Transfer Learning without Knowing: Reprogramming Black-box Machine Learning Models with Scarce Data and Limited Resources

Yun-Yun Tsai [1] **Pin-Yu Chen** [2] **Tsung-Yi Ho** [1]

Abstract

Current transfer learning methods are mainly based on finetuning a pretrained model with target-domain data. Motivated by the techniques from adversarial machine learning (ML) that are capable of manipulating the model prediction via data perturbations, in this paper we propose a novel approach, black-box adversarial reprogramming (BAR), that repurposes a well-trained black-box ML model (e.g., a prediction API or a proprietary software) for solving different ML tasks, especially in the scenario with scarce data and constrained resources. The rationale lies in exploiting high-performance but unknown ML models to gain learning capability for transfer learning. Using zeroth order optimization and multi-label mapping techniques, BAR can reprogram a black-box ML model solely based on its input-output responses without knowing the model architecture or changing any parameter. More importantly, in the limited medical data setting, on autism spectrum disorder classification, diabetic retinopathy detection, and melanoma detection tasks, BAR outperforms state-of-the-art methods and yields comparable performance to the vanilla adversarial reprogramming method requiring complete knowledge of the target ML model. BAR also outperforms baseline transfer learning approaches by a significant margin, demonstrating cost-effective means and new insights for transfer learning.

1. Introduction

Transfer learning is a widely used practical machine learning (ML) methodology for learning to solve a new task in a target domain based on the knowledge transferred from a source-domain task (Pan & Yang, 2009). One popular target-domain task is transfer learning of medical imaging with a large and rich benchmark dataset (e.g., ImageNet) as the source-domain task, since high-quality labeled medical images are often scarce and costly to acquire new samples (Raghu et al., 2019). For deep learning models, transfer learning is often achieved by finetuning a pretrained source-domain model with the target-domain data, which requires complete knowledge and full control of the pretrained model, including knowing and modifying the model architecture and pretrained model parameters.

In this paper, we revisit transfer learning to address two fundamental questions: (i) Is finetuning a pretrained model necessary for learning a new task? (ii) Can transfer learning be expanded to *black-box* ML models where nothing but only the input-output model responses (data samples and their predictions) are observable? In contrast, we call finetuning a *white-box* transfer learning method as it assumes the source-domain model to be transparent and modifiable.

Recent advances in adversarial ML have shown great capability of manipulating the prediction of a well-trained deep learning model by designing and learning perturbations to the data inputs without changing the target model (Biggio & Roli, 2018), such as prediction-evasive adversarial examples (Szegedy et al., 2014). Despite of the "vulnerability" in deep learning models, these findings also suggest the plausibility of transfer learning without modifying the pretrained model if an appropriate perturbation to the target-domain data can be learned to align the target-domain labels with the pretrained source-domain model predictions. Indeed, the *adversarial reprogramming* (AR) method proposed in (Elsayed et al., 2019) partially gives a negative answer to Question (i) by showing simply learning a universal target-domain data perturbation is sufficient to repurpose a pretrained source-domain model, where the domains and tasks can be different, such as reprogramming an ImageNet classifier to solve the task of counting squares in an image. However, the authors did not investigate the performance of AR on the limited data setting often encountered in transfer learning. Moreover, since the training of AR requires backpropagation of a deep learning model, AR still falls into the category of

[1]National Tsing Hua University, Hsinchu, Taiwan [2]IBM Research. Correspondence to: Yun-Yun Tsai <alice103000004@gmail.com>, Pin-Yu Chen <pin-yu.chen@ibm.com>, Tsung-Yi Ho <tyho@cs.nthu.edu.tw>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

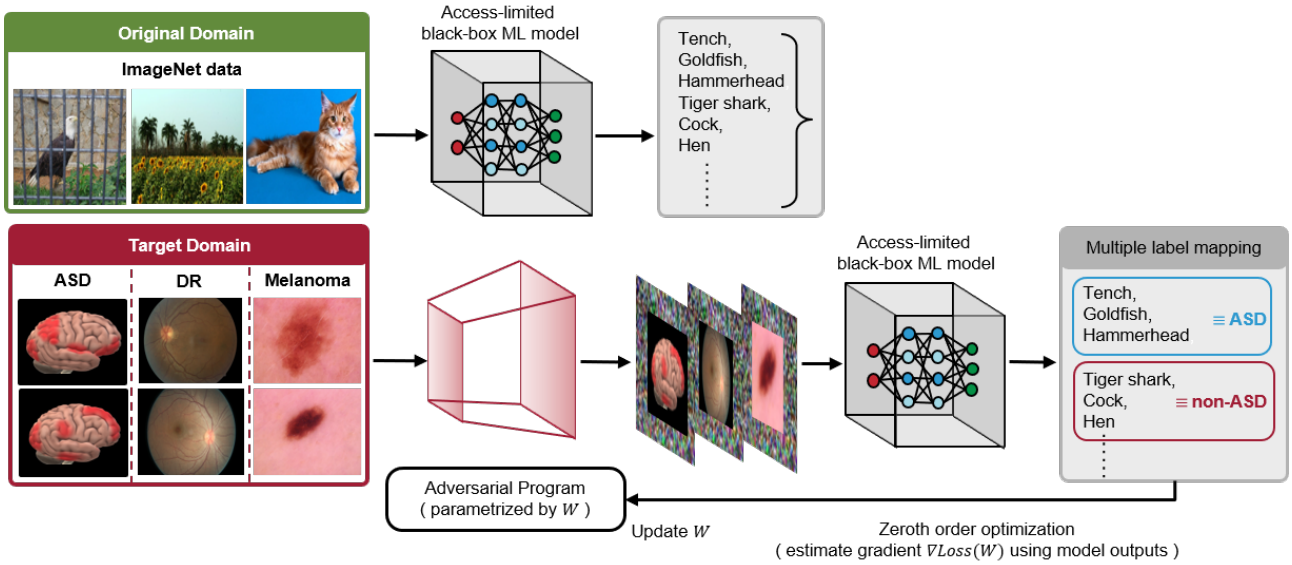

Figure 1. Schematic overview of our proposed black-box adversarial reprogramming (BAR) method.

white-box transfer learning methods and hence does not address Question (ii).

