

# Examining the Threat Landscape: Foundation Models and Model Stealing

Ankita Raj<sup>1</sup>  
ankita.raj@cse.iitd.ac.in  
Deepankar Varma<sup>†2</sup>  
satwikdpshrit@gmail.com  
Chetan Arora<sup>1</sup>  
chetan@cse.iitd.ac.in

<sup>1</sup> Indian Institute of Technology Delhi,  
India  
<sup>2</sup> Thapar Institute of Engineering and  
Technology,  
Patiala, India

---

## Abstract

Foundation models (FMs) for computer vision learn rich and robust representations, enabling their adaptation to task/domain-specific deployments with little to no fine-tuning. However, we posit that the very same strength can make applications based on FMs vulnerable to model stealing attacks. Through empirical analysis, we reveal that models fine-tuned from FMs harbor heightened susceptibility to model stealing, compared to conventional vision architectures like ResNets. We hypothesize that this behavior is due to the comprehensive encoding of visual patterns and features learned by FMs during pre-training, which are accessible to both the attacker and the victim. We report that an attacker is able to obtain 94.28% agreement (matched predictions with victim) for a Vision Transformer based victim model (ViT-L/16) trained on CIFAR-10 dataset, compared to only 73.20% agreement for a ResNet-18 victim, when using ViT-L/16 as the thief model. We arguably show, for the first time, that utilizing FMs for downstream tasks may not be the best choice for deployment in commercial APIs due to their susceptibility to model theft. We thereby alert model owners towards the associated security risks, and highlight the need for robust security measures to safeguard such models against theft. Code is available at [https://github.com/rajankita/foundation\\_model\\_stealing](https://github.com/rajankita/foundation_model_stealing).

## 1 Introduction

**Model Stealing Attacks.** Driven by a huge surge in the capabilities of Machine Learning (ML) techniques, many companies now deploy trained ML models on the cloud, and monetize by providing paid access to users via Application Programming Interfaces (APIs). The trained model and the training dataset are often the intellectual property of the company, and therefore the model internals, including the training dataset, model architecture, and weights are kept hidden, providing only black-box access to users. However, such models are vulnerable to model stealing attacks [28, 30, 41], wherein malicious users replicate the behavior of a model by querying the API on a select set of inputs, and training a substitute model on

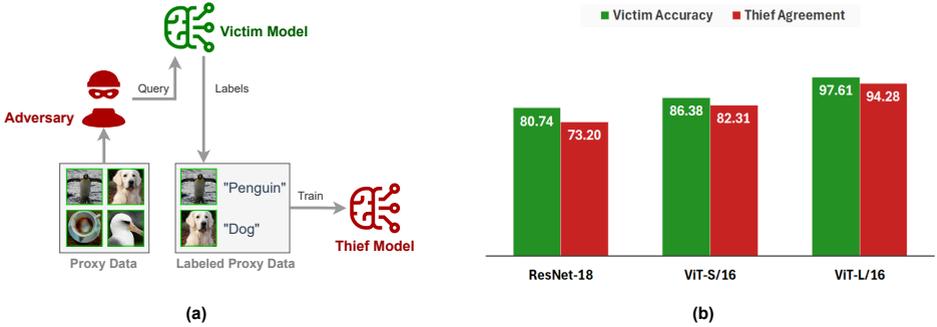


Figure 1: (a) Standard model stealing setup: An adversary picks images from a proxy dataset and queries from the victim model to obtain labels. This labeled proxy dataset is used to train the thief model. (b) Victims derived from foundation models are more prone to stealing: We steal three victim models trained on the CIFAR-10 dataset, using a ViT-L/16 thief. Even though using stronger victims based on foundation models like ViT-L/16 improves victim accuracy, but the agreement between the victim’s and thief’s predictions also increases at the same time, underlining the increased severity of the threat.

the acquired predictions (see Figure 1(a)). The model thus obtained can either be used as a substitute for the victim model, thereby extracting commercial value from it, or to launch further attacks like adversarial [13, 26] or model inversion [47] on the victim model (called adversarial transfer). In either case, the integrity of the victim model is compromised.

