DRMI: A Dataset Reduction Technology based on Mutual Information for Black-box Attacks

Yingzhe He[1,2], Guozhu Meng[1,2,*], Kai Chen[1,2,*], Xingbo Hu[1,2], and Jinwen He[1,2]

[1]*SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, China*

Abstract

It is non-trivial to attack deep neural networks in black-box settings without any model detail disclosed. Prior studies on black-box attacks leverage a number of queries to the target model for probing the target model or generating adversarial examples. Queries are usually limited and costly so that the adversary probably fails to mount an effective attack. However, not all the queries have to be made since there exist repetitions or redundancies that induce many inefficient queries. Therefore, it leaves a lot of room for data reduction and more efficient queries.

To this end, we first propose to use mutual information to measure the data redundancy between two data samples, and then develop a data reduction technique based on mutual information, termed as DRMI. We implement an efficient optimization algorithm in DRMI, so as to obtain a particular subset of data samples, of which the mutual information in between is minimized. We conduct extensive experiments on MNIST, CIFAR10, and ImageNet, and six types of deep neural networks, and evaluate DRMI in model extraction and adversarial attacks. The results demonstrate its high effectiveness in these attacks, surpassing a state-of-the-art approach by raising 7% of model accuracy and two times more transferability of adversarial examples. Through the comparison experiments with other three strategies, we identify what properties of data have been preserved and removed, to some extent reveal the essences of deep neural networks.

1 Introduction

Deep neural networks (DNNs) are now well known to be vulnerable to many attacks [5, 24, 36], such as adversarial attacks [9, 38, 57], model extraction attacks [58, 60], model inversion attacks [18, 51], and poisoning attacks [27, 49]. Unperceivable perturbations added into an image can deceive a classifier in an adversarial attack. Furthermore, these weaknesses in DNNs are considerably magnified along with the widespread deployment and commercialization of deep learning. To date, a line of research has successfully subverted the mainstream deep learning systems [33, 61, 64] that can endanger the users' daily life.

These attacks encounter several obstacles in black-box settings where most if not all information about model is unknown. Prior research has paved a way in solving them like *e.g.*, *transfer attacks* [44, 45] and *optimization attacks* [25, 59]. Both of these attacks have to query the target model as prerequisites, and then either train a substitute model [29, 45] or further optimize the queries. With a substitute model, attackers cannot only uncover the parameters and decision boundaries of the model, but also generate adversarial examples (AEs) in a white-box setting. However, in reality, a large number of queries to the model are costly and even infeasible. That motivates the research on reducing queries to the model.

For simplicity, we assume that attackers can access a similar dataset of the target model in this study. As such, to reduce the queries in a black-box attack, we can turn to selecting high quality data and eliminating redundancies from the original for substitute model training. Similar with our study, PRADA [29] manages to extract model information in black-box settings. It develops a Jacobian-based method to synthesize high quality data, and trains a substitute model with limited queries. Tested on the MNIST [35] dataset, the substitute model can still obtain a 90% accuracy with merely 1/300 of the data, and effectively facilitate the generation of adversarial examples. Gradient Estimation [7] also attacks black-box model with 61.5% success rate under 196 queries.

The motivation of our research is to reduce the query cost of training a substitute model in black-box settings without accessing the exact training data. The substitute model can also be used for other attacks, such as model inversion attacks [18, 51], adversarial attacks [9, 38]. To fulfill the reduction, we first propose mutual information (MI) [2] for measuring the redundancies in a data set. MI is a measure of the mutual dependence between two variables in information

* Corresponding authors.

theory. More dependent (or similar) variables indicate a larger MI in between, which induces data redundancy conceptually. Given this, we develop a data reduction technique based on mutual information (DRMI). In DRMI, we calculate the MI value between any two data samples, and search a subset of fixed size to ensure the sum of MI values among selected samples is minimized (see Section 4). In this way, the selected samples are more independent and informative for the good of substitute model training. In addition, we compare our DRMI with another three reduction techniques based on correlation matrix (CMAL) [65], class probability (CPB) [42], and activated neuron trace (TRACE) [16] in Section 5.4, showing that DRMI exhibits a more superior performance.

We design a set of experiments to evaluate DRMI comprehensively. These experiments are carried on the MNIST [35], CIFAR10 [32], and ImageNet [48] datasets. Six models, *i.e.*, LeNet-5 [34], C3F2 (detailed in Table 1), DNN5 (detailed in Table 2), ResNet18 [23], ResNet152 [23], and Inception-v3 [56], have been employed for substitute model training. In a nutshell, DRMI surpasses PRADA by 7% in the accuracy of substitute models, with only 50 queries on the MNIST dataset. Based on the substitute model, we generate adversarial examples and their transferability reaches up to 66%, three times more than PRADA. Under 600 queries on MNIST, DRMI achieves 97.3% model accuracy and 78.5% transferability using C3F2 architecture, improving 3.3% accuracy and 29.5% transferability than PRADA. Furthermore, DRMI also raises 11.7% attack success rate with even 46 fewer queries than Gradient Estimation. DRMI raises 1.1%, 11.2% attack success rate with 618, 1343 fewer queries than NES [25], AutoZoom [59] on the ImageNet dataset, respectively. Experiments prove that DRMI can effectively facilitate model extraction and adversarial attacks in black-box settings. Additionally, the comparison experiments with three other measures show that DRMI exceeds CMAL, CPB, TRACE methods with an average accuracy of 6.46%, 9.03%, and 26.53%, respectively. From the results, we identify several insights on interpretability of deep learning process in Section 5.4.

Contributions. We make the following contributions.

- We propose a novel data reduction technology based on mutual information dubbed DRMI. By solving the simplified dataset with the minimum value of the overall mutual information, we can form a rival model of >96% accuracy with only 1% of training data (Section 5.2).
- We conduct black-box attacks (Section 5.3) for extracting model information and generating adversarial examples based on the substitute model. The results show our approach outperforms PRADA in both model accuracy (+7%) and transferability (x3), and outperforms Gradient Estimation in success rate (+11.7%).
- We explore the interpretability of deep learning models from the perspective of data reduction (Section 5.4). The conclusions indicate the properties that are either reserved or wiped by deep neural networks, and facilitate an in-depth understanding.

2 Background

2.1 Dataset Reduction in Learning

Deep learning algorithms often require large datasets for training [17, 43]. That also results in the emerging of data augmentation for enriching the training data [15, 47]. However, the requirement brings new problems: collecting and labeling data cost tremendous time and resources; training model on a large dataset occupies huge computation; and a large volume of data is susceptible to poisoned data [39]. There have been already works on reducing training data to raise learning efficiency [12, 41]. These works explore how to simplify the training data without loss of model correction, and even defend poisoning attacks by eliminating low quality data.

High quality data means a specific set of samples which can well represent and sample the whole dataset with few redundancies and repetitions. As a kind of high-dimensional data, there are many similarity metrics between images, such as structural similarity (SSIM) and cosine similarity. The mostly used method is L_p-norm, which measures the perceptual similarity between original images and adversarial images [9, 19, 57, 63]. However, recent research [50] finds that L_p-norm is neither necessary nor sufficient for perceptual similarity, and new metrics need to be proposed for more accurate measurements [28]. In this paper, we propose a novel concept to connect mutual information measurement with image dataset quality. Our experiments prove that mutual information can measure the independence, diversity and representativeness of data. We tend to explore the application of mutual information in more fields, such as perceptual similarity.

In this paper, we propose a model-independent dataset reduction approach DRMI, which treats mutual information as an indicator to measure the common information shared by two samples. We also compare DRMI with three other measurements–correlation matrix (CMAL), class probability (CPB) and activated neuron trace (TRACE). CMAL constructs a matrix to present the correlation distribution among all data samples. It is still model-independent since it can be computed in advance of model training. Additionally, we observe the system states and outputs after training data is feed into the model. In particular, we record the activated neurons scattered in different layers, and the class probability for the input data. Based on these information, we implement the corresponding reduction techniques. As the information is processed by the model, we take them as model-dependent measures. Although CMAL, CPB, TRACE do not perform as well as DRMI, the results help us understand training data and models, and analyze interesting conclusions in the view of interpretability.

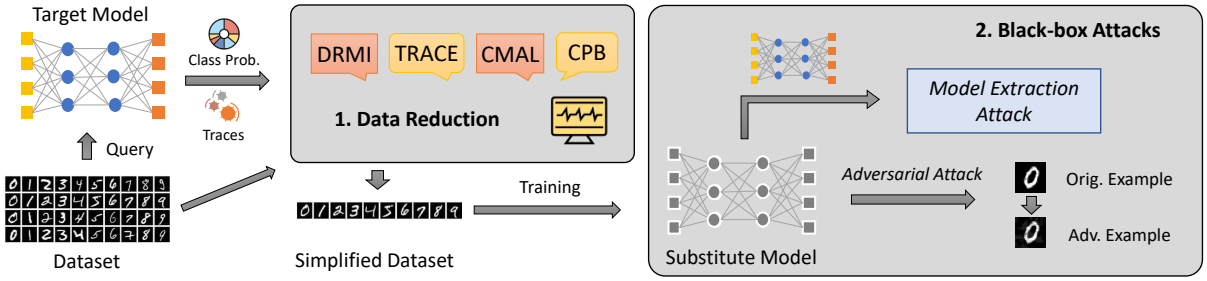

Figure 1: The workflow of our work

2.2 Black-box Attacks against DNNs

Black-box attacks against DNNs are of great variety [24, 36]. In this paper, we only focus on model extraction attacks and adversarial attacks.

Model Extraction Attack. It is an emerging technology to attack deep learning models in recent years. For deep neural networks, this attack tends to steal parameters [58], hyperparameters [60], architectures [40], decision boundaries [29, 44, 45], and functionalities [42]. However, it acquires a large number of queries to the target model for simulating models' behaviors. Reducing queries can not only avoid the attack being detected, but also save monetary costs.

Existing model extraction techniques commonly require training substitute models [42, 45]. Therefore, how to improve the effectiveness of substitute models with fewer queries has become the main focus for this attack. We propose a data reduction technique in this study, which enables a substitute model up to par with smaller datasets and fewer queries.

