Army of Thieves: Enhancing Black-Box Model Extraction via Ensemble based sample selection

Akshit Jindal
IIIT-Delhi
akshitj@iiitd.ac.in

Vikram Goyal
IIIT-Delhi
vikram@iiitd.ac.in

Saket Anand
IIIT-Delhi
anands@iiitd.ac.in

Chetan Arora
IIT Delhi
chetan@cse.iitd.ac.in

Abstract

Machine Learning (ML) models become vulnerable to Model Stealing Attacks (MSA) when they are deployed as a service. In such attacks, the deployed model is queried repeatedly to build a labelled dataset. This dataset allows the attacker to train a thief model that mimics the original model. To maximize query efficiency, the attacker has to select the most informative subset of data points from the pool of available data. Existing attack strategies utilize approaches like Active Learning and Semi-Supervised learning to minimize costs. However, in the black-box setting, these approaches may select sub-optimal samples as they train only one thief model. Depending on the thief model's capacity and the data it was pretrained on, the model might even select noisy samples that harm the learning process. In this work, we explore the usage of an ensemble of deep learning models as our thief model. We call our attack Army of Thieves(AOT) as we train multiple models with varying complexities to leverage the crowd's wisdom. Based on the ensemble's collective decision, uncertain samples are selected for querying, while the most confident samples are directly included in the training data. Our approach is the first one to utilize an ensemble of thief models to perform model extraction. We outperform the base approaches of existing state-of-the-art methods by at least 3% and achieve a 21% higher adversarial sample transferability than previous work for models trained on the CIFAR-10 dataset. Code is available at: https://github.com/akshitjindal1/AOT_WACV.

1. Introduction

Machine Learning (ML) models have become essential in various industries because they can learn from data and make accurate predictions. Many companies train and deploy models using cloud-based services such as Google Cloud Platform, Amazon web services etc. These services provide access to ML tools and infrastructure at a reasonable cost, making it viable to utilize ML commercially. A trained model's predictions are made available to the general public via an Application Programming Interface (API), where each API call incurs a cost to the user. This allows the model owner to monetize their trained model without exposing its inner workings. However, recent studies [18, 19, 24, 27] show that a malicious user can steal the deployed model's functionality even in such scenarios by querying it on a selected set of inputs. These types of at-

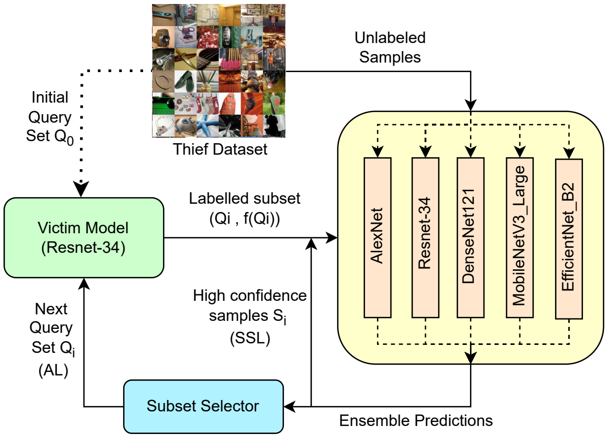


Figure 1. Overview of the Army of Thieves (AOT) Extraction Pipeline. The initial query set Q_0 is selected at random. Query outputs are then used to train the ensemble of five thief models. Based on the prediction confidence, highly confident samples are directly added to the labelled subset (Semi-Supervised Learning), while others are sent to the subset selector to curate the query set for the next training cycle (Active Learning).

tacks are called Model Extraction or Model Stealing attacks and constitute a significant threat to the intellectual property of the model owner. Moreover, stealing a model's functionality makes other attacks, such as membership inference and adversarial attacks, easier.

The model exposed to the risk of being stolen is referred to as the "victim" model, while the replica developed by the attacker is known as the "thief" model. The attacker aims to create a model with a comparable input-output behaviour to the victim's model. To accomplish this, the attacker assembles a pool of labelled or unlabeled data that is semantically similar or dissimilar to the victim's training data. Queries to the victim model are made using samples from this data pool, and the resulting predictions are utilized for training the thief model. The images in the attack dataset may be publicly available or artificially generated. Due to the limited resources of the attacker, it is not feasible to query every collected image blindly; thus, the attacker must carefully choose the images for each query. Researchers have used methods such as Active Learning [18] and Reinforcement Learning [17] to make each query as informative as possible, allowing the attacker to select the most useful subset of samples to query the victim efficiently. After exhausting the query budget, Semi-supervised learning techniques have been employed to effectively utilize the remaining unlabeled dataset [27].

The intelligent sample selection strategies have limitations in the black-box setting, where the attacker cannot access the model's architecture, training hyperparameters, and/or training data. In such a scenario, the attacker can query the model and observe only the output prediction, making it challenging to select informative samples. The amount of training data required to train a thief model rises with the rise in complexity of the thief model's architecture. This also leads to suboptimal sample selection in the first few cycles of the active learning process, as the model cannot accurately capture the required information from a low number of samples. As a result, previously proposed attacks require a large query budget before they can capture relevant information to train the thief model.

