IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 May 2024, accepted 26 May 2024, date of publication 30 May 2024, date of current version 14 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3407097

METHODS

D-BADGE: Decision-Based Adversarial Batch Attack With Directional Gradient Estimation

GEUNHYEOK YU, MINWOO JEON, AND HYOSEOK HWANG, (Member, IEEE)
Department of Software Convergence, Kyung Hee University, Yongin-si 17104, Republic of Korea

Corresponding author: Hyoseok Hwang (hyoseok@khu.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korean Government (MSIT) under Grant RS-2022-00167169, in part by IITP grant funded by Korean Government (MSIT) [Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University)] under Grant RS-2022-00155911, and in part by the Convergence Security Core Talent Training Business Support Program under Grant IITP-2023-RS-2023-00266615.

ABSTRACT The susceptibility of deep neural networks (DNNs) to adversarial examples has prompted an increase in the deployment of adversarial attacks. Image-agnostic universal adversarial perturbations (UAPs) are much more threatening, but many limitations exist to implementing UAPs in real-world scenarios where only binary decisions are returned. In this research, we propose D-BADGE, a novel method to craft universal adversarial perturbations for executing decision- To primarily optimize perturbation by focusing on decisions, we consider the direction of these updates as the primary factor and the magnitude of updates as the secondary factor. First, we employ Hamming loss that measures the distance from distributions of ground truth and accumulating decisions in batches to determine the magnitude of the gradient. This magnitude is applied in the direction of the revised simultaneous perturbation stochastic approximation (SPSA) to update the perturbation. This simple yet efficient decision-based method functions similarly to a score-based attack, enabling the generation of UAPs in real-world scenarios, and can be easily extended to targeted attacks. Experimental validation across multiple victim models demonstrates that the D-BADGE outperforms existing attack methods, even image-specific and score-based attacks. In particular, our proposed method shows a superior attack success rate with less training time. The research also shows that D-BADGE can successfully deceive unseen victim models and accurately target specific classes.

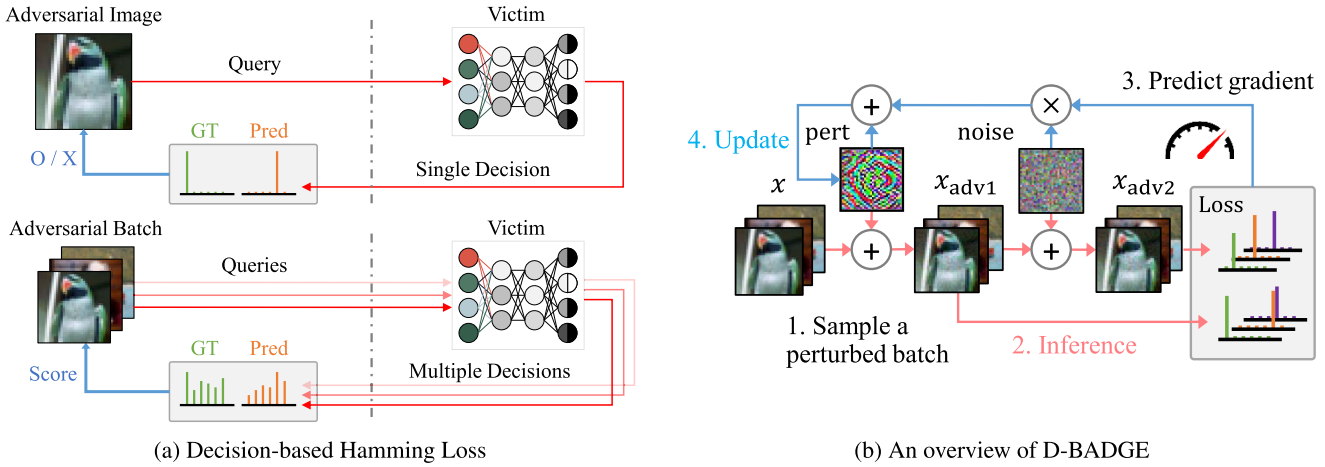
INDEX TERMS Deep neural networks, universal decision-based adversarial attack, image classification, representation learning, vulnerability, zeroth-order optimization.