Adversarial Attack. Adversarial attacks are the most significant threats to deep neural networks. Thousands of methods have been developed to subvert a well-trained deep learning model. In black-box settings, queries to the target model become indispensable for either training a substitute model [10, 29, 45] or estimating approximate gradients [11, 25, 59]. The substitute model, which behaves quite similarly with the target model, can be further used to find AEs in a white-box manner [9, 19, 37]. These samples can be used to attack the target model due to their transferability. In such a case, the limitations of queries undoubtedly raise the difficulties of attacks. Existing works have tried to increase query efficiency from the perspective of data distribution and properties [6, 8, 21, 53]. In this paper, our research proposes DRMI to quantify data redundancies and gets a much simplified dataset for querying.

3 Overview

In this paper, we aim to select a simplified and representative dataset from the original. It can not only spare the time and computing resources for training a model, but also empower black-box attacks with limited queries to the target model.

Figure 1 presents the workflow of our work. We start from a known dataset and develop a data reduction technique to obtain representative and reduced datasets. Then we use every reduced dataset to train a new model (a.k.a. substitute model), and adopt prediction accuracy to quantify the performance of substitute models. The substitute model with higher accuracy indicates that its training data is more representative for the original.

Threat Model. In this study, the adversary aims to launch black-box attacks, *e.g.*, adversarial attacks and model extraction attacks, against a public deep learning service. However, the adversary knows neither the internal structure and parameters of the target model, nor the exact training data. Even so, it is still able to obtain a small dataset that has the same distribution as the training data, or a larger one with a different distribution. The adversary can query the target model with the possessed data and then get prediction results. It is not necessary to acquire confidence scores for prediction although they are often provided by commercial services. Additionally, it has to limit the number of queries as too many queries are costly and probably constrained by some defense measures.

1 Data Reduction. Data reduction is a technique to remove out redundancies and repetitions from multitudinous amounts of data, but remain critical and representative data [22]. To explore the redundancy in deep learning, we use mutual information as a measure and develop a data reduction technique based on it (*i.e.*, DRMI). Moreover, we implement another three reduction techniques based on *correlation matrix* (CMAL), *class probability of prediction* (CPB) and *traces of activated neurons* (TRACE) for comparison. In particular, DRMI and CMAL are performed merely on the training data, and not related to deep training. Therefore, they are model-independent. CPB and TRACE both require to interact with the target model, *i.e.*, collecting the prediction result or internal states when one data sample passes through the model. As such, we regard them as being model-dependent. In this study, we employ all these four strategies to reduce the training data, and subsequently shape a substitute model.

2 Black-box Attacks. The trained substitute model can be applied for further black-box attacks against deep neural networks. More specifically, the substitute model is a close ap-

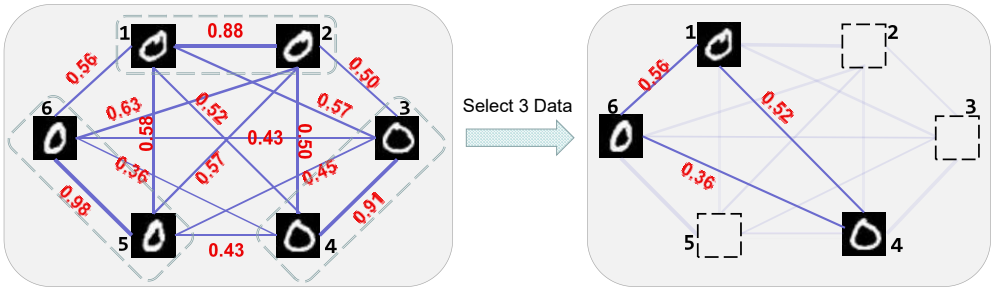

Figure 2: An illustrative example for DRMI. There are six images (noted as from 1 to 6) have quite similar appearance in pairs. The edge indicates the mutual information between two images. Thicker line indicates larger value. To form a subset with three images, we select images 1, 4, and 6 since the sum (1.44) of their MI values is minimal.

proximation of the target model in prediction. Hence, it helps to infer the parameters of the target model which is known as model extraction attacks [58, 60]. In this paper, we leverage prediction accuracy as the success rate for a model extraction attack. The substitute models created by the four techniques are compared, and the result shows DRMI has achieved the best performance (see Section 5.2 and 5.4). Based on the result, we also conclude a number of new views on the interpretability of deep neural networks.

On the other hand, the substitute model can be utilized for generating adversarial examples in black-box settings [8, 21, 53] or white-box settings [29,45]. Data reduction is especially beneficial for transfer attacks [6, 55] since it lowers the cost of model querying. Therefore, we conduct adversarial attack experiments based on our reduction techniques to evaluate its usefulness. We adopt the PGD method [37] to generate adversarial examples towards a substitute model, and test their transferability to the target model. Success rates are computed and compared with other state-of-the-art approaches.

4 The DRMI Approach

In this section, we detail the DRMI approach by formalizing the problem, analyzing its complexity and providing the solution.

4.1 Problem Formalization

We aim to select a more representative and reduced dataset through minimizing the mutual information value between any two data samples as shown in Figure 2. Assuming a big dataset D, and $n = |D|$, we intend to find a simplified dataset S, where $S \subset D, k = |S| < n$. For every two samples $u \in D$ and $v \in D$, we calculate the mutual information value $MI(u)(v)$ between them and get the MI matrix. According to the definition of mutual information in information theory [14], given the images u and v, we compute their MI value as:

$$MI(u)(v) = \sum_{i=0}^{R} \sum_{j=0}^{R} P_{uv}(i,j) \log \frac{P_{uv}(i,j)}{P_u(i)P_v(j)} \quad (1)$$

R is the maximum pixel intensity value. The marginal probability distribution $P_u(i)$ refers to the ratio of the pixels of intensity value i in image u to all the pixels in image u. P_{uv} is the joint probability distribution function between two certain images u and v. The probability $P_{uv}(i,j)$ refers to the ratio of the number of pixel points, where the pixel intensity value is i in image u and j in image v under the same coordinates, to the total number of pixels. If $P_{uv}(i,j) = 0$, we handle $P_{uv}(i,j) \log \frac{P_{uv}(i,j)}{P_u(i)P_v(j)} = 0$. For each pair $(u,v), u \in D, v \in D, u \neq v$, we calculate its MI value by Equation 1, and obtain the MI value matrix. Equation 1 considers not only the number of pixel intensity values, but also their positions.

For a seek of generalizability, we introduce a new matrix I and a hyperparameter α used to represent the weight of mutual information. The choice of α is discussed in Section 7. The correspondence between matrix I and mutual information is as follows:

$$I[u][v] = MI(u)(v)^{\alpha} \quad (2)$$

For convenience, we will use matrix I hereafter. Therefore, the process of sampling k data points with minimizing the sum of MI values between them can be formalized as:

$$\arg\min_{S} H = \frac{1}{2} \sum_{i \in S} \sum_{j \in S} I[i][j],\ i \neq j \quad (3)$$

We use H as this minimum and $1/2$ is multiplied to avoid redundant computation.

To solve the problem in Equation 3, we propose to formalize it as a graph theory problem. Let $G = \{V, E\}$ be a weighted undirected graph without self-loops and parallel edges. V is the set of vertices, and E is the set of edges. Each edge $e \in E$ is associated with a real number $w(e)$. With regard to this problem, we treat each data sample as a vertex v. For every

two samples v and u, we can link them up with their mutual information $I[u][v]$ as the weight. Therefore, a data set can be modeled as a undirected complete graph with weights. In the sequel, the problem can be converted as: *Given a weighted undirected complete graph G with n vertices, find an induced graph with k vertices ($k < n$), of which the sum of edge weights is minimal.* To gain an induced graph $G[S]$, we will address the following in this study.

$$\arg\min_{G[S]} H = \sum_{e=(u,v)} w(e), \quad u,v \in S, u \neq v, \text{ and } e \in E \tag{4}$$

4.2 Complexity Analysis

Unfortunately, the problem in Equation 4 is a NP-Complete problem. There is no optimal solution in polynomial time to date. We give a strict proof in the following.

Proof of NP. Given a subset $S \subset V$ with k vertices, we can calculate the sum of weights in the induced subgraph $G[S]$ in polynomial time using Equation 4. So we can verify every solution in polynomial time, proving that the problem is NP.

Proof of NP-Hard. Here we use another NP-Complete problem–the maximum independent set to complete the proof [30]. We need to prove that the maximum independent set problem can be reduced to our problem in polynomial time. In a simple unweighted undirected graph $G_i = \{V_i, E_i\}$, $e = (u,v) \in E_i,\ u,v \in V_i$, we call $S \subseteq V_i$ an independent set if and only if:

$$\forall u,v \in S, \quad (u,v) \notin E_i \tag{5}$$

Given a graph G_i and an integer $k < |V|$, the maximum independent set problem is to determine if there is an independent set S of at least size k.

Next, assuming that our problem is solvable, we use our problem to solve the maximum independent set problem. We convert the unweighted undirected graph $G_i = \{V_i, E_i\}$ into a weighted undirected complete graph $G_c = \{V_c, E_c\}(V_c = V_i)$ where $w(e)$ denotes the weight for edge $e = (u,v)$ and $e \in E_c$. The conversion satisfies the following constraints:

$$w(e) = \begin{cases} 1, & \forall u,v \in V_i \text{ and } (u,v) \in E_i \\ 0, & \forall u,v \in V_i \text{ and } (u,v) \notin E_i \end{cases} \tag{6}$$

If vertices u and v have an edge in graph G_i, we add a 1-weighted edge between them into graph G_c. If vertices u and v have no edge in G_i, we add a 0-weighted edge in G_c. Then we use Equation 4 to calculate the minimum H of a complete subgraph with k vertices on graph G_c. If $H = 0$, it means there exists k vertices in G_c, and the weight of any two vertices is 0. It indicates that there exists an independent set S with size k in graph G_i. Similarly, if $H > 0$, it means there does not exist any independent set S of at least size k in graph G_i.

Therefore, the maximum independent set problem can be reduced to our problem in polynomial time. Since the maximum independent set problem is NP-Hard, our problem is also NP-Hard. As NP-Complete is the intersection of NP and NP-Hard, the problem we need to solve is NP-Complete.

