Deep Neural Network Stealing Attacks

CSIT375/975 AI and Cybersecurity

Dr Wei Zong

SCIT University of Wollongong

Disclaimer: The presentation materials come from various sources. For further information, check the references section

Outline

- Introduction
- Knowledge distillation
- Knockoff nets
- Semi-supervised learning

Introduction

- Developing ML/DL models for commercial use is expensive: intense time, money, and human effort
    - Collecting a massive annotated dataset
        - Annotating medical images require human experts.
        - Confidential data such as medical images, financial records are not publicly available.
    - Identifying the right architecture, hyperparameters, etc.
    - Huge computational costs for training a model for the task.
        - ChatGPT-3 cost around $2 million to $4 million in 2020
        - PaLM (Pathways Language Model) in 2022 took between $3 million and $12 million to train when only looking at the cost of computing.
            - PaLM is a 540 billion-parameter transformer-based large language model (LLM) developed by Google AI.

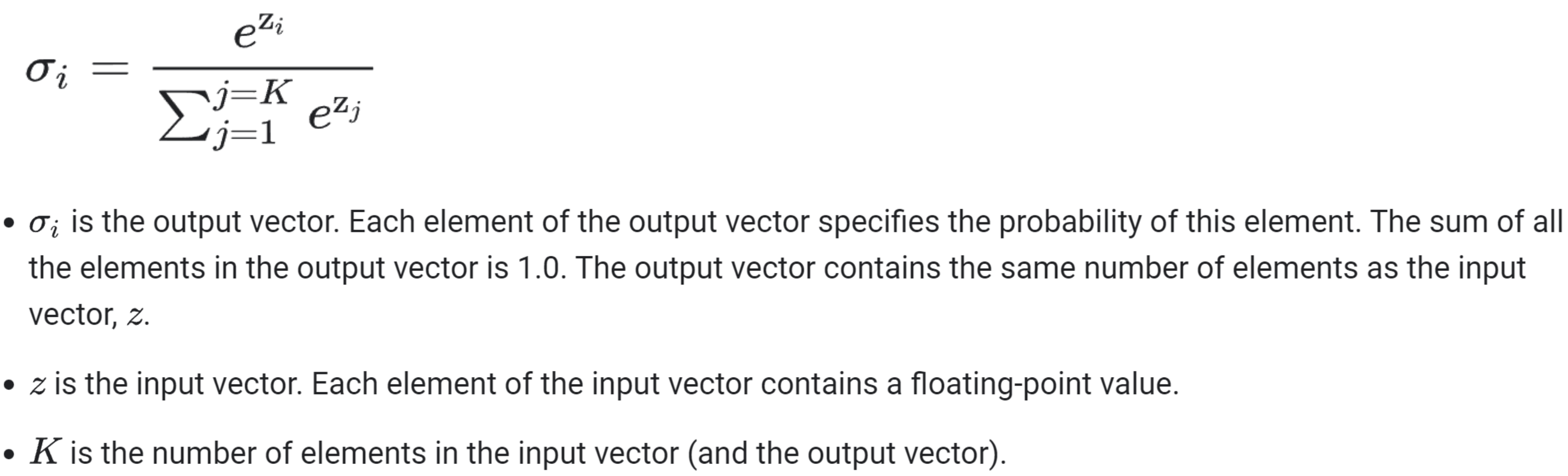
Introduction

- The details of the dataset, exact model architecture, and hyperparameters are naturally kept confidential
  - Protect the models' value.
  - To be monetized or simply serve a purpose, models are deployed function as black-boxes: input in, predictions out.

- Large-scale deployments of deep learning models in the wild
  - Can one create a copy of the black-box model solely based on **input-output pairs**?
    - This saves ("steals") huge money from developing and training models.
    - The answer is **YES**!
  - Model functionality stealing
    - Stealing functionality of complex black-box models
    - Different from "inference attacks"
      - Infer properties, e.g., training data, architecture, etc., about a black-box model.

Preliminaries: Softmax

- Softmax function: **multiclass** classification
  - Assign decimal probabilities to each class in a multi-class problem.
  - The decimal probabilities must add up to 1.0.