I. INTRODUCTION

Deep Neural Networks (DNNs) are considered among the most versatile and sophisticated machine learning architectures. Optimization algorithms refine the networks' parameters by autonomously identifying the optimal decision boundaries. For this reason, revolutionary progress has been achieved in numerous computer vision tasks [1], [2]. However, it has been proven that DNNs are highly vulnerable to adversarial examples (AEs), which are indistinguishable from the original image by adding a tiny amount of adversarial perturbation [3]. This can be critical, as adversarial attacks using AEs threaten the safety of DNN-based applications. They can confuse the networks [4], compromise privacy [5], [6], duplicate or steal a model [7], [8], and intentionally manipulate the model's decisions [9], [10], [11], [12], [13].

Since the introduction of Universal Adversarial Perturbation (UAP) by Moosavi-Dezfooli et al. [10], adversarial attacks have become even more menacing. AEs can be generated from any input image by simply applying an image-agnostic perturbation, allowing the UAP to capture the entire decision boundary of a victim model [14]. Unlike image-dependent methods, when generating an AE for an input image, once the UAP is created, it only needs to be added to the input image, ensuring that AEs are generated promptly. Furthermore, this fooling technique is easily transferable across other network models. When the DNNs prevail in industry, service providers can be requested to tune a

The associate editor coordinating the review of this manuscript and approving it for publication was Shovan Barma.



(a) Decision-based Hamming Loss

(b) An overview of D-BADGE

FIGURE 1. **(a) shows the difference between the binary decision loss and the proposed Hamming loss that aggregates multiple binary decisions to form a continuous score. (b) illustrates the D-BADGE framework that utilizes the SPSA algorithm with the Hamming loss.**

customer's network for generalization or fine-tuning. It is trivial that the more information provided to the service provider, the better the result will be presented.

Fortunately, adversarial attacks have yet to cause critical problems in real-world scenarios. This is due to two characteristics of a closer real-world scenarios: black-box and decision-based environments. From a certain perspective, attacks in this setting must be most considered because it can happen anywhere DNNs are used. In the black-box environment, the victim model is typically unknown. Therefore, the white-box approaches with the superior performance of optimizing perturbations via backpropagation through the model are no longer available [12], [13], [15], [16].

Papernot et al. [17] and Shi et al. [18] solved this problem by utilizing a substitute model which permits white-box access to the attacker. These kinds of methods allow the attacker to craft adversarial perturbations without accessing the victim model at all. However, the attackers have to prepare an appropriate substitute model for a black-box victim model. Or without, the perturbation should be able to generalize arbitrary victim networks. Both are extremely challenging. Recent focus has shifted towards techniques that identify decision boundaries within a black-box environment using minimal queries, highlighting a potential solution to this challenge [15], [19]. Although these methods hold promise for real-world applications, their utility in crafting UAPs is limited, as they produce perturbations specific to individual images.

Other methods that utilize the zero-order optimization methods demonstrate the feasibility of generating UAPs using only decisions. The optimization process is significantly constrained by the minimal information available from individual decisions, which adversely affects performance. Score-based approaches, which provide distributions rather than binary decisions in response to queries, offer a partial solution by delivering more precise guidance on the magnitude necessary for learning. Nonetheless, these approaches remain inadequately aligned with the real-world scenarios. Building on the idea that the problem can be tackled by estimating the output distribution exclusively through decisions, we approach this by treating the collective decisions within a batch as a measure of the distribution's spread.