In this study, we investigate the utilization of an ensemble of thief models for conducting model extraction attacks. An overview of our approach is given in Figure 1. Ensembling techniques leverage the collective intelligence of multiple models to improve overall performance. By combining the knowledge and predictions of multiple thief models, we can alleviate individual model limitations, reduce noise and uncertainty, and ultimately increase the extraction success rate. This approach holds promise for addressing challenges associated with model extraction, such as intricate model structures, defensive mechanisms, and noise in the extracted information. However, due to the constraint of maintaining query costs within a budget, selecting samples that facilitate improved learning for each member of the ensemble becomes challenging. Furthermore, transferring existing Semi-Supervised Learning (SSL) techniques to the ensemble approach is not a straightforward task.

To enhance diversity, we incorporate ensemble members of different sizes and complexities. Each member is trained on the same set of labelled samples; the ones already queried from the victim model. Sample selection is based on the disagreement observed among ensemble members, as it serves as a metric for quantifying sample complexity. Specifically, samples with high disagreement are considered more valuable for training since they pose greater classification challenges. This approach offers several advantages, including reduced bias towards a single thief model and the selection of samples that better capture the joint input-output distribution. Additionally, due to budget constraints, the model extraction pipeline resembles Few-shot learning methods, thereby increasing the risk of overfitting a single thief model on limited available data. Conversely, the ensemble learning approach acts as a regularization technique, mitigating overfitting. Moreover, the output labels of an ensemble are less noisy and have higher confidence than a single model. The higher confidence leads to better semi-supervised learning once the query budget is depleted. Our work is the *first* to explore the usage of an ensemble of thief models for performing a model extraction attack. We will release the code and pretrained models upon acceptance.

2. Related Work

We study existing attacks in the following aspects:

2.1. Attack Objective

While the primary objective of model extraction is to create a duplicate copy of the target or victim ML model, the secondary objective might differ. The three secondary objectives, according to Jagielski et al. [10], are a) accuracy close to that of the victim model **(Task accuracy)**, b) prediction agreement over the victim's private training data **(Fidelity)** and c) prediction agreement over the entire input domain **(Functional Equivalence)**. We propose that an ensemble-based approach to model stealing would cover all three objectives as each individual model targets accuracy and fidelity while the overall ensemble would lead to functional equivalence.

2.2. Attack Strategy

Tramer et al. [24] were the first to steal ML models via prediction APIs. Following their work, many model extraction attacks have been proposed over the years. These attacks can be broadly classified into two categories: Exact extraction and Equivalent extraction. Exact extraction [16, 24] aims to recover the exact parameters of the victim model, e.g., equation solving to recover a linear re-

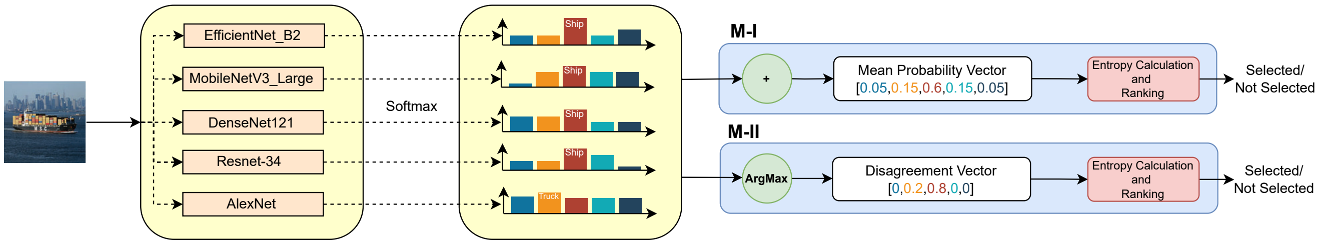


Figure 2. Detailed diagram of the Subset Selector module used for Active Learning. **M-I**: The mean Probability vector is computed for the Consensus Entropy method. **M-II**: Entropy of the Disagreement vector is calculated to measure Label Disagreement. Both methods are independent; only one is used for sample selection in a particular experiment.

gression model. However, extracting exact parameters for neural networks is impossible due to the sheer number of parameters and also due to the fact that different sets of parameters lead to similar outputs. Therefore, learning-based approaches [1, 18, 19, 22] have been proposed for equivalent extraction. These approaches treat the victim model as an oracle and treat a separate model using the model's outputs in a supervised learning fashion. Our ensemble-based approach falls under the equivalent extraction category.

2.3. Query Output from Victim

The amount of information provided by the victim model directly correlates to the extraction accuracy. Previous works that utilize the complete probability vector [17, 18, 25] perform far better than approaches that work in the top-1 or hard-label [2, 19, 20, 27] setting. However, the hard-label setting is closest to a real-world scenario as APIs often provide access only to the predicted label. In our work, we utilize only the predicted hard label from the victim.