Algorithm 1: Data Reduction on Mutual Information

Input: $G(V,E)$: a weighted undirected graph where $|V| = n$, k: the size of target subgraph
Output: S_{min} where $S_{min} \subset V \wedge |S_{min}| = k$

```
1 H_min ← MAXNUM;
2 S_min ← {};
3 for t ∈ V do
4     S_0 ← greedy_choice_initialization (t);
5     S, H ← one_hot_replacement_optimization (S_0);
6     if H < H_min then
7         H_min ← H;
8         S_min ← S;
9 return S_min
```

4.3 Our Solution

Since our problem is NP-Complete, there is no optimal solution in polynomial time. We propose a novel and effective heuristic algorithm to approximate the optimal solution, *i.e.*, obtaining the induced subgraph whose mutual information H approximates the minimal as Equation 4. First, we select an initial vertex t, and determine another $k-1$ vertices based on mutual information with a greedy strategy. As a consequence, we obtain an initial subset S_0 where $|S_0| = k$. Then, we optimize this subset iteratively in order to sustainedly lower the weights sum according to Equation 4. After a limited iterations, we are able to get a stable set S_F and H reaches its approximate optimal value.

Algorithm 1 presents the overall process of our data reduction technique. It proceeds with n iterations (Line 3). For each iteration, it selects one vertex in V and passes it to the *initialization* phase in Line 4. As such, the initial subset S_0 is obtained. S_0 goes through an *optimization* phase in Line 5, where the optimized subset S and the associated H are returned. Line 6-8 show that we will keep the superior solution while discard the inferior one. At last, S_{min} is the approximate optimal solution.

4.3.1 Initialization

In the phase of *initialization*, we construct a primary subset S_0, starting from the passed vertex t. The construction is realized with a greedy strategy, so the phase is termed as greedy-choice initialization. There are two methods to guide the greedy process as follows.

Min-sum method. Given the first vertex t, we initialize S_0 with t and additionally maintain a *sumI* array, of which $sumI[p]$ denotes the sum of MI values between data point p

Algorithm 2: Greedy-choice Initialization

Input: $G(V,E)$: a weighted undirected graph where $|V|=n$, k: the size of simplified set, t: the initial data point (vertex)

Output: S_0 where $S_0 \subset V \wedge |S_0| = k$

1 $S_0 \leftarrow \{t\}$;
2 $f(i) \leftarrow I[t][i],\ i \notin S_0$;
3 **for** $i = 2$ *to* k **do**
4 $\quad p' = \arg\min_p f(p),\ p \notin S_0$;
5 $\quad S_0 = S_0 \bigcup \{p'\}$
6 $\quad f(x) = g(f(x), I[p'][x]),\ x \notin S_0$;
7 **return** S_0

and every other element in set S_0:

$$sumI[p] = \sum_{q \in S_0} I[p][q],\ \ q \neq p \tag{7}$$

S_0 has only one element t at first. Then we select p' which minimizes $sumI[p']$ and $p' \notin S_0$. After that, we add p' into S_0 and maintain the $sumI$ array with Equation 7. We repeat the selection process and stop if $|S_0| = k$. This method takes seed t as the starting point, and selects the data outside set S_0 that has the least MI sum with all points in set S_0 at every time.

Min-max method. In min-sum method, we attempt to minimize the distance between the added sample and all samples already in the set. That is, the new sample has the least averaged similarity to the existing. While in min-max method, the new sample has the least maximum similarity to the existing samples. For example, if one sample is very similar to one in the set, we will not add it even though it is very different from any other samples. Correspondingly, we change the Equation 7 to the following $maxI$ array:

$$maxI[p] = \max_{q \in S_0} \{I[p][q]\},\ \ q \neq p \tag{8}$$

Algorithm 2 presents the process of obtaining a good initial set from the vertex t. S_0 is the initial set. At first, S_0 only has one vertex t (Line 1). Then we calculate $f(\cdot)$ (Line 2) where $f(\cdot)$ is either $sumI[\cdot]$ or $maxI[\cdot]$. Now the sum of the distances between each vertex i to all the vertices in S_0 is $I[t][i]$. Line 3 to 6 are a loop to add vertices into S_0. In each iteration, we find the vertex p' which has the minimum value in $f(p)$ at line 4. Then we add the new vertex into S_0 at line 5. The addition of p' needs an update to $f(x)$ by $I[p'][x]$ at line 6. If $f(\cdot)$ is $sumI[\cdot]$, $g(a,b) = a+b$. If $f(\cdot)$ is $maxI[\cdot]$, $g(a,b) = \max(a,b)$. Last, this algorithm returns the initial set S_0. The time complexity of Algorithm 2 is $O(kn)$.

4.3.2 Iterative Optimization

After getting an initial set S_0, we define two arrays In and Out. $In[i]$ expresses the sum of MI values between i and other points from S_0, which makes sense when $i \in S_0$. $Out[i]$ expresses the sum of MI values between i and all points from S_0, which makes sense when $i \notin S_0$. Then we can calculate the initial value H:

$$H = \frac{1}{2} \sum_{i \in S_0} \sum_{j \in S_0} I[i][j] = \frac{1}{2} \sum_{i \in S_0} In[i],\ \ j \neq i \tag{9}$$

Next, we need to adjust set $S(=S_0)$. Starting from S, we remove a data point with poorest performance in set S, and move into a data point with best performance outside set S. Here, poor performance means this point has the maximum In value, and good performance means the minimum Out value. If a swap (p,q) could make H decrease ($H' < H$ in Equation 10), we perform such an exchange.

$$H' = H + Out[q] - In[p] - I[p][q],\ p \in S,\ q \neq S \tag{10}$$

Then we repeat the above exchange process until H is no longer decreasing. We call this method of adjusting and optimizing the solution as one-hot replacement.

Algorithm 3 presents the one-hot replacement optimization, which is based on the exchange of vertices to optimize the solution. This algorithm needs to optimize the final set S and reduce H value according to the initial set. We give the initial set S_0 to the final S at line 1. For a vertex in S, we compute the sum of distances with other vertices in S (Line 3). For a vertex not in S, we compute the sum of distances with all vertices in S (Line 4). Then we calculate the initial H value.

Algorithm 3: One-hot Replacement Optimization

Input: $G(V,E)$: a weighted undirected graph where $|V|=n$, k: the size of simplified set, S_0: the initial set where $|S_0| = k$

Output: S, H where $S \subset V \wedge |S| = k$

1 $S \leftarrow S_0$;
2 $H \leftarrow 0$;
3 $In[t] = \sum_j I[t][j],\ t \in S, j \in S, j \neq t$;
4 $Out[t] = \sum_j I[t][j],\ t \notin S, j \in S$;
5 $H = H + \frac{1}{2}\sum_t In[t],\ t \in S$;
6 **while** *True* **do**
7 $\quad p = \arg\ \max_t In[t],\ t \in S$;
8 $\quad q = \arg\ \min_j Out[j] - I[p][j],\ j \notin S$;
9 $\quad$ **if** $Out[q] - I[p][q] >= In[p]$ **then**
10 $\quad\quad$ **break**;
11 $\quad H = H + Out[q] - I[p][q] - In[p]$;
12 $\quad S = S \bigcup \{q\} \setminus \{p\}$;
13 $\quad In[q] = Out[q] - I[p][q]$;
14 $\quad Out[p] = In[p] + I[p][q]$;
15 $\quad In[t] = In[t] - I[t][p] + I[t][q],\ t \in S, t \neq q$;
16 $\quad Out[t] = Out[t] - I[t][p] + I[t][q],\ t \notin S, t \neq p$;
17 **return** S, H

Line 6 to 16 are the loop to find set S with smaller H values. According to Equation 10, we first find the vertex p which has the maximum $In[p]$ in S, then the vertex q which has the minimum $Out[q] - I[p][q]$ not in S. Line 9 and 10 are the termination condition of the loop. If this condition is satisfied, H will not decrease after swapping vertices. Line 11 to 16 explain how to update variable values during the exchange process. Line 11 calculates the new H, and line 12 puts q in and puts p out to update S. Line 13 to 16 update the In or Out values for each vertex according to moving in a new vertex q and out an old vertex p. After the loop ends, the algorithm returns the minimum H and its corresponding set S at line 17. This algorithm can be terminated efficiently partially due to the greedy-choice initialization which offers an approximated optimal solution. It then takes only a few exchanges to reach a better solution. The transitivity of data similarity [62] further prevents one sample from being exchanged for multiple times. As a consequence, the replacement is expected to be terminated within $O(k)$ iterations. Our experiments with different datasets also confirm that the iteration number is lower than a constant (<10) multiple of k. Additionally, the worse-case complexity of the in-loop computation is $O(n)$. Therefore, the time complexity of one-hot replacement is $O(kn)$.

5 Evaluation

In this section, we describe the implementation details of our approach and the evaluation experiments.

Implementation. We implement DRMI with 2.5K lines of Python on top of PYTORCH [3]. The adversarial examples are evaluated by the targeted PGD [37] method using foolbox [4] library. The experiments are conducted on a server with 16 Intel(R) Xeon(R) CPUs of E5-2620 and 32GB memory, 2 NVIDIA GM200 [GeForce GTX TITAN X] GPUs and 1 ASPEED Video AST2400 GPU. These experiments are carried out to evaluate the efficiency and efficacy of DRMI. Through these experiments, we intend to answer:

RQ1. How effective is DRMI to reduce data for training?

RQ2. How does it facilitate black-box attacks?

RQ3. How is other reduction strategies, and what can be interpreted from the results?

5.1 Experiment Setup

Experiment Data. We conduct our experiments on MNIST [35], CIFAR10 [32], and ImageNet [48] (ILSVRC2012) datasets. The MNIST dataset contains 60,000 training images of 10 classes and 10,000 test ones. Its samples are 28×28 grey-scale images of handwritten digits. CIFAR10 contains 50,000 training samples of 10 classes and 10,000 test data. Its samples are 32×32 RGB images. ImageNet contains about 1,200,000 training data, 100,000 test data, and 50,000 validation data of 1,000 classes. Its samples are 224×224 RGB images. We train the substitute model on a simplified training dataset and test model on the test dataset.