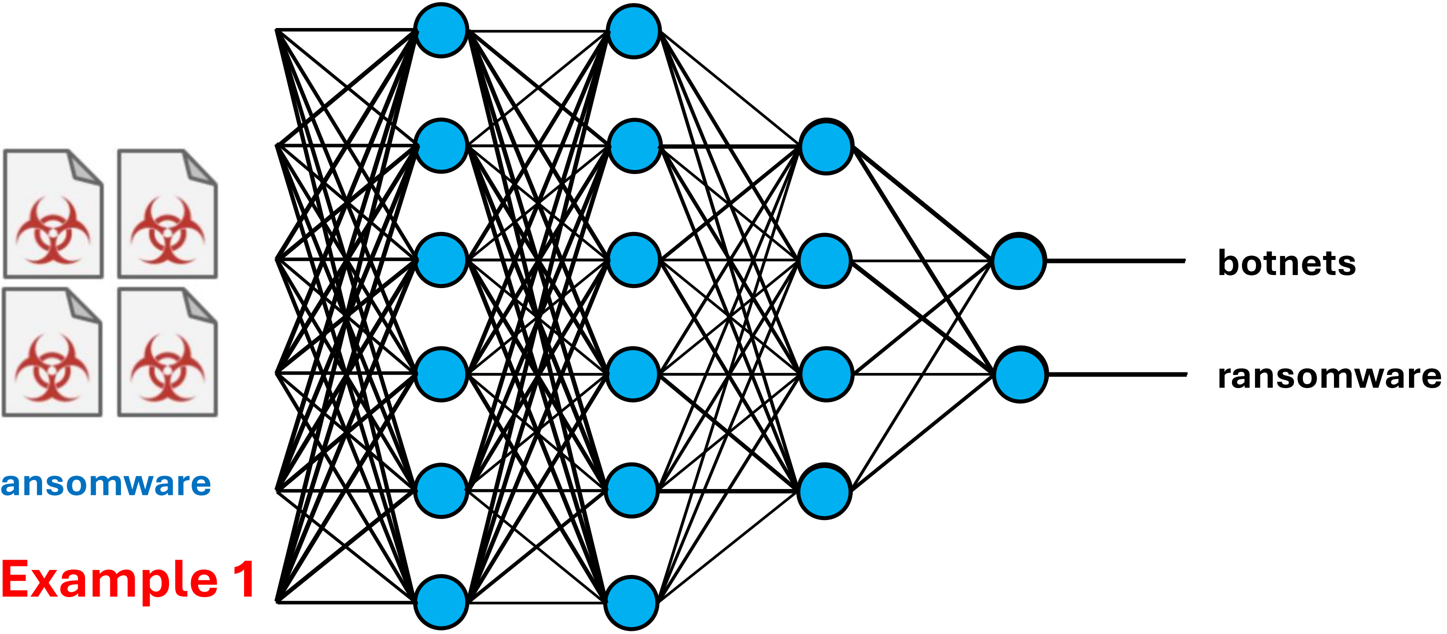


$$\sigma_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{j=K} e^{\mathbf{z}_j}}$$

- σ_i is the output vector. Each element of the output vector specifies the probability of this element. The sum of all the elements in the output vector is 1.0. The output vector contains the same number of elements as the input vector, z.
- z is the input vector. Each element of the input vector contains a floating-point value.
- K is the number of elements in the input vector (and the output vector).

Preliminaries: Softmax

$$\sigma_i = \frac{e^{z_i}}{\sum_{j=1}^{j=K} e^{z_j}}$$



Example 1

Preliminaries: Softmax

- Suppose the Input vector z is: [1.2, 2.5, 1.8]
- computes the denominator as: $\text{denominator} = e^{1.2} + e^{2.5} + e^{1.8} = 21.552$
- The softmax probability of each element is therefore:

$$\sigma_1 = \frac{e^{1.2}}{21.552} = 0.154$$

$$\sigma_2 = \frac{e^{2.5}}{21.552} = 0.565$$

$$\sigma_1 = \frac{e^{1.8}}{21.552} = 0.281$$

$$\sigma_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{j=K} e^{\mathbf{z}_j}}$$

- The output vector is: $\sigma = [0.154, 0.565, 0.281]$

Naïve Way

- Use the target model to label a dataset
    - Results highly depend on the quality of data
        - If data are the same or close to the original training data.
            - Can achieve comparable performance (maybe slightly worse).
        - If data are significantly different from the original training data.
            - Performance of the stolen model is low.
        - Impractical if data are expensive/difficult to obtain
            - medical images, financial records, etc.
    - Cannot save computational costs

Knowledge Distillation

- The original motivation of knowledge distillation (KD) is model compression.
  - Transfer knowledge of an ensemble of models into a small single model for deployment.
  - Exploited by adversaries for **model stealing**.
  - Proposed by **Hinton** et al.
    - Hinton was awarded **Nobel Prize 2024** for “foundational discoveries and inventions that enable machine learning with artificial neural networks”.
- Knowledge of a model
  - People tend to identify the knowledge in a trained model as the learned parameter values
    - This makes it hard to see how we can change the form of the model but keep the same knowledge.
  - Model knowledge is represented by the **learned mapping from input vectors to output vectors**.
    - The relative probabilities of incorrect answers tell us a lot about how a model generalizes.
      - An image of a BMW, may only have a very small chance of being mistaken for a garbage truck, but that mistake is still many times more probable than mistaking it for a carrot.

Hinton, G., 2015. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

Knowledge Distillation

- Use probabilities produced by the original model as “**soft targets**” for training the knock-off model.
  - Let the knock-off model mimic the original model.
  - We could use the original training set, or a separate “transfer” set, to produce “soft targets”.
    - The transfer set could consist entirely of **unlabeled** data.
  - This is normally done in addition to the cross-entropy loss if having access to labeled data:

$$\ell_{CE} = -\sum_{c \in \{1,2,\ldots,K\}} y_c \log(p_c)$$

- “Soft targets” are not always meaningful for simple tasks like MNIST
  - The original model almost always produces the correct answer with very high confidence
    - For example, one version of a 2 may be given a probability of 10^{-6} of being a 3 and 10^{-9} of being a 7.
  - Predictions are close to “hard targets” since these probabilities are nearly zero.
    - No obvious benefits to mimic the original model in such cases.

Hinton, G., 2015. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

Knowledge Distillation

- Distillation
  - Raise the **temperature** of the final softmax.

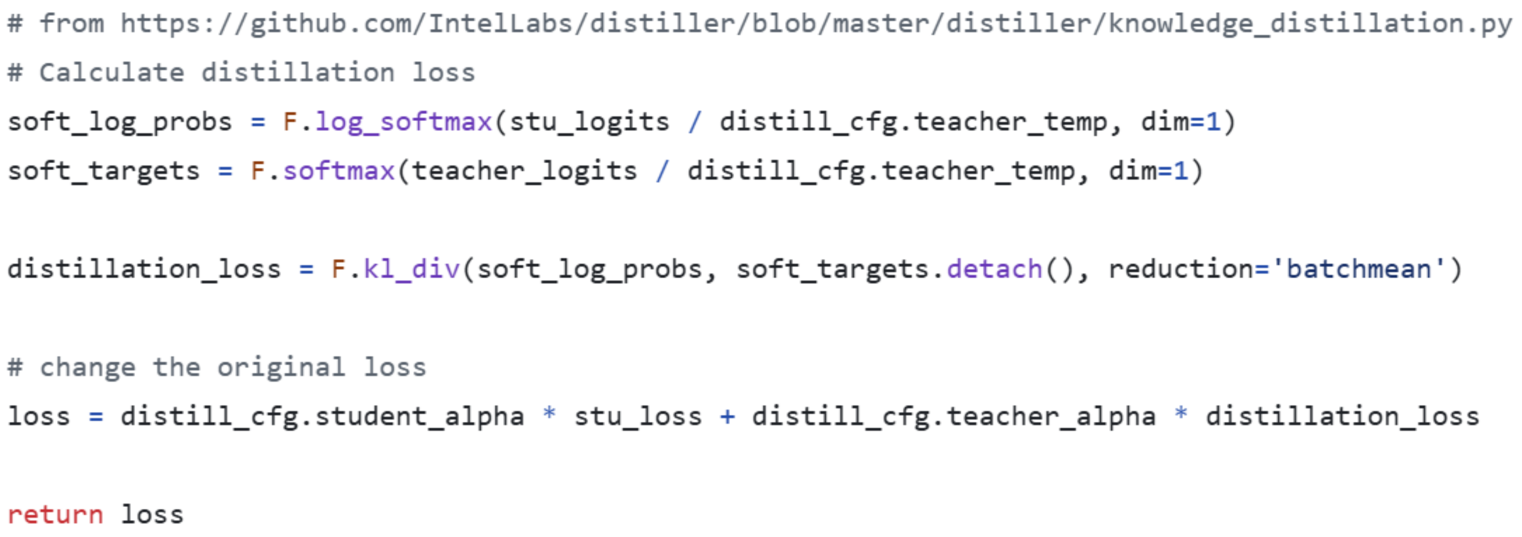
$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