2.4. Query dataset used

To perform queries to the victim model, previous works have utilized adversarial sample generation [8, 19], synthetic sample generation via GANs [20,25,29], and publicly available Out-of-Distribution datasets [7,17,18,27]. We utilize the publicly available dataset ImageNet [4] in our work for stealing Image Classification models.

2.5. Learning Paradigms used

Learning-based model extraction methods treat the victim model as an oracle. Various attacks utilize Active Learning [1, 10, 18] and Reinforcement Learning [17] to improve query efficiency. Moreover, Semi-supervised approaches have been used for Consistency Regularization [19, 27] after the query budget is exhausted to use the remaining data samples. So far, there have been no approaches that utilize an ensemble of thief models for performing model extraction though there have been approaches that steal an ensemble of models [14].

3. Proposed Attack

3.1. Recruiting Soldiers

It is a well-known fact that using an ensemble of machine learning models yields better results than using a single model, as the ensemble can handle noisy data and prevent overfitting. Thus we pick five models that have been widely accepted as standard architectures for image classification. All of the selected models are pretrained on the ImageNet dataset and are publically available via the Pytorch model hub. These models vary in their amount of trainable parameters and overall learning capacity, measured by their top-1 accuracy on the ImageNet dataset. However, we keep one model architecture the same as the victim model for comparison purposes.

3.2. Training the army

Active Learning Following the conventional model extraction pipeline, a random set of samples (Q_0) is selected from the unlabeled set (D_{UL}) for querying the victim model (f) in the first cycle. Each ensemble member (E_i) is then trained on the returned labels ($f(Q_0)$) in a supervised learning fashion. After each training cycle, the next set of unlabeled samples (Q_i) for querying the victim model is chosen in one of two ways:

- **Consensus Entropy** - For each unlabeled sample $D_{UL_i} \in D_{UL}$, the mean probability vector is computed from the softmax output of each ensemble member (p_{E_i}) to create a consensus vector (p_{cons}). Then the entropy of this consensus vector serves as the uncertainty measure for that sample. Samples with the highest consensus entropy are queried in the next cycle.

$$H(p_{cons}) = -\sum_{i=1}^{N} p_{cons_i} \log(p_{cons_i}) = H(\sum_{i=1}^{5} p_{E_i}) \tag{1}$$

Here N is the number of classes in the victim's output space and H is the entropy function.

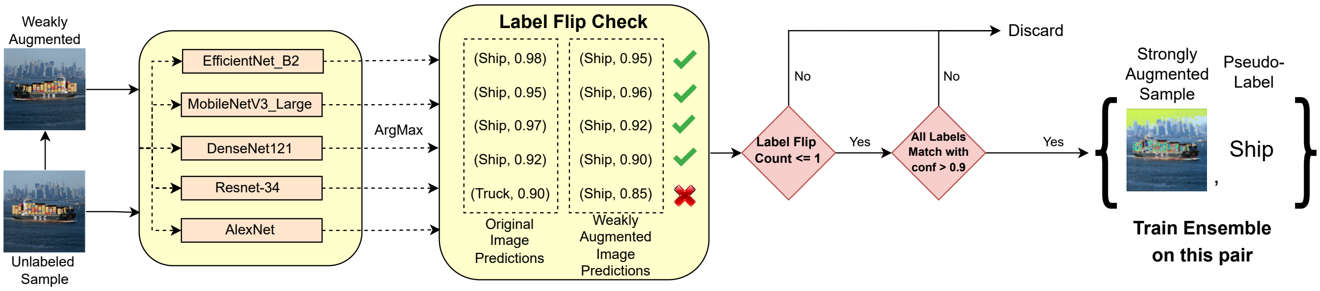


Figure 3. Detailed diagram of the Semi-Supervised Learning pipeline. Weak Augmentations like RandomHorizontalFlip are applied to each unlabeled sample, and the number of label changes are counted. Samples are selected iff the labels of at least 4 models do not change and each label is the same with a minimum confidence value of 0.9. The selected samples undergo a strong augmentation like RandAugment and the ensemble is trained on this (sample,pseudo-label) pair.

- **Label disagreement** - For each unlabeled sample, the disagreement between ensemble members' outputs serves as the metric for sample selection. If each ensemble member outputs a different label for a sample, it is likely a difficult sample and thus should be selected for querying. For each sample, we gather the output labels and create a label distribution vector p_{dis}. The entropy of this vector tells us the level of disagreement, as the entropy will be zero if all members output the same label and will be highest when each output label is different.

$$H(p_{dis}) = H([p_{c_1}, p_{c_1}, ..., p_{c_N}]) \tag{2}$$

where,

$$p_{c_i} = \frac{\text{Count of Label i in ensemble outputs}}{\text{Number of ensemble members}} \tag{3}$$

A detailed diagram explaining both selection strategies with an example is given in Figure 2. The samples selected at the end of the active learning cycle become part of the labelled subset D_L, i.e.

$$D_L = \bigcup_{k=1}^{cycles} Q_k$$

The number of cycles is usually kept equal to 10. Each sample belongs to the ImageNet dataset, but the labels from the victim model replace the corresponding output labels.