Table 1: Parameters of the C3F2 model

Layer Name	Output Dimensions
Input	1 * 28 * 28
Convolutional layer	16 * 24 * 24
Convolutional layer	32 * 20 * 20
Max-Pooling layer	32 * 10 * 10
Convolutional layer	64 * 6 * 6
Max-Pooling layer	64 * 3 * 3
Fully connected layer	100
Fully connected layer	10

Table 2: Parameters of the DNN5 model

Layer Name	Output Dimensions
Input	784
Fully connected layer 1	512
Fully connected layer 2	256
Fully connected layer 3	128
Fully connected layer 4	64
Fully connected layer 5	10

Target Model. We select LeNet-5 [34], C3F2 and DNN5 model structures on dataset MNIST. We adopt model ResNet18 [23] on dataset CIFAR10, and Inception-v3 [56], ResNet152 [23] on ImageNet. LeNet-5 is an efficient convolutional neural network for handwritten character recognition. It includes 2 convolutional layers, 2 pooling layers, and 3 fully connected layers. Table 1 shows C3F2's model architecture. It has 3 convolutional layers, 2 pooling layers, and 2 fully connected layers. Table 2 details DNN5's model architecture. It has 5 fully connected layers and no convolutional layer. ResNet is a residual network, which is used for more complex image classification.

Experiment Configuration. When training models on a simplified dataset, we set batch size to 4 on MNIST and 64 on CIFAR10. We use max-pooling in pooling layers, *cross entropy loss* to calculate losses. By default, we take *adaptive moment estimation (Adam)* as the optimizer and set the learning rate to 0.001. Our data selection is carried out under the same label. That is, we determine a simplified dataset for each category, and then glue them together into the training dataset for our experiments.

Baseline Method. We implement a baseline method in this paper to show to what extent our approach can raise in data reduction. In the baseline method, we randomly select a specific number of samples without any intelligence. Taking MNIST as an example, we select samples for each digit proportionally and randomly, and then train a substitute model as well as measuring its accuracy. This process is repeated for five times and the result is averaged in a comparison.

Manual Reduction Method. To verify whether our approach can excel manual efforts in data reduction, we invite two vol-

Table 3: Evaluations of model LeNet-5 on dataset MNIST. "Test Accuracy" means the substitute model accuracy on the test dataset. The optimal LeNet-5 model performance trained on the full dataset (60,000 data) reaches 99.17% accuracy. "Queries" is the number of queries to the original model, also the size of simplified set.

Method	α	Test Accuracy		
		Queries = 600	Queries = 300	Queries = 150
DRMI (min-sum)	1	95.59%	93.74%	88.01%
	2	95.84%	**94.29%**	**92.13%**
	4	**96.38%**	94.09%	91.35%
DRMI (min-max)	1	95.52%	91.99%	87.07%
	2	96.01%	93.49%	90.15%
	4	**96.41%**	94.14%	91.99%
manual reduction	-	94.65%	92.46%	86.57%
baseline	-	91.91%	88.48%	84.97%

Table 4: Evaluations of model ResNet18 on dataset CIFAR10. The original ResNet18 model trained on the full dataset (50,000 data) obtains 93.90% accuracy. "AD Size" is the dataset size of attackers can get. "Queries" is the number of queries to the original model, also the size of simplified set.

AD Size	Queries	Test Accuracy	
		DRMI (min-sum & α=2)	baseline
25,000	10,000	92.50%	80.05%
	4,000	89.74%	72.28%
	1,000	82.28%	55.72%
	500	73.46%	44.58%

unteers with normal eyesight and intelligence to collectively select typical and non-repetitive images from the MNIST dataset. If two images look similar in appearance, or are mirror symmetry, we remain only one image. The manually selected data will be tested and measured for comparison.

5.2 Effectiveness of Data Reduction

To answer RQ1, we train a substitute model on the simplified dataset with DRMI, and compute the accuracy and loss value of the model on the test dataset. The effectiveness of data reduction is evaluated threefold: *different datasets*, which we used to guide the optimization; *different reduction degrees*, to which we simplified the training data, *i.e.*, with only 1% or even 0.1% of the original data, and; *different target models*, to evaluate whether DRMI is widely applicable.

5.2.1 Different Datasets

Here we test different parameters and solutions in DRMI on different datasets. In Equation 2, we introduce α for MI value. When α is larger, a larger MI value has a greater effect on the result, but also means a larger penalty. Here we select $\alpha = 1, 2, 4$. We also adopt two initial solutions "min-sum" and "min-max" (see Algorithm 2) to evaluate different solutions.

Table 3 evaluates substitute models when adopting different parameters and solutions under the LeNet-5 model architecture. It is observed that the model gets the highest prediction accuracy when $\alpha = 4$ under 600 samples, and $\alpha = 2$ under 300 and 150 samples. $\alpha = 1$ performs worst on all sizes and algorithms. This indicates that we need to impose a heavier penalty on the larger MI value. Moreover, the two methods "min-sum" and "min-max" perform almost the same. Compared to the manual reduction method, DRMI has raised the accuracy by 1.76%, 1.83%, and 5.56% on 600, 300, and 150 sized samples. Compared with the baseline method, our methods have greatly raised the accuracy by 4.50%, 5.81%, and 7.16% on 600, 300, and 150 sized samples, respectively. Since the upper limit of test accuracy is 99.17%, our methods have improved the baseline method by 62%, 55%, and 51% on 600, 300, and 150 sized samples respectively in the whole improvable space.

Remark: From the experiments with varying parameters, it concludes that a higher power α for mutual information (i.e., greater penalties for large MI values) leads to a better reduction, where our two initial solutions both perform well. All of our best methods improve more than 50% from the baseline method within the improvable space.

We choose model ResNet18 to train substitute models on CIFAR10 and present the results in Table 4. Here we select the "min-sum" method and $\alpha = 2$. CIFAR10 images are more complex, so the training effect decreases. When querying 10,000 data, DRMI achieves 92.50% accuracy, only 1.40% gap to reach the original model. We also achieve 89.74%, 82.28%, and 73.46% accuracy with 4,000, 1,000, and 500 query, improving 17.46%, 26.56%, and 28.88% accuracy than the baseline method respectively. This shows that DRMI also works effectively on the CIFAR10 dataset.

Table 5 evaluates DRMI on the ImageNet dataset using Inception-v3 and ResNet152 models. When we use a simplified set with 100,000 data (8.3% of the target training set), DRMI still reaches 90.6% top-5 accuracy and 73.9% top-1 accuracy, while 100,000 random queries in baseline only gets

Table 5: Evaluations on ImageNet. The original Inception-v3 model reaches 94.5% top-5 accuracy and 79.2% top-1 accuracy. The original ResNet152 model reaches 94.0% top-5 accuracy and 78.8% top-1 accuracy. We adopt Inception-v3 and ResNet152 structures as substitute models respectively. Here we adopt the "min-sum" method and choose $\alpha = 2$. "AD Size" is the dataset size of attackers can get. "Queries" is the number of queries to extract a substitute model, also the size of simplified set.

AD Size	Queries	Inception-v3		ResNet152	
		Top-5 Acc.	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.
200,000	100,000	90.6%	73.9%	90.2%	73.6%
	50,000	87.7%	68.5%	87.2%	68.7%
50,000	20,000	82.3%	63.8%	80.8%	62.9%
	10,000	77.0%	57.9%	76.7%	57.4%
10,000	5,000	72.5%	48.4%	71.7%	47.8%
	2,000	61.8%	40.2%	60.5%	39.1%
baseline	100,000	73.8%	52.7%	72.2%	51.5%

Table 6: Evaluations when attackers only obtain limited data. The target model is trained on MNIST. Attackers get some data which matches the distribution of the target dataset (MNIST) in the left part, and obtain data from USPS (7291 data in total) which does not match the distribution in the right part. Here we use "min-sum" method and choose $\alpha = 2$. "AD Size" is attackers' dataset size. It means how many samples attackers can get. "150" means the attacker chooses 150 representative samples from his dataset using DRMI. "ALL." means attackers query the target model for all their data, which consumes lots of query overhead. The complete training dataset has 60,000 data. "Test Acc." means the substitute model accuracy on the test dataset.

AD Size	Test Acc. under different size of simplified set on MNIST						AD Size	Test Acc. under different size of simplified set on USPS					
	600	300	150	100	60	ALL.		600	300	150	100	60	ALL.
60,000	95.84%	94.29%	92.13%	88.09%	83.27%	99.13%	-	-	-	-	-	-	-
10,000	95.57%	93.01%	90.85%	87.97%	82.86%	98.27%	7,291	93.65%	92.15%	89.94%	86.69%	81.73%	95.56%
5,000	94.83%	92.40%	90.51%	87.77%	82.65%	97.56%	5,000	93.36%	91.88%	89.57%	86.24%	81.47%	94.69%
2,000	94.67%	92.05%	90.29%	86.38%	82.09%	96.33%	2,000	92.50%	91.20%	89.08%	85.41%	80.42%	93.17%
1,000	94.50%	91.76%	90.08%	86.13%	81.80%	95.36%	1,000	91.81%	90.89%	88.67%	85.11%	80.09%	92.26%
600	-	91.06%	88.58%	84.95%	80.42%	92.84%	600	-	90.23%	87.88%	84.30%	79.16%	90.83%

Table 7: Evaluations of C3F2 model and DNN5 model on MNIST. "Test Accuracy" means the substitute model accuracy on the test dataset. The original C3F2 model trained on the full dataset (60,000 data) reaches 99.28% accuracy. The original DNN5 model reaches 98.03% accuracy. The number of "Queries" is also the size of simplified set.

Model	Method	α	Test Accuracy		
			Queries = 600	Queries = 300	Queries = 150
C3F2	min-sum	1	95.10%	91.21%	89.98%
		2	96.54%	94.03%	86.59%
		4	97.25%	**94.57%**	90.49%
	min-max	1	96.05%	92.43%	88.89%
		2	96.40%	**94.57%**	89.02%
		4	**97.34%**	94.41%	**91.12%**
	baseline	-	92.40%	90.65%	85.18%
DNN5	min-sum	1	87.87%	80.80%	68.06%
		2	86.78%	82.79%	71.13%
		4	**90.11%**	**83.79%**	**74.77%**
	baseline	-	82.99%	73.87%	64.12%

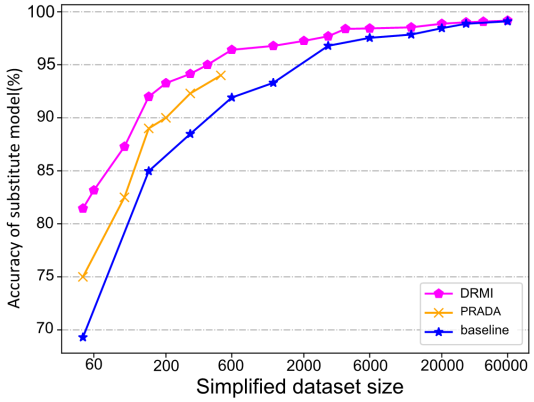

Figure 3: The curve of model accuracy under different dataset sizes on MNIST.

