mathcal{L}_{LR}$	70.17	80.97	48.21	64.10

Table 2. An ablation experiment showing the effectiveness of the two modules we designed on CUBS200 and Indoor67 datasets under both IB and XB settings with 10k queries.

$\tau$	IB		XB	
	CUBS200	Indoor67	CUBS200	Indoor67
0.5	70.68	78.81	48.43	60.90
0.7	71.37	80.00	48.36	63.51
0.9	<b>74.04</b>	81.19	48.52	63.06
<u>0.95</u>	73.23	<b>82.61</b>	<b>50.14</b>	<b>65.07</b>
0.99	73.31	81.19	49.86	61.72

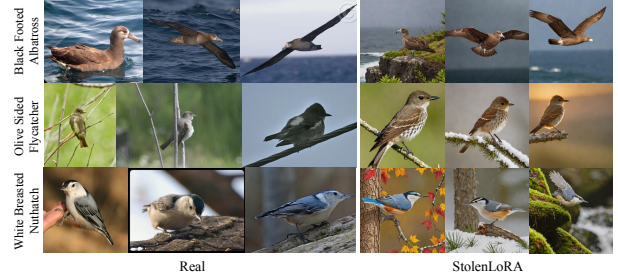
Table 3. Performance of StolenLoRA-DSL with varying confidence thresholds  $\tau$ . Results are reported on CUBS200 and Indoor67 under both IB and XB settings with 10k queries.

a more powerful LLM like GPT-4o generally yields better performance than Llama-3.1-8B[5], particularly in the more challenging XB setting. This highlights the benefit of utilizing a stronger LLM for prompt generation.

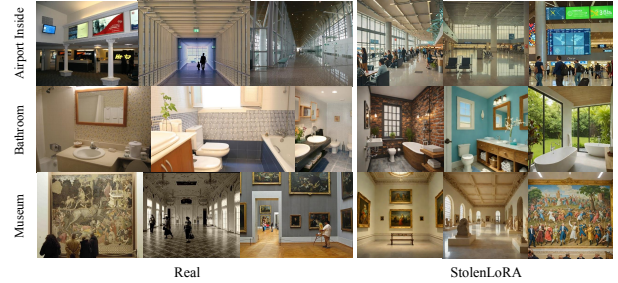
Disabling DSL and reverting to random querying results in a noticeable performance reduction, especially in the XB setting. This underscores the effectiveness of DSL in selectively querying informative samples. Similarly, removing SAT from the DSL framework also leads to a performance decrease, albeit less pronounced than removing DSL altogether. This indicates the benefit of label refining in mitigating the distribution shift between synthetic and real data.

We also analyze the impact of the confidence threshold  $\tau$  within DSL in Tab. 3. The results show that performance generally improves as  $\tau$  increases from 0.5 to 0.95. This indicates that focusing queries on samples where the substitute model is less confident leads to a more efficient use of the query budget. However, increasing  $\tau$  further to 0.99 results in a slight performance decrease, suggesting that an overly stringent threshold can exclude valuable samples from being queried. We therefore select  $\tau = 0.95$  as the optimal value for our experiments.

**Fidelity and Distribution Analysis of Synthetic Data.** StolenLoRA’s effectiveness relies on the quality of its synthetic training data. We evaluate this quality both visually and quantitatively. Visually, as shown in Fig. 4, synthetic images generated by StolenLoRA demonstrate a striking resemblance to real images from the target categories, capturing key features and background context crucial for train-



(a) Visualization on CUBS200



(b) Visualization on Indoor67

Figure 4. Visualization of StolenLoRA’s synthetic images and corresponding real images of the target category.

Method	CUBS200	Caltech256	Indoor67	Food101	Flowers102
KnockoffNets	51.44	8.94	22.01	50.04	55.77
ActiveThief	49.33	6.01	15.72	47.84	50.19
DFME	220.53	179.45	225.84	188.29	204.79
$E^3$	41.57	<b>4.29</b>	<b>6.68</b>	25.51	34.82
StolenLoRA	<b>2.14</b>	5.59	6.93	<b>19.54</b>	<b>16.63</b>

Table 4. FID scores ( $\downarrow$ ) between the datasets used by each attack method and the victim model’s training data distribution.

ing. This visual fidelity is corroborated by significantly lower Fréchet Inception Distance (FID) scores[13] (in Tab. 4) compared to other attack methods. For example, StolenLoRA achieves an FID of 2.14 compared to 51.44 for KnockoffNets on CUBS200, highlighting a much closer distributional match to the target domain. This demonstrates that StolenLoRA’s LLM-driven synthesis effectively generates high-fidelity, domain-specific data, even without access to the original training set, which is key to its superior extraction performance.

