Knockoff Nets: Stealing Functionality of Black-Box Models (Supplementary material)

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

A. Contents

The supplementary material contains:

B. Extended Descriptions

In this section, we provide additional detailed descriptions and implementation details.

B.1. Black-box Models

We supplement Section 5.1 by providing extended descriptions of the blackboxes listed in Table 1 of the main paper. Each blackbox F_V is trained on one particular image classification dataset.

P_A	P_V			
	`Caltech256` (K=256)	`CUBS200` (K=200)	`Indoor67` (K=67)	`Diabetic5` (K=5)
ILSVRC (Z=1000)	108 (42%)	2 (1%)	10 (15%)	0 (0%)
OpenImages (Z=601)	114 (44%)	1 (0.5%)	4 (6%)	0 (0%)

Table S1: Overlap between P_A and P_V.

Black-box 1: `Caltech256` [5]. Caltech-256 is a popular dataset for general object recognition gathered by downloading relevant examples from Google Images and manually screening for quality and errors. The dataset contains 30k images covering 256 common object categories.

Black-box 2: `CUBS200` [14]. A fine-grained bird-classifier is trained on the CUBS-200-2011 dataset. This dataset contains roughly 30 train and 30 test images for each of 200 species of birds. Due to the low intra-class variance, collecting and annotating images is challenging even for expert bird-watchers.

Black-box 3: `Indoor67` [11]. We introduce another fine-grained task of recognizing 67 types of indoor scenes. This dataset consists of 15.6k images collected from Google Images, Flickr, and LabelMe.

Black-box 4: `Diabetic5` [1]. Diabetic Retinopathy (DR) is a medical eye condition characterized by retinal damage due to diabetes. Cases are typically determined by trained clinicians who look for presence of lesions and vascular abnormalities in digital color photographs of the retina captured using specialized cameras. Recently, a dataset of such 35k retinal image scans was made available as a part of a Kaggle competition [1]. Each image is annotated by a clinician on a scale of 0 (no DR) to 4 (proliferative DR). This highly-specialized biomedical dataset also presents challenges in the form of extreme imbalance (largest class contains 30× as the smallest one).

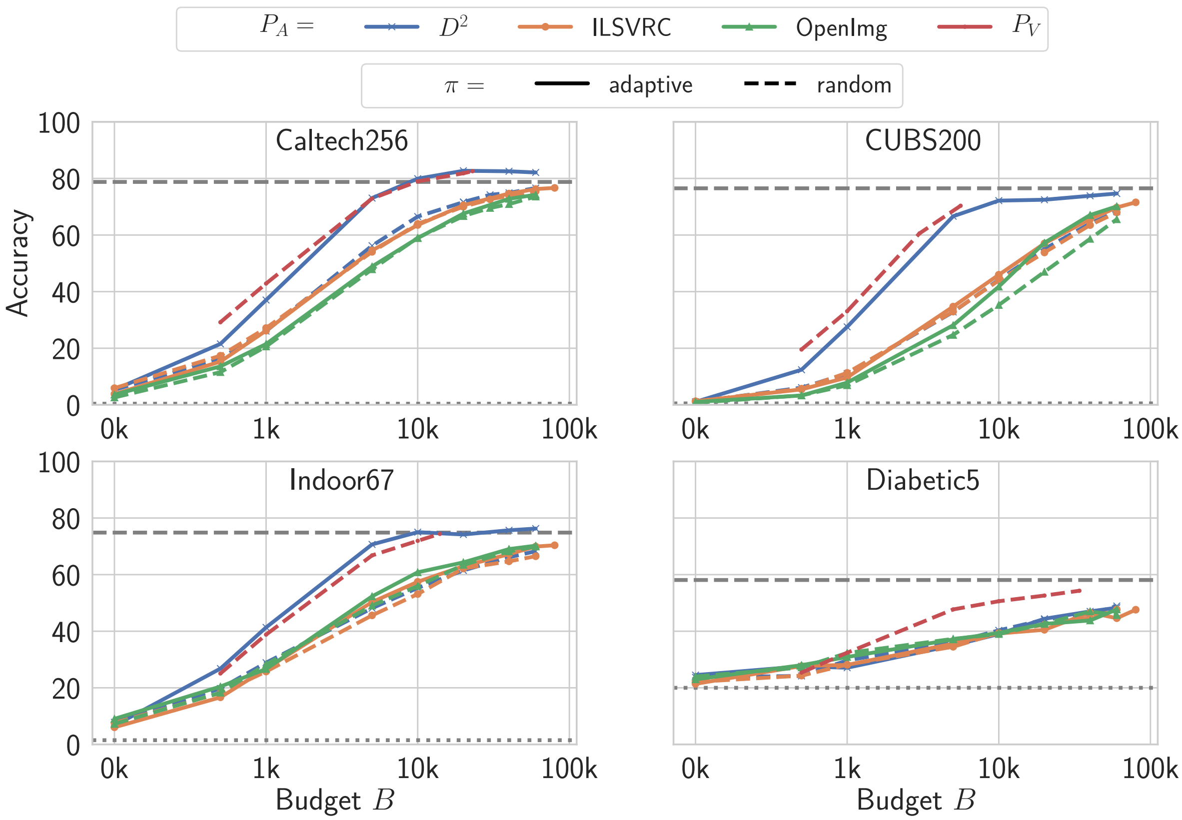


Figure S1: Performance of the knockoff at various budgets. (Enlarged version of Figure 5) Presented for various choices of adversary's image distribution (P_A) and sampling strategy π. - represents accuracy of blackbox F_V and ···· represents chance-level performance.