73.8% top-5 accuracy and 52.7% top-1 accuracy. When the attacker only uses 10,000 data, we can get a substitute model with 77.0% top-5 accuracy. Results show that DRMI also works on the ImageNet dataset.

Remark: Our DRMI *also performs well on CIFAR10 and more complex datasets like ImageNet.* DRMI *shows superior performance on different datasets.*

5.2.2 Different Reduction Degrees

In order to assess the relationship between reduced samples and corresponding accuracies, we conducted an experiment with different k for Equation 4. More specifically, we sample training data of varying sizes (*e.g.*, k=60, 600, or 6,000).The accuracies are measured for each training. Figure 3 shows the curve of the accuracy rates of substitute models with different dataset sizes. In PRADA, we only found results below 500 samples. Compared to other methods, our curve has high accuracy when the size is very small. It has reached 82% at 50 samples, 92% at 150, and 97% at 600. In PRADA [29], the accuracy of 50 samples is only 75%, 82.5% at 100, and 90% at 200. In the baseline method, the accuracy is lower than 70% at 50 samples, even 300 samples can only achieve an accuracy of 89%. This shows that DRMI can achieve high performance with small queries. The gap between DRMI and baseline is about 5% at 600 samples, 7.5% at 150, and more than 10% at 50. The gap between DRMI and PRADA is about 3.3% at 200 samples, 4.8% at 100, and 7% at 50. When the dataset size exceeds 2,000, the gap becomes smaller and is filled when the size is larger than 20,000.

Remark: Our DRMI *method can obtain a high accuracy with a small-sized dataset. When the dataset size is greater than 50, the smaller the dataset size, the greater the gap between other methods and ours.*

As claimed in "Threat Model" at Section 3, DRMI can still performs effectively when attackers can only access some data (may not in the training set) that has the same distribution with the training data. It is evaluated and presented as shown in Table 6. In this experiment, the dataset is randomly divided into two parts (except the "60,000" row). One part can be obtained by attackers, whose size is "AD Size" in Table 6, and the other is used to train a target model. This guarantees that attackers can only access the data of the same distribution with the training dataset, not the exact training data. The row of "60,000" is the situation when the attacker has all training samples. From the perspective of each column, the test accu-

racy only decreases slightly when the attacker has a smaller dataset. When the attacker has only one-tenth of previous data (from 10,000 to 1,000), the substitute model's accuracy only decreases 1.07%, 1.25%, 0.77%, 1.84%, 1.06% under 600, 300, 150, 100, 60 queries, respectively. When the attacker can only get 600 samples, DRMI also obtains 91.06% accuracy under 300 queries. The accuracy decline from 60,000 "AD Size" to 600 is between 2.85% and 3.55% under 300, 150, 100, and 60 queries. Results show that DRMI still performs well even when the attacker only has limited data. DRMI can select representative data from a small dataset, and the stolen substitute model still has a high accuracy rate.

We also do similar experiments on ImageNet in Table 5. The dataset size of the attacker varies from 200,000 to only 10,000. DRMI can achieve 72.5% top-5 accuracy through 5,000 data when the attacker only obtains 10,000 samples.

Remark: DRMI *performs well when attackers only have a very small dataset.* DRMI *also does not need attackers to know the exact training data.*

In order to explore the performance of DRMI when attackers obtain a different dataset that does not match the distribution of the training dataset, we choose another handwritten digits dataset USPS [1] and present the results in Table 6. Attackers utilize the USPS data to steal the target model trained on MNIST. Querying 7,291 data gets a 95.56% substitute model, while querying 600 representative samples using DRMI still reaches a 93.65% model. Compared to MNIST, using USPS data for attack only decreases 1.47%, 0.52%, 0.94%, 1.53%, 1.18% accuracy under 600, 300, 150, 100, 60 queries when attackers have 5,000 samples. Results show that using USPS data can still attack the target model, with a bit of accuracy decrease compared to using MNIST data.

Remark: DRMI *still works well when the attacker's dataset does not match the distribution of the target training dataset.*

5.2.3 Different Models

To evaluate its generality amongst varying models, we test our approach against C3F2 and DNN5 models on MNIST. Table 7 shows the results of training substitute model against a C3F2 model and a DNN5 model, spanning from size 150 to 600. We can see that the accuracies on C3F2 and LeNet-5 (see Table 3) models are all higher than that of DNN5 model, which is determined by model structure itself. The best C3F2 results are 7.23%, 10.78%, and 16.35% higher than the best DNN5 results on 600, 300, and 150 size.

In addition, the accuracy under $\alpha = 1$ still performs the worst, and there is a gap with cases of $\alpha = 2$ or 4. This also demonstrates the need to give high penalties (large α) to images with high similarity (large MI value) in the reduced dataset. Comparing the two algorithms, "min-max" and "min-sum" are still not far behind. On 600 dataset size of C3F2, we improve the test accuracy up to 97.34%, which is only fewer than two percentages away from the optimal model.

Compared to the baseline method, our approach on C3F2 has increased by 4.94, 3.92, and 5.94 percentages on 600, 300, and 150 dataset sizes, respectively. According to the upper limit of 99.28%, our improvement has reached 72%, 45%, and 42% on 600, 300, and 150 dataset sizes in the improvable space. On the DNN5 model, DRMI improves 7.12%, 9.92%, and 10.65% accuracy than baseline on 600, 300, and 150 size.

Table 5 shows attackers adopt an Inception-v3 and a ResNet152 network to steal the Inception-v3 target model on ImageNet. The top-5 and top-1 accuracy are very similar ($< 2\%$) on the two substitute model structures.

Remark: To sum up, our approach can be applied to a wide range of model structures (CNNs and DNNs), which proves the excellent generalizability of our DRMI *method. The attacker does not need to know the target model architecture. It is largely attributed to its model-independent property. As a result, given a dataset, we can extract a high-quality reduced dataset, which can be applied to different models.*

Jagielski *et al.* [26] also focuses on extracting high-accuracy substitute models with fewer queries. Their learning-based extraction adopts semi-supervised learning techniques. Here we make a comparison. On ImageNet, DRMI reaches 90.6% top-5 accuracy using about 8.3% data, and their method achieves 86.2% top-5 accuracy using 10% data. On CIFAR10 for 4,000 queries, DRMI reaches 89.74% accuracy, better than 86.51% in their fully supervised extraction, indicating that the quality of our queries is higher than theirs, but worse than 93.29% accuracy in their MixMatch extraction. This is mainly because they not only use query data for fully supervised learning, but also perform semi-supervised learning on the remaining unlabeled data in the training set.

5.3 Catalytic Effect for Black-box Attacks

We aim to answer RQ2 by evaluating how our approach facilitates black-box attacks. The accuracy evaluation proves that our substitute model is functionally similar to the target model. Here we evaluate the decision boundary similarity between them through attack success rate of adversarial examples (AEs). Adversarial attacks are a major technology to undermine the security of deep learning models. Training substitute models has been a method of black-box adversarial attacks. By querying the target model, attackers can obtain class probabilities of their inputs. Then they use these data to train a substitute model, and adopt white-box adversarial attacks to generate AEs on it. At last, attackers use these AEs to attack the target model and evaluate the success rate according to the transferability of AEs. In this process, training dataset quality and query numbers are particularly important. Attackers need to get a high quality dataset and use fewer queries for the target model.

Here we use our MI technique for black-box adversarial attacks. We adopt the simplified dataset produced by the MI method to query the target model, and train a substitute model

Table 8: Transferability of adversarial examples on target models generated by substitute models. Adversarial examples are generated by PGD. "Transferability" means success rate of adversarial examples on target model. "LeNet-5 (1,000)" means the attacker only has a small dataset with 1,000 data points. Experiments are under the same environments.

Queries	Target model	Transferability	Accuracy
50	LeNet-5	66.06%	82.27%
	C3F2	48.80%	80.96%
	LeNet-5 (1,000)	42.62%	80.40%
	PRADA [29]	22%	75%
	Practical [45]	19%	65%
150	LeNet-5	68.32%	92.13%
	C3F2	69.64%	91.12%
	LeNet-5 (1,000)	54.45%	90.08%
	PRADA	29%	89%
	Practical	27%	81.20%
200	LeNet-5	69.15%	93.27%
	C3F2	70.13%	92.18%
	LeNet-5 (1,000)	57.90%	91.13%
	PRADA	31%	90%
	Practical	28%	85%
300	LeNet-5	69.80%	94.34%
	C3F2	76.37%	94.57%
	LeNet-5 (1,000)	60.70%	91.76%
	PRADA	39%	91%
	Practical	33%	87%
600	LeNet-5	71.98%	96.49%
	C3F2	78.51%	97.34%
	LeNet-5 (1,000)	65.74%	94.50%
	PRADA	49%	94%
	Practical	39%	90%

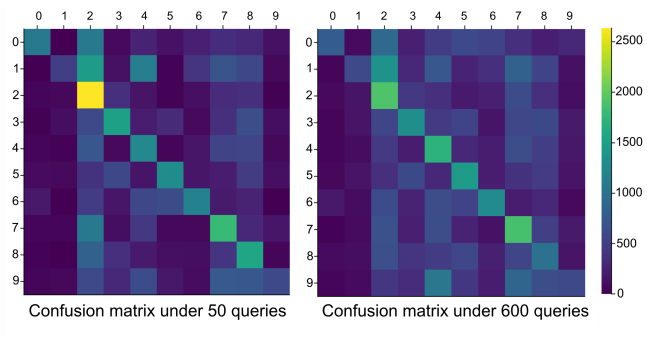

Figure 4: Confusion matrices of targeted adversarial examples attacking the target LeNet-5 model.