  - where T is a temperature that is set to 1 for naïve softmax.
    - Using a higher value for T produces a **softer** probability distribution over classes.
      - **Why?**
      - **How about using a smaller value for T?**
  - Transferring Knowledge
    - Training the knock-off model on a transfer set using soft target distributions produced by the original model with a **high** temperature.

Hinton, G., 2015. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

Knowledge Distillation

- Python code to implement KD



```
# from https://github.com/IntelLabs/distiller/blob/master/distiller/knowledge_distillation.py
# Calculate distillation loss
soft_log_probs = F.log_softmax(stu_logits / distill_cfg.teacher_temp, dim=1)
soft_targets = F.softmax(teacher_logits / distill_cfg.teacher_temp, dim=1)

distillation_loss = F.kl_div(soft_log_probs, soft_targets.detach(), reduction='batchmean')

# change the original loss
loss = distill_cfg.student_alpha * stu_loss + distill_cfg.teacher_alpha * distillation_loss

return loss
```

- **Kullback–Leibler (KL) divergence** measures the difference between two probability distributions
    - KL = 0 -> two distributions are the same
- stu_loss = F.cross_entropy(logits, target, reduction='mean')
    - Can add this small term to encourage the knock-off model to predict the true targets as well.

Hinton, G., 2015. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.

Knockoff Net

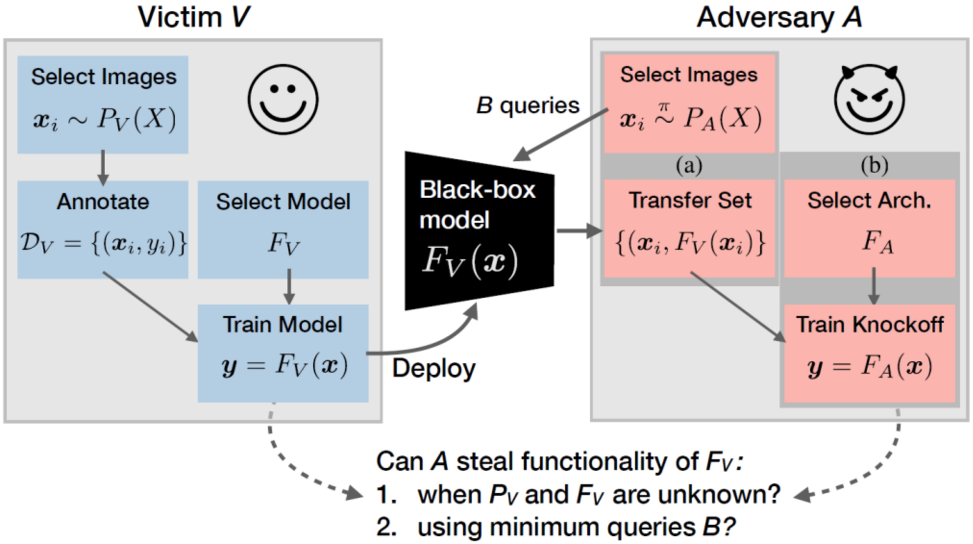
- Goal:
  - The adversary is presented with a black-box image classifier $F_v: X \rightarrow Y$
    - Given any image $x \in X$, it returns a K-dim probability vector $y \in [0,1]^K, \sum_k y_k = 1.$
  - Want to replicate its functionality using a knockoff model $F_A(X) \approx F_v(X)$

- Adversary's Unknowns:
  - The internals of F_v, e.g., hyperparameters or architecture.
  - The data used to train and evaluate the model.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net

- A two-player game between a victim V and an adversary A.
- Victim's move
  - The victim's end-goal is to deploy a trained CNN model F_v in the wild for a particular task
    - e.g., fine-grained bird classification.
  - To train this model, the victim
    - Collects task-specific images $x \sim P_V(X)$ and obtains expert annotations resulting in a training dataset.
    - Selects the model F_v that achieves best performance (accuracy) on a held-out test set.
    - The resulting model is deployed as a black-box which predicts probabilities given an image.