**Verifying the Distinction Hypothesis.** As mentioned in Sec. 3.1, we hypothesize that the performance of a perfectly extracted surrogate model will be consistent across In-Distribution (ID) data with the victim model, but not necessarily Out-of-Distribution (OOD) data, motivating us to perform extraction attacks using ID data. To empirically verify this conjecture, we train a substitute model using the XB setting on the victim model’s training dataset, aiming to achieve a near-perfect extraction of the LoRA adaptations. We then evaluate both the victim model and the extracted substitute on two sets of data: ID data, represented by the victim model’s held-out test set, and OOD data, sampled from the CC3M dataset. For each sample, we calculate the cross-entropy between the victim model’s predictions and

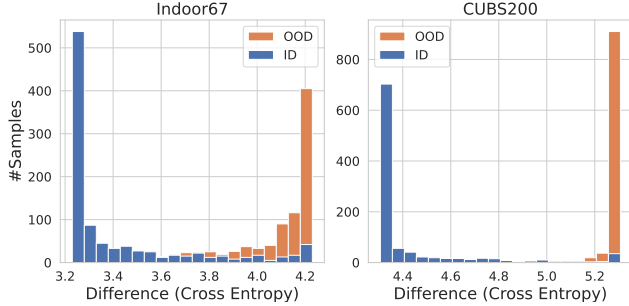


Figure 5. Cross-entropy difference between the victim model and a near-perfectly substitute model on In-Distribution (ID) and Out-of-Distribution (OOD) data. The substitute model is trained using the cross-backbone setting on the victim’s training dataset.

the substitute model’s predictions, providing a measure of their functional divergence. Fig. 5 presents the distribution of these differences. On both CUBS200 and Indoor67, the difference is smaller for the ID data compared to the OOD data. This observation supports our conjecture that even near-perfect LoRA extraction yields a substitute model that exhibits functional equivalence primarily within the ID domain. The divergence in OOD data likely stems from the inherent differences between the pre-trained backbone used by the victim and the substitute. This result underscores the importance of utilizing ID data for effective LoRA extraction, as the attacker primarily seeks to replicate the victim’s specialized adaptations rather than the general capabilities.

## 5. Defending Against LoRA Extraction

While this paper primarily focuses on the feasibility of LoRA extraction, we also explore a potential defense mechanism based on deploying multiple, diverse LoRA adapters to increase attacker uncertainty.

**Dual LoRA with Diversified Predictions.** Our proposed defense involves training two LoRA adapters, denoted as  $L_A$  and  $L_B$ , to maximize the difference in their output distributions while maintaining comparable performance on the target task. Instead of minimizing accuracy as initially conceived, we found that maintaining high accuracy while maximizing divergence is crucial for practical deployment. We achieve this by optimizing the following loss function:

$$\mathcal{L}' = \mathcal{L}(L_A) + \mathcal{L}(L_B) - \lambda \times \text{KL}(L_A || L_B) \quad (8)$$

where  $\mathcal{L}$  represents the cross-entropy loss for the target task, KL represents the Kullback-Leibler divergence between the output distributions of the two LoRAs, and  $\lambda$  is a hyperparameter controlling the balance between task performance and divergence. At deployment, we randomly select either  $L_A$  or  $L_B$  for each incoming query to generate the prediction. This random selection introduces uncertainty and makes it challenging for the attacker to extract the underlying LoRA function consistently.

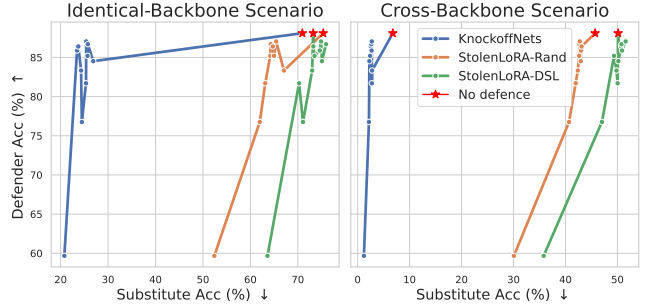


Figure 6. Defender Acc versus Substitute model Acc trade-off for defenses evaluated against StolenLoRA and KnockoffNets on the CUBS200 dataset with 10k queries.

**Experimental Evaluation.** We evaluate the proposed defense mechanism under both IB and XB settings with 10k queries. Fig. 6 illustrates the trade-off between defender Acc and substitute model Acc. The defense consistently degrades the substitute model’s performance. KnockoffNets experiences substantial accuracy drops (from 70.90% to 20.78% in IB and from 6.77% to 1.21% in XB), and StolenLoRA-Rand shows similar vulnerability (decreasing 22.95% in IB and 15.63% in XB). However, StolenLoRA-DSL demonstrates greater resilience, with comparatively smaller accuracy reductions (9.63% in IB and 14.34% in XB). This resilience is attributed to DSL’s efficient query selection and label refining, focusing on informative samples and mitigating the uncertainty introduced by the defense. While this defense effectively hinders less sophisticated attacks, it underscores the need for more robust mechanisms to counter advanced techniques like DSL that can learn effectively from limited queries.