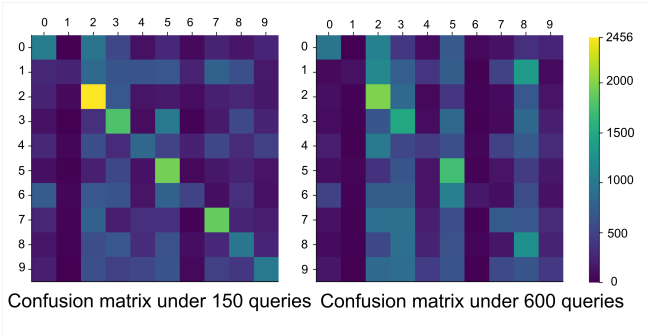

Figure 5: Confusion matrices of targeted adversarial examples attacking the target C3F2 model.

based on class probability information we obtained. Then we use the PGD [37] (projected gradient descent) method to generate targeted AEs on the substitute model. Finally, we apply targeted AEs which could successfully attack the substitute model to the target model, and evaluate its attack success rate. We choose the optimal model trained on the full dataset as the target model. PGD is an enhanced version of FGSM [19]. It is essentially projected gradient descent on negative loss function [37]. PGD can easily control the size of perturbations and is fast to compute. We import the PGD method from the foolbox [4] library. We set the upper perturbation (ε) limit to 128/255 after several attempts. As ε increases, the attack effect gets better, but as ε continues to increase, the effect does not change significantly. During an attack process, we randomly select 5,000 seed samples in the test dataset as a benchmark. For each sample, we generate 9 targeted AEs that are misclassified into all other categories by the substitute model. There are totally 45,000 targeted AEs, which take about 1.5 hours to generate (averagely 0.12s for one AE). Compared with untargeted AEs, targeted AEs not only make misclassifications, but also lead into the specified categories, which are more difficult for generation.

Through attacking the target model with AEs, we calculate transferability and draw confusion matrices. In Table 8, we evaluate LeNet-5 and C3F2 model structures and compare with state-of-the-art PRADA [29] and Practical [45]. The experimental environment is on the MNIST dataset. AEs are from 5,000 randomly selected normal samples in the test dataset. The upper perturbation size is set as 128/255. In DRMI, the AEs transferability reaches 66% under only 50 queries, while 22% in PRADA. Our approach is nearly three times as much as them. The accuracy of our substitute model is also 7% higher than them. As the number of queries increases, the transferability also increases, and both our transferability and accuracy are higher than PRADA. In 150, 200, and 300 queries, DRMI under LeNet-5 model all increase 3% accuracy than PRADA, and our attack success rates achieve 68%, 69%, and 70%, respectively, and increase 39%, 38%, and 30% than PRADA. Under 600 queries, our targeted AEs attack success rate is as high as 72%, 23 percentages higher than PRADA. In PRADA, the transferability reaches 64.64% under 3,200 queries. Even though the attacker only has a very small dataset (only 1,000 samples), DRMI still raises 20%, 25%, 21% in transferability, outperforming PRADA under 50, 150, 300 queries, and also has a better model accuracy. DRMI of LeNet-5(1,000) also raises 15.40%, 8.88%, 6.13%, 4.76%, 4.50% model accuracy and 23.62%, 27.45%, 29.90%, 27.70%, 26.74% attack success rate under 50, 150, 200, 300, 600 queries than Practical [45], respectively.

Among these model structures, the C3F2 model has a

higher attack success rate than LeNet-5 in most cases except 50 queries. With 50 queries, C3F2 reaches 48.80% transferability, two times as much as PRADA, but lower than LeNet-5. While with 150 and 200 queries, the success rate of C3F2 is slightly higher than that of LeNet-5, nearly 70%. Until 300 and 600 queries, C3F2 model reaches 76.37% and 78.51% transferability, both nearly 7 percentages higher than LeNet-5, and almost 30 percentages higher than PRADA with 600 queries. Results are affected by the model's complexity since C3F2 has one more convolutional layer than LeNet-5.

Remark: 1) Transferability increases as the query number increases. 2) Larger ε *helps transferability of AEs to a certain extent. 3) More complex model structure has better transferability.*

Figure 4 shows the confusion matrices of targeted AEs attacking target model with structure LeNet-5 under 50 and 600 queries. The value in i-th row, j-th column represents the number of samples whose original label is i and which is classified into j. The diagonal elements are failed attack numbers. Other elements are succeeded attack samples. Lighter color means larger value. As we can see in 50 queries, the (2,2) element is the lightest, which means many adversarial samples generated by images of label 2 did not succeed in the attack. Although the total attack success rate is 66% under 50 queries, the success rate is 36.4% for label 2. In 600 queries, we improve this situation. The (2,2) element turns darker and the success rate reaches 57.0% for label 2. In Figure 5, we can see the confusion matrices of targeted AEs attacking C3F2 under 150 and 600 queries. It achieves 69.6% accuracy at 150 queries, but AEs from different labels also have very different transferability. Label 2 still performs worst, its attack success rate is 44.8%. Label 3, 5, 7 also perform not well. Label 1 attacks best, whose AEs achieve 95.4% attack success rate. Confusion matrix under 600 queries contains a higher success rate of 78.5%. The (2,2) element is not so bright as in 150 queries. The attack success rate of label 2 achieves 53.9%. Label 1 also attacks best with 97.9% success rate. We can find that adversarial samples of label 2 are the most difficult to attack successfully. For other labels, we can intuitively feel that our attack success rate is high.

Remark: Different labels have different attack success rates of AEs. This is because different category has different boundaries, causing different density of AEs. This phenomenon is ubiquitous and does not affect the results, where the success rate is averaged on all labels.

We also generated untargeted adversarial examples to attack the target model, and compare with the state-of-the-art Gradient Estimation (GE) [7] in Table 9. GE queries the target model for 196 times and utilizes the acquired information to generate 1,000 untargeted AEs in 11s. These AEs achieve a 61.5% attack success rate on the target model. In DRMI, we use 150 queries to generate 1,000 untargeted AEs in 2 minutes. We achieve 71.3% success rate on LeNet-5 and 73.2% on C3F2. Our DRMI still improves the attack success rate by 11.7% than GE with even 46 fewer queries. The extra time is affordable. We can generate an AE in only about 0.12s. This comparison shows DRMI also performs effectively in a untargeted attack. Moreover, we perform untargeted attacks on the ImageNet dataset using Inception-v3 as shown in Table 10. One thousand images are randomly selected from the test set for evaluation. This experiment adopts the PGD [37] attack under L_2-norm. Results show that DRMI outperforms NES [25] and AutoZoom [59], and has similar performance with P-RGF [11].

Table 9: Attack success rates of untargeted AEs between DRMI and Gradient Estimation [7] on the MNIST dataset. We set ε (max perturbation) as 0.3, and test the success rate of 1,000 untargeted AEs for each experiment.

Method	Attack Success	Queries	Time per AE(s)
Gradient Estimation [7]	61.5%	196	0.011
DRMI on LeNet-5	71.3%	150	0.126
DRMI on C3F2	73.2%	150	0.113

Table 10: Attack success rates of untargeted AEs on the ImageNet dataset. We set perturbation ε as $\sqrt{0.001 \cdot D}$, and D is the input dimension ($\approx 270,000$) [48].

Method	Attack Success	Queries
NES [25]	95.5%	1718
AutoZoom [59]	85.4%	2443
P-RGF [11]	96.5%	1119
DRMI	96.6%	1100

Remark: Through these experiments, our substitute models have achieved a higher transferability with fewer queries, outperforming the state-of-the-art approaches. It proves that our substitute models generated by DRMI *can accurately imitate the decision boundaries of the target model, and thereby facilitate black-box attacks (e.g., adversarial attacks) against deep learning.*

5.4 Interpretability of Data Reduction

Training data can be reduced without losing too much accuracy, which implies the existence of redundancy in data. Therefore, data reduction can be regarded as redundancy eliminating. To answer RQ3, we implement another three metrics to measure data redundancy: *correlation matrix*, *class probability of prediction*, and *trace of activated neurons*. With these metrics, we evaluate their effectiveness in the same manner, and provide a number of insights on interpretability.

5.4.1 CMAL: Correlation Matrix

Correlation matrix reflects the overall correlation among data samples, and is a measure of data polymerization as a whole [65]. For a data point $\mathbf{x} = [x_1, x_2, ..., x_n]^{\mathrm{T}}$, its correlation

Table 11: Comparison between CMAL and DRMI on the MNIST dataset.

Method	Dataset Size	Test Accuracy	Test Loss	Epoch
CMAL	600 (1%)	90.16%	0.6071	30
DRMI	600 (1%)	**96.41%**	**0.1961**	30
CMAL	300 (0.5%)	89.81%	0.5392	20
DRMI	300 (0.5%)	**94.14%**	**0.2475**	20
CMAL	150 (0.25%)	83.32%	0.7078	15
DRMI	150 (0.25%)	**92.13%**	**0.2604**	15

Table 12: Effectiveness with class probability on MNIST. LCP means that data has low class probability, and the model classifies it correctly with low confidence.

Method	Dataset Size	Test Accuracy	Test Loss	Epoch
HCP	600 (1%)	90.96%	0.5288	26
LCP	600 (1%)	77.59%	0.8068	13
K-Means	600 (1%)	93.53%	0.3439	30
PCA + K-Means	600 (1%)	91.72%	0.4289	30
DRMI	600 (1%)	**96.41%**	**0.1961**	30
K-Means	300 (0.5%)	88.45%	0.6852	30
PCA + K-Means	300 (0.5%)	88.05%	0.5923	30
DRMI	300 (0.5%)	**94.14%**	**0.2475**	20
K-Means	150 (0.25%)	79.82%	1.1851	30
PCA + K-Means	150 (0.25%)	80.31%	0.6033	30
DRMI	150 (0.25%)	**92.13%**	**0.2604**	15

matrix is $\mathbf{x}\mathbf{x}^{\mathrm{T}}$. For a dataset $\mathbf{X}_m = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m]$, its corelation matrix is $\mathbf{R}(\mathbf{X}_m) = \mathbf{X}_m\mathbf{X}_m^{\mathrm{T}}/m$. CMAL selects a simplified dataset $\mathbf{S}$ from the whole dataset $\mathbf{D}$ which minimizes the value $||\mathbf{R}(\mathbf{S}) - \mathbf{R}(\mathbf{D})||^2$. CMAL tends to extract standard, moderate, and average-performing samples, rather than independent, diverse, and representative ones. We implement the correlation matching based active learning (CMAL) [65] and compare its performance with our approach.

