Zero-Shot Knowledge Distillation from a Decision-Based Black-Box Model

Zi Wang [1]

Abstract

Knowledge distillation (KD) is a successful approach for deep neural network acceleration, with which a compact network (student) is trained by mimicking the softmax output of a pre-trained high-capacity network (teacher). In tradition, KD usually relies on access to the training samples and the parameters of the white-box teacher to acquire the transferred knowledge. However, these prerequisites are not always realistic due to storage costs or privacy issues in real-world applications. Here we propose the concept of decision-based black-box (DB3) knowledge distillation, with which the student is trained by distilling the knowledge from a black-box teacher (parameters are not accessible) that only returns classes rather than softmax outputs. We start with the scenario when the training set is accessible. We represent a sample's robustness against other classes by computing its distances to the teacher's decision boundaries and use it to construct the soft label for each training sample. After that, the student can be trained via standard KD. We then extend this approach to a more challenging scenario in which even accessing the training data is not feasible. We propose to generate pseudo samples that are distinguished by the decision boundaries of the DB3 teacher to the largest extent and construct soft labels for these samples, which are used as the transfer set. We evaluate our approaches on various benchmark networks and datasets and experiment results demonstrate their effectiveness.

1. Introduction

Training compact deep neural networks (DNNs) (Howard et al., 2017) efficiently has become an appealing topic because of the increasing demand for deploying DNNs on resource-limited devices such as mobile phones and drones (Moskalenko et al., 2018). Recently, a large number of approaches have been proposed for training lightweight DNNs with the help of a cumbersome, over-parameterized model, such as network pruning (Li et al., 2016; He et al., 2019; Wang et al., 2021), quantization (Han et al., 2015), factorization (Jaderberg et al., 2014), and knowledge distillation (KD) (Hinton et al., 2015; Phuong & Lampert, 2019; Jin et al., 2020; Yun et al., 2020; Passalis et al., 2020; Wang, 2021). Among all these approaches, knowledge distillation is a popular scheme with which a compact student network is trained by mimicking the softmax output (class probabilities) of a pre-trained deeper and wider teacher model (Hinton et al., 2015). By doing so, the rich information learned by the powerful teacher can be imitated by the student, which often exhibits better performance than solely training the student with a cross-entropy loss. Many variants have been developed to improve the vanilla KD approach by not only mimicking the softmax output but also matching extra elements in the teacher.

The success of KD relies on three factors: (1) access to the teacher's training dataset, (2) the white-box teacher model, i.e., access to the teacher's parameters, and (3) the score-based outputs, i.e., class probabilities of the training samples outputted by the teacher. In real-world applications, however, these prerequisites are usually unrealistic. Due to storage costs of large training datasets (such as ImageNet (Deng et al., 2009)) or privacy issues (such as sensitive patient data or personal photos), accessing the training samples are sometimes not feasible. With this concern, the concept of zero-shot knowledge distillation (ZSKD) (Nayak et al., 2019; Chen et al., 2019; Yin et al., 2020; Wang, 2021) is proposed. ZSKD generates pseudo training samples via backpropagation with access to the parameters of the white-box teacher, which are then used as the transfer set for training the student model via KD. However, we argue that this scenario is still not realistic under certain circumstances.

In some cases, training samples are publicly available, but pre-trained models are not. For example, YouTube's recommendation system (Covington et al., 2016) is trained with tons of videos that can be accessed by any user. However, the trained model is a core competitiveness of the company and its parameters are not released. One can argue that a surrogate teacher can be trained locally with the accessible

[1]Department of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, TN, USA. Correspondence to: Zi Wang <zwang84@vols.utk.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

training set, but due to the limitations such as computing resources, its performance is usually not satisfactory compared to the provided powerful model with much more parameters and complicated architectures.

Moreover, a much more challenging scenario is that, in many real-world applications, none of the three factors mentioned above is available. A pre-trained model stored on the remote server may only provide APIs for inference, neither the model parameters nor the training samples are accessible to the users. Worse than that, these APIs usually return a category index for each sample (i.e., hard-label), rather than the class probabilities over all classes. For example, speech recognition systems like Siri and Cortana are trained with internal datasets and only return the results to users (López et al., 2017). Cloud-based object classification systems like Clarifai (Clarifai, 2020) just give the top-1 classes of the identified objects in the images uploaded by users.

With these concerns, we propose the concept of decision-based black-box knowledge distillation (DB3KD), i.e., training a student model by transferring the knowledge from a black-box teacher that only returns hard-labels rather than probability distributions. We start with the scenario when the training data is available. Our key idea is to extract the class probabilities of the training samples from the DB3 teacher. We claim that the decision boundary of a well-trained model distinguishes the training samples of different classes to the largest extent. Therefore, the distance from a sample to the targeted decision boundary (the boundary to the samples of a certain class) can be used as a representation of a sample's robustness, which determines how much confidence of a specific class is assigned to the sample. Based on this, the soft label of each training sample can be constructed with the value of sample robustness and used for training the student via KD.

We further extend DB3KD to the scenario when training data are not accessible. As the decision boundary makes every effort to differentiate the training samples of all classes, samples used for training the teacher tend to be with longer distances to the boundary than others. We propose to optimize randomly generated noises away from the boundary to obtain robust pseudo samples that simulate the distribution of the training samples. This is achieved by iteratively estimating the gradient direction on the boundary and pushing the samples away from the boundary in that direction. After that, pseudo samples are used for training the student via DB3KD. To our best knowledge, this is the first study of KD from a DB3 teacher, both with and without access to the training set.

The contribution of this study is summarized as follows. (1) We propose the concept of decision-based black-box knowledge distillation for the first time, with which a student is trained by transferring knowledge from a black-box teacher that only returns hard-labels. (2) We propose to use sample robustness, i.e., the distance from a training sample to the decision boundaries of a DB3 teacher, to construct soft labels for DB3KD when training data is available. (3) We extend the DB3KD approach to a more challenging scenario when accessing training data is not feasible and name it zero-shot decision-based black-box knowledge distillation (ZSDB3KD). (4) Extensive experiments validate that the proposed approaches achieve competitive performance compared to existing KD methods in more relaxed scenarios.