**Foundation Models.** Recent development in foundation models such as Vision Transformers [10], and powerful pre-trained encoders like CLIP [33] and ALIGN [20] have greatly advanced the field of Computer Vision. The availability of foundation models pre-trained on massive datasets has greatly enhanced the capabilities of model owners, who can train task-specific models on downstream applications easily by fine-tuning these models. The resulting downstream models boast high accuracy, causing model owners to be more inclined towards choosing foundation models over conventional vision architectures [16, 39]. While ViTs [10] are known to be more robust compared to convolutional architectures in terms of both adversarial attacks [37] and common corruptions [30], the model stealing risks associated with downstreaming these models for commercial APIs have not been systematically investigated.

**Our Focus.** In this paper, we study the vulnerability of image classification APIs based on foundation models to model stealing attacks. The victim models are foundation models (ViTs) fine-tuned on downstream datasets that are made accessible via black-box APIs. We consider both final layer fine-tuning, aka *linear probing*, and *full fine-tuning*. Firstly, we show that using more accurate victim models may not always be the best bet for model owners from the perspective of model stealing. This is because, not only the model owners, but attackers can also avail of these powerful pre-trained models. Assuming a well equipped thief, which has access to at least as strong a foundation model as the victim, we show that victim models obtained by fine-tuning from foundation models have a heightened susceptibility to theft compared to smaller models like ResNets. This is due to the extensive knowledge encapsulated within these foundation models, which is now available to both victim and adversary/thief models. Secondly, attackers armed with access to a foundation

model, by virtue of stronger representation available, can execute model theft with relative ease, when compared to shallower thieves.

**Higher Accuracy or Better Privacy?** By carrying out this study, we aim to raise awareness about the risks associated with using foundation models in the Machine Learning as a Service (MLaaS) setup. While it is always beneficial for an adversary to use foundation models for stealing, the victim needs to be more careful in their choice, and must choose between higher accuracy and better privacy. Our study also underscores the necessity for robust security protocols and countermeasures in the deployment and utilization of these models.

**Contributions.** We make the following contributions: **(1)** We conduct a thorough systematic study on three datasets, seven victim models, and four thief models to evaluate the model stealing vulnerability of victim models obtained from foundation models, particularly ViTs, either via linear probing or by fine-tuning all layers. **(2)** Our studies conclude that, under a strong attack wherein the thief has access to pre-trained foundation models, models fine-tuned from ViTs are more vulnerable to theft as compared to models fine-tuned from convolutional architectures. Using a ViT-L/16 thief, we report agreements of 94.28%, 60.52% and 62.94% for victim models based on ViT-L/16 model trained on CIFAR-10, Indoor-67 and Caltech-256 datasets respectively, compared to agreements of 73.20%, 40.22% and 46.23% respectively for ResNet-18 based victims. **(3)** Even when the victim model is not fine-tuned from a foundation model, we show that foundation models greatly enhance the attacker’s capabilities. When stealing a ResNet-18 victim trained on CIFAR-10 dataset, we report an agreement of 77.12% for a ViT-B/16 CLIP-based thief, compared to 64.53% agreement for a ResNet-34 thief (refer Figure 3).

## 2 Related Work

**Model Stealing Attacks.** Model stealing attacks aim to replicate black-box machine learning models, either by extracting exact weights or hyperparameters of the victim model [0, 0, 01, 18, 19, 27], or by approximately mimicking the victim model’s predictions [0, 01, 18]. The latter method involves iteratively querying the victim on a designated set of inputs, often referred to as a "proxy" dataset, and subsequently training a substitute model based on the acquired predictions. Some studies [0, 28, 29, 45] use natural images from publicly available datasets as the proxy data, devising techniques to select the most informative samples to minimize the attacker’s cost. Another school of methods [3, 21, 35, 42, 44, 48] generates synthetic data for querying the victim, eliminating the reliance on natural images but incurring significant costs for the attacker due to the necessity of millions of queries. In this work, we adopt an approximate model stealing setup utilizing natural images, recognizing its practicality and resemblance to real-world attack scenarios.