To bridge this gap, we propose a novel approach, named black-box adversarial reprogramming (BAR), to reprogram a deployed ML model (e.g., an online image classification service) for black-box transfer learning. Comparing to the vanilla (white-box) AR approach, our BAR has the following substantial differences and unique challenges:

1. **Black-box setting.** The vanilla AR method assumes complete knowledge of the pretrained (target) model, which precludes the ability of reprogramming a well-trained but access-limited ML models such as prediction APIs or proprietary softwares that only reveal model outputs based on queried data inputs.

2. **Data scarcity and resource constraint.** While data is crucial to most of ML tasks, in some scenarios such as medical applications, massive data collection can be expensive, if not impossible, especially when clinical trials, expert annotation or privacy-sensitive data are involved. Consequently, without transfer learning, the practical limitation of data scarcity may hinder the strength of complex (large-scaled) ML models such as deep neural networks (DNNs). Moreover, even with moderate amount of data, researchers may not have sufficient computation resources or budgets to train a DNN as large as a commercial ML model or perform transfer learning on a large pretrained ML model.

Our proposed BAR tackles these two challenges in a cost-effective manner, which not only firstly extends white-box transfer learning to the black-box regime but also "unlocks" the power of well-trained but access-limited ML models for transfer learning. In particular, we focus on adversarial reprogramming of black-box image classification models for solving medical imaging tasks, as image classification is one of the most mature AI applications and many medical ML tasks often entail data scarcity challenges. As will be evident in the Experiments section (Sec. 4), BAR can successfully leverage the powerful feature extraction capability of black-box ImageNet classifiers to achieve high performance in three medical image classification tasks with limited data.

Figure 1 provides an overview of our proposed BAR method. To adapt to the black-box setting, we leverage zeroth-order optimization (Ghadimi & Lan, 2013) on iterative input-output model responses to enable black-box transfer learning. We also use multi-label mapping of source-domain and target-domain labels to enhance the performance of BAR. We summarize our main contributions as follows.

- We propose BAR, a novel approach to reprogram black-box ML models for transfer learning. To the best of our knowledge, BAR is the first work that expands transfer learning to the black-box setting without knowing or finetuning the pretrained model.

- We evaluate the performance of BAR using three different medical imaging tasks for transfer learning from pretrained ImageNet models: (a) autism spectrum disorder (ASD) classification; (b) diabetic retinopathy (DR) detection; and (c) melanoma detection. The results show that our method consistently outperforms the state-of-the-art methods and improves the accuracy of the finetuning approach by a significant margin. We also explain the success of BAR through a representa-

tion analysis and several ablation studies.

- We demonstrate the practicality and effectiveness of BAR by reprogramming real-life image classification APIs from Clarifai.com[1] and Microsoft Custom Vision[2], which is infeasible for the vanilla white-box AR method due to the black-box setting. In terms of total expenses, it only costs less than $24 US dollars to reprogram these two APIs for ASD classification.

2. Related Work

2.1. Adversarial ML and Reprogramming

Adversarial ML mainly studies how to manipulate the decision-making of a target model and develop countermeasures (Biggio & Roli, 2018). In particular, several works have identified the vulnerability of DNNs to different types of adversarial threats, such as crafting prediction-evasive adversarial examples (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2017; Chen et al., 2018) or poisoning training data and implanting backdoors in the downstream ML models (Muñoz-González et al., 2017; Chen et al., 2017b; Shafahi et al., 2018; Gu et al., 2019; Xie et al., 2020), to name a few.

Adversarial reprogramming (AR) is a recently introduced technique that aims to reprogram a target ML model for performing a different task (Elsayed et al., 2019). Different from typical transfer learning methods that modify the model architecture or parameters for solving a new task with target-domain data, AR keeps the model architecture unchanged. Instead, AR uses a trainable adversarial program and a designated output label mapping on the target-domain data samples to perform reprogramming. Intuitively, the adversarial program serves as a parametrized and trainable input transformation function such that when applied to the target-domain data (e.g., images having squares), the *same* target model will be reprogrammed for the new task (e.g., the output label "dog" of a programmed data input translates to "3 squares"). The work in (Neekhara et al., 2019) demonstrates AR of text classification but still assumes white-box access to the target ML models.

2.2. Zeroth Order Optimization for Black-box Setting

It is worth noting that the vanilla AR method proposed in (Elsayed et al., 2019) requires complete access to the target ML model to allow back-propagation for training the parameters of adversarial program. In other words, vanilla AR lacks the ability to reprogram an access-limited ML model such as prediction API, owing to prohibited access to the target model disallowing back-propagation. To bridge this gap and empower reprogramming advanced yet access-limited ML models trained with tremendous amount of data and considerable computation resources (e.g., Google cloud vision API), we use zeroth order optimization techniques to enable black-box AR for transfer learning.

In contrast to the conventional first order (gradient-based) optimization methods such as stochastic gradient descent (SGD), zeroth order optimization (Ghadimi & Lan, 2013) achieves gradient-free optimization by merely using numerical evaluations of the same training loss function instead of gradients, making it a powerful tool for the black-box setting. As the gradients of black-box ML models are infeasible to obtain, the main idea of zeroth order optimization is to replace the true gradients in first-order algorithms with gradient estimates from function evaluations. Prior arts in adversarial ML have shown that zeroth order optimization can be used to generate adversarial examples (Chen et al., 2017a; Tu et al., 2019; Brendel et al., 2018; Ilyas et al., 2018; Cheng et al., 2019; 2020), known as black-box adversarial attacks. Moreover, advanced zeroth order optimization methods can provide query-efficient solutions for black-box ML tasks (Liu et al., 2018; 2019; 2020).