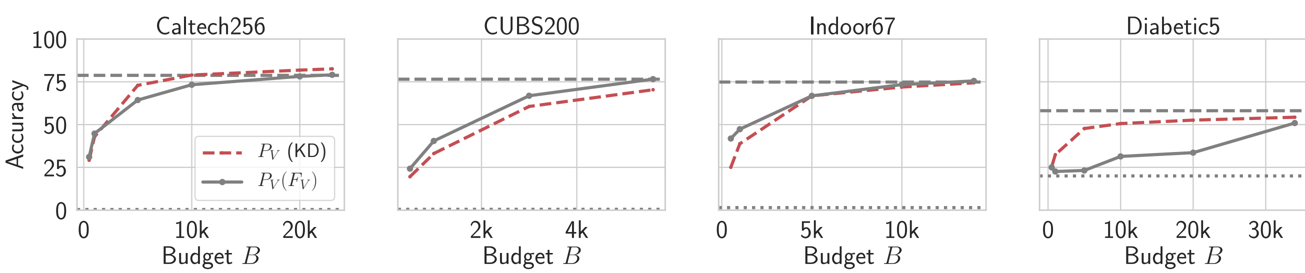


Figure S2: Training on GT vs. KD. Extension of Figure 5. We compare sample efficiency of first two rows in Table 2: "$P_V(F_V)$" (training with GT data) and "P_V (KD)" (training with soft-labels of GT images produced by F_V)

B.2. Overlap: Open-world

In this section, we supplement Section 5.2.1 in the main paper by providing more details on how overlap was calculated in the open-world scenarios. We manually compute overlap between labels of the blackbox (K, e.g., 256 Caltech classes) and the adversary's dataset (Z, e.g., 1k ILSVRC classes) as: $100 \times |K \cap Z|/|K|$. We denote two labels $k \in K$ and $z \in Z$ to overlap if: (a) they have the same semantic meaning; or (b) z is a type of k e.g., z = "maltese dog" and k = "dog". The exact numbers are provided in Table S1. We remark that this is a soft-lower bound. For instance, while ILSVRC contains "Hummingbird" and CUBS-200-2011 contains three distinct species of hummingbirds, this is not counted towards the overlap as the adversary lacks annotated data necessary to discriminate among the three species.

B.3. Dataset Aggregation

All datasets used in the paper (expect OpenImages) have been used in the form made publicly available by the authors. We use a subset of OpenImages due to storage con-

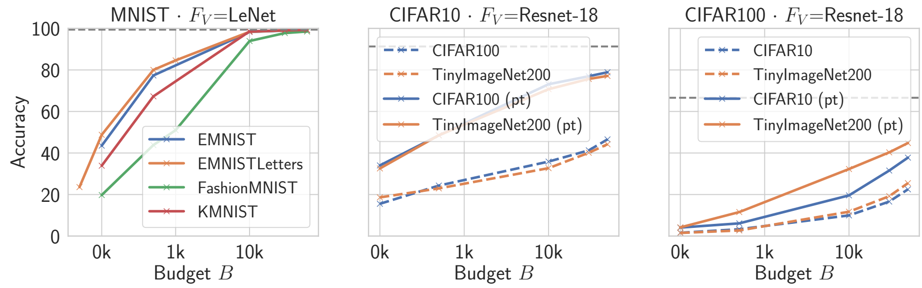


Figure S3: Training with non-ImageNet initializations of knockoff models. Shown for various choices of blackboxes F_V (subplots) and adversary's image distribution P_A (lines). All victim blackbox models are trained from scratch; test accuracy indicated by --- . All knockoff models are either trained from scratch, or pretrained on the corresponding P_A task (suffixed with '(pt)').

straints imposed by its massive size (9M images). The description to obtain these subsets are provided below.

OpenImages. We retrieve 2k images for each of the 600 OpenImages [8] "boxable" categories, resulting in 554k unique images. ∼19k images are removed for either being corrupt or representing Flickr's placeholder for unavailable images. This results in a total of 535k unique images.

OpenImages-Faces. We download all images (422k) from OpenImages [8] with label "`/m/0dzct: Human face`" using the OID tool [13]. The bounding box annotations are used to crop faces (plus a margin of 25%) containing at least 180×180 pixels. We restrict to at most 5 faces per image to maintain diversity between train/test splits. This results in a total of 98k faces images.

B.4. Additional Implementation Details

In this section, we provide implementation details to supplement discussions in the main paper.

Input Transformations. While training the blackbox models F_V we augment training data by applying input transformations: random 224×224 crops and horizontal flips. This is followed by performing normalizing the image using standard Imagenet mean and standard deviation values. While training the knockoff model F_A and for evaluation, we resize the image to 256×256, obtain a 224×224 center crop and normalize as before.

Training F_V = `Diabetic5`. We train this model using a learning rate of 0.01 (while this is 0.1 for the other models) and a weighted loss. Due to the extreme imbalance between classes of the dataset, we weigh each class as follows. Let n_k denote the number of images belonging to class k and let $n_{\min} = \min_k n_k$. We weigh the loss for each class k as $n_{\min}/n_k$. From our experiments with weighted loss, we found approximately 8% absolute improvement in overall accuracy on the test set. However, the training of knockoffs of all blackboxes are identical in all aspects, including a non-weighted loss irrespective of the victim blackbox targeted.

