Knockoff Nets: Stealing Functionality of Black-Box Models

Tribhuvanesh Orekondy[1] Bernt Schiele[1] Mario Fritz[2]

[1] Max Planck Institute for Informatics [2] CISPA Helmholtz Center for Information Security
Saarland Informatics Campus, Germany

1. Introduction

Developing and engineering ML models for commercial use (e.g., image recognition APIs) is a product of intense time, money, and human effort – ranging from collecting a massive annotated dataset to tuning the right model for the task. Once developed, they are deployed (e.g., on devices or internet) to function as blackboxes: input in, predictions out. In this work, we ask: can one create a "knock-off" of deployed ML models based solely on blackbox access? We work towards purely stealing *functionality* of complex blackbox models by making minimal assumptions.

We formulate model functionality stealing as a two-step approach: (i) querying a set of input images using a strategy $\mathbb{P}_\pi$ to the blackbox victim model to obtain predictions; and (ii) training a "knockoff" with queried image-prediction pairs. What makes it particularly challenging in our scenario is that the adversary does not have access nor knowledge of the training image data distribution used to train the victim model. The adversary's goal is for the knockoff (created using as few queries as possible) to compete with the victim model at the victim's task.

We make multiple remarkable observations: (a) querying random images from a different distribution than that of the blackbox training data results in a well-performing knockoff; (b) this is possible even when the knockoff is represented using a different architecture; and (c) our RL approach additionally improves query sample efficiency in certain settings and provides performance gains. We validate model functionality stealing on a range of datasets and tasks, as well as show that a reasonable knockoff of an image analysis API could be created for as little as \$30.

2. Learning to Knockoff

We now present the problem (§2.1) and our approach (§2.2) to perform model functionality stealing.

2.1. Problem Statement

We set up the task as shown in Figure 1. Given blackbox access to a "victim" model $F_V : \mathcal{X} \to \mathcal{Y}$, the (adversary's) goal is to replicate its functionality using "knockoff" model F_A of the adversary. We evaluate the knockoff performance on the victim's held-out test set $\mathcal{D}_V^{\text{test}}$.

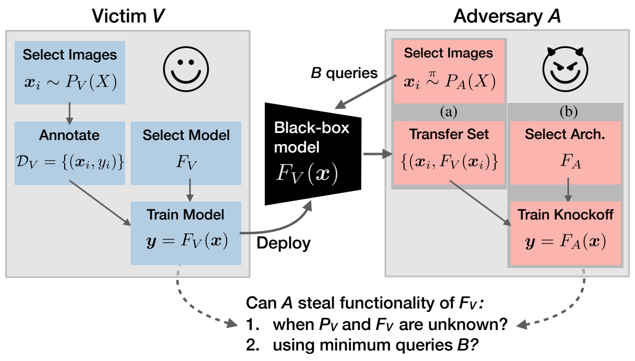
Victim V
Select Images $x_i \sim P_V(X)$
Annotate $\mathcal{D}_V = \{(x_i, y_i)\}$
Select Model F_V
Train Model $y = F_V(x)$
Deploy
B queries
Black-box model $F_V(x)$
Adversary A
Select Images $x_i \overset{\pi}{\sim} P_A(X)$
(a)
Transfer Set $\{(x_i, F_V(x_i))\}$
(b)
Select Arch. F_A
Train Knockoff $y = F_A(x)$
Can A steal functionality of F_V:
1. when P_V and F_V are unknown?
2. using minimum queries B?

Figure 1: Problem Statement. Laying out the task of model functionality stealing in the view of two players - victim V and adversary A. We group adversary's moves into (a) Transfer Set Construction (b) Training Knockoff F_A.

Assumptions We assume a CNN blackbox $\boldsymbol{y} = F_V(\boldsymbol{x})$, which given *any* image $\boldsymbol{x} \in \mathcal{X}$ returns a K-dim posterior probability vector $\boldsymbol{y} \in [0,1]^K$, $\sum_k y_k = 1$. We also consider relaxations, such as by truncating $\boldsymbol{y}$. We assume remaining aspects to be unknown to the adversary: (i) the internals of F_V e.g., hyperparameters or architecture; (ii) the data $\mathcal{D}_V = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}, \boldsymbol{x}_i \sim P_V(X)$ used to train and evaluate the model; and (iii) semantics over the K classes.

Threat Model To train the knockoff model, the adversary: (i) interactively queries images $\{\boldsymbol{x}_i \overset{\pi}{\sim} P_A(X)\}$ using strategy π to obtain a "transfer set" of images and pseudo-labels $\{(\boldsymbol{x}_i, F_V(\boldsymbol{x}_i))\}_{i=1}^B$; and (ii) selects an architecture F_A for the knockoff and trains it to mimic the behaviour of F_V on the transfer set.

2.2. Approach

We briefly introduce our approach here and refer the reader to the main paper [1] for more details.