3. Black-box Adversarial Reprogramming (BAR): Method and Algorithm

This section presents our proposed method and algorithm, named BAR, for reprogramming black-box ML models. A schematic overview of BAR is illustrated in Figure 1.

3.1. Problem Formulation

Black-box setting. We consider the problem of reprogramming a black-box ML classification model denoted by $F : \mathcal{X} \mapsto \mathbb{R}^K$, where it takes a data sample $X \in \mathcal{X}$ as an input and gives a vector of confidence scores $F(X) = [F_1(X), F_2(X), \ldots, F_K(X)] \in \mathbb{R}^K$ as its output, where $\mathcal{X}$ denotes the space of feasible data samples (e.g., image sizes and pixel value ranges) and K is the number of classes. Similar to the access rights of a regular user when using a prediction API, one is able to observe the model output $F(X)$ for any given $X \in \mathcal{X}$, whereas inquiring the gradient $\nabla F(X)$ is inadmissible.

Adversarial program. To reprogram a black-box ML model, we use the same form of adversarial program in (Elsayed et al., 2019) as an input transformation function to translate the data of the target domain to the input space of the source domain. Without loss of generality, let $\mathcal{X} = [-1, 1]^d$ denote the scaled input space of an ML model F, where d is the (vectorized) input dimension. We also denote the set of data from the target domain by $\{D_i\}_{i=1}^n$, where $D_i \in [-1, 1]^{d'}$ and $d' < d$ to allow extra dimensions

[1] https://www.clarifai.com

[2] https://azure.microsoft.com/en-us/services/cognitive-services/custom-vision-service/

for reprogramming. For each data sample $i \in [n]$, where $[n]$ denotes the integer set $\{1, 2, \ldots, n\}$, we let X_i be the zero-padded data sample containing D_i, such as embedding a brain-regional correlation graph of size 200×200 to the center of a 299×299 (width $\times$ height) image, as shown in Figure 1. Let $M \in \{0, 1\}^d$ be a binary mask function indicating the common embedding location for $\{D_i\}_{i=1}^n$, where $M_j = 0$ means the j-th dimension is used for embedding and $M_j = 1$ otherwise. The transformed data sample for AR is defined as

$$\widetilde{X}_i = X_i + P \text{ and } P = \tanh(W \odot M), \tag{1}$$

where P is called an adversarial program to be learned and is universal to all target data samples $\{X_i\}_{i=1}^n$, $W \in \mathbb{R}^d$ is a set of trainable parameters for AR, $\odot$ denotes the Hadamard (entry-wise) product, and $\tanh \in [-1, 1]$ ensures $\widetilde{X}_i \in [-1, 1]^d$. Note that the binary mask function M in P ensures the target data samples $\{D_i\}$ embedded in $\{\widetilde{X}_i\}$ are intact during AR.

Multi-label mapping (MLM). As illustrated in Figure 1, in addition to input data transformation via an adversarial program, for AR we also need to map the source task's output labels (e.g., different objects) to the target task's output labels (e.g., ASD or non-ASD). Evaluated on three medical tasks (see Section 4), we find that multiple-source-labels to one-target-label mapping can further improve the accuracy of the target task when compared to one-to-one label mapping. For instance, the prediction of a transformed data input from the source label set {Tench,Goldfish,Hammerhead} will be reprogrammed for predicting the target class ASD. Let K (K') be the total number of classes for the source (target) task. We use the notation $h_j(\cdot)$ to denote the k-to-1 mapping function that averages the predictions of a group of k source labels as the prediction of the j-th target domain's label. For example, If the source labels {Tench,Goldfish,Hammerhead} map to the target label {ASD}, then $h_{\text{ASD}}(F(X)) = [F_{\text{Tench}}(X) + F_{\text{Goldfish}}(X) + F_{\text{Hammerhead}}(X)]/3$. More generally, if a subset of source labels $\mathcal{S} \subset [K]$ map to a target label $j \in [K']$, then $h_j(F(X)) = \frac{1}{|\mathcal{S}|} \sum_{s \in S} F_s(X)$, where $|\mathcal{S}|$ is the set cardinality. Furthermore, we propose a frequency-based label mapping scheme by matching target labels to source labels according to the label distribution of initial predictions on the target-domain data before reprogramming. We find that it improves the accuracy over random label mapping. We defer the readers to Section 4 for more details and an ablation study on multi-label mapping in Sec. 4.5.

Loss function for AR. Here we formally define the training loss for AR. Without loss of generality, we assume the model output is properly normalized such that $\sum_{j=1}^K F_j(X) = 1$ and $F_j(X) \geq 0$ for all $j \in [K]$, which can be easily satisfied by applying a softmax function to the model output. Let $\{y_i\}_{i=1}^n$ with $y_i = [y_{i1}, \ldots, y_{iK'}] \in \{0, 1\}^{K'}$ denote the one-hot encoded label for the target domain task and let $h(F(X)) = [h_1(F(X)), \ldots, h_{K'}(F(X))]$ be a surjective multi-label mapping function from the model prediction $F(X)$ of the source domain to the target domain. For training the adversarial program P parametrized by W, we use the focal loss (Lin et al., 2017) as empirically it can further improve the performance of AR/BAR (see Sec. 4.5). The focal loss (F-loss) aims to penalize the samples having low prediction probability during training, and it includes the conventional cross entropy loss (CE-loss) as a special case. The focal loss of the ground-truth label $\{y_i\}_{i=1}^n$ and the transformed prediction probability $\{h\left(F(X_i + P)\right)\}_{i=1}^n$ is

$$-\sum_{i=1}^{n} \sum_{j=1}^{K'} \omega_j (1 - h_j)^\gamma y_{ij} \log h_j \left(F(X_i + P)\right), \tag{2}$$

where $\omega_j > 0$ is a class balancing coefficient, $\gamma \geq 0$ is a focusing parameter which down-weights high-confidence (large h_j) samples. When $\omega_j = 1$ for all j and $\gamma = 0$, the focal loss reduces to the cross entropy. In our implementation, we set $\omega_j = 1/n_j$ and $\gamma = 2$, where n_j is the number of samples in class j, as suggested in (Lin et al., 2017).