Creating ILSVRC Hierarchy. We represent the 1k labels of ILSVRC as a hierarchy (Figure 4b) in the form: root node "entity" → N coarse nodes → 1k leaf nodes. We obtain N (30 in our case) coarse labels as follows: (i) a 2048-d mean feature vector representation per 1k labels is obtained using an Imagenet-pretrained ResNet ; (ii) we cluster the 1k features into N clusters using scikit-learn's [10] implementation of agglomerative clustering; (iii) we obtain semantic labels per cluster (i.e., coarse node) by finding the common parent in the Imagenet semantic hierarchy.

`Adaptive` Strategy. Recall from Section 6, we train the knockoff in two phases: (a) *Online*: during transfer set construction; followed by (b) *Offline*: the model is retrained using transfer set obtained thus far. In phase (a), we train F_A with SGD (with 0.5 momentum) with a learning rate of 0.0005 and batch size of 4 (i.e., 4 images sampled at each t). In phase (b), we train the knockoff F_A from scratch on the transfer set using SGD (with 0.5 momentum) for 100 epochs with learning rate of 0.01 decayed by a factor of 0.1 every 60 epochs. We used Δ=25.

C. Extensions of Existing Results

In this section, we present extensions of existing results discussed in the main paper.

C.1. Qualitative Results

Qualitative results to supplement Figure 6 are provided in Figures S4-S7. Each row in the figures correspond to an output class of the blackbox whose images the knockoff has never encountered before. Images in the "transfer set" column were randomly sampled from ILSVRC [4, 12]. In contrast, images in the "test set" belong to the victim's test set (Caltech256, CUBS-200-2011, etc.).

C.2. Sample Efficiency: Training Knockoffs on GT

We extend Figure 5 in the main paper to include training on the same ground-truth data used to train the blackboxes.

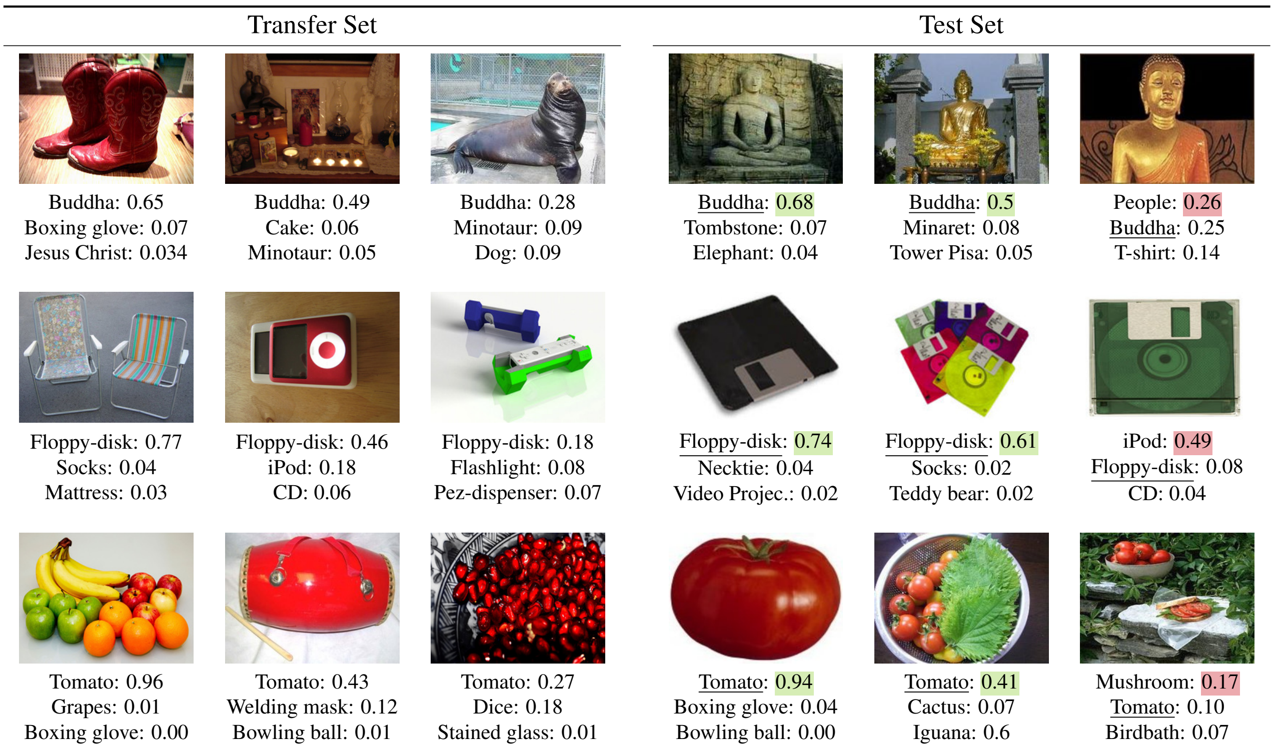


Figure S4: Qualitative results: `Caltech256`**.** Extends Figure 6 in the main paper. GT labels are underlined, correct knockoff top-1 predictions in green and incorrect in red.