**Foundation Models in Computer Vision.** Foundation models, initially conceived within the NLP domain, have found widespread adoption within the computer vision community. Trained on massive amounts of data in a supervised or self-supervised manner, these models learn rich, generalized representations that can serve as the backbone for various downstream tasks. Among these, Vision Transformers (ViTs) [01] stand out as prominent foundation models in vision research that excel in capturing global dependencies and contextual information from visual data using self-attention mechanisms. Since the original Vision Transformer (ViT) [01] model, several variants have been proposed including DeiT [40], Swin Transformer [29], DINO [5], etc. Additionally, models like CLIP [33] and ALIGN [20] have

further expanded the horizon of foundation models in vision, by aligning visual and textual representations into a shared embedding space.

**Foundation Models and Model Stealing.** In the NLP domain, Krishna *et al.* [22] demonstrated that models fine-tuned from large pretrained language models like BERT can be stolen by issuing semantically irrelevant input queries. In computer vision, model stealing attacks have been studied only for CNN based victim models for the image classification task. Battis *et al.* evaluated the role of ViTs in model stealing, but only from the attacker’s perspective. An emerging thread of work [12, 24, 36] deals with stealing large pre-trained image encoders like SimCLR [8], MoCo [17], BYOL [15] and CLIP [63]. These methods aim to steal general-purpose encoders that return rich feature embeddings rather than model posteriors or labels, rendering them highly vulnerable to theft. Differently, in our work, we steal classification models fine-tuned from large foundation models on downstream tasks.

## 3 Methodology

### 3.1 Victim Model

The victim model is a neural network  $f_V$  trained for image classification. The model owner trains  $f_V$  on labeled images from training data distribution  $P_V(X)$ . To ease the training, the model owner uses an open-source pre-trained model, and fine-tunes it on their dataset, called victim dataset. The pre-trained model could either be conventional architectures popularly used in computer vision, e.g., VGG [68] or ResNet [16], or modern large foundation models like ViT [11] or CLIP [63] which are pretrained on huge datasets and come with strong representation power. The output of the model,  $y \in \{1, \dots, K\}$ , is a distribution over  $K$  classes.

The trained victim model is deployed on a cloud service platform as a black-box. In this setting, the victim’s architecture and weights are hidden, but a user can query the model via an API and obtain its predictions on a given image. This setup allows the victim to monetize from their model by charging the user on query basis. Several previous works assume the availability of the full probability vector  $f_V(x) \in \mathbb{R}^K$  (also known as *soft labels*) as the victim’s output. However, many real-world APIs only return the topmost prediction  $\text{argmax}_{i \in \{1, \dots, K\}} f_V(x)_i$  for a queried image, also known as *hard-label*. We adapt the hard-label setup in our experiments, on account of it being closest to real-world scenario.

### 3.2 Attacker’s Goal

The attacker’s objective is to replicate the behaviour of the victim model. We don’t mean to replicate the exact weights of the victim neural network, but to train a substitute/thief model  $f_T$  that is functionally equivalent to the victim model in terms of predictions on the victim’s held-out test set.

### 3.3 Attack Method

A major constraint for the attacker is that it does not have access to the victim’s training data distribution  $P_V(X)$ . It therefore, uses a *proxy* distribution  $P_A(X)$  to query the victim model. The attacker selects a subset of images  $\{x^i\}_{i=1}^m$  from  $P_A(X)$  and receives predictions for the queried images, thus constructing a labeled set  $\mathcal{D}_l = \{(x^i, f_V(x^i))\}_{i=1}^m$  of size  $m$  which is then used to train the thief model  $f_T$ . Fundamentally, a thief model uses information obtained

from the victim in the form of queried labels to learn similar decision boundaries as the victim model. The performance of the thief model depends on the following factors:

**Proxy dataset.** For computer vision applications, the proxy dataset can be constructed from publicly accessible natural images, or by generating synthetic images. In line with previous works [28, 29], we use large scale publicly available datasets of natural images as the proxy distribution.

**Query selection method.** For practical model stealing attacks, the thief has to work under limited query budgets. As such, there is a large body of works dedicated to selecting the best set of samples from the proxy dataset to query the victim model. This includes the reinforcement-learning methods by [28], active learning based methods by [29] and GRAD-CAM based methods by [45]. In this work, we adopt the simple yet effective Random selection strategy [28] for most of our experiments. The impact of changing the sample selection technique is studied in Section 5.4.