Note that the loss function is a function of W since from (1) the adversarial program P is parametrized by W, and W is the set of optimization variables to be learned for AR. The loss function can be further generalized to the minibatch setting for stochastic optimization.

3.2. Zeroth Order Optimization for BAR

In the white-box setting assuming complete access to the target ML model F, optimizing the loss function in (2) and retrieving its gradient for AR are straightforward via back-propagation. However, when F is a black-box model and only the model outputs $F(\cdot)$ are available for AR, back-propagation through F is infeasible since the gradient $\nabla F(\cdot)$ is inadmissible. In our BAR framework, to optimize the loss function in (2) and update the parameters W of the adversarial program, we propose to use zeroth order optimization to solve for W. Specifically, there are two major components to enable BAR: (i) gradient estimation and (ii) gradient descent with estimated gradient.

Query-efficient gradient estimation. Let $f(W)$ be the $Loss$ defined in (2) and W be the optimization variables. To estimate the gradient $\nabla f(W)$, we use the one-sided averaged gradient estimator (Liu et al., 2018; Tu et al., 2019) via q random vector perturbations, which is defined as

$$\bar{g}(W) = \frac{1}{q} \sum_{j=1}^{q} g_j, \tag{3}$$

where $\{g_j\}_{j=1}^q$ are q independent random gradient estimates of the form

$$g_j = b \cdot \frac{f(W + \beta U_j) - f(W)}{\beta} \cdot U_j, \tag{4}$$

where b is a scalar balancing bias and variance trade-off of the estimator, $W \in \mathbb{R}^d$ is the set of optimization variables in vector form, β is the smoothing parameter, and $U_j \in \mathbb{R}^d$ is a vector that is uniformly drawn at random from a unit Euclidean sphere. The mean squared estimation error between $\bar{g}(W)$ and the true gradient $\nabla f(W)$ has been characterized in (Tu et al., 2019) with mild assumptions on f. In our experimental setup, we set $b = d$ in order to obtain an unbiased gradient estimator of a smoothed function of ∇f (Gao et al., 2014), and we set β to be of the order $1/d$ (i.e., $\beta = 0.01$) following the analysis in (Liu et al., 2018) and set U_j to be a realization of a standard normal Gaussian random vector divided by its Euclidean norm. By construction, for each data sample X_i, $i \in [n]$, the averaged gradient estimator takes $q + 1$ queries from the ML model F. Smaller q can reduce the number of queries to the target model but may incur larger gradient estimator error. We will study the influence of q on the performance of BAR in the next section.

BAR algorithm. Using the averaged gradient estimator $\bar{g}$, our BAR algorithm is compatible with any gradient-based training algorithm by simply replacing the inadmissible gradient $\nabla Loss$ with $\bar{g}$ in the gradient descent step. The corresponding algorithmic convergence guarantees have been proved in recent works such as (Liu et al., 2018; 2019) in both the convex loss and non-convex loss settings. In this paper, we use stochastic gradient descent (SGD) with $\bar{g}$ to optimize the parameters W in BAR, which are updated by

$$W_{t+1} = W_t - \alpha_t \cdot \bar{g}(W_t), \tag{5}$$

where t is the t-th iteration for updating W with a minibatch sampled from $\{X_i\}_{i=1}^n$ (we set the minibatch size to be 20), α_t is the step size (we use exponential decay with initial learning rate η), and $\bar{g}(W_t)$ is the gradient estimate of the loss function at W_t using the t-th minibatch. Note that since the loss function defined in (2) is a function of the target ML model F's input and output, and the parameters W of the adversarial program only associate with the input of F, the entire gradient estimation and training process for BAR is indeed operated in a black-box manner. That is, BAR only uses input-output responses of F and does *not* assume access to the model internal details such as model type, parameters, or source-domain data. The entire training process for BAR takes $\#\ iterations \times mini\ batch\ size \times\ (q+1)$ queries to F. Algorithm 1 summarizes our proposed BAR method. For the ease of description, the minibatch size is set to be the training data size n in Algorithm 1.

Algorithm 1 Training algorithm of black-box adversarial reprogramming (BAR)

Input: black-box ML model F, AR loss function $Loss(\cdot)$, target domain training data $\{D_i, y_i\}_{i=1}^n$, maximum number of iterations T, number of random vectors for gradient estimation q, multi-label mapping function $h(\cdot)$, step size $\{\alpha_t\}_{t=1}^T$

Output: Optimal adversarial program parameters W

1: Randomly initialize W; set $t = 1$
2: Embed $\{D_i\}_{i=1}^n$ with mask M to create $\{X_i\}_{i=1}^n$
3: **while** $t \leq T$ **do**
4: **# Generate adversarial program**
$P = \tanh(W \odot M)$
 # Generate q perturbed adversarial programs
$\widetilde{P}_j = \tanh((W + U_j) \odot M)$ for all $j \in [q]$
$\{U_j\}_{j=1}^q$ are random vectors defined in (4)
5: **# Function evaluation for gradient estimation**
Evaluate $Loss$ in (2) with W and $\{X_i + P\}_{i=1}^n$
Evaluate $Loss$ in (2) with $W + U_j$ and $\{X_i + \widetilde{P}_j\}_{i=1}^n$ for all $j \in [q]$
6: **# Optimize adversarial program's parameters:**
Use Step 5 and (3) to obtain estimated gradient $\bar{g}(W)$
$W \leftarrow W - \alpha_t \cdot \bar{g}(W)$
$t \leftarrow t + 1$
7: **end while**

4. Experiments

This section presents the following experiments for performance evaluation and comparison. We release our code to encourage future work on reprogramming other classification tasks[3].