2. Related Work

Knowledge distillation. Knowledge distillation is first introduced in (Buciluǎ et al., 2006) and generalized in (Ba & Caruana, 2014; Hinton et al., 2015), which is a popular network compression scheme to train a compact student network by mimicking the softmax output predicted by a high-capacity teacher or ensemble of models. Besides transferring the knowledge of class probabilities, many variants have been proposed to add extra regulations or alignments between the teacher and the student to improve the performance (Romero et al., 2014; Yim et al., 2017; Kim et al., 2018; Heo et al., 2019). For example, FitNet (Romero et al., 2014) introduces an extra loss term that matches the values of the intermediate hidden layers of the teacher and the student, which allows fast training of deeper student models. (Zagoruyko & Komodakis, 2016) defines the attention of DNNs and uses it as the additional transferred knowledge.

Knowledge distillation with limited data. To mitigate the storage and transmission costs of large training datasets, several studies propose the concept of few-shot KD, which generates pseudo samples with the help of a small number of the original training samples (Kimura et al., 2018; Wang et al., 2020; Li et al., 2020). Another study suggests that instead of the raw data, some surrogates with much smaller sizes (also known as metadata) can be used to distill the knowledge from the teacher. (Lopes et al., 2017) leverages the statistical features of the activations of the teacher to train a compact student without access to the original data. However, releasing this kind of metadata along with the pre-trained teacher is usually not a common scenario.

Zero-shot knowledge distillation. To deal with the scenario when training data is not accessible, (Nayak et al., 2019) proposes zero-shot knowledge distillation (ZSKD). The authors model the softmax output space of the teacher with a Dirichlet distribution and samples soft labels as the targets. Randomly generated noise inputs are optimized towards these targets via backpropagation and are used as the transfer set. (Wang, 2021) replaces the Dirichlet distribution with a multivariate normal distribution to model the softmax output space of the generated samples. Therefore, pseudo samples of different classes can be generated simul-

taneously rather than one after another as in (Nayak et al., 2019). Generative adversarial networks (GANs) (Goodfellow et al., 2014) are leveraged in (Chen et al., 2019; Micaelli & Storkey, 2019) to solve this task so that pseudo sample synthesis and student network training can be conducted simultaneously. Another study (Yin et al., 2020) proposes to use the features in the batch normalization layers to generate pseudo samples. However, these methods still need access to the parameters of the teacher for backpropagation, which is unrealistic in many cases.

Black-box knowledge distillation. Although the vanilla KD is built with a black-box teacher (Hinton et al., 2015), the whole training dataset is used for training. (Wang et al., 2020) investigates the possibility that a student is trained with limited samples and a black-box teacher. Other than zero-shot KD methods that generate pseudo inputs, (Orekondy et al., 2019) proposes to sample from a large pool (such as ImageNet) to get the transfer set to train the student. Therefore, there is no need to access the teacher's parameters. Although the prerequisites in these methods are relaxed, weak assumptions on the training samples and a score-based teacher that outputs class probabilities are still needed. Different from these studies, we consider a much more challenging case in which knowledge is transferred from a black-box teacher that only returns top-1 classes.

Decision-based adversarial attack. Our approach leverages the distance from a sample to the decision boundary for soft label construction, which is related to the research of decision-based black-box adversarial attack (Brendel et al., 2017; Cheng et al., 2018; 2019; Liu et al., 2019). These methods aim to add some imperceptible perturbations to the inputs to create adversarial samples that fool a well-trained DNN with high confidence. This is achieved by identifying the points on the decision boundary with minimal distance to the original inputs. Inspired by these studies, we use the distance from a sample to the targeted decision boundaries as a representation of a sample's robustness against other categories, which can be converted to a probability distribution of all classes with proper operations.

3. Methodology

We first formulate KD in its standard form and present our approach that creates soft labels of the training samples with a DB3 teacher. Finally, we extend our approach to the scenario in which the training set is not accessible.

3.1. Knowledge Distillation

KD is used for training a compact student by matching the softmax outputs of a pre-trained, cumbersome teacher (Hinton et al., 2015) (Fig. 1(left)). For an object classification task, denote $F_t(x)$ and $F_s(x)$ the teacher and the student DNNs, respectively, which take an image x as the input, and output a vector $P \in [0,1]^L$, i.e., $F_t(x) = P_t = \text{softmax}(a_t)$, $F_s(x) = P_s = \text{softmax}(a_s)$, where L is the number of classes and a is the pre-softmax activation. In a KD procedure, a temperature τ is usually introduced to soften the softmax output, i.e., $P^\tau = \text{softmax}(a/\tau)$, which is proved to be efficient to boost the training process. The student is trained by minimizing the loss function in Eq. (1).

$$\mathcal{L} = \mathcal{L}_{CE}(P_s, y) + \lambda \mathcal{L}_{KD}(P_t^\tau, P_s^\tau), \tag{1}$$

where y is the ground truth label, $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$ are the cross-entropy loss and the distillation loss. A scaling factor λ is used for balancing the importance of the two losses.

3.2. Decision-Based Black-Box Knowledge Distillation

As mentioned, in many real-world applications, users are prohibited from querying any internal configuration of the teacher except for the final decision (top-1 label). Denote $F_t^B(x)$ the DB3 teacher, then $F_t^B(x) = l, l \in \{1, 2, \cdots, L\}$. In this case, P_t cannot be obtained and the student cannot be trained with Eq. (1). We claim that a sample's robustness against a specific class can be used as a representation of how much confidence should be assigned to this class, with proper post-operations. Therefore, we extract the sample's robustness against each class from the DB3 teacher and convert it to a class distribution $\hat{P}_t$ as an estimate of P_t (Fig. 1(bottom)). In the following, we propose three metrics to measure sample robustness and present how to construct class distributions with the sample robustness measurements. Intuitively, if a sample is closer to some points in the region of a specific class, it is more vulnerable to this class and thus should be assigned higher confidence.