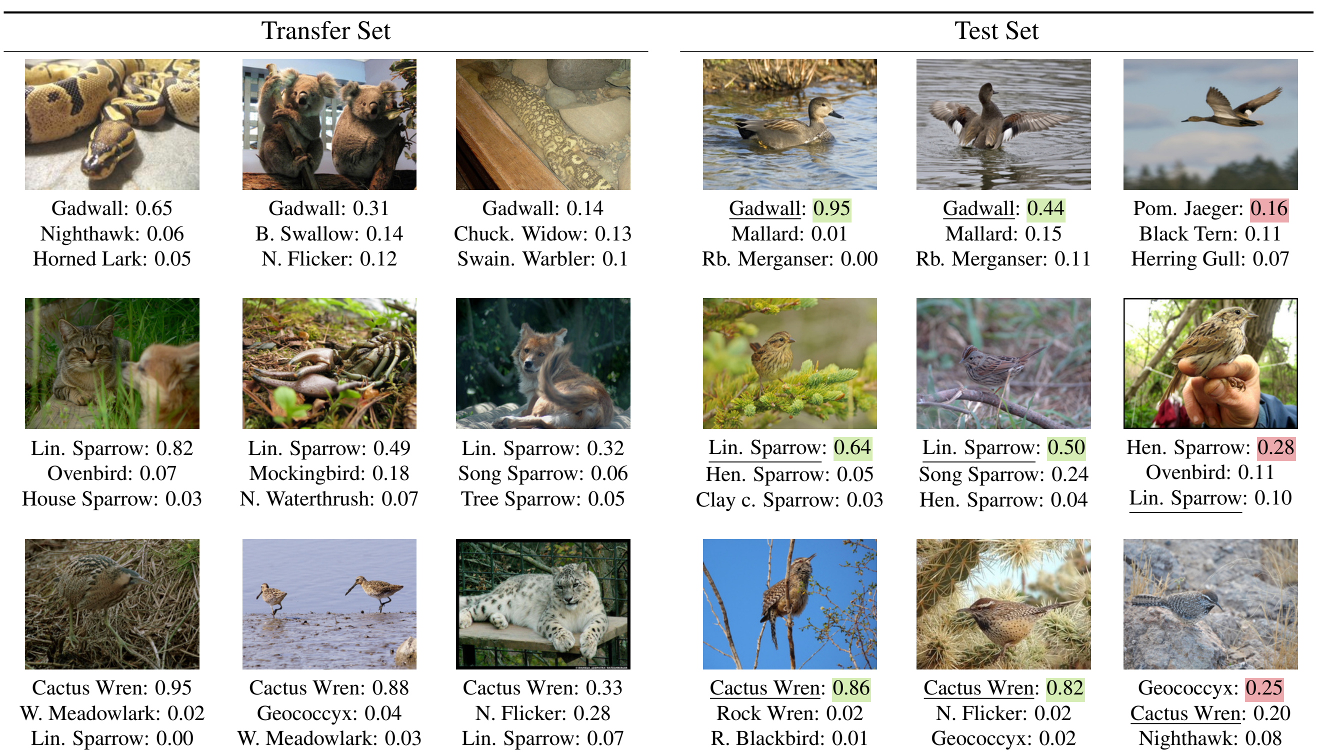


Figure S5: Qualitative results: `CUBS200`**.** Extends Figure 6 in the main paper.