1. Reprogramming three pretrained black-box ImageNet classifiers (1000-object recognition task) from (N.Silberman & S.Guadarrama, 2016), including ResNet 50 (He et al., 2016), Inception V3 (Szegedy et al., 2016) and DenseNet 121 (Iandola et al., 2014), for three medical imaging classification tasks, including Autism Spectrum Disorder (ASD) classification (2-classes), Diabetic Retinopathy (DR) detection (5-classes) and Melanoma detection (7-classes).

2. Reprogramming two online Machine Learning-as-a-Service (MLaaS) toolsets, including Clarifai.com[1] and Microsoft Custom Vision[2], for medical imaging tasks and reporting the expenses.

3. Sensitivity analysis on the influence of number of random vectors q and multi-label mapping (MLM) size m for BAR, and ablation studies in terms of different loss functions (CE-loss v.s. F-loss) and label mapping methods (random mapping v.s. frequency mapping).

For implementing BAR and AR, we use the focal loss in (2) and frequency-based MLM derived from the initial pre-

[3] https://github.com/yunyuntsai/Black-box-Adversarial-Reprogramming

dictions of the target-domain data before reprogramming. Their ablation studies will be discussed in Section 4.5. We also highlight the results of BAR in boldface.

Baselines. To benchmark the performance of BAR, we compare it with three baselines. For fair comparisons, all methods use the same training/testing data and we do not use any data augmentation nor model ensemble techniques.

- Vanilla adversarial reprogramming (white-box AR): It assumes white-box access to the target ML model and optimizes the AR training loss in (2) using the ADAM optimizer (Kingma & Ba, 2015). Its accuracy serves as an upper bound of BAR as BAR only assumes black-box access.
- Transfer learning: We finetune the same pretrained models following the implementation in tensorflow tutorial[4]. The details are given in the supplementary material. We also implement another baseline that trains the model from scratch. Ideally, in the limited data setting training from scratch serves as a lower bound on BAR's accuracy due to insufficient data. We use the original target-domain data (without zero padding) for these two transfer learning baselines because the resulting performance is better than that with zero padding.
- State-of-the-art (SOTA): For each task, we implement the SOTA methods in the literature but disable any data augmentation or model ensemble techniques.

4.1. Autism Spectrum Disorder Classification

Classifying Autism Spectrum Disorder (ASD) is a challenging task. ASD is a complex developmental disorder that involves persistent challenges in social interaction, speech and nonverbal communication, and restricted/repetitive behaviors. It affects about 1% of the global population. Currently, the only clinical method for diagnosing ASD are standardized ASD tests, which require prolonged diagnostic time and considerable medical costs. Therefore, ML can play an important role in providing cost-effective means of detecting ASD. We use the dataset from the Autism Brain Imaging Data Exchange (ABIDE) database (Craddock et al., 2013). The preprocessed dataset[5] is split into 10 folds and contains 503 individuals suffering from ASD and 531 non-ASD samples. The data sample is a 200×200 brain-regional correlation graph of fMRI measurements, which is embedded in each color channel of ImageNet-sized inputs. In this task, we assign 5 separate ImageNet labels to each ASD label (i.e., ASD/non-ASD) for MLM and set the parameters $\eta = 0.05$ and $q = 25$. Table 1 reports the 10-fold cross validation test accuracy, where the averaged test data size is 104. The accuracy of BAR is comparable to white-box AR, and their accuracy outperforms the SOTA performance as reported in (Heinsfeld et al., 2018; Eslami et al., 2019). The performance of finetuing and training from scratch is merely close to random guessing due to limited data, and BAR's accuracy is 17%-18% better than that of transfer learning.

Table 1. Performance comparison (10-fold averaged test accuracy) on autism spectrum disorder classification task.

Model	Accuracy	Sensitivity	Specificity
ResNet 50 (BAR)	**70.33%**	**69.94%**	**72.71%**
ResNet 50 (AR)	72.99%	73.03%	72.13%
Train from scratch	51.55%	51.17%	53.56%
Transfer Learning (finetuned)	52.88%	54.13%	54.70%
Incept. V3 (BAR)	**70.10%**	**69.40%**	**70.00%**
Incept. V3 (AR)	72.30%	71.94%	74.71%
Train from scratch	50.20%	51.43%	52.67%
Transfer Learning (finetuned)	52.10%	52.65%	54.42%
SOTA 1. (Heinsfeld et al., 2018)	65.40%	69.30%	61.10%
SOTA 2. (Eslami et al., 2019)	69.40%	66.40%	71.30%

Table 2. Test accuracy on diabetic retinopathy detection task. The notation $*$ denotes the network used in SOTA method.

Model	From Scratch	Finetuning	AR	BAR
ResNet 50*	73.44%	76.63%	80.48%	**79.33%**
Incept. V3	72.10%	74.20%	76.42%	**74.33%**
DenseNet 121	67.22%	71.29%	75.22%	**72.33%**

4.2. Diabetic Retinopathy Detection

The task of Diabetic Retinopathy (DR) detection is to classify high-resolution retina imaging data collected from a Kaggle challenge[6]. The goal is to predict different scales ranging from 0 to 4 corresponding to the rating of presence of DR. Note that collecting labeled data for diagnosing DR is a costly and time-consuming process, as it requires experienced and well-trained clinicians to make annotations on the digital retina images. The collected dataset contains 5400 data samples and we hold 2400 data samples as the test set. In this task, we set the parameters $\eta = 0.05$, $q = 55$ and use 10 labels per target class for MLM. Table 2 shows the test accuracy of reprogramming different pretrained classifiers, including ResNet 50, Inception V3 and DenseNet 121. BAR can achieve 79.33% accuracy, which is 2.7% better than SOTA and nearly the same as white-box AR (80.48%). We note that even without complicated techniques such as data augmentation and model emsemble, the performance of AR/BAR is close to the current best reported accuracy (81.36%) in the literature (Sarki et al., 2019) using single model without ensemble approach, which requires specifically designed data augmentation with fine-tuning on

[4] https://www.tensorflow.org/tutorials/images/transfer_learning

[5] http://preprocessed-connectomes-project.org/abide

[6] https://www.kaggle.com/c/diabetic-retinopathy-detection

Table 3. Test accuracy on melanoma detection task. The notation $*$ denotes the network used in SOTA method.