In this paper, we propose ***D**ecision-based **B**atch **A**ttack with **D**irectional **G**radient **E**stimation (D-BADGE)*, which aims to efficiently craft universal adversarial perturbations in the decision-based black-box attacks. We propose the Hamming loss function, based on the Hamming distance, to precisely measure the loss magnitude from binary decisions, which accumulates the distance between distributions on a mini-batch as illustrated in Figure 1a. The Hamming loss is applied to the revised SPSA and utilized to determine the magnitude of the update (see Figure 1b). The proposed method is effectively extensible, allowing other distance metrics and attack methods, such as targeted attacks or score-based attacks, to be easily applied. The proposed method achieves a white-box level attack success rate with a similar number of perturbation updates. The main contributions of this study can be summarized as follows:

- We mainly propose a novel method to improve optimization performance using the batch Hamming loss with a distribution of decisions in a decision-based black-box attack, where our knowledge is confined to the decisions of queries.
- We mathematically formulated the loss function in the context of error detection based on the Hamming distance and compared it with other existing loss functions.
- We analyzed the effectiveness of a straight forward combination of SPSA and Adam optimization algorithms on decision-based universal attacks.
- Our method overperforms other methods in terms of training time efficiency while achieving white-box level attack success rate on both convolutional networks and transformer-based networks.

II. RELATED WORKS

A. UNIVERSAL ADVERSARIAL ATTACK

Traditional adversarial attack methods are image-dependent attacks. It refers to exploiting the victim model by optimizing perturbation for one image. UAP based on DeepFool (DF-UAP) [10] was introduced in which a perturbation can be applied to any input image in contrast to image-dependent perturbations by iterative boundary search [9]. Singular value-based UAP [20], which is relatively data-efficient to build compared to UAP, was demonstrated. Network Adversary Generation (NAG) [21] is an adversarial example generator based on Generative Adversarial Networks (GAN) [22]. They demonstrated that generators can capture the perturbation geometry and achieve high fooling transferability. Generative Adversarial Perturbation (GAP) is another GAN-based adversarial perturbation generator [23]. NAG and GAP are capable of generating not only image-dependent adversaries but also UAPs.

B. BLACK-BOX ADVERSARIAL ATTACK

The attacker cannot access the victim's architecture, gradient, or training process in a black-box attack. We classified some approaches by the provided information to the attacker.

1) TRANSFER-BASED METHODS

Transfer-based methods do not directly access the victims but require a substitute model. Local Substitute Network [17] trained perturbations using a known network and attacked an unknown victim model. Curls and Whey [18] is another black-box attack method using a substitute network. The authors tried to boost the attack process by finding a faster trajectory to make the data point to cross the decision boundary. Translation-Invariant Attack [24] tackled true translation-invariance to attack convolutional neural networks properly.

2) SCORE-BASED METHODS

These methods take the rich confidence score for each query. Zeroth-order Optimization (ZOO) [25] proposed a black-box attack method by optimizing randomly selected pixels using Newton's Method. Simple Black-box Adversarial Attacks (SimBA) [26] is another powerful black-box attack method. They proposed to use a set of orthonormal vectors as a direction set for local search.

3) DECISION-BASED METHODS

In a real-world scenario, the victim service provides decisions only. Therefore, decision-based attacks have been researched from the perspective of query efficiency. RGF optimization was applied to solve this problem [27], [28]. The local search algorithm was also applied [16] but they further applied the Biased Boundary Attack to build low-frequency perturbations. Reliable attack precisely selects the magnitude of an update [15], [19]. The genetic algorithm was proposed to be applied, finding the optimal step on a sparse decision space [29]. Wu et al. introduced Decision-based Universal Attack (DUAttack), which is an algorithm to build a universal perturbation using decisions [11]. They aggregated multiple images into a mini-batch to properly update the perturbations. To the best of our knowledge, this was the only successful approach to solving the decision-based universal attack problem. However, DUAttack does not successfully perform on transformer-based victim networks. Self-attention, the atomic operation of transformers [30], [31], [32], first patchifies the input to project each of those, and outputs the weighted sum of them. This discards the benefit of diagonal perturbation and DUAttack becomes less effective to transformer-based networks compared to CNNs.