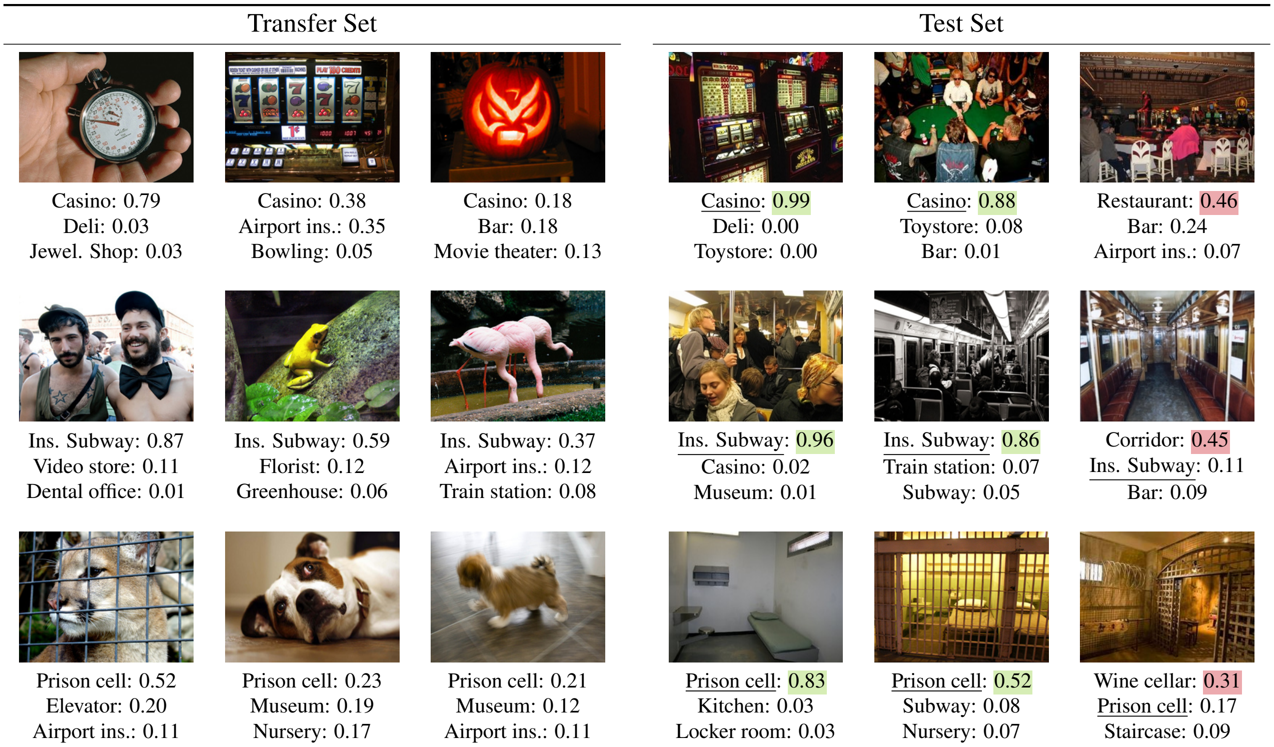


Figure S6: Qualitative results: `Indoor67`. Extends Figure 6 in the main paper. GT labels are underlined, correct top-1 knockoff predictions in green and incorrect in red.

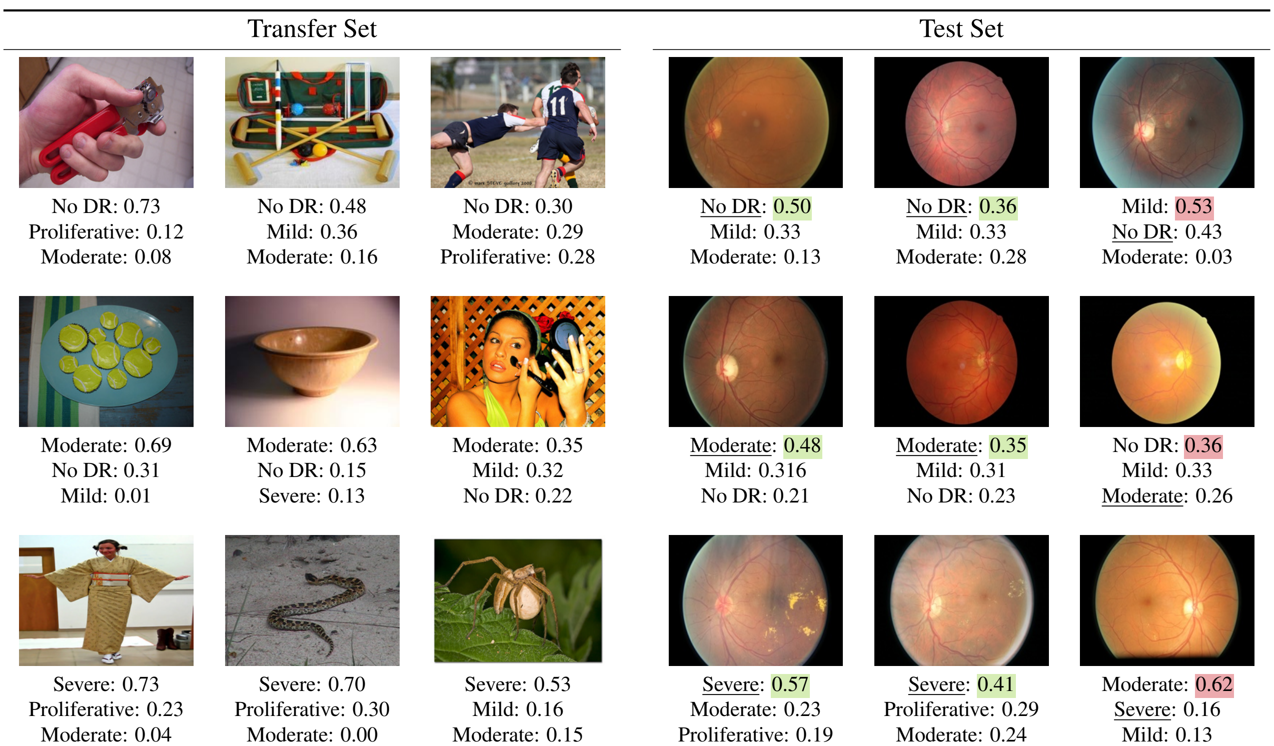
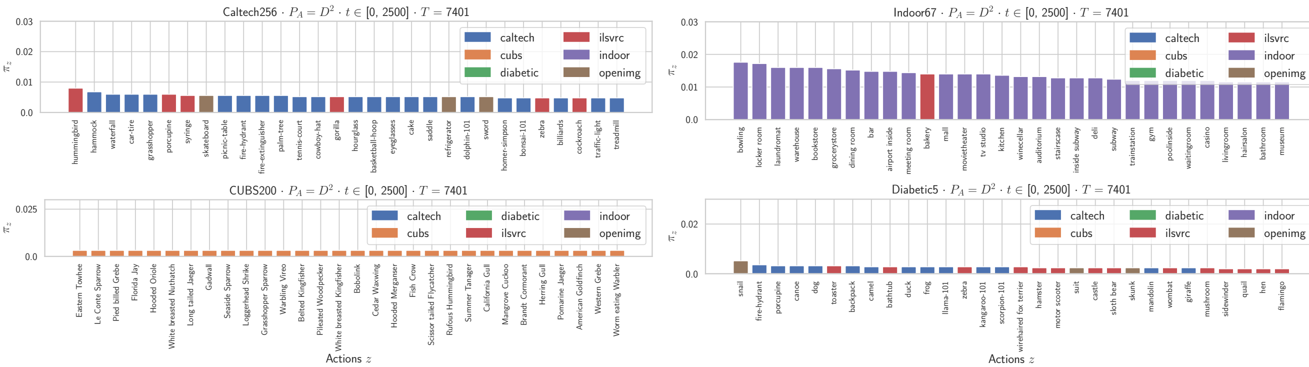
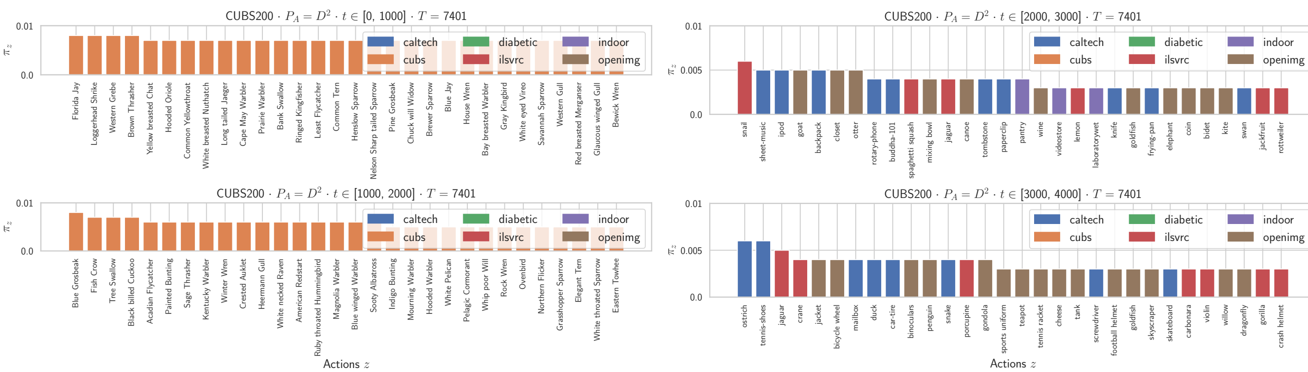