3.2.1. Sample Robustness

Sample Distance (SD). The most straightforward way to quantify the sample robustness is to compute the minimal ℓ_2-norm distance from a sample to those of other classes (Fig. 2(left)). Denote $x_0^m \in \mathbb{R}^{C\times W\times H}$ a sample of the m-th class, $\mathbf{x}^n = \{x_1^n, x_2^n, \cdots, x_S^n\}$ a batch of S samples from the n-th class, where $n \neq m$, C, W, H are the number of channels, width and height of the sample, respectively. The robustness of x_0^m against class n is computed with Eq. (2).

$$r_0^{m,n} = \min_{1\leq i\leq S} ||x_i^n - x_0^m||_2. \tag{2}$$

The advantage of using SD is it can be implemented without querying from the teacher. However, SD is a rough estimate of sample robustness since it does not mine any information from the teacher. Therefore, we introduce two advanced strategies to measure sample robustness.

Boundary Distance (BD). To obtain better representation of sample robustness, we propose to leverage the distances

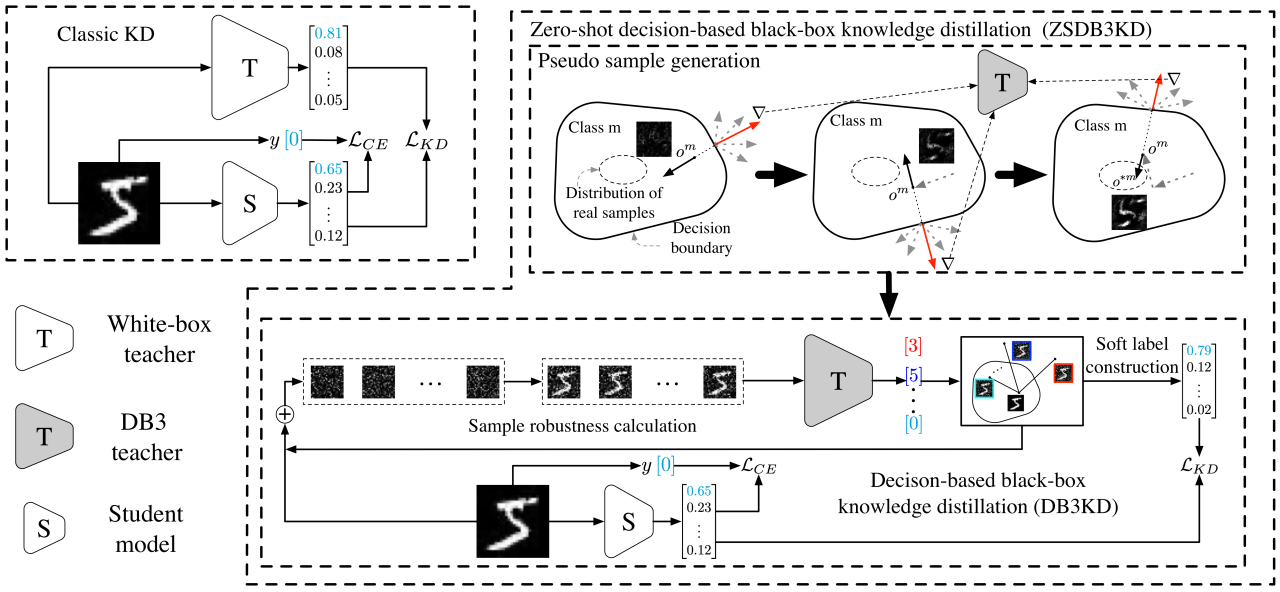


Figure 1. The overall workflow of the proposed approach. Left: classic KD. Bottom: decision-based black-box KD (DB3KD). Samples are iteratively fed to the DB3 teacher to compute the sample robustness, which is transformed as soft labels for training the student via KD. Right: Zero-shot DB3KD (ZSDB3KD). Pseudo samples are generated by moving random noises away from the decision boundary and approaching the distribution of the original training samples, which are used as the transfer set for training the student via DB3KD.

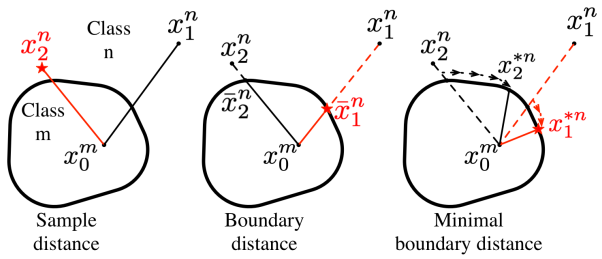


Figure 2. Strategies for computing sample robustness.

from a sample to the targeted decision boundaries of the teacher (Fig. 2(middle)). For each $x_i^n \in \mathbf{x}^n$, we implement a binary search in the direction $(x_i^n - x_0^m)$ and find the corresponding point $\bar{x}_i^n$ on the decision boundary (Eq. (3)).

$$\bar{x}_i^n = \min_{\alpha}(x_0^m + \alpha \cdot \frac{x_i^n - x_0^m}{||x_i^n - x_0^m||_2}), i = 1, 2, \cdots, S, \quad \text{s.t. } F_t^B(\bar{x}_i^n + \epsilon) = n, \quad ||\epsilon||_2 \to 0. \tag{3}$$

We then compute the sample robustness with Eq. (2) in which x_i^n is replaced by $\bar{x}_i^n$.

Minimal Boundary Distance (MBD). Inspired by recent studies of decision-based black-box adversarial attack (Brendel et al., 2017; Cheng et al., 2018; Liu et al., 2019; Cheng et al., 2019), we further optimize $\bar{x}_i^n$ by moving it along the decision boundary to the point x_i^{*n} where $||x_i^{*n} - x_0^m||_2$ is minimized (Fig. 2(right)). Starting from $\bar{x}_i^n$, we first estimate the gradient of the boundary $\nabla F_t^B(\bar{x}_i^n)$ via zeroth order optimization (Wang et al., 2018), which is achieved by sampling Q Gaussian random vectors $\mathbf{u}_q \in \mathbb{R}^{C \times W \times H}$ $(q = 1, 2, \cdots, Q)$ and averaging them (Fig. 3, Eq. (4)).