C. ZEROTH-ORDER OPTIMIZATION

Several methods have been studied to apply gradient-based optimization to craft adversarial perturbations without the gradient of victim models. Natural Evolutionary Strategy (NES) was employed to adversarial attack to optimize without calculating gradients [33], [34]. Some researchers addressed this problem by employing the RGF algorithm [27]. RGF is a basic zeroth-order optimization algorithm based on so-called "try one direction, adopt as much it worths" strategy. Most adversarial attacks based on the RGF aim to identify effective random steps. Randomly sampled steps on a hypersphere can be used as an update with the momentum optimization method [28]. The generator was also employed in zeroth-order optimization to generate a step using neural networks [35]. These methods effectively addressed score-based attack problems using the RGF. SPSA is another method for solving zeroth-order optimization problems [36]. SPSA tries one random direction and its opposite direction to precisely measure the value of the direction. Then, the magnitude of the reciprocal of the sampled direction is adopted. It can be applied to address black-box adversarial attack problems [37], [38]. SPSA with gradient correction (SPSA-GC) [39] improved the direction of the steps in the black-box prompt tuning task by integrating the SPSA algorithm with the Nesterov Accelerated Gradient optimization algorithm [40]. In black-box attacks, it was proven that selecting better directions improves performance. Specifically, geometrical approaches achieved adversarial attack in perspective of crossing over the decision boundaries [41], [42], [43]. Thus far, the methods NES, RGF, and SPSA have proven unsuitable for decision-based attacks, as they are unable to accurately evaluate the direction, particularly in the context of universal attacks.

III. METHODS

In this section, we introduce D-BADGE, a decision-based method incorporating a revised version of the SPSA algorithm. This revised version aggregates decisions in a batch to recover the lost distribution. We propose the Hamming loss function, a simple, yet highly effective tool for comparing two distributions. Furthermore, we provide a mathematical proof to demonstrate the feasibility of our loss function.

Model fooling is to deceive a victim classifier into making a misclassification by adding a tiny amount of perturbation to the input. Let the original input image, the perturbation vector, and the classifier be $\mathbf{x}_i, \mathbf{p}_i \in \mathbb{R}^{N_{\text{in}}}$ and $\mathcal{C} : \mathbb{R}^{N_{\text{in}}} \mapsto [0, 1]^{N_{\text{cls}}}$, respectively. As long as our objective is to make UAPs, one perturbation should be able to fool as many images as possible. Moreover, in the real-world scenario, the attacker can only acquire the victim's decision. Therefore, our objective is formulated following:

$$\min \sum_{i=1}^{N} \langle \mathbb{D}(\mathbf{x}_i), \mathbb{D}(\mathbf{x}_i + \mathbf{p}) \rangle \quad \text{subject to: } \|\mathbf{p}\|_\infty \leq \epsilon \tag{1}$$

for N input images, where $\langle \cdot, \cdot \rangle$ is inner product between two vectors and $\| \cdot \|_l$ is L_l-norm of a vector, $\mathbb{D}(\mathbf{x})$ denotes the one-hot vector corresponding to the top-1 element of $\mathcal{C}(\mathbf{x})$.

A. HAMMING LOSS

The decision of the victim model forgets score distribution. This hinders the calculation of losses by the difference between the distributions. Our crucial concept involves computing and updating the distribution in mini-batches, rather than deriving it from a single image. This allows us to calculate the distance of two batches' decision distributions and formulate another *score*. There are several distance metrics for distribution, such as Cross Entropy (CE), Kullback-Leibler Divergence (KLD), Earth Mover's Distance (EMD), and the Hamming distance. However, not all metrics are suitable for addressing this particular issue. Hamming distance is a distance metric for binary vectors introduced by Hamming. This metric is essentially the exclusive OR operation that counts the number of elements with different values. It was first proposed to measure the amount of error in data signal to precisely correct the error [44], [45]. This attribute makes it fit a decision-based attack that uses binary vectors as decisions. When attempting to apply SPSA-based optimization, the loss function also must be convex to prevent getting trapped in local minima, and it should be Lipschitz-continuous to ensure stable optimization [46].