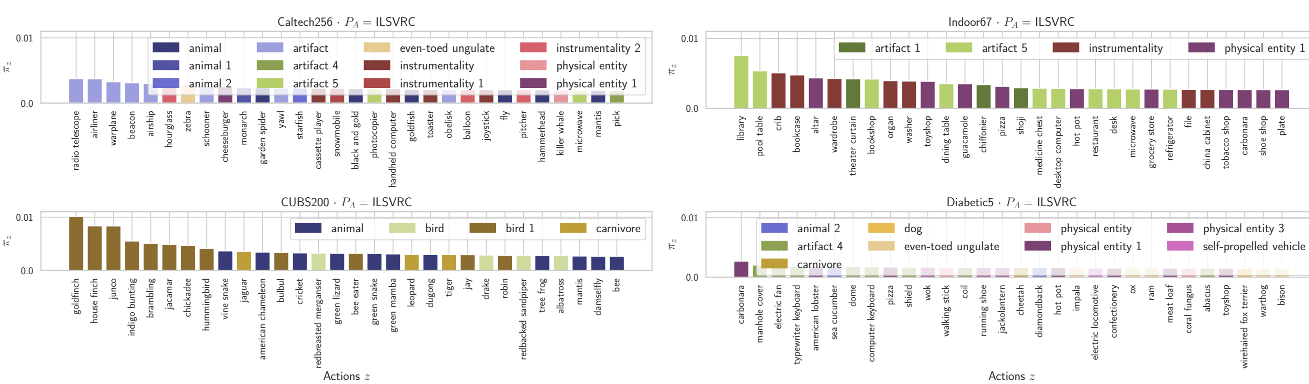
Figure S7: Qualitative results: `Diabetic5`. Extends Figure 6 in the main paper.



(a) Closed world.



(b) Closed world. Analyzing policy over time t for `CUBS200`.



(c) Open world.

Figure S8: Policies learnt by `adaptive` strategy. Supplements Figure 7 in the main paper.

This extension "$P_V(F_V)$" is illustrated in Figure S2, displayed alongside KD approach. The figure represents the sample-efficiency of the first two rows of Table 2. Here we observe: (i) comparable performance in all but one case (`Diabetic5`, discussed shortly) indicating KD is an effective approach to train knockoffs; (ii) we find KD achieve better performance in `Caltech256` and `Diabetic5` due to regularizing effect of training on soft-labels [6] on an imbalanced dataset.

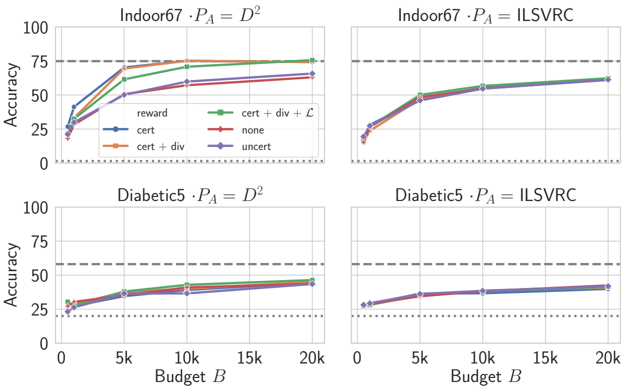


Figure S9: Reward Ablation. Supplements Figure 8 in the main paper.