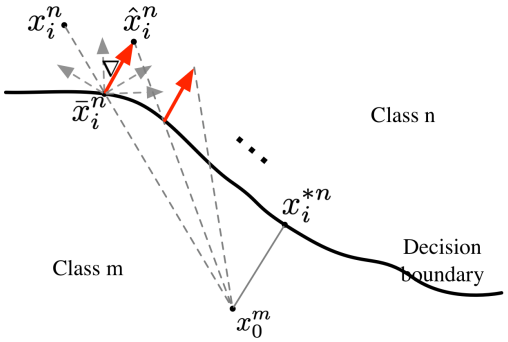


Figure 3. The iterative procedure for the optimization of MBD.

$$\nabla F_t^B(\bar{x}_i^n) = \frac{1}{Q}\sum_{q=1}^{Q} \text{sign}(\bar{x}_i^n + \epsilon_g \mathbf{u}_q)\mathbf{u}_q, \tag{4}$$

where ϵ_g is a very small scalar, and $\text{sign}(x_i^n + \epsilon_g \mathbf{u}_q)$ is a sign function, i.e,

$$\text{sign}(x_i^n + \epsilon_g \mathbf{u}_q) = \begin{cases} +1, & F_t^B(\bar{x}_i^n + \epsilon_g \mathbf{u}_q) = n, \\ -1, & \text{Otherwise.} \end{cases} \tag{5}$$

Once the gradient is determined, we get a new sample outside the decision boundary $\hat{x}_i^n \leftarrow \bar{x}_i^n + \xi_d \nabla F_t^B(\bar{x}_i^n)$ with a step size ξ_d. Then we conduct the same binary search procedure (Eq. (3)) in the direction $(\hat{x}_i^n - x_0^m)$ and obtain an updated $\bar{x}_i^n$. Since the search is within a very small region, the decision boundary in such a region is smooth. Therefore, the new $\bar{x}_i^n$ has a smaller distance to x_0^m (Fig. 3). We repeat the procedure above to get the optimal solution $x_i^{*n} = \bar{x}_i^n$

until $||\bar{x}_i^n - x_0^m||_2$ cannot be further minimized or the query limit is reached. Finally, we compute the sample robustness with Eq. (2) in which x_i^n is replaced by x_i^{*n}.

3.2.2. Soft Label Construction

After obtaining all the samples' robustness on all classes, we construct the soft labels for them with proper manipulations. We start with the pre-softmax activations for better illustration. Suppose the pre-softmax activation of a sample x_s^m is $\mathbf{a_s^m} = \{a_{s,1}^m, a_{s,2}^m, \cdots, a_{s,L}^m\}$. Then the pre-softmax activation and the sample robustness should be in correlation with the following conditions. (1) $\text{argmax}_i a_{s,i}^m = m$. It is obvious that $a_{s,m}^m$ should be the largest number to ensure that the sample is assigned to the correct class. (2) If $r_s^{m,j} > r_s^{m,k}$, then $a_{s,j}^m < a_{s,k}^m$. This is because bigger sample robustness indicates longer distance to the targeted decision boundary, which means that the sample is more robust against the certain class and should be assigned a lower confidence. (3) If $\sum_{j=1}^L r_s^{m,j} > \sum_{j=1}^L r_p^{m,j}, j \neq m$, then $a_{s,m}^m > a_{p,m}^m$. This is because when the sum of a sample's distances to its targeted decision boundaries is larger, the probability mass of this sample is more concentrated in its top-1 class. Otherwise, the mass is more dispersed among all elements.

With the above design philosophy, to meet requirement (1) and (2), we define $\hat{a}_{s,n}^m (n = 1, 2, \cdots, L)$ in Eq. (6).

$$\hat{a}_{s,n}^m = \begin{cases} \frac{1}{r_s^{m,n}}, & \text{for } n \neq m, \\ \sum_{i=1}^L \frac{1}{r_s^{m,i}}, i \neq m, & \text{for } n = m. \end{cases} \tag{6}$$

$\hat{a}_{s,n}^m$ is then divided by $(\sum_{i=1}^L \frac{1}{r_s^{m,i}})^2$ to meet requirement (3), as presented in Eq. (7).

$$a_{s,n}^m = \frac{\hat{a}_{s,n}^m}{(\sum_{i=1}^L \frac{1}{r_s^{m,i}})^2}, \ i \neq m, \text{ for } n = 1, 2, \cdots, L. \tag{7}$$

Finally, we get $\hat{P}_t = \text{softmax}(\mathbf{a}_s^m)$ for sample x_s^m.

3.2.3. Training of Student Model

Once the soft labels of all the training samples are constructed with the above approach, we can train the student with standard KD, using the objective function in Eq. (1).

3.3. Zero-shot Decision-Based Black-Box Knowledge Distillation

In zero-shot KD, pseudo samples are usually generated by optimizing some noise inputs via backpropagation towards some soft labels sampled from a prior distribution, which are then used as the transfer set. However, with a DB3 teacher, backpropagation cannot be implemented and the prior distribution cannot be obtained, which makes ZSDB3KD a much more challenging task. Since the teacher is trained to largely distinguish the training samples, the distance between a training sample to the teacher's decision boundary is usually much larger than the distance between a randomly generated noise image to the boundary. With this claim, we propose to iteratively push random noise inputs towards the region that is away from the boundary to simulate the distribution of the original training data (Fig. 1(right)).

Denote o_0^m and $\mathbf{o}^{\bar{m}} = [o_1^{\bar{m}}, o_2^{\bar{m}}, \cdots, o_T^{\bar{m}}]$ a random noise input of the m-th class and a batch of T random noises with any other class, respectively. Similar but slightly different from Eq. (3), for each $o_i^{\bar{m}} \in \mathbf{o}^{\bar{m}}$, we first identity its corresponding points on the boundary $\bar{o}_i^m$ with Eq. (8).

$$\begin{aligned} \bar{o}_i^m = \min_{\alpha} (o_0^m + \alpha \cdot \frac{o_i^{\bar{m}} - o_0^m}{||o_i^{\bar{m}} - o_0^m||_2}), i = 1, 2, \cdots, T, \\ \text{s.t. } F_t^B(\bar{o}_i^m + \epsilon) \neq m, \quad ||\epsilon||_2 \to 0. \end{aligned} \tag{8}$$

Similarly, the MBDs of o_0^m, i.e., o_i^{*m}, can be iteratively estimated with Eq. (4) and (5). Let o^{*m} be the one of o_i^{*m} $(i = 1, 2, \cdots, T)$ such that $||o^{*m} - o_0^m||_2$ attains its minimal value, i.e., $||o^{*m} - o_0^m||_2 = \min_i ||o_i^{*m} - o_0^m||_2$. We then estimate the gradient at the boundary $\nabla F_t^B(o^{*m})$ with Eq. (4) and update o^m as $o^m \leftarrow o^m - \xi_o \nabla F_t^B(o^{*m})$ with the step size ξ_o. The new o^m is usually with longer distance to the boundary. We repeat the above process until $||o^{*m} - o^m||_2$ cannot be further maximized or the query limit is reached. Finally, we used the generated pseudo samples with the DB3KD approach to train the student as described in Section 3.2.