Model	From Stratch	Finetuning	AR	BAR
ResNet 50	72.10%	76.90%	82.05%	**81.71%**
Incept. V3	70.91%	68.63%	82.01%	**80.20%**
DenseNet 121*	70.22%	68.88%	80.76%	**78.33%**

ResNet 50.

4.3. Melanoma Detection

Skin cancer is the most common type disease, with over 5 million newly diagnosed cases in the United States every year. However, visual inspection of the skin and differentiating the type of skin diseases still remains as a challenging problem. ML-based approaches have been actively studied to address this challenge. Here the target-domain dataset is extracted from the International Skin Imaging Collaboration (ISIC) (Codella et al., 2019; Tschandl et al., 2018) dataset, containing 10015 images of 7 types of skin cancer. The average image size is 450×600 pixels. We resize these data samples to be 64×64 pixels and embed them in the center of ImageNet-sized inputs. Since the data distribution is imbalanced (70% data samples belong to one class), we perform re-sampling on the training data to ensure the same sample size for each class. Finally, the training/testing data samples are 7800/780. In this task, we assign 10 separate ImageNet labels to each target-domain label for MLM and set the parameters $\eta = 0.05$ and $q = 65$. Table 3 reports the test accuracy of different methods. Consistent with previous findings, BAR attains similar accuracy as AR. More importantly, their accuracy significantly increases the accuracy of finetuning by a significant margin (5-10%), especially for Inception V3 and DenseNet 121 models. Training from scratch again suffers from insufficient data samples and hence has low accuracy. The performance of BAR/AR even outperforms the best reported accuracy (78.65%) in the literature, which uses specifically designed data augmentation with finetuning on DenseNet (Li & Li, 2018).

4.4. Reprogramming Real-life Prediction APIs

To further demonstrate the practicality of BAR in reprogramming access-limited (black-box) ML models, we use two real-life online ML-as-a-Service (MLaaS) toolkits provided by Clarifai.com and Microsoft Custom Vision. For Clarifai.com, a regular user on an MLaaS platform can provide any data input (of the specified format) and observe a model's prediction via Prediction API but has no information about the model and training data used. For Microsoft Custom Vision, it allows users to upload labeled datasets and trains a ML model for prediction, but the trained model is unknown to users. We aim to show how BAR can "unlock" the inference power of these unknown ML models and reprogram them for Autism spectrum disorder classification or Diabetic retinopathy detection tasks. Note that white-box AR and current transfer learning methods are inapplicable in this setting as acquiring input gradients or modifying the target model is inadmissible via prediction APIs.

Table 4. Performance of BAR on Clarifai.com APIs.

Orig. Task to New Task	q	# of query	Accuracy	Cost
NSFW to ASD	15	12.8k	64.04%	$14.24
	25	24k	**65.70%**	$23.2
Moderation to ASD	15	11.9k	65.14%	$13.52
	25	23.8k	**67.32%**	$23.04
Moderation to DR	15	15.2k	71.03%	$18.24
	25	26.4k	**72.75%**	$31.68

Table 5. Performance of BAR on Microsoft Custom Vision API.

Orig. Task to New Task	q	# of query	Accuracy	Cost
Traffic sign classification	1	1.86k	48.15%	$3.72
to	5	5.58k	62.34%	$11.16
ASD	10	10.23k	**67.80%**	$20.46

Clarifai Moderation API. This API can recognize whether images or videos have contents such as "gore", "drugs", "explicit nudity", or "suggestive nudity". It also has a class called "safe", meaning it does not contain the aforementioned four moderation categories. Therefore, in total there are 5 output class labels for this API.

Clarifai Not Safe For Work (NSFW) API. This API can recognize images or videos with inappropriate contents (e.g., "porn", "sex", or "nudity"). It provides the prediction of two output labels "NSFW" and "SFW".

Here, we separate the ASD dataset into 930/104 and the DR dataset into 1500/2400 for training and testing, and in BAR we use random label mapping instead of frequency mapping to avoid extra query cost. The test accuracy, total number of queries and the expenses of reprogramming Clarifai.com are reported in Table 4. For instance, to achieve 67.32% accuracy for ASD task and 72.75% for DR task, BAR only costs $23.04 US dollars and $31.68 for reprogramming the Clarifai Moderation API. Setting a larger q value for a more accurate gradient estimation can indeed improve the accuracy but at the price of increased query and expense costs. We expect the accuracy of BAR can be further enhanced if we use frequency-based multi-label mapping or reprogram prediction APIs with more source labels.

Microsoft Custom Vision API. We use this API to obtain a black-box traffic sign image recognition model (with 43 classes) trained with GTSRB dataset (Stallkamp et al., 2012). We then apply BAR with different number of random vectors q (1/5/10) and a fixed number of random label mapping $m = 6$ to reprogram it for ASD task. As shown in Table 5, the test accuracy achieves 69.15% when q is set to 10 and the overall query cost is $20.46 US dollars.