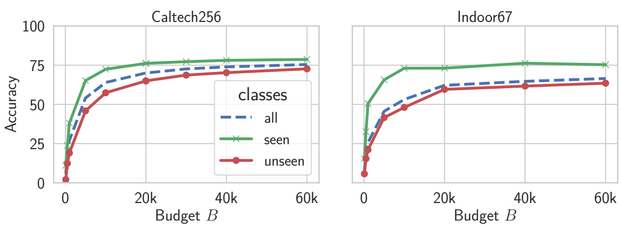


Figure S10: Per class evaluation. Per-class evaluation split into seen and unseen classes.

C.3. Policies learnt by `Adaptive`

We inspected the policy π learnt by the `adaptive` strategy in Section 6.1. In this section, we provide policies over all blackboxes in the closed- and open-world setting. Figures S8a and S8c display probabilities of each action $z \in Z$ at $t = 2500$.

Since the distribution of rewards is non-stationary, we visualize the policy over time in Figure S8b for `CUBS200` in a closed-world setup. From this figure, we observe an evolution where: (i) at early stages ($t \in [0, 2000]$), the approach samples (without replacement) images that overlaps with the victim's train data; and (ii) at later stages ($t \in [2000, 4000]$), since the overlapping images have been exhausted, the approach explores related images from other datasets e.g., "ostrich", "jaguar".

C.4. Reward Ablation

The reward ablation experiment (Figure 8 in the main paper) for the remaining datasets are provided in Figure S9. We make similar observations as before for `Indoor67`. However, since F_V = `Diabetic5` demonstrates confident predictions in all images, we find little-to-no improvement for knockoffs of this victim model.

D. Auxiliary Experiments

In this section, we present experiments to supplement existing results in the main paper.

D.1. Effect of CNN Initialization

In our experiments (Section 6), the victim and knockoff models are initialized with ImageNet pretrained weights[1], a de facto when training CNNs with a limited amount of data. In this section, we study influence of different initializations of the victim and adversary models.

To achieve reasonable performance in our limited data setting, we perform the following experiments on comparatively smaller models and datasets. We choose three victim blackboxes (all trained after random initialization) using the following datasets: MNIST [9], CIFAR10 [7], and CIFAR100 [7]. We train a LeNet-like model[2] for MNIST, and Resnet-18 models for CIFAR-10 and CIFAR-100.

While we use the same blackbox model architecture for the knockoff, we either randomly initialize them or pretrain them on a different task. Consequently, in the following experiments, both the victim and knockoff have different initializations. We repeat our experiment using random policy (Section 4.1.1) and using as the query set P_A: (a) when P_V=MNIST: EMNIST [3] (superset of MNIST containing alpha numeric characters [A-Z, a-z, 0-9]), EMNISTLetters ([A-Z, a-z]), FashionMNIST [15] (fashion items spanning 10 classes e.g., trouser, coat) and KMNIST [2] (Japanese Hiragana characters spanning 10 classes); (b) when P_V=CIFAR10: CIFAR100 [7] and TinyImageNet200[3] (subset of ImageNet with 500 images per each of 200 classes); and (c)when P_V=CIFAR100: CIFAR10 and TinyImageNet200. Note that the output classes between CIFAR10 and CIFAR100 are disjoint.

From Figure S3, we observe: (i) model stealing is possible even when the knockoffs are randomly initialized. For instance, when stealing MNIST, we recover 0.98$\times$ victim accuracy across all choices of P_A; (ii) pretraining the knockoff model – even on a different task – improves sample efficiency of model stealing attacks e.g., when F_V=CIFAR10-resnet18, querying images from P_V improves the knockoff accuracy after 50k queries from 46.5% to 78.9%.

D.2. Seen and Unseen classes

We now discuss evaluation to supplement Section 5.2.1 and Section 6.1.

In Section 6.1, we highlighted strong performance of the knockoff even among classes that were never encountered (see Table S1 for exact numbers) during training. To elaborate, we split the blackbox output classes into "seen" and "unseen" categories and present mean per-class accuracies in Figure S10. Although we find better performance on

[1] Alternatives for ImageNet pretrained models across a wide range of architectures were not available at the time of writing

[2] `https://github.com/pytorch/examples/blob/master/mnist/main.py`

[3] `https://tiny-imagenet.herokuapp.com/`

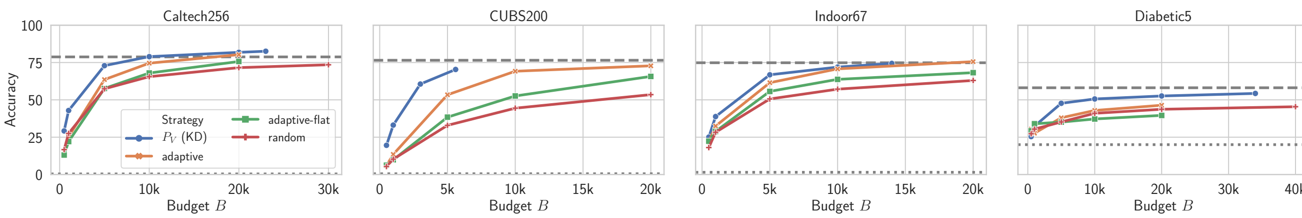


Figure S11: Hierarchy. Evaluating `adaptive` with and without hierarchy using $P_A = D^2$. -- represents accuracy of blackbox F_V and represents chance-level performance.