4. Experiments

In this section, we first demonstrate the performance of DB3KD when training samples are accessible. Then we show the results of ZSDB3KD under the circumstance that training data is not accessible.

4.1. Experiment Setup of DB3KD

We demonstrate the effectiveness of DB3KD with several widely used DNNs and datasets as follows. (1) A LeNet-5 (LeCun et al., 1998) with two convolutional layers is pre-trained on MNIST (LeCun et al., 1998) as the teacher, following the configurations in (Lopes et al., 2017; Chen et al., 2019). A LeNet-5-Half and a LeNet-5-1/5 are designed as the student networks, which contains half and 1/5 number of convolutional filters in each layer compared to LeNet-5, respectively. (2) The same teacher and student networks as in (1) are used but are trained and evaluated on the Fashion-MNIST dataset. (3) An AlexNet (Krizhevsky et al., 2012) pre-trained on CIFAR-10 (Krizhevsky et al., 2009) is used as the teacher. An AlexNet-Half and an AlexNet-Quarter with half and 25% filters are used as student networks. (4) A ResNet-34 (He et al., 2016) pre-trained on the

Algorithm	MNIST		Fashion-MNIST		CIFAR10		FLOWERS102
	LeNet5 -half	LeNet5 -1/5	LeNet5 -half	LeNet5 -1/5	AlexNet -half	AlexNet -quarter	ResNet-18
Teacher CE	99.33%	99.33%	91.63%	91.63%	79.30%	79.30%	95.07%
Student CE	99.11%	98.77%	90.21%	88.75%	77.28%	72.21%	92.18%
Standard KD	99.33%	99.12%	90.82%	89.09%	77.81%	73.14%	94.05%
Surrogate KD	99.13%	98.85%	90.27%	88.72%	77.49%	72.49%	92.93%
Noise logits	99.01%	98.72%	89.81%	88.20%	77.04%	72.06%	91.99%
DB3KD-SD	99.15%	98.98%	90.86%	89.31%	77.66%	72.78%	93.18%
DB3KD-BD	99.51%	99.19%	90.68%	89.47%	77.92%	72.94%	93.30%
DB3KD-MBD	99.52%	99.22%	91.45%	89.80%	78.30%	73.78%	93.77%

Table 1. Performance evaluation of the proposed DB3KD approach.

high-resolution, fine-grained dataset FLOWERS102 (Nilsback & Zisserman, 2008) is used as the teacher, and the student is a ResNet-18.

We evaluate our approach with the three strategies for sample robustness calculation as described in Section 3.2.1, represented as DB3KD-SD, DB3KD-BD, and DB3KD-MBD, respectively. For DB3KD-SD, we use 100 samples from each class to compute the sample robustness r for MNIST, Fashion-MNIST, and CIFAR-10. Since there are only 20 samples in each class of FLOWERS102, we use all of them. Starting with these samples, ϵ is set to $1e^{-5}$ as the stop condition of the binary search in DB3KD-BD. In DB3KD-MBD, we use 200 Gaussian random vectors to estimate the gradient and try different numbers of queries from 1000 to 20000 with $\xi_d = 0.2$ to optimize the MBD and report the best test accuracies. The sample robustness are calculated in parallel with a batch size of 20 with FLOWERS102, and 200 with the other datasets.

With the constructed soft labels, we train the student networks for 100 epochs, using an Adam optimizer (learning rate $5e^{-3}$), for all the datasets except for FLOWERS102, which is trained for 200 epochs. The scaling factor λ is set to 1 for simplicity. Since Eq. (7) has the similar functionality with the temperature τ, τ is not need to be as large as in previous studies (Hinton et al., 2015).With a hyperparameter search, we find that smaller τs between 0.2 and 1.0 leads to good performance. We use $\tau = 0.3$ in our experiments. All experiments are evaluated for 5 runs with random seeds.

4.2. Performance Evaluation of DB3KD

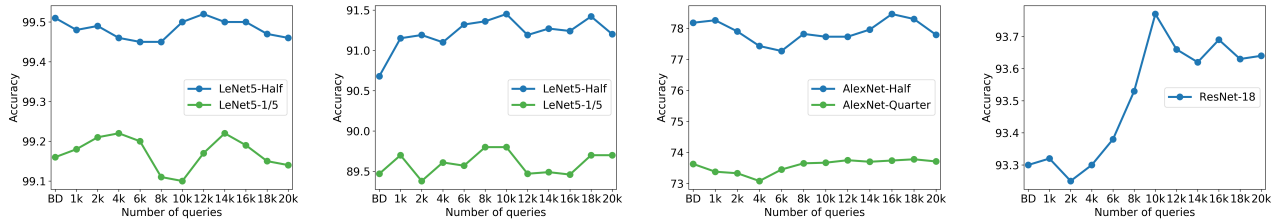
The performance of DB3KD is presented in Table 1. To understand the proposed approach better, we also present the performance of the following training strategies. (1) The teacher and the student networks trained solely with the cross-entropy loss. (2) The standard KD with Eq. (1) (Hinton et al., 2015). (3) Training the student network via KD with a surrogate white-box teacher (Surrogate KD in Table 1), which is used for simulating the scenario in which one can train a smaller but affordable surrogate model with full access to its parameters compared to the powerful DB3 teacher. Here the surrogate has the same architecture with the student. The performance of surrogate KD is considered as the lower bound of DB3KD. (4) Training with the soft labels constructed with randomly generated sample robustness (Noise logits in Table 1), which is used for verifying the effectiveness of DB3KD for soft label construction.