**Thief architecture.** Typically, the attacker has no knowledge of the victim model’s architecture or hyper-parameters. However, several works on model stealing assume that the attacker uses the same architecture as the victim model, for ease of experimentation, and also owing to the belief that the thief model’s architecture does not significantly affect the model stealing performance. However, given the easy availability of large and powerful foundation models, it would be prudent for an attacker to use these larger models instead. We therefore assume that an informed thief is able to use open-source pretrained models that are available on the internet, and fine-tunes the model on the labeled dataset  $\mathcal{D}_l$  constructed from the proxy data. For completeness, we also study the special case scenario when the thief model has the same architecture as the victim (in the Supplementary).

## 4 Experimental Setup

### 4.1 Victim Datasets and Architectures

**Datasets.** We adopt three commonly-used image recognition datasets to train our victim models. **CIFAR-10** [23], **Caltech-256** [24], and **Indoor-67** [52]. Of these, CIFAR-10 and Caltech-256 are general-purpose object recognition datasets with 10 and 256 object classes respectively, while Indoor-67 is a fine-grained classification dataset comprising 67 types of indoor scenes.

**Architectures.** We work with seven different victim architectures in our experiments. These include four conventional CNN architectures: **ResNet-18**, **ResNet-34**, **ResNet-50** and **ResNet-101** [16] pre-trained on ImageNet-1K dataset [54] containing 1.2M images, and three vision foundational models: **ViT-S/16**, **ViT-B/16** and **ViT-L/16** [10] pre-trained on the larger ImageNet-21K dataset [9] containing 14M images from 21,841 classes. The power of these models comes not only from the deeper architectures, but also from stronger pre-training on massive datasets.

**Training details.** Fine-tuning on ImageNet-pretrained weights is a widely adopted method of training CNN models. However, with the advent of powerful foundation models, fine-tuning only the final layer (a.k.a linear probing) is gaining popularity, as it offers good results at low training costs. We, therefore, consider both scenarios in our experiments. All models are initialized using publicly available ImageNet [54] pre-trained weights: the ResNet models are pre-trained on ImageNet-1K, and the ViTs are pre-trained using the larger ImageNet-

21K dataset. All pre-trained weights are obtained from open-source libraries like Pytorch hub [10] or timm [46]. All victim models are trained using SGD optimizer for 100 epochs, with a momentum of 0.9, weight decay of  $5 \times 10^{-4}$ , and batch size of 128. An initial learning rate is 0.002 is decayed by a factor of 10 after every 30 epochs. All input images are resized to  $224 \times 224$  and augmented using random cropping and random horizontal flipping.

## 4.2 Thief Datasets and Architectures

**Dataset.** We use publicly available unannotated images from the training set of ILSVRC-2012 challenge dataset [39] as the thief’s proxy dataset. Commonly known as ImageNet, the dataset contains 1.2 million images of varying sizes, of which we only use a subset of 128K images resized to  $224 \times 224$ .

**Architectures.** We consider multiple model architectures for the thief model. These include a CNN model ResNet-34 [16], two ViT models: ViT-B/16 and ViT-L/16 [10], as well as a multimodal foundation model ViT-B/16 CLIP [33]. In general, we assume that the thief has access to all publicly available pre-trained models that the victim has access to.

**Training details.** Thief models are trained using SGD optimizer for 100 epochs, with a momentum of 0.9, weight decay of  $5 \times 10^{-4}$ , and a batch size of 128 (for ResNet-34), 32 (for ViT-B/16 and ViT-B/16 CLIP) or 64 (ViT-L/16). The initial learning rate is 0.001, and it is decayed by a factor of 10 after 50 epochs. For stealing CIFAR-10 victim, input images are resized to  $224 \times 224$ , and augmented using random cropping, random rotation and random horizontal flip. For stealing other victims, we use random crop, random horizontal flip and RandAugment [8]. All networks are initialized from ImageNet-pretrained weights. For ResNet thieves, all layers are fine-tuned, whereas for ViT’s only the final layer is fine-tuned.