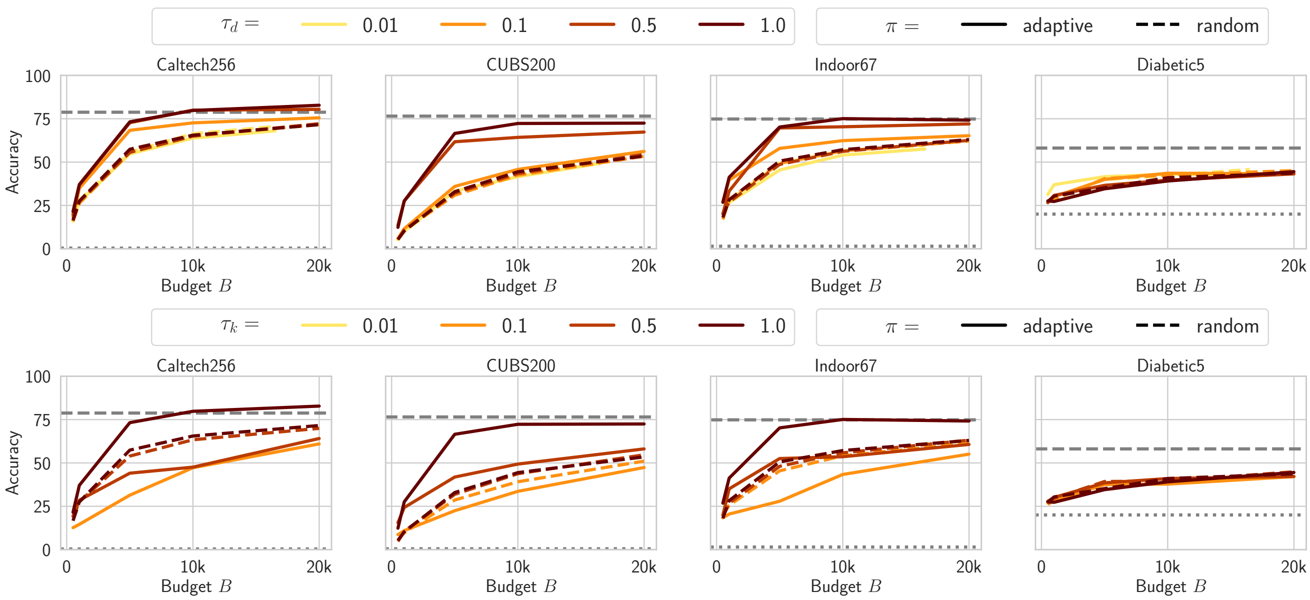


Figure S12: Semi-open world: τ_d and τ_k.

classes seen while training the knockoff, performance of unseen classes is remarkably high, with the knockoff achieving >70% performance in both cases.

D.3. `Adaptive`: With and without hierarchy

The `adaptive` strategy presented in Section 4.1.2 uses a hierarchy discussed in Section 5.2.2. As a result, we approached this as a hierarchical multi-armed bandit problem. Now, we present an alternate approach `adaptive-flat`, without the hierarchy. This is simply a multi-armed bandit problem with $|Z|$ arms (actions).

Figure S11 illustrates the performance of these approaches using $P_A = D^2$ ($|Z| = 2129$) and rewards {certainty, diversity, loss}. We observe `adaptive` consistently outperforms `adaptive-flat`. For instance, in `CUBS200`, `adaptive` is 2× more sample-efficient to reach accuracy of 50%. We find the hierarchy helps the adversary (agent) better navigate the large action space.

D.4. Semi-open World

The closed-world experiments ($P_A = D^2$) presented in Section 6.1 and discussed in Section 5.2.1 assumed access to the image universe. Thereby, the overlap between P_A and P_V was 100%. Now, we present an intermediate overlap scenario **semi-open world** by parameterizing the overlap as: (i) τ_d: The overlap between *images* P_A and P_V is $100 \times \tau_\text{d}$; and (ii) τ_k: The overlap between *labels* K and Z is $100 \times \tau_\text{k}$. In both these cases $\tau_\text{d}, \tau_\text{k} \in (0, 1]$ represents the fraction of P_A used. $\tau_\text{d} = \tau_\text{k} = 1$ depicts the closed-world scenario discussed in Section 6.1.

From Figure S12, we observe: (i) the `random` strategy is unaffected in the semi-open world scenario, displaying comparable performance for all values of τ_d and τ_k; (ii) τ_d: knockoff obtained using `adaptive` obtains strong performance even with low overlap e.g., a difference of at most 3% performance in `Caltech256` even at $\tau_\text{d} = 0.1$; (iii) τ_k: although the `adaptive` strategy is minimally affected in few cases (e.g., `CUBS200`), we find the performance drop due to a pure exploitation (certainty) that is used. We observed recovery in performance by using all rewards indicating exploration goals (diversity, loss) are necessary when transitioning to an open-world scenario.

References

[1] Eyepacs. `https://www.kaggle.com/c/diabetic-retinopathy-detection`. Accessed: 2018-11-08. 1

[2] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018. 7

[3] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017. 7

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[5] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 1

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 6

[7] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 7

[8] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 3

[9] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010. 7

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[11] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 1

[12] 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. 3

[13] Angelo Vittorio. Toolkit to download and visualize single or multiple classes from the huge open images v4 dataset. `https://github.com/EscVM/OIDv4_ToolKit`, 2018. 3

[14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1

[15] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 7