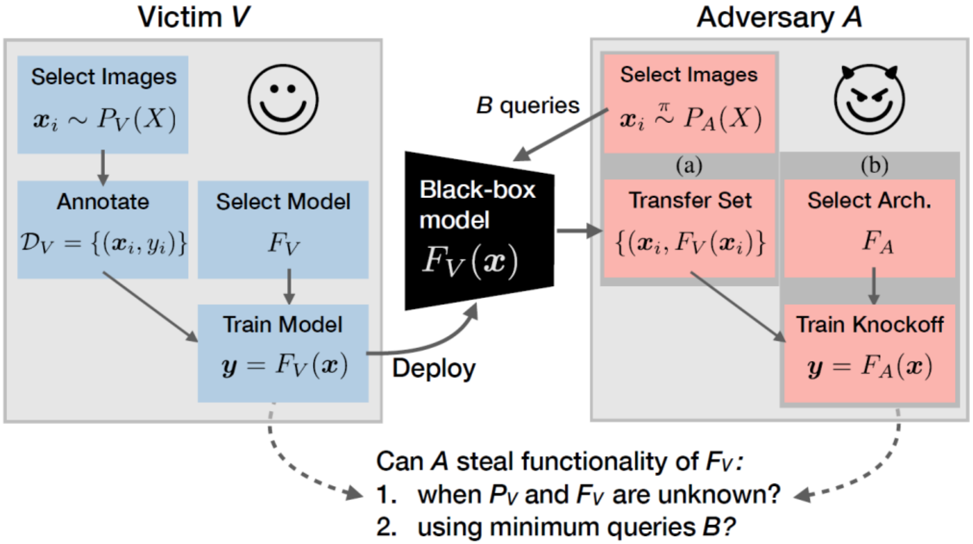

Victim V
Select Images $x_i \sim P_V(X)$
Annotate $\mathcal{D}_V = \{(x_i, y_i)\}$
Select Model F_V
Train Model $y = F_V(x)$
Deploy
Black-box model $F_V(x)$
B queries
Adversary A
Select Images $x_i \overset{\pi}{\sim} P_A(X)$
(a)
Transfer Set $\{(x_i, F_V(x_i))\}$
(b)
Select Arch. F_A
Train Knockoff $y = F_A(x)$

Can A steal functionality of F_V:
1. when P_V and F_V are unknown?
2. using minimum queries B?

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net

- Adversary’s attack to train a knockoff model
  - Interactively query images using a specific strategy to obtain a “transfer set” of images and pseudo-labels
    - Pseudo-labels are probabilities predicted by the victim model.
    - Each prediction incurs a cost, e.g., money, latency.
  - Select an architecture for the knockoff model and train it to mimic the behavior of the victim on the transfer set.



Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net

- Transfer set construction (given an image dataset P_A)
  - Random strategy
    - Randomly sample images (without replacement) to query F_v
    - There is a risk that the adversary samples images irrelevant to learning the task
      - E.g., querying dog images to a bird classifier.
  - Adaptive strategy
    - Encourage **confident predictions** by the victim
      - Hence indicating the domain F_V was trained on
    - Encourage **diversity** in images
      - To avoid images to come from a single or only a few labels
    - Encourage **different predictions** between the knockoff model and victim
      - To facilitate the training of the knockoff model.
      - F_A is being trained during the transfer set construction.
        - It will be retrained later.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

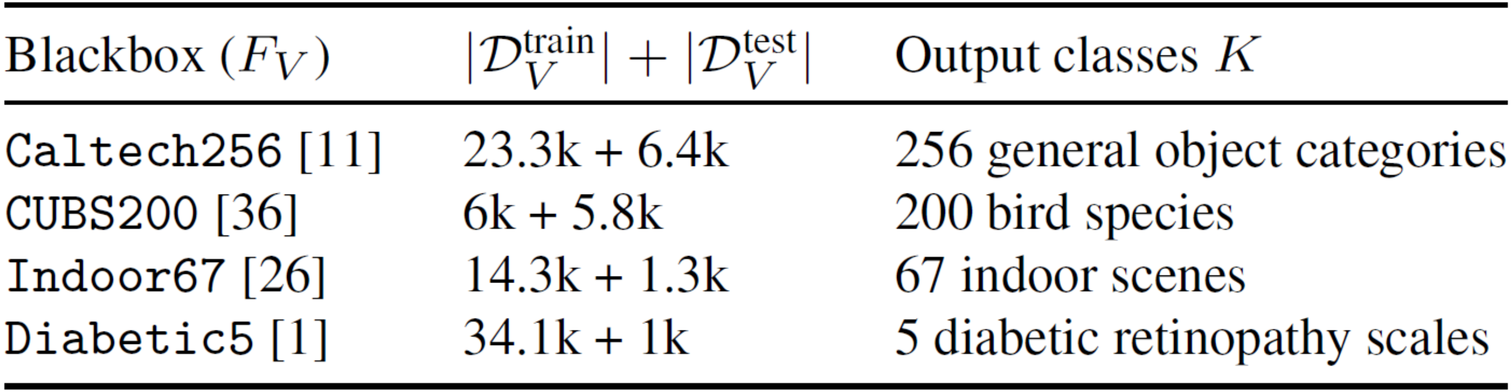
Knockoff Net

- Knowledge Distillation
  - No temperature is applied (i.e., T is set to 1) because only probabilities are returned.
  - $\ell_{CE} = -\sum_{c \in \{1,2,\ldots,K\}} p_c^V \log(p_c^A)$
    - Modified **cross-entropy** loss: weight predictions with the confidence of the victim's output instead of ground truth.
    - **Cross-entropy** and **KL divergence** are identical up to an additive constant when the target distribution is fixed.
      - Optimization results in the same solution.
- The knockoff models are trained in two phases:
  - Online: transfer set construction
    - F_A is trained during the transfer set construction.
  - Offline: the model is **retrained** using transfer set obtained thus far.
    - Each image is saved together with corresponding victim predictions.
    - F_A is pre-trained on **ImageNet**
- Selecting Architecture
  - Selecting a reasonably complex architecture e.g., VGG or ResNet.
  - Existing findings indicate robustness to choice of reasonably complex student models.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net Results

- Datasets



Blackbox (F_V)	$\lvert\mathcal{D}_V^{\text{train}}\rvert + \lvert\mathcal{D}_V^{\text{test}}\rvert$	Output classes K
`Caltech256` [11]	23.3k + 6.4k	256 general object categories
`CUBS200` [36]	6k + 5.8k	200 bird species
`Indoor67` [26]	14.3k + 1.3k	67 indoor scenes
`Diabetic5` [1]	34.1k + 1k	5 diabetic retinopathy scales

- Victim models
  - All models are trained using a **ResNet-34** architecture
    - with **ImageNet** pretrained weights