The Hamming distance is a distance metric for two arrays consisting of binary elements, and it just fits into our problem domain because decisions are Bernoulli-distributed random variables. The range of the Hamming distance is the set $\{0, 1\}$ because the maximum value of the score survives as 1 and others turn into 0 in classification. Therefore, the Hamming distance, in this case, means the top-1 accuracy, which completely equals to the logical *and* operation of two distributions. The accuracy is a discrete distance metric; therefore, we need to transform the accuracy into continuous space in order to analyze as a loss function. We define the Hamming loss function L_H that utilizes the Hamming distance consisting of two distinct planes:

$$L_H(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{N_{\text{cls}}} \sum_{i=1}^{N_{\text{cls}}} \max(0, \mathbf{y}_{1,i} + \mathbf{y}_{2,i} - 1), \tag{2}$$

Algorithm 1 D-BADGE Algorithm

Input: $X, Y \leftarrow T$ input, ground truth batches
Parameters: $\beta_1 = 0.5, \beta_2 = 0.999, \eta = 10^{-8}$, $\delta = 0.01, \gamma = 0.001$
Output: $\mathbf{p}_{\text{uni}} \leftarrow$ optimized UAP

$\mathbf{p} \leftarrow \mathbf{0}$
$\mathbf{m}_0 \leftarrow \mathbf{0}, \mathbf{v}_0 \leftarrow \mathbf{0}$ ▷ initialize moment vectors
for t **in** 1 **to** T **do**
  $\mathbf{x}, \mathbf{y} \leftarrow X_t, Y_t$ ▷ iterating over batches

  $\mathbf{u} \leftarrow$ random-select$(-\delta, \delta)$
  $\mathbf{p}^-, \mathbf{p}^+ \leftarrow$ clip$(\mathbf{p} - \mathbf{u})$, clip$(\mathbf{p} + \mathbf{u})$
  $\mathbf{x}^-, \mathbf{x}^+ \leftarrow \mathbf{x} + \mathbf{p}^-, \mathbf{x} + \mathbf{p}^+$
  $\mathbf{x}^- \leftarrow$ clamp$(\mathbf{x}^-, \min(\mathbf{x}), \max(\mathbf{x}))$ ▷ clamp x_*
  $\mathbf{x}^+ \leftarrow$ clamp$(\mathbf{x}^+, \min(\mathbf{x}), \max(\mathbf{x}))$

  $\hat{\mathbf{y}}^-, \hat{\mathbf{y}}^+ \leftarrow \mathbb{D}(\mathbf{x}^-), \mathbb{D}(\mathbf{x}^+)$
  $\mathbf{g}_t \leftarrow \frac{L_H(\hat{\mathbf{y}}^-, \mathbf{y}) - L_H(\hat{\mathbf{y}}^+, \mathbf{y})}{\gamma \mathbf{u}}$ ▷ apply L_H

  $\mathbf{m}_t \leftarrow \beta_1 \times \mathbf{m}_{t-1} + (1 - \beta_1) \times \mathbf{g}_t$ ▷ apply Adam
  $\mathbf{v}_t \leftarrow \beta_2 \times \mathbf{v}_{t-1} + (1 - \beta_2) \times \mathbf{g}_t^2$
  $\hat{\mathbf{m}}_t, \hat{\mathbf{v}}_t \leftarrow \mathbf{m}_t / (1 - \beta_1^t), \mathbf{v}_t / (1 - \beta_2^t)$

  $\mathbf{p} \leftarrow$ clip$(\mathbf{p} + \alpha \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \eta))$
end for
$\mathbf{p}_{\text{uni}} \leftarrow \mathbf{p}$

where $\mathbf{y}_{j,i} \in \{0, 1\}$ denotes the i-th element of the vector $\mathbf{y}_{j \in \{1,2\}}$.

Convexity and Lipschitz-continuity are indicators of the feasibility of a loss function in convex optimization. Both two properties of the Hamming loss function can be shown.

Theorem 1: L_H is a convex function.

Proof: The Hessian matrix of L_H is a 2×2 zero matrix because it consists of planes.

$\therefore L_H$ is a convex function because the Hessian matrix of it is positive semi-definite 1. □

Lemma 1: Given a function $f : \mathbb{R}^{N_{in}} \mapsto \mathbb{R}^{N_{out}}$ if f is semi-positive definite $\Rightarrow f$ is a convex function.