Semi-Supervised Learning After the query budget has been exhausted, we utilize the remaining unlabeled data via Semi-supervised learning. This allows us to further improve the accuracy of the thief models without querying the victim. Our semi-supervised approach is inspired by FixMatch [21], as shown in Figure 3. A concise version of our sample selection algorithm is provided in Algorithm 1. The exact steps are as follows:

1. First, we apply a weak augmentation, such as a Random horizontal flip, to our entire unlabeled set D_{UL} (nearly a million images). We infer each original sample $\mathbf{x}$ and its weakly augmented counterpart $\mathbf{x}'$ from our trained AOT and store the output probability vectors $\mathbf{p_i}$ for each ensemble member.

2. Secondly, we remove any samples for which the labels of one or more ensemble members change after weak augmentation, i.e. $argmax(p_i(\mathbf{x})) \neq argmax(p_i(\mathbf{x}'))$. The change shows a lack of robustness in the ensemble's predictions for such samples, and we don't want to reinforce such knowledge.

3. Then we filter those samples for which all ensemble members unanimously agree upon the output label $\mathbf{y}$. Furthermore, we select only those samples for which each model's output label confidence is above a certain confidence threshold, i.e. $\max(\mathbf{p_i}) \geq$ threshold.

4. The chosen samples $\mathbf{x}$ and their corresponding pseudo-labels $\mathbf{y}$ are then utilized for training the ensemble. Like the FixMatch approach, we apply strong augmentations such as RandAugment [3] to each sample before input and train the model to output the pseudo-labels.

5. To prevent forgetting, the learning rate is set to a small value of the order 1e-3, and each epoch consists of traversing both the labelled subset and the pseudo-label set once. The losses for each set are combined as per equation 4 and subsequently backpropagated for training the thief models.

$$\text{Total Loss} = Loss(D_L) + \lambda * Loss(D_{pseudo}) \tag{4}$$

To sum up, we use the ensemble's collective decision to initially select samples for querying and then select the most confident samples for consistency regularization. The former extracts the maximum amount of accuracy possible and the latter helps with the generalization ability of each ensemble member.

Algorithm 1 Sample Selection for Semi-supervised Learning

```
Require: Unlabeled set $D_{UL}$, Ensemble members $E$, confidence threshold t
Train the ensemble of models $E = \{E_1, E_2, \ldots, E_k\}$ on labeled set $D_L$ with cross-entropy loss.
for each sample $\mathbf{x}$ in $D_{UL}$ do
    $\mathbf{x}' = \text{WeakAugment}(\mathbf{x})$
    for each ensemble member $E_i$ in $E$ do
        $p_i(\mathbf{x}) = E_i(\mathbf{x})$
        $p_i(\mathbf{x}') = E_i(\mathbf{x}')$
        $l_i(\mathbf{x}) = \text{argmax}(p_i(\mathbf{x}))$
        $l_i(\mathbf{x}') = \text{argmax}(p_i(\mathbf{x}'))$
    end for
    changes $= \sum_{i=1}^{k} \mathbb{I}\{l_i(\mathbf{x}) \neq l_i(\mathbf{x}')\}$
    if changes $\leq 1$ then
        if $\forall i, l_i(\mathbf{x}) = y$ and $p_i(l_i) \geq t$ then
            Add sample $(\mathbf{x}, \mathbf{y})$ to $D_{pseudo}$
        end if
    end if
end for
```

4. Experiments

4.1. Victim Models

We train the victim Resnet-34 [6] model on four public image classification datasets: CIFAR-10 [12], CIFAR-100 [12], Caltech-256 [5] and CUBS-200 [26] to study and compare the proposed attack's effectiveness. We choose the above architecture and datasets as they have been used for similar studies in the past. The test accuracies for each model are 92.18%, 61.38%, 78.43%, and 77.11%, respectively. All models were trained using an SGD optimizer with a momentum of 0.5 for 200 epochs. We started with a base learning rate of 0.1, which was decayed by a factor of 0.1 every 30 epochs. After training, all models are treated as black-box APIs, i.e., we only utilize the final one-hot predictions for every input image.

4.2. Thief Model architectures

Previous works utilize the same model as the victim or a smaller model as the thief model. In our work, we use an ensemble of five models, namely Resnet-34 [6], Alexnet [13], MobilenetV3 [11], Densenet121 [9] and EfficientNet_B2 [23], as our thief model. The model sizes and the number of parameters are given in Table 1. All members are pretrained on the Imagenet dataset. One ensemble member, i.e., Resnet-34, was kept the same as the victim model. Other members were chosen based on their size, learning capacity, and architecture to enforce diversity.