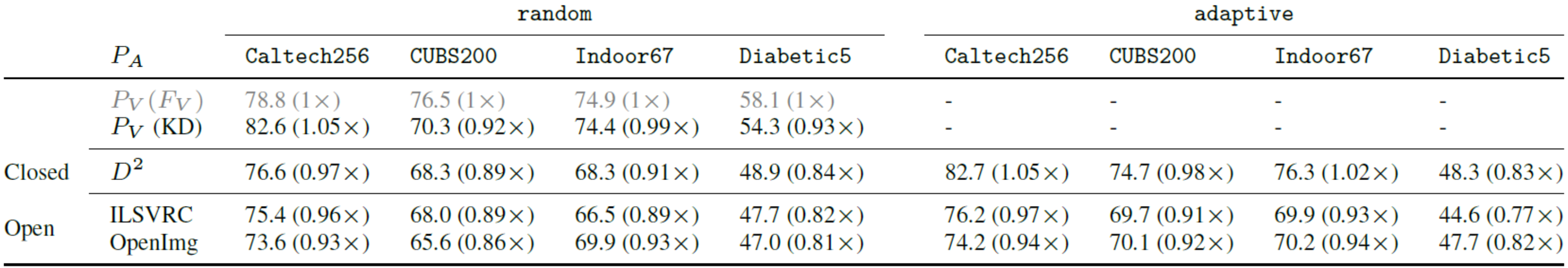
Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net Results

- Choices of P_A for transfer set construction
  - P_A = P_V
    - Images queried are identical to the ones used for training F_V .
    - There is a 100% overlap but P_A only contains **unlabeled** data.
  - Closed-world ($P_A = D^2$):
    - The adversary has access to all images in the “universe”.
      - 2.2M images and 2129 classes
    - Blackbox train data P_V is a subset of the image universe P_A.
    - There is a 100% overlap.
  - Open-world (P_A ∈ {ILSVRC−2012, OpenImages}):
    - ILSVRC-2012:1.2M images over 1000 categories
    - OpenImages: 550K images over 600 categories
    - Overlap between PV and PA is purely coincidental:
      - Caltech256 (42% ILSVRC, 44% OpenImages),
      - CUBS200 (1%, 0.5%),
      - Indoor67 (15%, 6%),
      - and Diabetic5 (0%, 0%).

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net Results

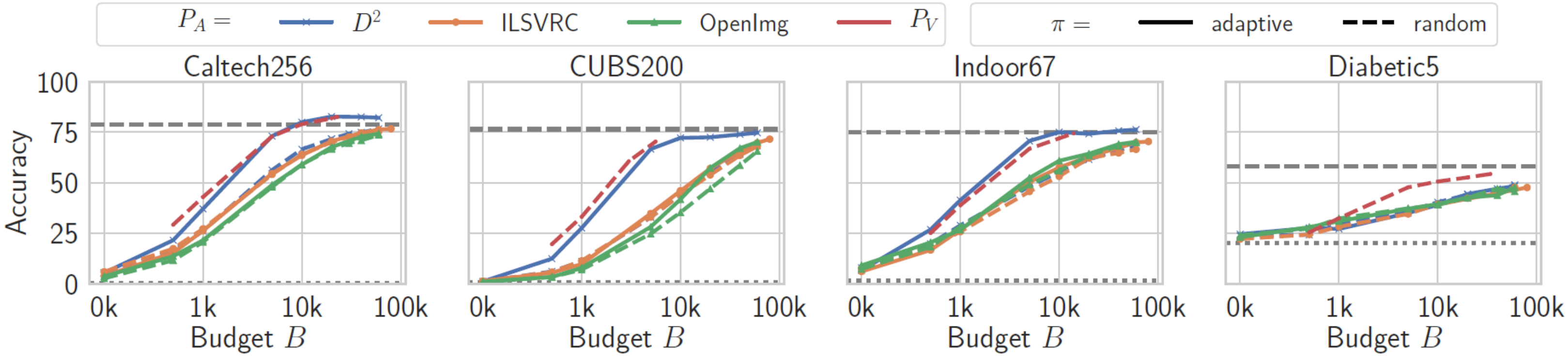


		random				adaptive			
	P_A	Caltech256	CUBS200	Indoor67	Diabetic5	Caltech256	CUBS200	Indoor67	Diabetic5
Closed	$P_V\,(F_V)$	78.8 (1×)	76.5 (1×)	74.9 (1×)	58.1 (1×)	-	-	-	-
	P_V (KD)	82.6 (1.05×)	70.3 (0.92×)	74.4 (0.99×)	54.3 (0.93×)	-	-	-	-
	D^2	76.6 (0.97×)	68.3 (0.89×)	68.3 (0.91×)	48.9 (0.84×)	82.7 (1.05×)	74.7 (0.98×)	76.3 (1.02×)	48.3 (0.83×)
Open	ILSVRC	75.4 (0.96×)	68.0 (0.89×)	66.5 (0.89×)	47.7 (0.82×)	76.2 (0.97×)	69.7 (0.91×)	69.9 (0.93×)	44.6 (0.77×)
	OpenImg	73.6 (0.93×)	65.6 (0.86×)	69.9 (0.93×)	47.0 (0.81×)	74.2 (0.94×)	70.1 (0.92×)	70.2 (0.94×)	47.7 (0.82×)

- Random strategy
  - P_A = P_V (KD using unlabeled data at temperature = 1)
    - All knockoff models recover 0.92-1.05X performance of F_V ;
    - Can even achieve a better performance than F_V itself.
  - Closed-world (#queries=60k)
    - The knockoff reasonably imitates all the black-box models, recovering 0.84-0.97X performance.
  - Open-world (#queries=60k)
    - The knockoff is able to obtain 0.81-0.96X performance.
    - Results marginally vary (at most 0.04×) between ILSVRC and OpenImages, indicating any large diverse set of images makes for a good transfer set.
- Adaptive Strategy
  - Adaptive display improved performance (up to 4.5%) consistently across all choices of F_V.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