Theorem 2: L_H is a 1-Lipschitz-continuous function.

proof:

$$\nabla L_H = \begin{cases} \mathbf{0}, & \text{if } \mathbf{y}_{1,i} + \mathbf{y}_{2,i} < 1, \\ (1, 1), & \text{otherwise}, \end{cases} \tag{3}$$

where ∇ denotes the Jacobian of a matrix. The slope of a straight line passes through two points on L_H lies between 0 and 1 (inclusive).

$\therefore L_H$ is a 1-Lipschitz-continuous function. □

Therefore, it is plausible that the Hamming loss function can be optimized using SPSA-based optimization.

B. SPSA WITH ADAPTIVE MOMENTUM

SPSA is a robust algorithm, but this controls the magnitude of updates only based on the loss value without any

corrections. We formulated SPSA with Adaptive Momentum (SPSA-AM), which is a combination of SPSA and Adam optimization algorithms. Decision-based attack using SPSA-AM is illustrated in Algorithm 1. We initialize the first perturbation $\mathbf{p} = \mathbf{0}$ and then randomly sample $\mathbf{u} \in \{\pm\delta\}^{N_{\text{in}}}$. Then, $\mathbf{u}$ serves as the step in the optimization process. We have two different perturbations: $\mathbf{p}^+$ and $\mathbf{p}^-$, by adding and subtracting the same step $\mathbf{u}$ to $\mathbf{p}$. However, this does not guarantee $\|\mathbf{p}^+\|_l, \|\mathbf{p}^-\|_l \leq \epsilon$. Therefore, we clipped the perturbation using the following equation:

$$\text{clip}(\mathbf{p}) = \epsilon \frac{\mathbf{p}}{\|\mathbf{p}\|_l}, \quad l \in \{2, \infty\}. \tag{4}$$

And build the adversarial examples $\mathbf{x}^+$ and $\mathbf{x}^-$ with two opposite directions $\mathbf{p}^+$ and $\mathbf{p}^-$ are added to $\mathbf{x}$. We calculate the pseudo-gradient $\mathbf{g}$ using the decision of the two perturbations and a batch of images. The gradient $\mathbf{g}$ does not require the decision of the clean images. Note that we clamped the adversarial examples using the lower and upper bounds of the original images. Finally, we update $\mathbf{p}$ using $\mathbf{g}$ and step size α. We clipped the perturbation again because it does not guarantee the budget constraint. The algorithm returns the final UAP $\mathbf{p}_{\text{uni}}$ after updating for all batches.

C. EXTENSIONS TO OTHER ATTACK TASKS

1) TARGETED ATTACK

We have addressed non-targeted attacks, so we did not specify the target category. While non-targeted attack aims to fool the victim, targeted attack aims to make the victim misclassify as a specific category. In other words, the Hamming loss function should be modified to count decisions not equal to the target. Therefore, the target Hamming loss function L_H^{target} can be formulated as:

$$L_H^{\text{target}}(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{N_{\text{cls}}} \sum_{i=1}^{N_{\text{cls}}} (1 - \max(0, \mathbf{y}_{1,i} + \mathbf{y}_{2,i} - 1)). \tag{5}$$

2) SCORE-BASED ATTACK

D-BADGE functions similarly to score-based attacks with decision-based universal adversarial perturbation using the Hamming loss function. However, our method can be directly applied to score-based methods as well by simply not applying $\mathbb{D}$ to $\mathcal{C}(\mathbf{x})$. Score-based attacks are discussed in more detail in the Comparison of Loss Functions section.

IV. EXPERIMENTS

In this section, we conduct extensive experiments on various datasets and victim models to evaluate the performance of D-BADGE in terms of attack success rate, the norm of the perturbation, and the training time. We note that there are few methods for crafting UAPs in a real-world environment, so we included other methods in our experiments, such as white-box UAP and score-based SimBA. We also demonstrate the transferability and performance of the proposed method when applied to a targeted attack. Furthermore, we analyze the effectiveness when applying various batch sizes, loss functions, and optimization algorithms.