## 6. Conclusion

This paper introduces *LoRA Extraction*, a novel model extraction attack targeting the widespread practice of adapting large vision models with LoRA. We present *StolenLoRA*, a highly effective method leveraging LLM-driven Stable Diffusion to generate task-specific, high-fidelity synthetic training data, bypassing the limitations of traditional extraction techniques reliant on large-scale real datasets or unreliable GANs. Furthermore, StolenLoRA incorporates *Disagreement-based Semi-supervised Learning (DSL)* to efficiently query the victim model, focusing on uncertain predictions and iteratively refining labels, enabling successful extraction with limited queries. Extensive experiments demonstrate StolenLoRA’s strong performance across diverse datasets and in challenging cross-backbone scenarios. We also explore a preliminary defense based on diversified LoRA deployments, showing promise in mitigating these attacks. This work represents a crucial first step towards understanding and addressing the safety risks associated with LoRA adaptation, paving the way for more secure deployments of efficient fine-tuning methods.

## Acknowledgments

This work is in part supported by National Key R&D Program of China (Grant No. 2022ZD0160103) and National Natural Science Foundation of China (Grant No. 62276067), and Shanghai Artificial Intelligence Laboratory.

## References

- [1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017. 2
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *ECCV*, 2014. 5
- [3] Eric L Buehler and Markus J Buehler. X-lora: Mixture of low-rank adapter experts, a flexible framework for large language models with applications in protein mechanics and molecular design. *APL Machine Learning*, 2024. 2
- [4] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1
- [5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 7
- [6] Ricard Durall, Avraam Chatzimichailidis, Peter Labus, and Janis Keuper. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. *arXiv preprint arXiv:2012.09673*, 2020. 2
- [7] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *CVPR*, 2024. 2
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2
- [9] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 5
- [10] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024. 2
- [11] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024. 1
- [12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 5
- [13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 7
- [14] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019. 2
- [15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1, 2
- [16] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*, 2021. 2
- [17] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *Usenix Security*, 2020. 1
- [18] Sanjay Kariyappa, Atul Prakash, and Moinuddin Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *CVPR*, 2021. 1, 2
- [19] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *CVPR*, 2023. 1
- [21] Youssef Kossale, Mohammed Airaj, and Aziz Darouichi. Mode collapse in generative adversarial networks: An overview. In *ICOA*, 2022. 2
- [22] Zi Liang, Qingqing Ye, Yanyun Wang, Sen Zhang, Yaxin Xiao, Ronghua Li, Jianliang Xu, and Haibo Hu. Alignment-aware model extraction attacks on large language models. *arXiv preprint arXiv:2409.02718*, 2024. 1, 2
- [23] Zijun Lin, Ke Xu, Chengfang Fang, Huadi Zheng, Aneez Ahmed Jaheezuddin, and Jie Shi. Quda: Query-limited data-free model extraction. In *ACM Asia CCS*, 2023. 1, 2
- [24] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5
- [25] Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhan Zhao, et al. Safety at scale: A comprehensive survey of large model safety. *arXiv preprint arXiv:2502.05206*, 2025. 1
- [26] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian conference on computer vision, graphics & image processing*, 2008. 5
- [27] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>, 2024. 5
- [28] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *CVPR*, 2019. 1, 2, 5
- [29] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *AAAI*, 2020. 1, 2, 5
- [30] Jie Qin, Jie Wu, Weifeng Chen, Yuxi Ren, Huixia Li, Hefeng Wu, Xuefeng Xiao, Rui Wang, and Shilei Wen. Diffusiongpt: Llm-driven text-to-image generation system. *arXiv preprint arXiv:2401.10061*, 2024. 2

- [31] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 5
- [32] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys*, 2021. 2
- [33] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lih Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 5
- [34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2
- [35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5
- [36] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 2, 5
- [37] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5
- [38] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 5
- [39] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, 2016. 1
- [40] Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *CVPR*, 2021. 1, 2, 5
- [41] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5
- [42] Yixu Wang, Jie Li, Hong Liu, Yan Wang, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Black-box dissector: Towards erasing-based hard-label model stealing attack. In *ECCV*, 2022. 1, 2
- [43] Yixu Wang, Tianle Gu, Yan Teng, Yingchun Wang, and Xingjun Ma. Honey-potnet: Backdoor attacks against model extraction. In *AAAI*, 2025. 1
- [44] Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *ICLR*, 2023. 2
- [45] Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. Real-fake: Effective training data synthesis through distribution matching. In *ICLR*, 2024. 2
- [46] Yunlong Zhao, Xiaoheng Deng, Yijing Liu, Xinjun Pei, Jiazhi Xia, and Wei Chen. Fully exploiting every real sample: Superpixel sample gradient model stealing. In *CVPR*, 2024. 1, 2
- [47] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *CVPR*, 2020. 1, 2
- [48] Hongyu Zhu, Wentao Hu, Sichu Liang, Fangqi Li, Wenwen Wang, and Shilin Wang. Efficient and effective model extraction. *arXiv preprint arXiv:2409.14122*, 2024. 5