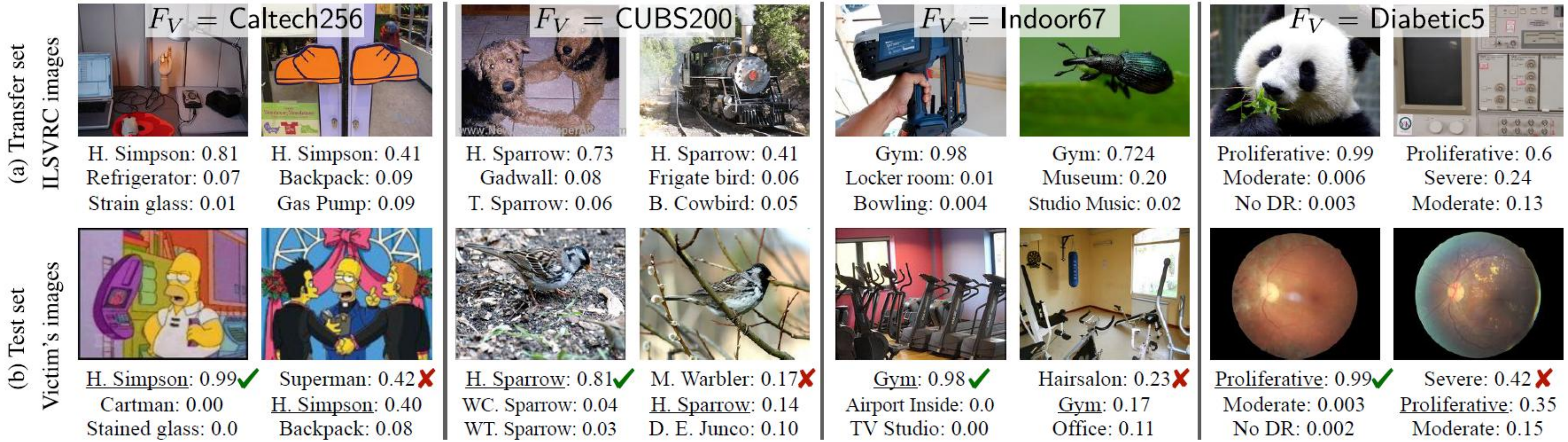
Knockoff Net Results (various number of queries)



- Closed-world
  - Adaptive is extremely sample-efficient in all but one case, i.e., Diabetic5.
    - Its performance is comparable to P_A = P_V, although samples are drawn from a 36-188× larger image distribution.
    - Significant sample-efficiency improvements
      - E.g., while CUBS200-random reaches 68.3% at B=60k, adaptive achieves this 6X quicker at B=10k.
    - Comparably low performance in Diabetic5
      - The black-box exhibits confident predictions for all images resulting in poor feedback signal.
- Open-world
  - Adaptive has marginal improvements over random in this challenging scenario
    - 1.5X quicker to reach an accuracy 57% on CUBS200 with OpenImages.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

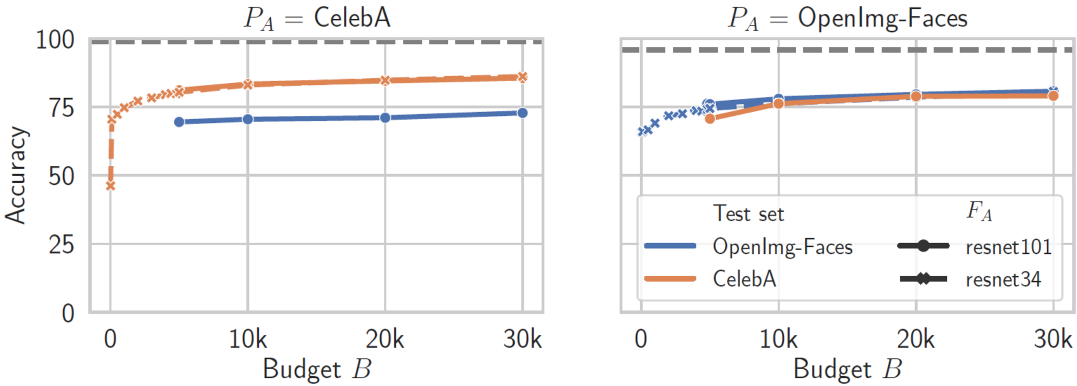
Knockoff Net



- (a) Transfer set
  - Samples from the transfer set displayed for four output classes (one from each F_V)
    - 'Homer Simpson', 'Harris Sparrow', 'Gym', and 'Proliferative DR'.
- (b) Test set
  - The knockoff F_A trained on the transfer set but evaluated on victim's test set.
  - <u>Ground truth labels</u> are underlined.
  - Objects from these classes were never encountered while training F_A.

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Knockoff Net



- Stealing functionality of a real-world black-box Face Reconigtion (random strategy)
  - Strong performance of the knockoffs achieving 0.76-0.82X performance as that of the API on the test sets.
  - The diverse nature OpenImages-Faces helps improve generalization resulting in 0.82X accuracy of the API on both testsets.
  - The complexity of F_A does not play a significant role.
    - Both Resnet-34 and Resnet-101 show similar performance
    - Indicating a compact architecture is sufficient to capture discriminative features for this particular task.
  - An inexpensive knockoff can be trained
    - Using victim API queries amounting to only USD $30 ($1-2 per 1k queries).

Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).

Can we do better?

Unlabeled data improves query efficiency

- Semi-supervised learning
  - Consider a learner with some labeled data, but much more unlabeled data
    - Unlabeled data can normally be collected automatically.
  - The learner seeks to leverage the unlabeled data to improve performance
    - For example, by training on guessed labels
  - Assuming access to unlabeled **task-specific** data, semi-supervised learning can be used to improve model extraction attacks.

- MixMatch
  - An effective semi-supervised learning technique.
  - Use a combination of successful techniques
    - including training on "guessed“ labels, regularization, and image augmentations.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A. and Papernot, N., 2020. High accuracy and high fidelity extraction of neural networks. In 29th USENIX security symposium (USENIX Security 20) (pp. 1345-1362).