Model Arch.	Size (MB)	Parameters (millions)	ImageNet Acc@1
AlexNet	233.08	61.1	56.522
Resnet-34	83.28	21.79	73.314
Densenet121	30.99	7.97	74.434
MobilenetV3	21.12	5.48	75.274
EfficientNet_B2	35.2	9.1	80.608

Table 1. Ensemble Members Details. We choose models based on their number of parameters and learning capacity. We use the Imagenet top-1 accuracy metric as our measure of learning capacity.

Alexnet has the lowest learning capacity due to its simpler architecture despite having the largest number of trainable parameters. On the other hand, EfficientNet has the highest learning capacity, as is evident by its performance on the ImageNet dataset. MobileNet has a relatively lower learning capacity but is useful in low-resource environments. DenseNet has a moderate number of parameters and is able to capture complex patterns in the data. Thus, each member contributes to the diversity of the ensemble in different ways.

4.3. Thief Dataset

As our task is to extract Image Classification models, we use the complete ILSVRC-2012 challenge dataset (1.2M images) [4] as our attack dataset, as per previous work. This is similar to an actual attack scenario, as the attacker might scrape the web to collect images. The Imagenet is a comprehensive dataset in terms of the number of classes and image resolution. Appropriate model-specific transformations are applied to each image before inputting to the victim and thief models.

4.4. Training the ensemble

For a valid comparison, we limit our maximum query budget to 30K, similar to previous works. 10% of the budget is set aside as the validation set whose accuracy is monitored while training the ensemble. The number of query cycles is set to 10, allowing us 2.7K queries per cycle. That means every cycle, we select 2.7K samples out of the 1.2 million ImageNet images to query the victim model and then train our ensemble using the victim's output. The initial selection is done at random.

For all ensemble members, during the active learning stage, we use the SGD optimizer with a momentum of 0.9 for 200 epochs. The learning rate is set to 0.01 for Alexnet and 0.02 for others. The learning rate is decayed by a factor of 0.1 after every 30 epochs. Weight decay is set to 0 for all datasets. After ten cycles, each ensemble member has seen 27K samples chosen via the specified strategy. This leaves us with roughly 1.25 million samples for semi-supervised learning.

Method	CIFAR-10		CIFAR-100		Caltech-256		CUBS-200	
	Acc	Agr	Acc	Agr	Acc	Agr	Acc	Agr
Random Selection	75.64	76.88	43.0	42.78	59.01	61.17	33.78	36.9
ActiveThief(Entropy)	74.21	75.26	40.59	40.8	54.14	56.28	29.43	32.05
ActiveThief(Kcenter)	74.24	75.71	42.29	42.77	58.84	61.19	34.64	37.68
BBD	**80.47**	**82.14**	14.32	15.37	61.41	63.61	36.28	39.07
+Kcenter	79.27	80.84	12.2	13.38	**63.75**	**66.34**	**44.43**	**48.46**
AOT Consensus	81.9	82.97	48.62	49.9	62.76	64.92	38.83	41.68
+SSL	82.5	83.62	**50.09**	**51.76**	60.81	63.03	39.29	42.19
+KCenter	81.59	82.64	49.68	50.44	**68.21**	**69.65**	**45.85**	**50.17**
AOT Voting	82.93	83.82	47.86	47.95	64.75	66.1	33.32	36.46
+SSL	**83.06**	**84.12**	49.9	49.85	62.57	63.78	30.79	33.58

Table 2. Experimentation Results. Accuracy (Acc) and Agreement (Agr) of the thief model are reported for each dataset. The highest results for our method and previous works are highlighted.

During the semi-supervised learning stage, we impose a selection limit of 100 samples for each class to avoid class imbalance. The learning rate is set to 0.002 to prevent catastrophic forgetting. Each epoch in this stage consists of a pass over the labelled as well as the unlabeled set.

4.5. Ensemble output and Comparison

Majority voting is used as the ensemble's final decision-making strategy, i.e. if at least three models output the same label, it is considered the ensemble's output label. We compare our method with ActiveThief [18] and Black-Box Dissector (BBD) [27]. We focus on the accuracy (Acc) and agreement (Agr) metrics as proposed by [18] for each ensemble member and the ensemble as a whole. As test sets for the victim dataset are seldom available in a real-world setting, we pick the best model for each ensemble member based on its validation set accuracy in each cycle. Since none of the previous works utilizes our chosen thief model architectures for extraction, their accuracies are reported only for the scenario where the thief model architecture is the same as the victim model.