Approach	Teacher	Student	Accuracy
Cross-entropy	ResNet-34	-	78.63%
Cross-entropy	ResNet-18	-	75.91%
Standard KD	ResNet-34	ResNet-18	77.18%
Surrogate KD			76.52%
BAN*			76.84%
TF-KD			77.23%
SSKD			76.20%
DB3KD			77.31%
DB3KD	ResNet-50	ResNet-18	78.65%

Table 2. Performance comparison to self-distillation approaches with ResNet on CIFAR-100. $*$ indicates the results are based on our own implementation.

We observe from the results that DB3KD works surprisingly well. With the most straightforward strategy SD, our approach still achieve competitive performance on all experiments compared to standard KD and outperform surrogate KD. When using MBD to compute sample robustness, DB3KD-MBD outperforms standard KD on all the experiments except for FLOWERS102. On FLOWERS102, the performance of DB3KD is slightly worse due to the complexity of the pre-trained teacher model. However, DB3KD still outperforms the surrogate KD with a clear margin. These results validate the effectiveness of DB3KD and indicates that sample robustness with proper post-operation provides an informative representation of a sample's probabilities over all classes and can be used as an alternative to the softmax output when only a DB3 teacher is provided.

We also observe the following phenomena in the experiments. (1) Training with noise logits via KD does not work,



(a) LeNet5-MNIST (b) LeNet5-Fashion-MNIST (c) AlexNet-CIFAR10 (d) ResNet-FLOWERS102

Figure 4. Performance comparison with different numbers of queries for computing sample robustness.

but even results in worse performance than training with cross-entropy. It indicates noise logits cannot capture the distribution of class probabilities, but are even harmful due to the wrong information introduced. (2) Training a student with a surrogate teacher not only results in unsatisfactory performance, but is also a difficult task due to the low capacity of the surrogate model. Also, the performance is sensitive to hyperparameter selection (λ, τ, learning rate, etc.). Therefore, training an extra affordable surrogate teacher is not an optimal solution compared to DB3KD.

We notice that in some experiments, surprisingly, DB3KD even works better than standard KD, though the models are trained with a more challenging setting. A reasonable hypothesis is that, for some problems, the distance between a training sample to the decision boundary may provide more information than the softmax output. These results provide future research directions that the dark knowledge behind the teacher's decision boundary is more instructive compared to the teacher's logits in certain cases.

4.3. Comparison with Self-Distillation Approaches

Similar to our proposed scenario, in the absence of a pre-trained teacher, self-knowledge distillation aims to improve the performance of the student by distilling the knowledge within the network itself (Furlanello et al., 2018). Since self-distillation approaches can also deal with our proposed scenario, we compare the performance of DB3KD to recent self-distillation approaches, including born-again neural networks (BAN) (Furlanello et al., 2018), teacher-free knowledge distillation (TF-KD) (Yuan et al., 2020), and self-supervision knowledge distillation (SSKD) (Xu et al., 2020). We use ResNet-34/18 as the teacher and the student on CIFAR-100 for illustration. For further comparison, we also implement DB3KD with a ResNet-50 teacher.

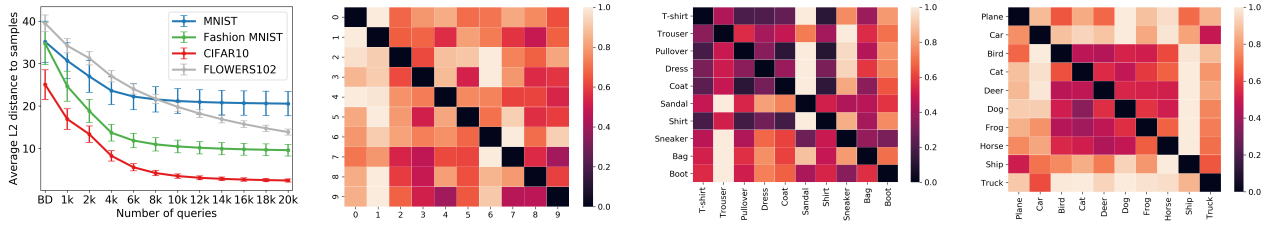
The results are shown in Table 2. It is observed that our approach is still competitive in this case. With the same network configuration, our student achieves a test accuracy of 77.31%, which outperforms other self-distillation approaches, even with a DB3 teacher. It is also worth mentioning that, given a fixed student, the performance of self-distillation has an upper bound because it is teacher-free. One advantage of our approach is that the student can leverage the information from a stronger teacher and its performance can be further improved. As an example, we substitute the DB3 teacher with a ResNet-50 network and keep other other configurations unchanged, the performance of our student network is further increased by 1.34%, which outperforms self-distillation approaches with a clear margin.

4.4. Ablation Studies and Analyses of DB3KD

We conduct several ablation studies and analyses for further understanding of the effectiveness of DB3KD.

Number of queries in label construction. We first investigate whether different numbers of queries used for computing sample robustness has any influence on the performance. For each dataset, we query from the teacher for a variety of times from 1000 to 20000 to compute the sample robustness (Fig. 4). It can be observed that with more queries, the student models perform slightly better, especially for deeper architectures (ResNet) and high-resolution datasets (FLOWERS102). In general, the student models perform well with various numbers of queries. Even using a binary search with around 100 queries (DB3KD-BD), the performance are satisfactory on all student models. This is because the quality of a sample's soft label is largely related to its robustness against different classes. Moreover, the MBD used for computing sample robustness shows a highly positive correlation with the number of queries (Fig. 5(a)). The ratios of sample robustness against different classes remain stable against the number of queries. Therefore, it is not necessary to optimize the MBD with a large number of queries, which indicates that DB3KD is query efficient. It is also worth noting that the performance is not linearly correlated with the query numbers. This is because for all experiments, we use the same set of hyperparameters for fair comparison, which may not be optimal as the query number increases. However, we'd like to emphasize the performance is not sensitive to query numbers and is satisfactory with a wide range of numbers (from 2k to 20k).