**Sample Selection.** For majority of our experiments, we consider the random-adversary based sample selection method from Knockoff Nets [28], and limit the thief’s query budget to 5000 samples. Of these, 10% samples are set aside as validation data for hyper-parameter selection, and the rest are used for training. In Section 5.4, we evaluate the impact of the sample selection method and query budget on our findings.

## 4.3 Evaluation Metrics

We use two evaluation metrics prevalent in the literature. *Accuracy* measures the performance of the thief model on the victim model’s held-out test set, while *Agreement* measures how often the thief’s prediction matches the victim’s. Note that the held-out victim dataset is not accessible to the thief, and is only used for evaluation.

# 5 Experiments and Results

## 5.1 Foundation Model Victims are Easy to Steal

In this section, we try to answer the question: *Is a victim fine-tuned from a foundation model more vulnerable to model stealing compared to a victim fine-tuned from a shallower model?* To do so, we train multiple victim models, including both conventional architectures and foundation models. For the thief model, we assume a well-equipped attacker who utilizes large, publicly available pre-trained foundation models, as reasoned in 3.3.

	Victim architecture	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ViT-S/16	ViT-B/16	ViT-L/16
	Params (M)	11.18	21.29	23.53	42.52	21.66	85.80	303.31
	Pretraining dataset	IN-1K	IN-1K	IN-1K	IN-1K	IN-21K	IN-21K	IN-21K
CIFAR	Victim accuracy	80.74	81.96	80.24	84.64	86.38	94.12	97.61
	Thief accuracy	82.97	88.92	81.17	83.92	90.49	93.79	94.53
	Thief agreement	73.20	77.85	71.56	75.39	82.31	91.10	94.28
Indoor	Victim accuracy	54.70	58.06	45.97	51.49	78.51	82.76	87.39
	Thief accuracy	36.87	39.70	34.10	29.85	51.49	46.57	59.03
	Thief agreement	40.22	39.63	44.70	36.72	49.70	46.79	60.52
Caltech	Victim accuracy	67.75	74.36	58.47	74.78	84.78	87.67	94.13
	Thief accuracy	47.42	54.86	35.38	53.63	58.20	57.91	62.83
	Thief agreement	46.23	52.39	44.80	56.75	54.22	54.72	62.94

Table 1: Model stealing results for **linear-probed victim** models on three victim datasets. Thief model is ViT-L/16. IN stands for ImageNet.

**Linear-probed Victims.** We first consider victim models that are trained by fine-tuning the last layer of the respective pre-trained models, which is a common practice with foundation models. The thief model is a ViT-L/16, which is also linear-probed. We report the victim models’ accuracy, along with the accuracy and agreement of the thief models in Table 1. We draw the following observations. Firstly, using foundation models leads to an increase in victim model accuracy, which is expected. Secondly, the ViT-L/16 thief’s accuracy is higher for victims derived from foundation models (ViT-S/16, B/16 and L/16). This can be attributed partly to the higher victim model accuracy for foundation models. Therefore, we also report the agreement between the victim and thief models, which directly quantifies the similarity between the predictions of the victim and thief. Thirdly, we see that agreements are also higher for ViT victims compared to ResNets (see Figure 2). On the Indoor-67 dataset for instance, agreement for a ResNet-18 victim is only 40.22%, as compared to 60.52% for a ViT-L/16 victim. These findings suggest that while foundation models like ViTs offer higher accuracy for the victim, they are also more susceptible to theft compared to non-foundation models, particularly when targeted by well-equipped attackers who utilize foundation models themselves.

**Fully Fine-tuned Victims.** In this section, we evaluate victims obtained by fine-tuning all layers of the backbone models, while the thief is still a linear-probed ViT-L/16. We show the results for stealing multiple victim models using a ViT-L/16 thief in Table 2, and once again, observe higher thief agreements for ViT victims compared to ResNets.

## 5.2 Foundation Models make Strong Thieves

So far, we have analyzed the impact of using foundation models on the victim’s end. In this section, we look at the other perspective and try to answer the question: *How do foundation models affect the capabilities of a thief?* We vary the thief’s model architecture and study its impact on model stealing performance for various victim models. We compare the agree-

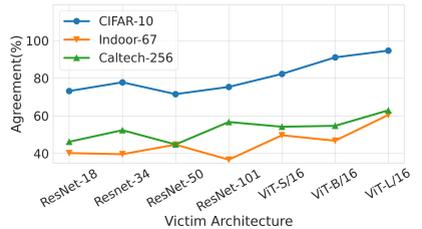


Figure 2: Agreement between multi-linear-probed victims (trained on three datasets) and a linear-probed ViT-L/16 thief.