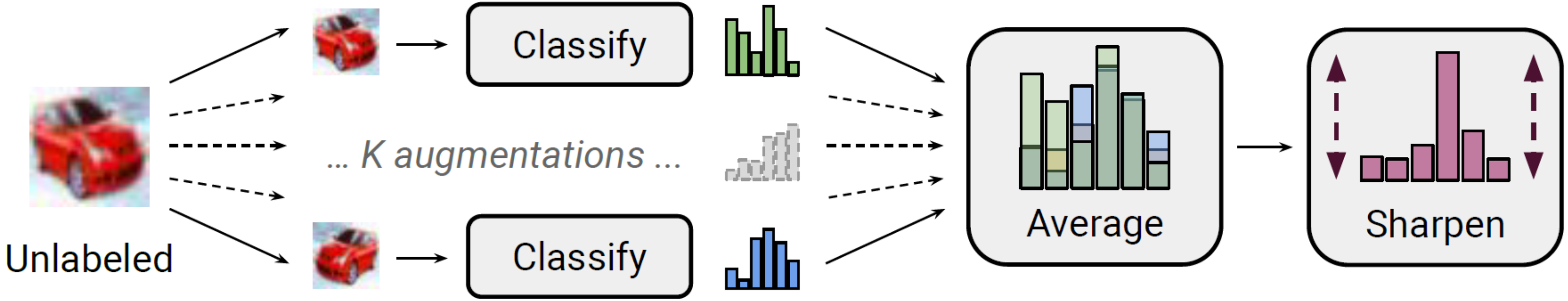
Unlabeled data improves query efficiency

- MixMatch
  - MixMatch is a “holistic” approach which incorporates ideas and components from the dominant paradigms for Semi-supervised learning.
  - Overview: given a batch X of labeled examples with one-hot targets (representing one of L possible labels) and an equally-sized batch U of unlabeled examples
    - MixMatch produces a processed batch of augmented labeled examples X',
    - and a batch of augmented unlabeled examples with “guessed” labels U'.
    - X' and U' are then used in computing **labeled** and **unlabeled** loss terms.

- MixMatch Step 1: augmentation
  - For each x_b in the batch of labeled data X, we generate a transformed version.
  - For each u_b in the batch of unlabeled data U, we generate K augmentations
    - They are used together to generate a “guessed label” u_b.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Unlabeled data improves query efficiency



- MixMatch Step 2: Label Guessing and sharpening for unlabeled data u_b
  - Each augmented image is fed through the classifier.
  - The average of these K predictions (probabilities for each category) is “sharpened” by adjusting the distribution’s temperature.
    - The sharpening function lowers the “temperature” T of predictions (T=0.5 in experiments).
    - $T \rightarrow 0$: output → one-hot encoding.
    - Sharpened predictions are used as soft labels for each augmented unlabeled data.

$$Sharpen(p,T) = \frac{p_i^{\frac{1}{T}}}{\sum_{j\in\{1,2,\ldots,L\}} p_j^{\frac{1}{T}}}$$

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

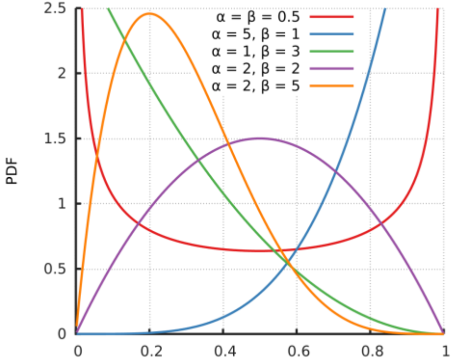
Unlabeled data improves query efficiency

- MixMatch Step 3: MixUp
  - Collect all augmented labeled and unlabeled data into a set W.
  - Blend each augmented labeled and unlabeled data with a random sample from W
    - Blending two samples x_1 and x_2 with predictions p_1 and p_2, respectively:

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1 - \lambda)$$
$$x' = \lambda' x_1 + (1 - \lambda') x_2$$
$$p' = \lambda' p_1 + (1 - \lambda') p_2$$

    - The max function ensures that x' is closer to x1 than to x2.
      - Labeled data and unlabeled data use different loss functions.
      - This ensures that correct loss is used.

Beta distribution



Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Unlabeled data improves query efficiency

- MixMatch Step 4: calculating loss
  - Given processed batches X' and U', the standard semi-supervised loss is used

$$\mathcal{X}', \mathcal{U}' = \operatorname{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \mathrm{H}(p, \mathrm{p}_{\text{model}}(y \mid x; \theta))$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \| q - \mathrm{p}_{\text{model}}(y \mid u; \theta) \|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

    - where H(p, q) is the cross-entropy between distributions p and q,
    - L represents the number of classes.
    - T, K, α, and λ_U are hyperparameters
  - L_U corresponds to a form of consistency regularization
    - Key idea: a classifier should output the same class distribution for an unlabeled example even after it has been augmented.
      - K Augmented unlabeled data share the same guessed labels before the MixUp step.

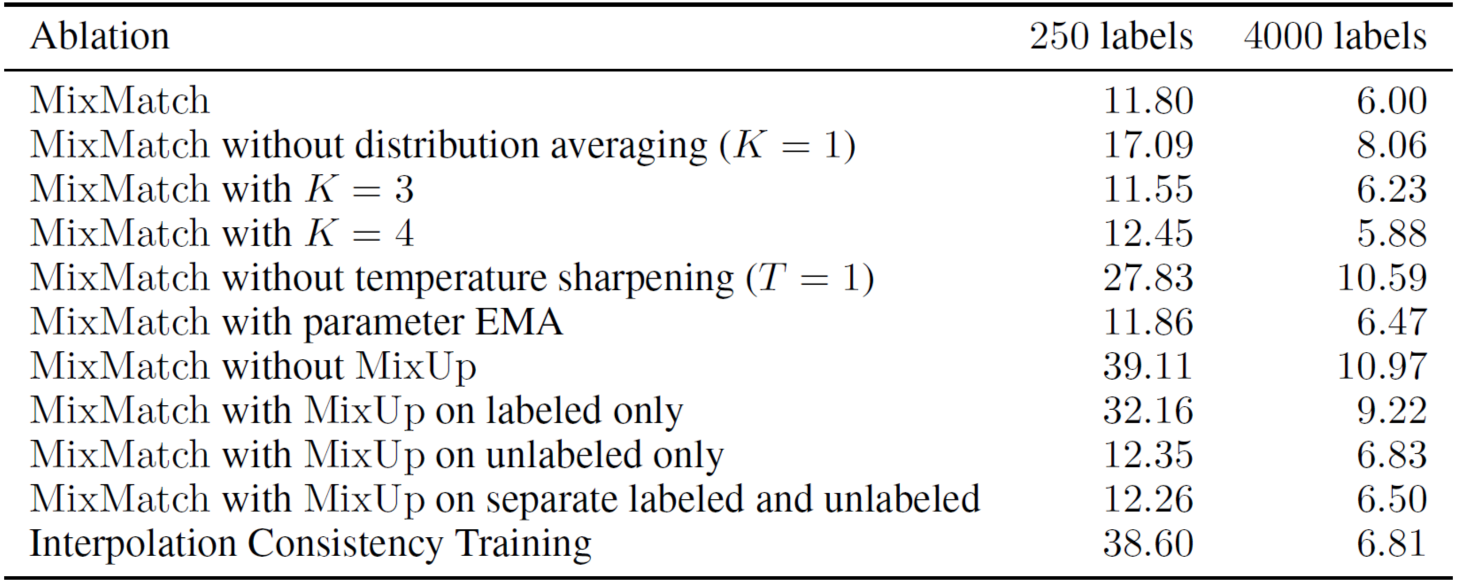
Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Unlabeled data improves query efficiency