In Table 11, we adopt a LeNet-5 model to evaluate the accuracy and loss value of the DRMI and CMAL methods. We find that our method performs much better (*i.e.*, higher accuracy yet lower loss) than CMAL in all dataset sizes. DRMI increases 6%, 5%, and 9% accuracy on 600, 300, and 150 dataset sizes than CMAL, respectively.

Remark: According to our investigation, the reason why CMAL *performs worse is that this sampling is prone to choosing more averaged than diverse data. Although the selected data follows a similar distribution with the whole dataset, the model cannot learn distinctive features from them and thereby performs under our exceptions. As a result, it proves that the correlation matrix based reduction likely removes distinctions that could degrade the performance of data reduction.*

5.4.2 CPB: Class Probability of Prediction

High class probability (hereafter referred to as HCP) of data indicates that the model classifies it correctly with high confidence. In our experiments, HCP data points are first sorted

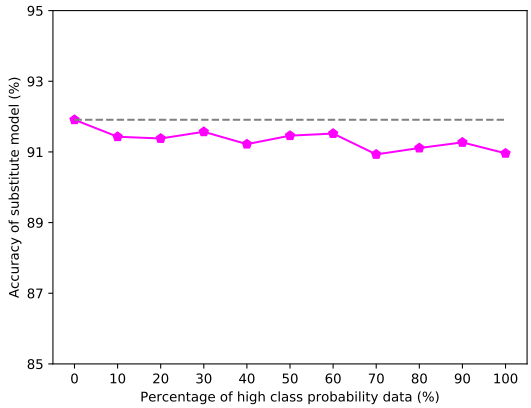

Figure 6: The effect of high class probability data on the accuracy of substitute models.

in order of confidence scores of the correct class from high to low, and then selected in order until filling the simplified training set of fixed size. In our general cognition, the data with higher class probabilities during the testing process can reflect the logical relationship with the target model much better. In [42], they also use class probability returned by the target model as a measure. Here we are eager to verify more directly whether HCP data is more useful for training substitute models.

In Figure 6, the gray dotted line is the model accuracy from a randomly reduced dataset, and x-axis is the percentage of HCP data. For a dataset with randomly selected 600 samples, we start to replace a portion ($10\% \sim 100\%$) of data with HCP and observe the impact of HCP data on the accuracy of the substitute model. We find that the increase of HCP data does not raise the accuracy of the substitute model, but lowers it down slightly. It shows that HCP data does not contribute more than random data for training substitute models.

Furthermore, we try to categorize data based on class probabilities by K-Means clustering. We treat prediction confidence scores after the softmax layer as feature vectors, use L_2 to measure the distance between two points, and perform K-Means to form k independent clusters. For each cluster, we select the data that is nearest to the centroid, and finally obtain a reduced dataset with k samples. In Table 12, we test K-Means on varying sizes from 150 to 600, which performs worse than DRMI with decreasing the accuracy by 2.9%, 5.7%, and 12.3% on sizes 600, 300, and 150, respectively.

Remark: We investigated the formed k clusters and finally selected samples in the experiment to explain its unsatisfied performance. We find that the selected samples are more likely to be picked at random, seriously deviating from our expectations. It is due to the features of high-dimensional data: the points (under this context) in the high-dimensional space have nearly equal euclidean distances between each other. Therefore, K-Means cannot effectively separate these

Table 13: Effectiveness using activated neurons trace information under 600 dataset size. In Target, "MIN" means we find the minimum hamming distance sum, while "MAX" refers to the maximum hamming distance sum.

Method	Target	Initial Solu.	Test Acc.	Test Loss	Epoch
TRACE	MIN	min-sum	67.21%	2.3855	15
TRACE	MIN	min-max	60.53%	3.3914	15
TRACE	MAX	min-sum	79.10%	0.9326	15
TRACE	MAX	min-max	72.67%	1.6328	15
DRMI	-	-	96.41%	0.1961	30

samples. It reveals class probability has been pruned with the diversity in euclidean space.

To solve the curse of dimensionality, we apply a principal components analysis (PCA) before K-Means. However, it still brings no noticeable improvement in Table 12. The CPB method, even with PCA, fails largely due to the deep transformation from input to output by DNNs. As claimed in [54], the original data features fade away but the essential features for abstract outputs remain and get enhanced during training. Data redundancy is apparently discarded in the course, so that using class probability can only tell how different of their predictions but definitely not the input data.

5.4.3 TRACE: Trace of Activated Neurons

DNN is one kind of data model which transforms a sort of data into another. Generally, there are scattering lots of neurons internally to accomplish the transformation. When a data sample enters the model, it will activate a number of neurons, and then reach the final result. As such, it leaves a trace during passing through the deep learning model. This kind of traces have been employed for multiple purposes [46, 52]. Here we explore whether it is suitable for measuring data redundancy.

For simplicity, we use $M = (L^i),\ i < n$ to denote a n-layer DNN, where L^i is i-th layer in the model. For each layer, there may be varying numbers of neurons. We define $L^i = (s^i_1, s^i_2, ...s^i_m)$ as the i-th layer with m neurons, and s^i_j denotes the activation state of neurons. If the current neuron is activated, the value of s^i_j is 1, otherwise 0. Hence L^i is a binary string of length m, and m is the neuron number in the i-th layer. We assume Tr is a binary string of length l, and l is the total number of neurons in the model. Binary string Tr_a represents the activated neurons path for data a. Then we calculate the Hamming distance (performing an xor operation on two strings and counting the number of "1"s in the result) of Tr_a and Tr_b, to represent the distance of data a and b. Then we replace the MI matrix with the Hamming distance as follows:

$$I[a][b] = Hamming(Tr_a)(Tr_b) \tag{11}$$

Finally, we adopt Algorithm 1 to obtain a simplified dataset. Here we try two directions–smallest and largest Hamming distance sets. In Table 13, we test TRACE methods under 600 samples. All of TRACE methods perform worse than DRMI, decreasing 17.3% to 35.9% accuracy. In TRACE, we find "MAX" target performs better than "MIN" target, increasing 11.89% in min-sum and 12.14% in min-max initial solution. This indicates the set with larger hamming distance has better effect. We need to make traces of activated neurons more diverse and cover as more neurons as possible.

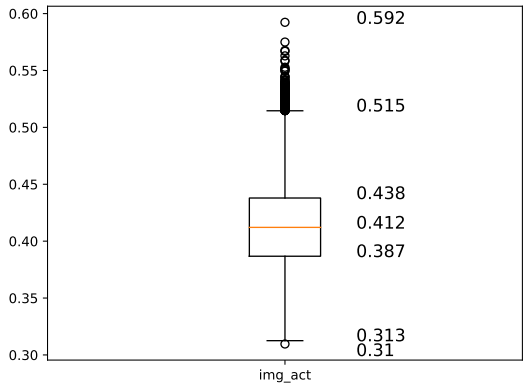

Figure 7: The box-plot of the proportion of activated neurons in all data. The ordinate is the proportion of activated neurons.

In order to study why activation traces could not filter out a good simplified training dataset, we analyze the distribution of the proportion of activated neurons and draw a box-plot in Figure 7. The proportions are almost all concentrated at $[0.313, 0.515]$, within a small interval. Even 50% data activated neurons proportions are concentrated at $[0.387, 0.438]$, a very small interval. This may be the cause of poor performance to select data through the activation neuron trace.

Remark: The TRACE method by considering the Hamming distance between activated neurons traces performs worse than DRMI. *The proportions of activated neurons in all predicted samples are almost concentrated in a small interval.*

6 Related Work

There has been a line of related research described as below.

Black-box Attacks. Many black-box attacks (*e.g.*, adversarial attacks) need to train a substitute model [29, 44, 45]. Techniques have been developed to reduce queries as much as possible. Papernot *et al.* [44] adopted *reservoir sampling* method and successfully reverse-engineered two machine learning classification systems. In order to reduce the query number, Papernot *et al.* [45] adopted Jacobian-based dataset augmentation (JbDA) to create synthetic data for training DNNs on MNIST. Based on JbDA, Juuti *et al.* [29] proposed Jb-topk and Jb-self methods to synthesize samples for substitute model training. *Differently,* DRMI *relies on data reduction from a large dataset for querying. But PRADA augments data locally for training that may induce wrongly labeled data. Through the experiments in Section 5.2, it proves* DRMI *can*

achieve more accurate substitute models using the same or fewer queries. Orekondy *et al.* [42] stole the functionality of target models by querying. They use three metrics to choose images: images with higher class probabilities, images with diverse labels, and images which imitates badly. *According to our results, the sole selection of images with higher class probabilities cannot augment accuracy of the trained model.*

Jagielski *et al.* [26] propose a learning-based extraction method using semi-supervised learning techniques: rotation loss and MixMatch. For adversarial capabilities, they need both labels and scores from the original model, while DRMI only needs labels. Their adversary has access to the same training set without labels, but DRMI does not need the exact training data. They can save the query costs because much unlabeled data does not need to be queried in semi-supervised learning. Based on this analysis, we can incorporate their method into ours in future: use DRMI to select the query data for fully supervised learning, and perform semi-supervised learning on the remaining unlabeled data.

Data reduction. Eschrich *et al.* [17] reduced the amount of clustering data by aggregating similar samples and using weighted samples. Ougiaroglou *et al.* [43] reduced data in clustering by producing homogeneous clusters. It reduced storage requirements and had low pre-processing cost. Chouvatut *et al.* [13] proposed a graph-based optimum-path forest to reduce the size of training sets. They utilized the segmented least square algorithm to estimate the tree's shape. In DNNs, Zheng *et al.* [65] proposed a correlation matching based active learning technique to label the most informative data and simplify the dataset. *We implemented it in our experiments for a comparison. Results show* DRMI *performs remarkably better than it in CNNs.* Katharopoulos *et al.* [31] found that not all samples in the training phase are equal. Hence, they adopted importance sampling to identify informative examples, which can reduce the variance of a SGD process. DRMI *aims to reduce the queries and the reduction can be completed before training, therefore,* DRMI *is model-independent, i.e., not affected by model structures and training processes.*

7 Discussion

Effectiveness of DRMI. In this study, we use mutual information to measure the data redundancy of a dataset, and then find a subset to minimize the summed mutual information. As claimed in Section 4.2, the problem is NP-Complete and cannot be solved in polynomial time. Therefore, we propose DRMI to solve the intractableness. Its effectiveness is twofold. On one hand, DRMI can find an approximate optimal solution by enumerating the starting point (Algorithm 1) and filling an initial subset for representative data (Algorithm 2) to avoid the trap of local optimum. On the other hand, one-hot replacement (Algorithm 3) replaces the vertices that incur large mutual information and identify the optimal solution in the current setting. Based on the complexity analysis for each algorithm at Section 4.3, the overall complexity of DRMI is $O(kn^2)$, which can be further optimized to $O(kn\log n)$ with more efficient sorting algorithms. It is also confirmed by the experiments on three diverse and large-scale datasets.