	Victim architecture	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ViT-S/16	ViT-B/16
	Params (M)	11.18	21.29	23.53	42.52	21.66	85.80
	Pretraining dataset	IN-1K	IN-1K	IN-1K	IN-1K	IN-21K	IN-21K
CIFAR	Victim accuracy	96.54	97.41	97.62	98.42	98.33	98.33
	Thief accuracy	92.17	94.74	94.74	93.80	94.67	91.48
	Thief agreement	90.89	93.96	93.80	93.10	94.01	91.06
Indoor	Victim accuracy	75.60	78.73	81.64	82.61	87.01	86.19
	Thief accuracy	43.21	41.64	47.16	49.63	55.52	52.83
	Thief agreement	41.27	41.42	48.28	49.48	55.97	53.96
Caltech	Victim accuracy	76.78	81.39	86.00	88.70	93.44	95.14
	Thief accuracy	47.73	56.14	61.77	63.20	63.00	57.57
	Thief agreement	44.58	52.78	60.23	61.67	62.25	56.98

Table 2: Model stealing results for **fully fine-tuned victim** models on three different victim datasets. Thief model is ViT-L/16. IN stands for ImageNet.

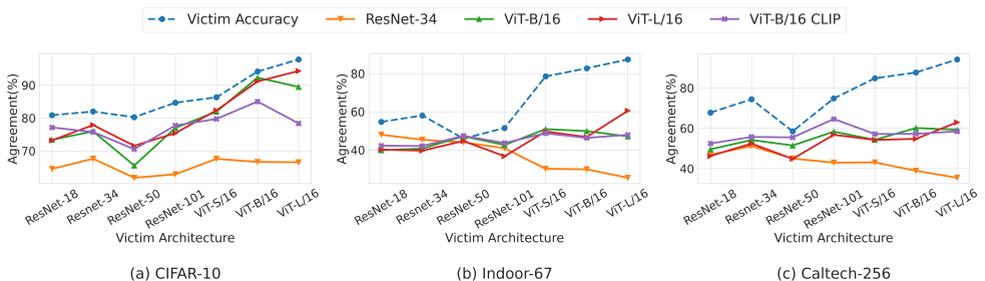


Figure 3: Agreement for thieves based on foundation models (ViTs) vs. a ResNet-34 based thief. Victim models are **linear-probed**.

ments for four thief architectures: ResNet-34, ViT-B/16, and ViT-L/16, and ViT-B/16 CLIP, for linear-probed victims in Figure 3 and fully fine-tuned victims in Figure 4. We can see that for a given victim architecture, especially deeper architectures, a foundation model thief achieves higher agreement as compared to a regular ResNet thief. More importantly, the gap in performance between the ResNet thief and ViT thieves increases, as victim models become deeper. When stealing smaller capacity victim models, the ResNet thief performs as well as, or even better than the stronger ViT thieves, especially for Indoor-67 and Caltech-256 datasets. But when the victim model itself is a higher capacity model, the ResNet thief manages to recover only a small proportion of the victim model’s accuracy. For instance, in Figure 3, when stealing a ViT-B/16 victim trained on Indoor-67 dataset, a ResNet-34 thief achieves an agreement of only 25.44%, whereas a ViT-L/16 thief achieves 46.79% agreement. Overall, we observe that foundation models serve as better thieves, particularly when the victims are also derived from foundation models.