5. Results and Analysis

5.1. Performance comparison

Table 2 shows the results of our method in comparison with previous methods. We call our attack the Army of Thieves (AOT) attack. The results for the two sample selection strategies, consensus entropy, and label disagreement, are shown separately. As semi-supervised learning (SSL) was applied after budget exhaustion, the results are shown in separate rows. From the table, it is evident that our method outperforms previous work in terms of accuracy and agreement on CIFAR-10 and CIFAR-100. We achieved a higher accuracy on Caltech-256, and our agreement value is on par with previous work. We perform better than the base approach of BBD on CUBS-200 but fall short when they use the KCenter algorithm. We infer this is due to the inherent complexity of the CUBS-200 dataset, which is also evident from the low extraction accuracies of all previous approaches, the best being 44%. The graphs showing each ensemble member's accuracy per active learning cycle are given in figures 4a and 4b.

Dataset	Model Arch.	Max Accuracy	Max Agreement
CIFAR-10	**Alexnet**	63.93	64.31
	DenseNet121	80.55	81.3
	EfficientNet_B2	87.61	87.12
	MobileNetV3_large	79.53	80.62
	Resnet32	74.21	75.33
CIFAR-100	**Alexnet**	41.56	43.06
	DenseNet121	44.60	45.19
	EfficientNet_B2	50.33	48.68
	MobileNetV3_large	43.62	45.90
	Resnet34	43.77	44.36
Caltech-256	**Alexnet**	41.87	43.45
	DenseNet121	60.75	62.43
	EfficientNet_B2	65.75	63.92
	MobileNetV3_large	60.9	60.95
	Resnet34	59.37	61.93
CUBS-200	**Alexnet**	21.12	22.36
	DenseNet121	36.64	39.52
	EfficientNet_B2	36.33	37.31
	MobileNetV3_large	33.7	36.27
	Resnet34	36.29	39.16

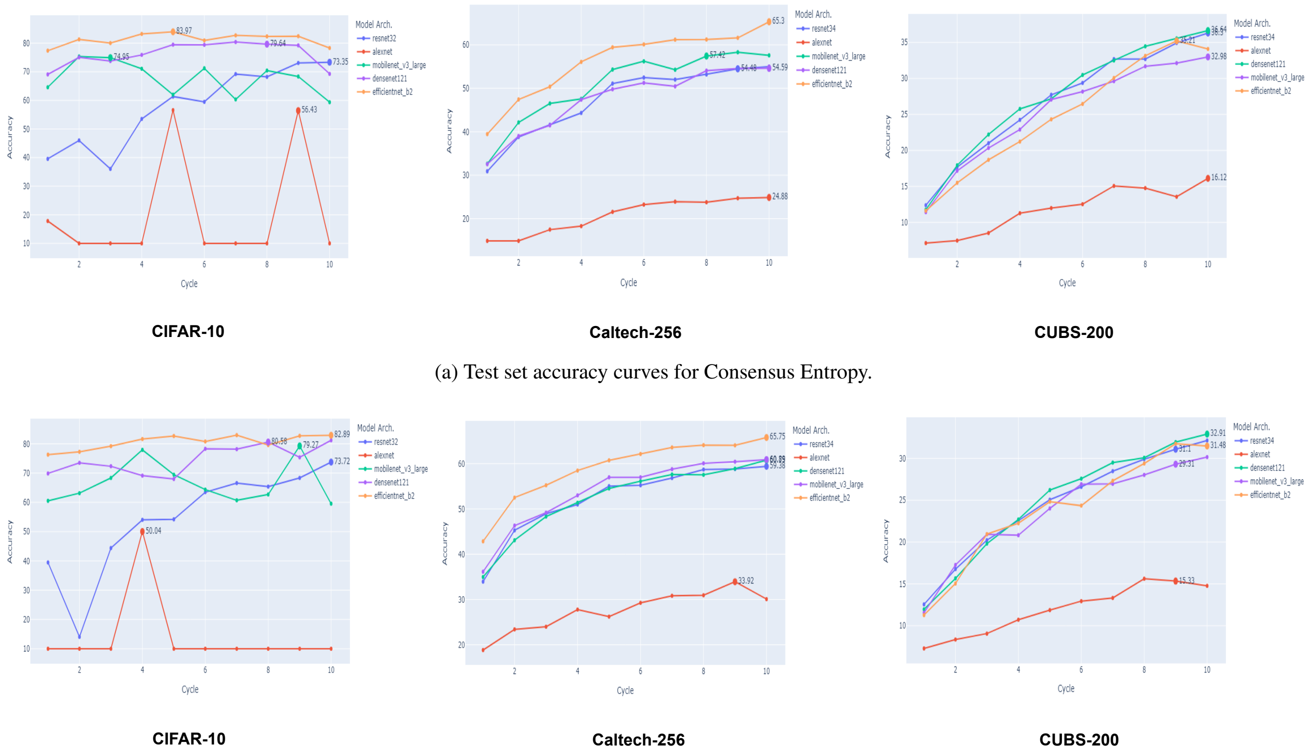
Table 3. Individual model best accuracies

5.2. Performance Analysis

In this section, we study the performance of our attack in terms of accuracy and agreement metrics. For CIFAR-10 and Caltech-256, we obtain the best performance when the selection strategy is disagreement-based instead of consensus-based. We observe that the semi-supervised learning approach helps the CIFAR-10 ensemble but fails for Caltech-256. We hypothesize that the class imbalance of the selected samples adversely affects the model. Due to the high number of classes (256), more samples get selected for the few highly confident classes in Caltech-256 as compared to CIFAR-10, which has only ten classes that are almost uniformly distributed. Due to the class imbalance, the model forgets information relevant to the less prevalent classes and leads to lower performance.