- MixMatch ablation study
  - MixMatch combines various semi-supervised learning mechanisms.
    - Need to know which component really contributes to the performance.
  - Ablation study reveals the effect of removing or adding components.
    - Using the mean class distribution over K augmentations or using the class distribution for a single augmentation (i.e. setting K = 1).
    - Removing temperature sharpening (i.e. setting T = 1).
    - Using an exponential moving average (EMA) of model parameters when producing guessed labels, as is done by Mean Teacher (not covered in this subject).
    - Performing MixUp between labeled examples only, unlabeled examples only, and without mixing across labeled and unlabeled examples.
    - Using Interpolation Consistency Training (not covered in this subject)
      - This can be seen as a special case of this ablation study where only unlabeled mixup is used, no sharpening is applied and EMA parameters are used for label guessing.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Unlabeled data improves query efficiency



Ablation	250 labels	4000 labels
MixMatch	11.80	6.00
MixMatch without distribution averaging ($K = 1$)	17.09	8.06
MixMatch with $K = 3$	11.55	6.23
MixMatch with $K = 4$	12.45	5.88
MixMatch without temperature sharpening ($T = 1$)	27.83	10.59
MixMatch with parameter EMA	11.86	6.47
MixMatch without MixUp	39.11	10.97
MixMatch with MixUp on labeled only	32.16	9.22
MixMatch with MixUp on unlabeled only	12.35	6.83
MixMatch with MixUp on separate labeled and unlabeled	12.26	6.50
Interpolation Consistency Training	38.60	6.81

- MixMatch ablation study
  - Error rates on CIFAR-10 with 250 and 4000 labels.
    - T = 0.5 and K = 2 for MixMatch
  - Each component contributes to MixMatch’s performance,
    - with the most dramatic differences in the 250-label setting.

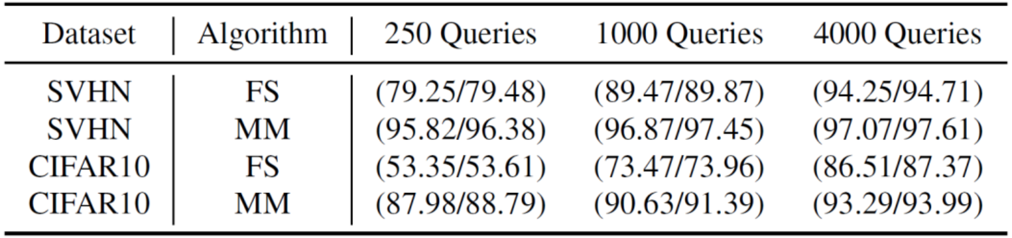
Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Unlabeled data improves query efficiency

- Include MixMatch into model stealing attack
    - Two Datasets of 32x32 images belonging to one of 10 classes
        - SVHN
            - The training contains 73257 images and the test set contains 26032 images.
        - CIFAR10
            - The training set contains 50000 images and the test set contains 10000 images.
    - Adversary
        - Has access to the same training set.
        - Only needs to query the oracle on a small subset of these training points to extract a model
            - Accuracy on the task is comparable to the victim.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A. and Papernot, N., 2020. High accuracy and high fidelity extraction of neural networks. In 29th USENIX security symposium (USENIX Security 20) (pp. 1345-1362).

Unlabeled data improves query efficiency



Dataset	Algorithm	250 Queries	1000 Queries	4000 Queries
SVHN	FS	(79.25/79.48)	(89.47/89.87)	(94.25/94.71)
SVHN	MM	(95.82/96.38)	(96.87/97.45)	(97.07/97.61)
CIFAR10	FS	(53.35/53.61)	(73.47/73.96)	(86.51/87.37)
CIFAR10	MM	(87.98/88.79)	(90.63/91.39)	(93.29/93.99)

Performance (accuracy/fidelity) of fully supervised (FS) and MixMatch (MM) extraction on SVHN and CIFAR10. Fidelity means label agreement. FS uses KD on data labeled by the victim (prediction probabilities are provided).

- Victim model (WideResNet-28-2)
  - 97.36% accuracy on SVHN; 95.75% accuracy on CIFAR10.
- Results.
  - with only 250 queries (293x smaller label set than the SVHN oracle and 200x smaller for CIFAR10)
    - MixMatch reaches 95.82% test accuracy on SVHN and 87.98% accuracy on CIFAR10.
    - This is higher than fully supervised training that uses 4000 queries.
  - With 4000 queries
    - MixMatch is within 0.29% of the accuracy of the victim on SVHN, and 2.46% on CIFAR10.
  - These gains come from the prior MixMatch is able to build using the unlabeled data, making it effective at exploiting few labels.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A. and Papernot, N., 2020. High accuracy and high fidelity extraction of neural networks. In 29th USENIX security symposium (USENIX Security 20) (pp. 1345-1362).

References

- Hinton, G., 2015. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.
- Orekondy, T., Schiele, B. and Fritz, M., 2019. Knockoff nets: Stealing functionality of black-box models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4954-4963).
- Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C.A., 2019. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.
- Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A. and Papernot, N., 2020. High accuracy and high fidelity extraction of neural networks. In 29th USENIX security symposium (USENIX Security 20) (pp. 1345-1362).