### 5.3 Qualitative Analysis

To understand why stealing victims fine-tuned from foundation models is more successful, we visualize embeddings of the pretrained backbone models and the victim models in Figure 5. Using the CIFAR-10 dataset for ease of visualization, we select five victim models trained by linear probing. For the backbone, we plot t-SNE embeddings [13] of the penulti-

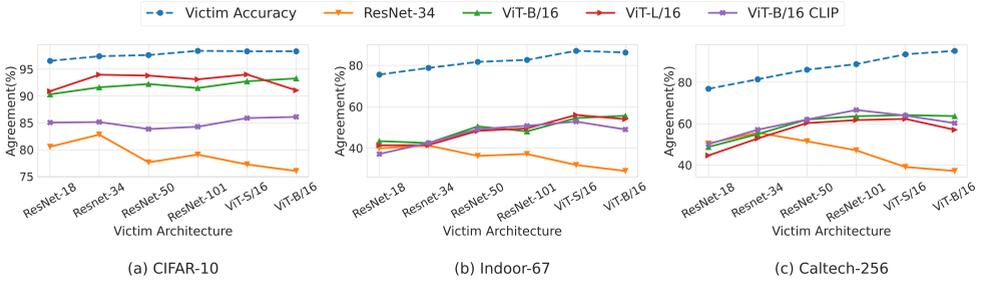


Figure 4: Agreement for thieves based on foundation models (ViTs) vs. a ResNet-34 based thief. Victim models are **fully fine-tuned**.

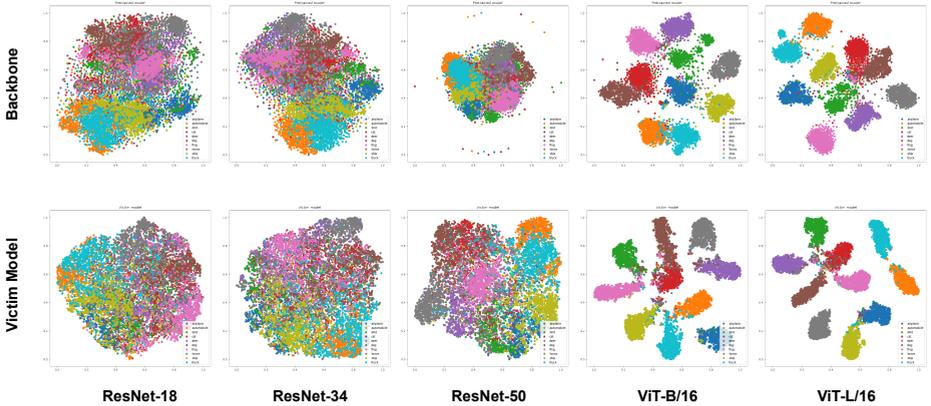


Figure 5: t-SNE [14] visualizations of embeddings for backbone models (top row), and corresponding victim models (bottom row) trained on CIFAR-10 dataset using linear probing method. Observe that the clusters from the backbone models are much more well-separated for the foundation models (ViT-B/16 and ViT-L/16) compared to ResNets.

mate layer on CIFAR-10 test set. Since the victim models are obtained by fine-tuning only the classification head, we do not expect a change in embeddings of the penultimate layer. We therefore, plot the final classification layer embeddings for the victim models. Notably, for foundation models like ViT-B/16 and ViT-L/16, the backbone is already strong enough to form well-separated clusters on CIFAR-10 despite not having been trained on CIFAR-10. This clear separation of classes extends to the corresponding victim models. We hypothesize that due to the rich representations captured by the backbone architecture in foundation models, the primary computational burden in the victim model resides within this backbone structure, with relatively minimal reliance on the classification head. Consequently, when the thief model possesses a similarly robust backbone architecture, stealing the classification head becomes significantly easier. This stands in contrast to architectures such as ResNet-18, where the pre-trained features within the backbone are comparatively less potent, thereby rendering the task of stealing the classification head more challenging.

## 5.4 Ablation Study

**Impact of query budget.** We vary the query budget in the range {2K, 5K, 10K, 20K} and compute agreement for different linear-probed victims on the Indoor-67 dataset, and a ViT-B/16 thief. Our findings indicate that the same trend holds for all query budgets: foundation models can be stolen with higher agreement compared to ResNets. While the gap in performance of ResNets and ViTs is less for lower budgets, increasing the number of queries proves more detrimental for ViTs than for ResNets (figure in Supplementary).

**Impact of sample selection method.** So far, we had used Knockoff Nets' random sample selection method [28] to select query points from the proxy dataset. We deploy two more query-set selection strategies: Entropy-based and kCenter-based methods from ActiveThief [29] and observe similar trends in agreement for foundation models and ResNets, for linear-probed victims on the Indoor-67 dataset, stolen using a ViT-B/16 thief (figure in Supplementary), proving the generality of our findings.