Although the boundary may be complex in the pixel domain and the boundary sample may be fragile, what we actually care about is the minimal boundary distance (MBD). It actu-



(a) (b) MNIST (c) Fashion-MNIST (d) CIFAR10

Figure 5. (a) The average minimal boundary distances over number of queries. Error bar indicates one standard deviation. (b-d) Normalized average minimal boundary distances of the samples of different classes. Darker colors indicate smaller distances between two classes.

Algorithm	Data	Model	MNIST	FMNIST
Teacher CE	Yes	White	99.33%	91.63%
Student CE	Yes	White	99.11%	90.21%
Standard KD	Yes	Black-S	99.33%	90.82%
FSKD	Few	White	86.70%	72.60%
BBKD	Few	Black-S	98.74%	80.90%
Meta KD	Meta	White	92.47%	-
DAFL	No	White	98.20%	-
ZSKD	No	White	98.77%	79.62%
DFKD	No	White	99.08%	-
ZSDB3KD	No	Black-D	96.54%	72.31%

Table 3. Result of ZSDB3KD with MNIST and Fashion-MNIST. S: score-based teacher. D: decision-based teacher.

Algorithm	Data	Model	Accuracy
Teacher CE	Yes	White	79.30%
Student CE	Yes	White	77.28%
Standard KD	Yes	Black-S	77.81%
FSKD	Few	White	40.58%
BBKD	Few	Black-S	74.60%
DAFL	No	White	66.38%
ZSKD	No	White	69.56%
DFKD	No	White	73.91%
Noise input	No	Black-S	14.79%
Noise input	No	Black-D	13.53%
ZSDB3KD	No	Black-D	59.46%

Table 4. Result of ZSDB3KD on AlexNet with CIFAR-10.

ally measures how fragile a training sample is against other classes and is a robust measurement. As supplementary evidence, the standard deviations of the MDBs are relatively small (shown with the error bars in Fig. 5(a)), indicating the robustness of the proposed approach.

Correlation between sample robustness and class probability. To further analyze the effectiveness of DB3KD for constructing soft labels, we visualize the normalized average MBDs of the samples with different classes (Fig. 5(b-d)). It is observed that classes semantically closer with each other are with smaller distances to their decision boundary. For example, in MNIST, the distance between '8' and '9' is smaller than '8' and '1' because '8' looks more like '9' than '1'. Therefore, a sample of '8' is assigned higher confidence in class '9' than '1'. Similarly, in Fashion-MNIST, 'T-shirt' looks more like 'shirt' than 'sneaker' so that their distance are smaller. In CIFAR-10, samples of the 'dog' class are with smaller distances to the boundary with 'cat' than 'truck' since 'dog' and 'cat' are semantically closer. These analyses confirm the consistency between sample robustness and class probability distribution.

4.5. Experiment Setup of ZSDB3KD

We evaluate ZSDB3KD with (1) a LeNet-5 and a LeNet-5-Half (on MNIST and Fashion-MNIST), and (2) an AlexNet and an AlexNet-Half (on CIFAR-10) as the teacher and the student. The networks are the same as in Section 4.1.

We optimize the pseudo samples for 40 ($\xi_o = 0.5$) and 100 iterations ($\xi_o = 3.0$) for the two LeNet-5 and the AlexNet experiments, respectively. The query is limited to 5000 when iteratively searching for the MBD. We generate 8000 samples for each class with a batch size of 200 for all the experiments. We use data augmentation to enrich the transfer set (see Appendix). We use 5000 queries for computing the sample robustness since we have shown the number of queries is trivial. Other parameters are the same as the DB3KD experiments. We compare the performance of ZSDB3KD with several popular KD approaches in more relaxed scenarios, including FSKD (Kimura et al., 2018), BBKD (Wang et al., 2020), Meta KD (Lopes et al., 2017), DAFL (Chen et al., 2019), ZSKD (Nayak et al., 2019) and DFKD (Wang, 2021).

4.6. Performance Comparison of ZSDB3KD

The performance of ZSDB3KD on MNIST and Fashion-MNIST, and CIFAR-10 presented in Table 3 and 4 show that ZSDB3KD achieves competitive performance. The accuracies of the student networks are 96.54% and 72.31% on MNIST and Fashion-MNIST, which are quite close to other KD approaches with more relaxed scenarios (training data or the teacher's parameters are accessible). On CIFAR-10, our AlexNet-Half model achieves an accuracy

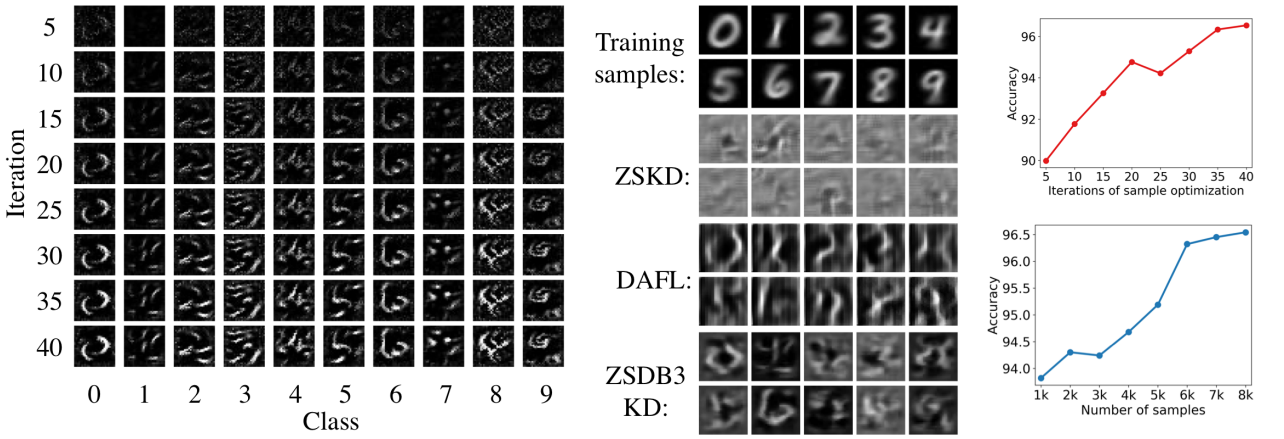


Figure 6. Analysis and ablation study of ZSDB3KD with MNIST. Left: evolution of pseudo images over iterations. Middle: averaged images compared to other white-box zero-shot KD approaches. Upper right: the accuracies with different iterations of sample generation. Bottom right: the accuracies with different numbers of samples used for training the student.

of 59.46% without accessing any training samples and the softmax outputs of the teacher. It is worth noting that using random noise as the input results in very poor performance with a DB3 teacher. These results indicate that the samples generated with our proposed approach indeed capture the distribution of the samples used for training the teachers.