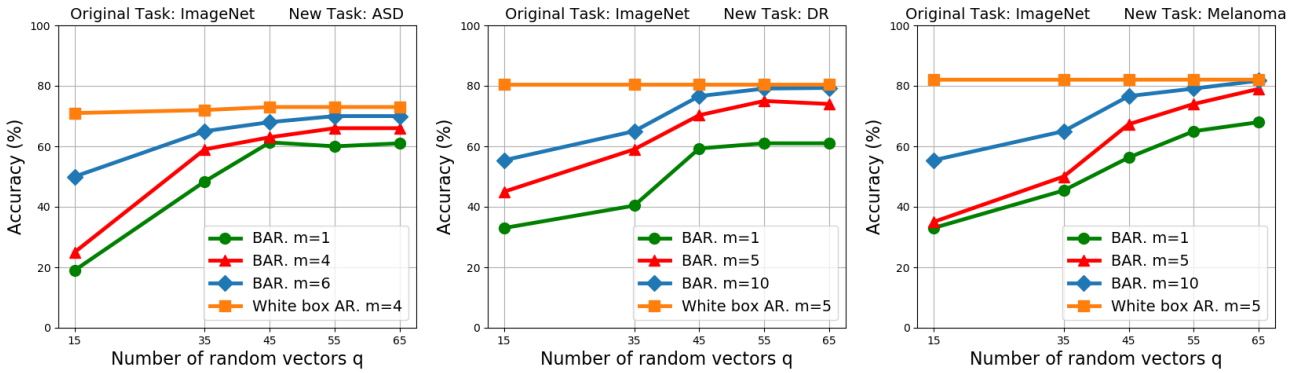

Figure 2. Sensitivity analysis of BAR on the number of random vectors q for gradient estimation and the frequency-based multi-label mapping size m. The white-box AR is shown as a reference. The accuracy of BAR improves as q or m increases.

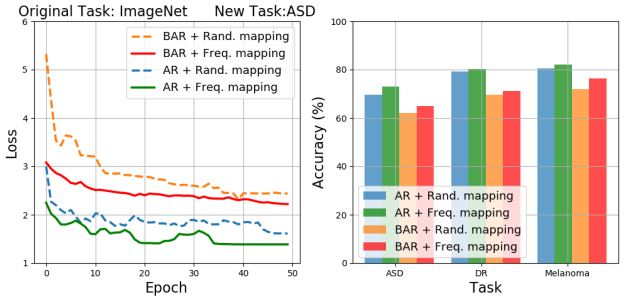

Figure 3. Ablation study on random and frequency multi-label mapping for AR and BAR. With frequency mapping, the accuracy and training performance of AR and BAR can be further improved. Left: training loss over epochs for ASD task. Right: Test accuracy upon convergence for each task.

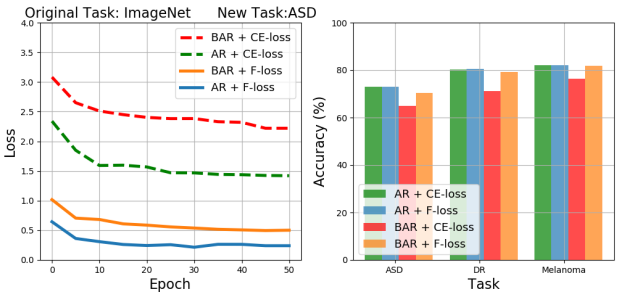

Figure 4. Analysis on cross entropy (CE-loss) and focal loss (F-loss) for AR and BAR in each task. By using F-loss, the loss converges faster than using CE-loss for both AR and BAR methods. Using F-loss, the performance of BAR can be significantly improved in each task.

4.5. Ablation Study and Sensitivity Analysis

Number of random vectors (q) and mapping size (m). In our BAR method, we use the one-sided averaged gradient estimator in (3) via q random vector perturbations. Here, we empirically investigate the sensitivity of q and m on the accuracy of BAR when reprogramming the pretrained Resnet 50 ImageNet model to perform ASD classification, DR detection, and Melanoma detection with different q and m values. As shown in Figure 2, the test accuracy of BAR is low when $q = 15$, suggesting that insufficient gradient estimation will undermine the performance. On the other hand, the accuracy indeed increases with q and then saturates for different mapping sizes. For a fixed q number, we can conclude that increasing the label mapping size m for each target-domain label can improve the accuracy.

Random and frequency multi-label mapping. We perform an ablation study on two multiple-label mapping schemes – random mapping and frequency mapping – for both AR and BAR on Resnet 50 with the three medical imaging learning tasks. For random mapping, for each target-domain class we randomly assign m separate labels from the source domain. For frequency mapping, in each task, we first obtain the source-label prediction distribution of the target-domain data before reprogramming. Based on the distribution, we then sequentially assign the most frequent source-label to the corresponding dominating target-label until each target-label has been assigned with m source-labels. Figure 3 shows the training loss over training epochs (left diagram) on ASD and the resulting test accuracy (right diagram) upon convergence for all tasks. Comparing to random mapping, we find that frequency mapping leads to faster and better convergence results for both AR and BAR, thereby yielding roughly 3% to 5% gain in test accuracy. Similar trends in convergence are observed in DR and Melanoma detection tasks.

Cross entropy loss (CE-loss) and focal loss (F-loss). Here we compare the performance of AR and BAR using CE-loss and F-loss. As shown in Figure 4 (left diagram), on ASD we find that using F-loss can converge faster and better than using CE-loss for both AR and BAR. Similar observations are made for DR and Melanoma tasks. The performance gain when using F-loss can be explained by the fact that it is designed for improving dense object detection, with the capability of better differentiating foreground-background