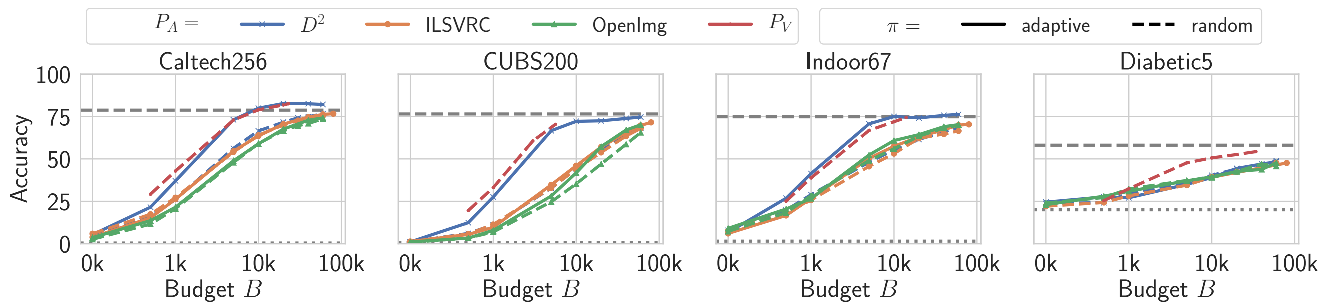

Figure 2: Performance of the knockoff at various budgets. Across choices of adversary's image distribution (P_A) and sampling strategy π. -- represents accuracy of blackbox F_V and ····· represents chance-level performance.

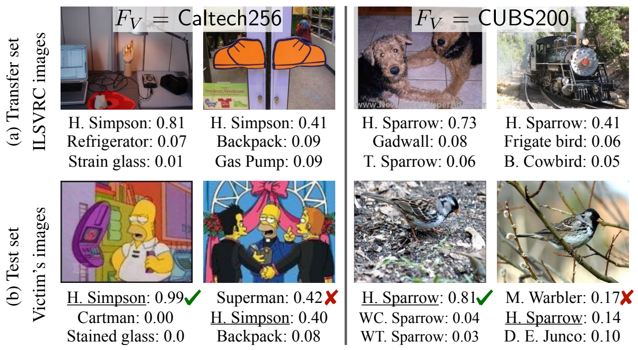

Figure 3: Qualitative Results. (a) Samples from the transfer set ($\{(\boldsymbol{x}_i, F_V(\boldsymbol{x}_i))\}, \boldsymbol{x}_i \sim P_A(X)$) displayed for two output classes: 'Homer Simpson' and 'Harris Sparrow'. (b) With the knockoff F_A trained on the transfer set, we visualize its predictions on victim's test set ($\{(\boldsymbol{x}_i, F_A(\boldsymbol{x}_i))\}, \boldsymbol{x}_i \sim \mathcal{D}_V^{\text{test}}$). Ground truth labels are underlined. Objects from these classes, among numerous others, were never encountered while training F_A.

Selecting $P_A(X)$. The adversary first selects an image distribution to sample images. We consider this to be a large discrete set of images. For instance, one of the distributions P_A we consider is 1.2M images of ImageNet.

Transfer Set Construction. We consider two strategies π to construct the transfer set of blackbox input-output pairs $\{(\boldsymbol{x}_i, F_V(\boldsymbol{x}_i))\}_{i=1}^{B}$, where $\{\boldsymbol{x}_i \overset{\pi}{\sim} P_A(X)\}$: (i) `random`: at each time-step, the adversary queries an image drawn i.i.d from $P_A(X)$; and (ii) `adaptive`: we incorporate a feedback signal as a result of querying an image at each time-step. This signal is used to learn a policy π in order to improve sample-efficiency of queries.

Training Knockoff F_A. Given the transfer set, we now train a knockoff F_A to imitate the image-prediction pairs. But, how do we model F_A? While some works argue to choose the architecture and hyperparameters by reverse-engineering blackbox, we find it orthogonal to our requirement of simply stealing the functionality. Instead, we model F_A with a reasonably complex architecture e.g., VGG or ResNet. Existing findings in knowledge distillation and model compression indicate robustness to choice of reasonably complex (student) models. We investigate the choice under weaker knowledge of the teacher (F_V) e.g., training data and architecture is unknown.

3. Teaser Results

We evaluate model functionality stealing of four blackbox models: Caltech256 (classification of 256 general object categories), CUB-200 (200 bird species), Indoor Scenes (67 indoor scenes), and Diabetic Retinopathy (5 diabetic retinopathy scales). Within this setup, we highlight some remarkable observations from the main paper: (i) *effective model stealing of complex models*: As can be seen in Figure 2, our stealing attacks (colored lines) in all cases achieve over 0.82× test accuracy of the victim model (gray dashed line). This is in particular remarkable, given that the knockoff model has *never* encountered images of numerous classes that appear at test-time e.g., >90% of the bird classes in `CUBS200`. As an example, consider the classification of class 'Homer Simpson' in Caltech256. The knockoff achieves 92% test-time accuracy for this class (test-time predictions in Fig. 3b), albeit learning only from non-'Homer Simpson' images (most confident transfer set examples in Fig. 3a); (ii) *`adaptive` improves sample-efficiency*: e.g., while `random` reaches 68.3% at B=60k, `adaptive` achieves this 6× quicker at B=10k. However, generally, we surprisingly find `random` to be highly competitive baseline; (iii) *stealing transfers across architectures:* for any reasonably complex choice of F_A (e.g., Resnet-34), we consistently achieved over 0.95× accuracy for various choices of F_V; and (iv) *stealing is possible in spite of simple defenses*: we observed 0.76× knockoff accuracy on Caltech256 in spite of blackbox returning argmax predictions.

Our results indicate an inexpensive knockoff can be trained which exhibits strong performance, using only a reasonable number of victim API queries. Thereby, circumventing monetary and labour costs of collecting images, obtaining expert annotations, and tuning a model.

References

[1] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *CVPR*, 2019. 1