A. EXPERIMENT SETTINGS

1) VICTIM MODELS AND DATASETS

We evaluated convolutional networks: ResNet18 (RN18), ResNet20 (RN20) [47], VGG19 [48], MobileNet_v2 (MVN_v2) [49], ResNeXt29_2 × 64d (RNX29) [50], and transformers: Vision Transformer (ViT-T) [31] and Swin Transformer (Swin-T) [32] for the CIFAR-10 [51] dataset and a simple toy convolutional network for the MNIST [52] dataset. We also evaluated D-BADGE on large-scale datasets: ImageNet-1k [53] and ImageNet-12 [54]. CIFAR-10 contains 60,000 images with 32 × 32 size, arranged by ten classes. CIFAR-10 is widely used in evaluating the feasibility and analyzing the characteristics of proposed methods. ImageNet-1k is a large-scale dataset that contains 1,431,167 images arranged by 1k classes. The size of the images in ImageNet-1k is not fixed but usually resized into 224 × 224 or 32 × 32 according to the purpose. ImageNet-1k is the most common dataset for evaluating the capability of a model on the large-scale data. ImageNet-12 is a subset of ImageNet-1k with 12 classes proposed to evaluate adversarial attack and defense techniques. VGGNet is a deep stack of convolution layers and ResNet is a VGGNet with skip connections. MobileNet_v2 significantly reduced the number of operations by adopting inverted residual blocks. ResNeXt grouped the channels of feature maps and improved the capability of each convolution layer. ViT is a stack of transformer encoder blocks with self-attention. Swin Transformer is based on ViT, but employs hierarchical architecture and shifted multi-head self-attention. These differences make networks work differently and that is why multiple architectures have to be experimented with validate adversarial attack methods. In the CIFAR-10 dataset experiment, we compared D-BADGE with DF-UAP, SimBA, and DUAttack attack methods. UAP, DUAttack, and D-BADGE used the CIFAR-10 training set to create universal perturbation, and SimBA built image-dependent perturbations using the validation set. All four methods were evaluated using the CIFAR-10 validation set. Additionally, we attacked the CIFAR-100 dataset, which is more challenging. Unless otherwise stated, our experiments were primarily conducted using ResNet18.

2) TRAINING DETAILS

We trained our perturbations exactly like training a classifier, but with zeroth-order optimization instead of backpropagation. Previous works (DeepFool [9], SimBA [26], and DF-UAP [10]) update perturbation with a single image over multiple iterations and do the same with another image. This can be interpreted as an inverted procedure of ours. We controlled the step size using cosine annealing [55] scheduling. We had α to decay from 10^{-4} to 10^{-3} once over entire epochs. We decayed δ using step scheduling [56]

TABLE 1. Performance comparison with other methods. (a) ResNet18 (RN18), VGG19, MobileNet_v2 (MBN_v2), (b) Vision Transformer (ViT-T) and Swin Transformer (Swin-T) were tested as victim models. UAP, SimBA, and DUAttack are the baselines. The number inside the parenthesis refers to the accuracy of the victim model with clean inputs. Note that DF-UAP is a white-box attack method and SimBA is an image-dependent attack method. (c) shows the ASR of perturbations with five different random noises. The l_2-norms of the random noises were made equal to the l_2-norms of optimized perturbations. Bold font numbers indicate the best result in each evaluation metric.

	RN18 (95.48 %)				VGG19 (93.86 %)				MBN_v2 (92.61 %)			
	# UDT(↓)	Time(↓)	l_2(↓)	ASR(↑)	# UDT(↓)	Time(↓)	l_2(↓)	ASR(↑)	# UDT(↓)	Time(↓)	l_2(↓)	ASR(↑)
DF-UAP	4.42E+5	91,945	**338.45**	73.99	6.31E+5	132,863	380.43	29.36	6.14E+5	226,809	**276.58**	71.54
SimBA	1.91E+7	120,206	393.52	78.25	1.98E+7	115,208	**248.84**	49.44	1.43E+7	150,357	410.87	81.79
DUAttack	1.56E+5	28,852	541.77	85.10	1.56E+5	22,516	554.24	59.60	1.56E+5	30,129	535.86	82.49
D-BADGE (ours)	**1.56E+5**	**6,541**	409.05	**85.88**	**1.56E+5**	**5,978**	425.73	**63.53**	**1.56E+5**	**7,034**	398.89	**88.06**

(a) Comparison among convolution-based architectures.