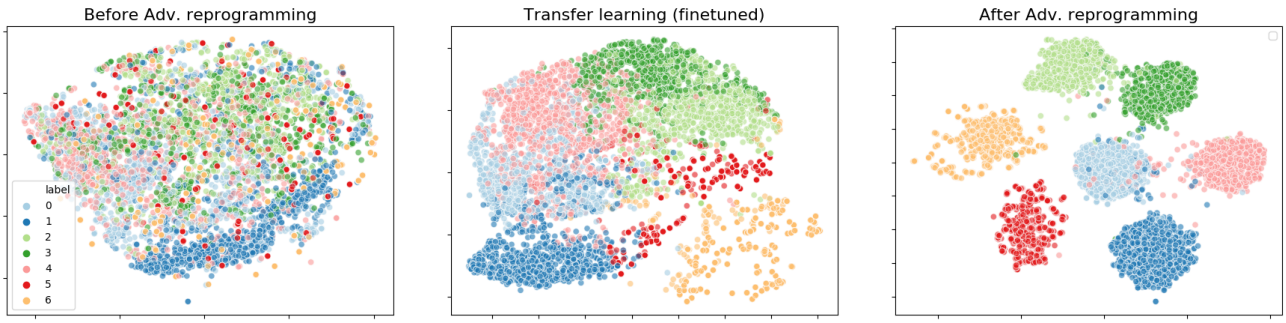

Figure 5. Comparison of t-SNE embedding plots using ResNet 50 and the training data of Melanoma Detection task. Colors represent 7 different class labels. Adversarial reprogramming indeed learns better data representations for solving the target-domain task.

variances, which well maps to our AR setting (foreground being the embedded target-domain data and background being the learned universal adversarial program). Comparing the test accuracy on different tasks (right diagram), we find that F-loss greatly improves the accuracy of BAR by 3%-5%, while the gain in AR's accuracy can be marginal.

Representation analysis. To validate that AR/BAR indeed learns useful data representations for transfer learning, in Figure 5 we visualize the data representations of before/after AR and finetuning using t-distributed stochastic neighbor embedding (t-SNE), where the melanoma training data representations are extracted from the ResNet 50 feature maps of the pre-logit layer. We can observe that before AR, the data representations are non-separable, whereas after AR they become highly clustered and well separated, leading to high predictability. In contrast, finetuning has worse representation learning performance relative to AR. The t-SNE plots of other datasets are shown in the supplementary material.

5. Conclusion

In this paper, we proposed BAR, a novel approach to adversarial reprogramming of black-box ML models via zeroth order optimization and multi-label mapping techniques. Comparing to the vanilla AR method assuming complete knowledge of the target ML model, our BAR method only required input-output model responses, enabling black-box transfer learning of access-limited ML models. Evaluated on three data-scarce medical ML tasks, BAR showed comparable performance to the vanilla white-box AR method and outperformed the respective state-of-the-art methods as well as the widely used finetuning approach. We also demonstrated the practicality and effectiveness of BAR in reprogramming real-life online image classification APIs with affordable expenses, and performed in-depth ablation studies and sensitivity analysis. Our results provide a new perspective and an effective approach for transfer learning without knowing or modifying the pre-trained model.

Acknowledgements

This work was based on a joint study agreement between IBM Research and National Tsing Hua University, Taiwan. The work of Yun-Yun Tsai and Tsung-Yi Ho was supported by the Taiwan Ministry of Science and Technology under Grant MOST 108-2218-E-007-031. Pin-Yu Chen would like to thank Payel Das at IBM Research for her inputs on the autism spectrum disorder classification task.

References

Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402, 2013.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *International Conference on Learning Representations*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017a.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. EAD: elastic-net attacks to deep neural networks via adversarial examples. *AAAI*, 2018.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017b.

Cheng, M., Le, T., Chen, P.-Y., Yi, J., Zhang, H., and Hsieh, C.-J. Query-efficient hard-label black-box attack: An optimization-based approach. *International Conference on Learning Representations*, 2019.

Cheng, M., Singh, S., Chen, P. H., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-OPT: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020.

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Craddock, C., Benhajali, Y., Chu, C., Chouinard, F., Evans, A., Jakab, A., Khundrakpam, B. S., Lewis, J. D., Li, Q., Milham, M., Yan, C., and Bellec, P. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, (41), 2013.

Elsayed, G. F., Goodfellow, I., and Sohl-Dickstein, J. Adversarial reprogramming of neural networks. In *International Conference on Learning Representations*, 2019.

Eslami, T., Mirjalili, V., Fong, A., Laird, A. R., and Saeed, F. Asd-diagnet: A hybrid learning approach for detection of autism spectrum disorder using fmri data. *Frontiers in Neuroinformatics*, 13, Nov 2019.

Gao, X., Jiang, B., and Zhang, S. On the information-adaptive variants of the admm: an iteration complexity perspective. *Optimization Online*, 12, 2014.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Heinsfeld, A. S., Franco, A. R., Craddock, R. C., Buchweitz, A., and Meneguzzi, F. Identification of autism spectrum disorder using deep learning and the abide dataset. In *NeuroImage: Clinical*, 2018.

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., and Keutzer, K. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. *International Conference on International Conference on Machine Learning*, 2018.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Li, K. M. and Li, E. C. Skin lesion analysis towards melanoma detection via end-to-end deep learning of convolutional neural networks. *arXiv preprint arXiv:1807.08332*, 2018.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., and Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 3731–3741, 2018.

Liu, S., Chen, P.-Y., Chen, X., and Hong, M. signsgd via zeroth-order oracle. *International Conference on Learning Representations*, 2019.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero, A., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning. *IEEE Signal Processing Magazine*, 2020.

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. In *ACM Workshop on Artificial Intelligence and Security*, pp. 27–38, 2017.

Neekhara, P., Hussain, S., Dubnov, S., and Koushanfar, F. Adversarial reprogramming of text classification neural networks. *EMNLP*, 2019.

N.Silberman and S.Guadarrama. Tensorflow-slim image classification model library. 2016. URL https://github.com/tensorflow/models/tree/master/research/slim.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems*, pp. 3342–3352, 2019.

Sarki, R., Michalska, S., Ahmed, K., Wang, H., and Zhang, Y. Convolutional neural networks for mild diabetic retinopathy detection: an experimental study. *bioRxiv*, pp. 763136, 2019.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pp. 6103–6113, 2018.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0): –, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016. URL http://www.sciencedirect.com/science/article/pii/S0893608012000457.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Tschandl, P., Rosendahl, C., and Kittler, H. The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, 03 2018. doi: 10.1038/sdata.2018.161.

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. *AAAI*, 2019.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.