## 6 Conclusion

We studied the susceptibility of image classification models fine-tuned from powerful foundation models to model stealing attacks. Our findings reveal that victim models derived from foundation models exhibit greater vulnerability to such attacks from strong, foundation-model based thieves, compared to those derived from shallower backbones. In stark contrast to the celebrated robustness of foundation models against adversarial and natural corruptions, our study sheds light on their heightened susceptibility in the model stealing context. This highlights a crucial trade-off between privacy and accuracy inherent in deploying models fine-tuned from foundation models in commercial APIs, and emphasizes the necessity for enhanced security measures in model deployment strategies.

## Acknowledgement

We acknowledge and thank the funding support from AIIMS Delhi-IIT Delhi Center of Excellence in AI funded by Ministry of Education, government of India, Central Project Management Unit, IIT Jammu with sanction number IITJMU/CPMU-AI/2024/0002.

## References

- [1] Pytorch hub. URL <https://pytorch.org/hub/>.
- [2] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. Csi neural network: Using side-channels to recover your artificial neural network information. *arXiv preprint arXiv:1810.09076*, 2018.
- [3] James Beetham, Navid Kardan, Ajmal Mian, and Mubarak Shah. Dual student networks for data-free model stealing. In *Dual Student Networks for Data-Free Model Stealing*. Eleventh International Conference on Learning Representations (ICLR), 2023.

- [4] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. In *Annual International Cryptology Conference*, pages 189–218. Springer, 2020.
- [5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [7] Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F de Souza, and Thiago Oliveira-Santos. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
- [8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [11] Vasisht Duddu, Debasis Samanta, D Vijay Rao, and Valentina E Balas. Stealing neural networks via timing side channels. *arXiv preprint arXiv:1812.11720*, 2018.
- [12] Adam Dziedziec, Nikita Dhawan, Muhammad Ahmad Kaleem, Jonas Guan, and Nicolas Papernot. On the difficulty of defending self-supervised learning against model extraction. In *International Conference on Machine Learning*, pages 5757–5776. PMLR, 2022.
- [13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [14] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- [15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [18] Weizhe Hua, Zhiru Zhang, and G Edward Suh. Reverse engineering convolutional neural networks through side-channel information leaks. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2018.
- [19] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pages 1345–1362, 2020.
- [20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [21] Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13814–13823, 2021.
- [22] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*, 2019.
- [23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [24] Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. Stolenencoder: stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2115–2128, 2022.
- [25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [26] Mantas Mazeika, Bo Li, and David Forsyth. How to steer your adversary: Targeted and efficient model stealing defenses with gradient redirection. In *International Conference on Machine Learning*, pages 15241–15254. PMLR, 2022.
- [27] Smitha Milli, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9, 2019.
- [28] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963, 2019.

- [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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 865–872, 2020.
- [30] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.
- [31] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.
- [32] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [34] 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. *International journal of computer vision*, 115 (3):211–252, 2015.
- [35] Sunandini Sanyal, Sravanti Addepalli, and R Venkatesh Babu. Towards data-free model stealing in a hard label setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15284–15293, 2022.
- [36] Zeyang Sha, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. Can’t steal? cont-steal! contrastive stealing attacks against image encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16373–16383, 2023.
- [37] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-jui Hsieh. On the adversarial robustness of vision transformers. In *Annual Conference on Neural Information Processing Systems*, 2022.
- [38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.
- [40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

- [41] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.
- [42] Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4771–4780, 2021.
- [43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [44] Wenxuan Wang, Bangjie Yin, Taiping Yao, Li Zhang, Yanwei Fu, Shouhong Ding, Jilin Li, Feiyue Huang, and Xiangyang Xue. Delving into data: Effectively substitute training for black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4761–4770, 2021.
- [45] 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 *European Conference on Computer Vision*, pages 192–208. Springer, 2022.
- [46] Ross Wightman. Pytorch image models. <https://github.com/huggingface/pytorch-image-models>, 2019.
- [47] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.
- [48] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2020.