Parameter Choice (α, ε). α is used to tune the variable relationship between mutual information and data redundancy. It indicates a linear relationship if $\alpha = 1$. Moreover, we explore whether there are non-linear relationships by augmenting α to 2 and 4. The results in Table 3 and 7 show the increase of α (from 2 to 4) hardly improves accuracy, but the extra overhead caused by exponent computation cannot be offset by accuracy gains. Similarly, it is experimentally confirmed that this principle also applies to the other datasets CIFAR10 and ImageNet. As for the max perturbation ε, it reflects the balance between attack effect and imperceptibility of AEs. Larger ε can raise the attack effect, but reduce the imperceptibility. For ease of comparison, we set $\varepsilon = 0.3$ for untargeted AEs, consistent with [7] and $\varepsilon = 0.5$ for targeted AEs on the MNIST dataset. It is because targeted AEs usually need larger perturbations for generation, and the value is also aligned to PRADA. Inspired by [11], we set $\varepsilon = \sqrt{0.001 \cdot D}$ on the ImageNet dataset, and D is the input dimension ($\approx 270,000$).

Robustness of DRMI. There is a line of work to defend such black-box attacks. For instance, [58] proposes a number of strategies to prevent model stealing, including rounding confidence scores, providing fake or no class probability. However, we show that DRMI is still effective without class probability in Section 5.2, making this defense ineffective. PRADA [29] also proposes a defensive method by detecting abrupt changes in the distribution of queried samples. It detects PRADA's attack after 100 queries on MNIST. We re-implement this method to detect DRMI. In our experiments, we assume that normal users submit random queries, which reduces the detection difficulty. Through our results, DRMI can successfully create a high quality substitute model after only a few hundred queries on MNIST, while it takes about 32,000 queries for PRADA to detect our attack. So PRADA is not effective at stopping our attack. Our queries are not easily detected by AEs detection methods, such as adversarial training, defensive distillation, and input transformation [20]. Because our queried samples contain no adversarial perturbations. In addition, we use mainstream methods (*e.g.*, PGD [37]) to generate AEs. Although they are likely to be detected by defensive methods like [20], it is not the concern of this study.

One possible defense is to measure the redundancy of queries from one client, just alike DRMI. Generally, the queries of DRMI have a much smaller MI value compared to the normal samples of the same number, since normal data have relatively more repetitions. However, this method needs to count many queries and establish a distribution of MI values. In our test, the defender needs to have more than 100 times malicious queries for detection. It inevitably brings huge computational cost. Additionally, this defense becomes more infeasible in front of distributed queryings.

8 Conclusion

This paper proposes a novel dataset reduction technology based on mutual information DRMI, which can be used in black-box attacks. With this approach, we can accurately measure the overall quality of dataset, identifying redundancies and repetitions therein. Compared with other three techniques, it proves that our approach achieves the best performance in the selection of representative and distinct data for DNN training. Moreover, we apply DRMI to reduce queries in model extraction and adversarial attacks. The results show a superior ability of DRMI in data reduction while maintaining a high model accuracy and transferability of adversarial examples.

Acknowledgement

We thank our shepherd David Wagner for his valuable guidance and assistance and all the anonymous reviewers for their constructive feedback. The authors are supported in part by the National Key Research and Development Program of China under Grant No.2020AAA0107800, NSFC U1836211, NSFC 61902395, Beijing Natural Science Foundation (No.JQ18011), National Top-notch Youth Talents Program of China, Youth Innovation Promotion Association CAS, Beijing Academy of Artificial Intelligence (BAAI), CCF-Tencent Open Fund, and a research grant from Huawei.

References

[1] USPS dataset. https://www.kaggle.com/bistaumanga/usps-dataset.

[2] Mutual information. https://en.wikipedia.org/wiki/Mutual_information, 2019.

[3] Pytorch. https://pytorch.org/, 2020.

[4] Welcome to foolbox native. https://foolbox.readthedocs.io/en/latest/, 2020.

[5] Naveed Akhtar and Ajmal S. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[6] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *CoRR*, abs/1912.00049, 2019.

[7] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Exploring the space of black-box attacks on deep neural networks. *CoRR*, abs/1712.09491, 2017.

[8] Thomas Brunner, Frederik Diehl, Michael Truong-Le, and Alois Knoll. Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks. *CoRR*, abs/1812.09803, 2018.

[9] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP, San Jose, USA*, pages 39–57.

[10] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *29th USENIX Security Symposium, August 12-14, 2020*, pages 2667–2684.

[11] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *NeurIPS 19, Vancouver, Canada*, pages 10932–10942, 2019.

[12] Kashyap Chitta, Jose M. Alvarez, Elmar Haussmann, and Clement Farabet. Training data distribution search with ensemble active learning, 2019.

[13] V. Chouvatut, W. Jindaluang, and E. Boonchieng. Training set size reduction in large dataset problems. In *2015 International Computer Science and Engineering Conference (ICSEC)*, pages 1–5, Nov 2015.

[14] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley series in telecommunications. Wiley, 1991.

[15] Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, Workshop Track Proceedings*.

[16] Xiaoning Du, Xiaofei Xie, Yi Li, Lei Ma, Yang Liu, and Jianjun Zhao. Deepstellar: Model-based quantitative analysis of stateful deep learning systems. In *27th ACM Joint Meeting on ESES/FSE*, New York, NY, USA, 2019.

[17] Steven Eschrich, Jingwei Ke, Lawrence O. Hall, and Dmitry B. Goldgof. Fast accurate fuzzy clustering through data reduction. *IEEE Trans. Fuzzy Systems*, 11(2):262–270, 2003.

[18] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *2018 ACM SIGSAC Conference on CCS*, pages 619–633, Oct. 2018.

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd ICLR, San Diego, CA, USA, 2015*.

[20] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th ICLR, Vancouver, BC, Canada, 2018*.

[21] Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace Attack: Exploiting Promising Subspaces for Query-Efficient Black-box Attacks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3825–3834. 2019.

[22] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier, 2012.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. Towards Security Threats of Deep Learning Systems: A Survey. *IEEE Transactions on Software Engineering (TSE)*, pages 1–28, 2020.

[25] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden*, pages 2142–2151, 2018.

[26] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1345–1362. USENIX Association, August 2020.

[27] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy, San Francisco, USA*, pages 19–35.

[28] Uyeong Jang, Xi Wu, and Somesh Jha. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *33rd Annual Computer Security Applications Conference, Orlando, FL, USA*, pages 262–277.

[29] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. PRADA: protecting against DNN model stealing attacks. In *IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden*, pages 512–527.

[30] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, New York, USA*, pages 85–103, 1972.

[31] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden*.

[32] Alex Krizhevsky. The CIFAR-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html/.

[33] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, Workshop Track Proceedings*.

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

[35] Yann LeCun. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[36] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. DEEPSEC: A uniform platform for security analysis of deep learning model. In *2019 IEEE Symposium on Security and Privacy, SP, San Francisco, USA*, pages 673–690.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada*.

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*, pages 86–94.

[39] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *AISec@CCS 2017, Dallas, TX, USA*, pages 27–38.

[40] Seong Joon Oh, Max Augustin, Mario Fritz, and Bernt Schiele. Towards reverse-engineering black-box neural networks. In *International Conference on Learning Representations*, 2018.

[41] Lucila Ohno-Machado, Hamish S. F. Fraser, and Aleksander Øhrn. Improving machine learning performance by removing redundant cases in medical data sets. In *AMIA 1998, Lake Buena Vista, FL, USA*.

[42] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA*, pages 4954–4963.

[43] Stefanos Ougiaroglou and Georgios Evangelidis. Efficient dataset size reduction by finding homogeneous clusters. In *Balkan Conference in Informatics, BCI 2012, Novi Sad, Serbia*.

[44] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

[45] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM AsiaCCS 2017*, pages 506–519.

[46] Haekyu Park, Fred Hohman, and Duen Horng Chau. Neuraldivergence: Exploring and understanding neural networks by comparing activation distributions. *CoRR*, abs/1906.00332, 2019.

[47] Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *NeurIPS 2017, Long Beach, USA*, pages 3236–3246.

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[49] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS 2018, Montréal, Canada*, pages 6106–6116.

[50] Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *CVPR Workshops 2018, Salt Lake City, UT, USA*, pages 1605–1613.

[51] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, San Jose, USA*, pages 3–18.

[52] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *34th ICML*, volume 70, pages 3145–3153, 2017.

[53] Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Black-box adversarial attacks with bayesian optimization. *CoRR*, abs/1909.13857.

[54] Congzheng Song and Vitaly Shmatikov. Overlearning reveals sensitive attributes. In *8th ICLR, Addis Ababa, Ethiopia, April 26-30, 2020*.

[55] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid Batch Attacks: Finding Black-box Adversarial Examples with Limited Queries. In *29th USENIX Security Symposium*, 2020.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR 2016, Las Vegas, NV, USA*, pages 2818–2826.

[57] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd ICLR, 2014, Banff, AB, Canada*.

[58] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium, 2016, Austin, TX, USA*, pages 601–618.

[59] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *31rd AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA, 2019*, pages 742–749.

[60] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP), San Francisco, California, USA*, pages 36–52.

[61] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A. Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, USA*, pages 49–64.

[62] L.A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3(2):177 – 200, 1971.

[63] Mingming Zha, Guozhu Meng, Chaoyang Lin, Zhe Zhou, and Kai Chen. Rolma: A practical adversarial attack against deep learning-based lpr systems. In *Information Security and Cryptology (Inscrypt)*, pages 4701–4708, Dec 2019.

[64] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *ACM CCS 2019, London, UK*, pages 1989–2004.

[65] Jian Zheng, Wei Yang, and Xiaohua Li. Training data reduction in deep neural networks with partial mutual information based feature selection and correlation matching based active learning. In *IEEE ICASSP 2017, New Orleans, LA, USA*.