From the accuracy curves, it can be seen that accuracy for some models drops as more samples are selected. This contradicts the belief in active learning that accuracy should improve as more informative samples are selected with each cycle. We argue this happens primarily due to the nature of our ensemble, as a selected sample might not be equally informative for all models. This leads to minor drops in accuracy in the later stages of active learning. We also observe that Alexnet fails to learn optimally as the amount of available data is insufficient.

For comparison with the ensemble, max accuracy and agreement metrics are reported for each individual model in Table 3. we observe that EfficientNet alone outperforms the ensemble for the CIFAR-10 and Caltech-256 datasets. Due to the differences in architectures of the victim and thief



(a) Test set accuracy curves for Consensus Entropy.

(b) Test set accuracy curves for Label Disagreement.

Figure 4. Accuracy of ensemble members in each cycle of the Active Learning process for a)Consensus Entropy and b) Label Disagreement on all selected datasets. AlexNet shows sudden spikes in accuracy for the CIFAR-10 dataset but is stable for other datasets. The best accuracy for each model is mentioned on the corresponding scatter line. We choose the best model based on the validation set accuracy as test sets aren't available in a real-world scenario.

models, it becomes hard to verify whether the thief model has captured the exact decision boundaries as the victim or learnt entirely new ones due to the inherent capability of the architecture. We attempt to verify this by evaluating the adversarial sample transferability of each individual thief model. Adversarial samples created using MobileNet and EfficientNet do not transfer well to the victim Resnet-34 architecture. Both models generalize well in terms of accuracy and agreement but are not a valid "copy" of the victim model as they fail to capture its decision boundaries.

5.3. Adversarial Sample Transfer

Adversarial Sample Transferability is measured by counting how many failures are shared across the victim and thief model. As showcased in Table 4, our method achieves superior adversarial sample transferability than other methods when the thief model architecture matches the victim model. Even for DenseNet, the transferability is better than in previous work. All the adversarial samples were generated using the Projected Gradient Descent (PGD) [15] with a maximum L_∞-norm of 8/255 as per previous work in the field [27]. For EfficientNet, MobileNet, and AlexNet, the transferability numbers are 54.81%, 52.41%, and 59.66%, respectively. We argue that the transferability is low when the thief architecture does not match the victim, as the decision boundaries learned differ from the victim Resnet-34. The models provide high accuracy and agreement but cannot capture decision boundaries similar to the victim model.

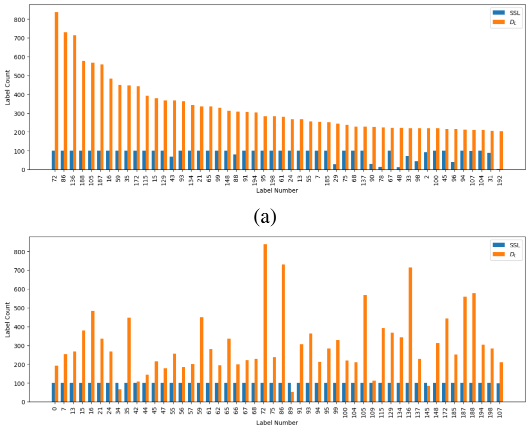
Method	**Substitute's Architecture**	
	Resnet-32	DenseNet
ActiveThief(k-center)	57.44%	60.72%
ActiveThief(Entropy)	63.56%	62.05%
BBD	76.63%	66.96%
Ours	**97.64%**	**71.64%**

Table 4. Adversarial Sample Transfer for the CIFAR-10 dataset using PGD attack. The numbers for previous works have been directly taken from the BBD [27] paper.

We also compare our transferability with the cross-domain Beyond ImageNet Adversarial Attack (BIA) [28] in Table 5. We generate adversarial versions of each sample in the CIFAR-10 using pretrained generators provided by the authors. From the table, it is evident that the samples

Model	Victim (Resnet-32)	Thief (Resnet-32)	Thief (DenseNet121)
Clean	92.18	73.54	79.64
PGD (Resnet-32)	**0.6**	0 (97.64)	-
PGD (DenseNet121)	**24.43**	-	0 (71.64)
BIA (Resnet152)	53.76	46.82 (61.60)	48.04 (64.58)
BIA (DenseNet169)	60.55	48.67 (55.22)	51.55 (58.68)

Table 5. We report the top-1 accuracy after performing adversarial attacks on models trained on the CIFAR-10 dataset. The adversarial transferability is also reported in parenthesis. PGD attacks on our thief models are more effective as compared to the BIA attack.