4.7. Ablation Studies and Analyses of ZSDB3KD

In this subsection, we perform several studies to understand the effectiveness of ZSDB3KD, using LeNet-5-Half trained on MNIST as an example.

Iteration of sample generation. We first evaluate the performance of the student with pseudo samples generated with different iterations (Fig. 6(upper right)). As expected, the performance is improved as the samples are optimized away from the decision boundaries with more iterations. As shown in Fig. 6(left), with more steps, more pixels in the pseudo samples are activated, with sharper edges and recognizable digits, which indicates that the samples become more robust as we keep moving them to the opposite of the gradient direction on the decision boundaries.

Number of samples used for training. We then investigate the effect of the number of pseudo samples used for training on the performance of the student network. The results of training the student network with different numbers of generated samples (from 1k to 8k per class) are presented in Fig. 6(bottom right). Not surprisingly, with more samples, the test accuracy increases. Even with a small number of samples (1k per class), the student network can still achieve a competitive performance of 94% test accuracy. With 8k samples per class, the student's performance gets saturated and is comparable to the performance of standard KD.

Visualization of generated samples. As mentioned above, we have shown the evolution of individual samples over iterations (Fig. 6(left)), which gradually exhibits clear digits. To have a further visualization of the generated pseudo samples, we further average 1k samples for each class as shown in Fig. 6(middle). Even though generated with a DB3 teacher, the samples are with a satisfactory quality compared with the averaged samples generated with ZSKD and DAFL that use white-box teachers.

5. Conclusion

In this study, we introduced KD from a decision-based black-box teacher for the first time. We proposed DB3KD to deal with this problem, which uses sample robustness to construct the soft labels for the training samples by iteratively querying from the teacher. We also extend DB3KD to a much more challenging scenario in which the training set is not accessible and named it Zero-shot DB3KD (ZSDB3KD). Experiments on various networks and datasets validated the effectiveness of the proposed approaches.

Our study motivated a new line of research on KD, in which the black-box teacher only returns top-1 classes. It is a much more challenging scenario because the class probabilities of the training samples need to be constructed by iteratively querying from the DB3 teacher. With the training set accessible, our DB3KD achieved competitive performance on FLOWERS102, in which samples largely overlap with ImageNet. We believe that DB3KD can work effectively on large-scale datasets. With the training samples not available, like most of the existing works, a large amount of computing resource is required for pseudo sample generation, making zero-shot KD hard to accomplish with large-scale datasets. With a DB3 teacher, even more iterations are needed compared to learning from a white-box model. Although we proposed the first principled solution, we hope it helps to raise attention in this area and promote efficient approaches.

References

Ba, J. and Caruana, R. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.

Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., and Tian, Q. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3514–3522, 2019.

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. *arXiv preprint arXiv:1807.04457*, 2018.

Cheng, M., Singh, S., Chen, P., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

Clarifai, I. Clarifai: Computer vision and ai enterprise platform. 2020. URL http://www.clarifai.com.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2019.

Heo, B., Lee, M., Yun, S., and Choi, J. Y. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3779–3787, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Jaderberg, M., Vedaldi, A., and Zisserman, A. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

Jin, X., Lan, C., Zeng, W., and Chen, Z. Uncertainty-aware multi-shot knowledge distillation for image-based object re-identification. *arXiv preprint arXiv:2001.05197*, 2020.

Kim, J., Park, S., and Kwak, N. Paraphrasing complex network: Network compression via factor transfer. In *Advances in neural information processing systems*, pp. 2760–2769, 2018.

Kimura, A., Ghahramani, Z., Takeuchi, K., Iwata, T., and Ueda, N. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Li, T., Li, J., Liu, Z., and Zhang, C. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14639–14647, 2020.

Liu, Y., Moosavi-Dezfooli, S.-M., and Frossard, P. A geometry-inspired decision-based attack. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4890–4898, 2019.

Lopes, R. G., Fenu, S., and Starner, T. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

López, G., Quesada, L., and Guerrero, L. A. Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*, pp. 241–250. Springer, 2017.

Micaelli, P. and Storkey, A. J. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems*, pp. 9551–9561, 2019.

Moskalenko, V., Moskalenko, A., Korobov, A., Boiko, O., Martynenko, S., and Borovenskyi, O. Model and training methods of autonomous navigation system for compact drones. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp. 503–508. IEEE, 2018.

Nayak, G. K., Mopuri, K. R., Shaj, V., Babu, R. V., and Chakraborty, A. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114*, 2019.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Orekondy, T., Schiele, B., and Fritz, M. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4954–4963, 2019.

Passalis, N., Tzelepi, M., and Tefas, A. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2339–2348, 2020.

Phuong, M. and Lampert, C. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pp. 5142–5151. PMLR, 2019.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Wang, D., Li, Y., Wang, L., and Gong, B. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1498–1507, 2020.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1356–1365, 2018.

Wang, Z. Data-free knowledge distillation with soft targeted transfer set synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10245–10253, 2021.

Wang, Z., Li, C., and Wang, X. Convolutional neural network pruning with structural redundancy reduction. *arXiv preprint arXiv:2104.03438*, 2021.

Xu, G., Liu, Z., Li, X., and Loy, C. C. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pp. 588–604. Springer, 2020.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141, 2017.

Yin, H., Molchanov, P., Alvarez, J. M., Li, Z., Mallya, A., Hoiem, D., Jha, N. K., and Kautz, J. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Yuan, L., Tay, F. E., Li, G., Wang, T., and Feng, J. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

Yun, S., Park, J., Lee, K., and Shin, J. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13876–13885, 2020.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.