(a)

(b)

Figure 5. a) Labels with highest frequencies in the Labeled set are selected as part of the Semi-supervised Learning set. b) Labels with high frequency in the Semi-supervised set almost always have high frequencies in the Labeled set. Both graphs are for the CUBS-200 dataset.

generated using the PGD attack on our thief models transfer better to the victim model and cause larger drops in accuracy. The best transferability is again observed when the thief and victim architectures are similar.

5.4. Qualitative Analysis

During semi-supervised learning, the selection of samples is typically based on the high confidence exhibited by the model towards certain classes. This approach establishes a direct relationship between the number of samples chosen per class through active learning and the model's confidence in those samples belonging to the respective class. We showcase this relationship for the CUBS-200 dataset in Figure 5. Consequently, an overemphasis on these particular classes can occur, as they are repeatedly selected for semi-supervised learning, essentially reinforcing the existing knowledge of the model. Unfortunately, this leads to the loss of information pertaining to other classes, consequently resulting in a decline in the overall accuracy of the model.

6. Conclusion

We explore the usage of an ensemble of thief models for the task of model extraction in the hard-label setting as a solution to the problems of noisy sample selection and the overfitting of thief models. Our method, called Army of Thieves, performs better than previous methods in terms of extracted accuracy and query efficiency. Even though we focus only on Image Classification in this work, our ensemble approach is generic enough to be applied to any domain like text, speech etc., given the presence of appropriate active and semi-supervised learning methods. The selection of ensemble members can be streamlined by an effective metric that quantifies the information-capturing ability of the architecture for the downstream task. We leave this task to future work and invite other researchers to work in this direction.

7. Limitations and Discussions

The foremost consideration when forming an ensemble of thief models is the selection of appropriate models. In our study, we opted for architectures with varying sizes and complexities. However, due to computational resource constraints, we could not include larger models such as Vision Transformer (ViT) and InceptionNet in our ensemble. Furthermore, the maximum ensemble size was limited to five members. It is worth noting that in the context of replicating the victim model, the size or efficiency of the models does not significantly impact the outcome. While larger models may possess the ability to capture more information and exhibit improved generalization, they do not serve a purpose from an adversarial perspective. Ultimately, the primary objective of replicating the victim model is not effectively achieved by utilising larger models.

Prior studies in model extraction have incorporated the k-centre or core-set method alongside their proposed techniques to enhance active learning on extensive datasets. Our algorithm surpasses previous methodologies that employ the k-centre method on CIFAR-10, CIFAR-100 and Caltech-256 datasets. Upon incorporation of k-centre algorithm into our pipeline, our approach outperforms all previous SOTA approaches on the CUBS-200 dataset as well. We postulate that the performance increase of 6.5% stems from the unique characteristics of the CUBS-200 dataset, where distinctions between classes are subtle, necessitating fine-grained classification. In the end, we would still argue that employing an ensemble approach generally yields superior results for model extraction.

Acknowledgements

The authors acknowledge the support of Infosys Centre for AI (CAI) at IIIT-Delhi and iHub-Anubhuti-IIITD Foundation set up under the NM-ICPS scheme of the DST.

References

[1] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1309–1326, 2020. 3

[2] 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. 3

[3] 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. 4

[4] 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. 3, 5

[5] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 5

[6] 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. 5

[7] Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. Model extraction and adversarial transferability, your bert is vulnerable! *arXiv preprint arXiv:2103.10013*, 2021. 3

[8] Sanghyun Hong, Michael Davinroy, Yiğitcan Kaya, Dana Dachman-Soled, and Tudor Dumitraş. How to 0wn nas in your spare time. *arXiv preprint arXiv:2002.06776*, 2020. 3

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[10] 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. 2, 3

[11] Brett Koonce and Brett Koonce. Mobilenetv3. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pages 125–144, 2021. 5

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 5

[14] Z. Ma, X. Liu, Y. Liu, X. Liu, Z. Qin, and K. Ren. Divtheft: An ensemble model stealing attack by divide-and-conquer. *IEEE Transactions on Dependable and Secure Computing*, (01):1–13, jan 5555. 3

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 7

[16] Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144. Springer, 2019. 2

[17] 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. 2, 3

[18] 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. 1, 2, 3, 6

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

[20] 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. 3

[21] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 4

[22] Xuxiang Sun, Gong Cheng, Hongda Li, Lei Pei, and Junwei Han. Exploring effective data for surrogate training towards black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15355–15364, 2022. 3

[23] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[24] 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. 1, 2

[25] 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. 3

[26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[27] 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. 1, 2, 3, 6, 7

[28] Qilong Zhang, Xiaodan Li, Yuefeng Chen, Jingkuan Song, Lianli Gao, Yuan He, and Hui Xue. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains, 2022. 7

[29] 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. 3