	ViT-T (62.45 %)				Swin-T (77.59 %)			
	# UDT(↓)	Time(↓)	l_2(↓)	ASR(↑)	# UDT(↓)	Time(↓)	l_2(↓)	ASR(↑)
DF-UAP	3.72E+5	100,708	368.03	48.31	5.45E+5	342,664	314.64	30.91
SimBA	2.33E+7	173,816	**117.27**	23.36	2.60E+7	492,098	**90.10**	17.86
DUAttack	1.56E+5	33,012	553.75	36.19	1.56E+5	36,249	554.24	42.66
D-BADGE (ours)	**1.56E+5**	**9,890**	405.04	**57.67**	**1.56E+5**	**11,010**	427.92	**65.70**

(b) Comparison among transformer-based architectures.

Architecture	ASR
RN18	10.53
VGG19	10.13
MBN_v2	24.11
ViT-T	7.02
Swin-T	10.98

(c) ASR using random perturbations.

from 0.01 with a 0.9 decay ratio. γ hyperparameter was set to 10^{-3}. The default batch size was set to 256 by heuristic compromise. All training and evaluation samples are 8-bit images with values under 255. The l_∞-norm of perturbation was limited to 10.0. All experiments were conducted on Ubuntu Server 18.04 with an Intel Xeon Gold 6226R 2.90GHz and NVIDIA RTX 3090.

B. EVALUATION METRICS

1) ATTACK SUCCESS RATE (ASR, %)

ASR is our primary evaluation metric for non-targeted attacks, defined as the ratio of the number of changed decisions over the total number of adversarial examples. Similarly, we define target accuracy as an evaluation metric for a targeted attack. Target accuracy is defined as the number of adversarial examples classified to the targeted class.

2) NORM OF PERTURBATION (L_*)

The norm value of a perturbation is an intuitive evaluation metric for how the perturbation is recognizable by humans. l_∞-norm is the highest entry in the vector space. l_∞-norm essentially determines the maximum magnitude of a given vector. l_2-norm, the Euclidean or Frobenius norm, is the shortest distance between two vectors. It is calculated as the distance between the original and adversarial examples in the adversarial setting. We primarily adopted l_2-norm to measure the overall distortion of adversarial examples.

3) THE NUMBER OF UPDATES (# UDT)

This metric indicates how many times the perturbation requires to reach a certain ASR. The lower # UDT suggests that the update directions to the perturbation were a better direction under a similar ASR.

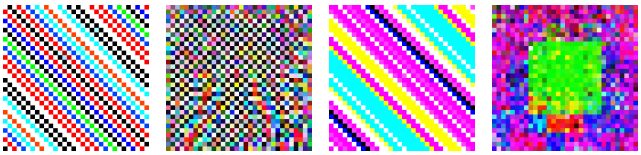
4) TRAINING TIME (TIME, SECONDS)

Training time indicates how long a method takes to generate adversarial examples for the entire validation set. Therefore, it must be measured differently depending on image dependency. The perturbation optimization time could be the training time for optimizing universal perturbation and the time to build adversarial perturbations for all images in the validation set. The shorter training time suggests that the method can capture a victim's decision boundary in shorter periods of time.

C. NON-TARGETED ATTACK

We tested four attack methods: UAP, SimBA, DUAttack, and D-BADGE to five victim models: ResNet18, VGG19, MBN_v2, ViT-T, and Swin-T on the CIFAR-10 dataset. UAP and D-BADGE generated universal perturbations using a training set (50,000 images) and evaluated using the validation split. All experiments were performed under $l_\infty = 10.0$ constraint. As shown in Table 1, D-BADGE achieved higher ASR with fewer updates and shorter training time, even compared to the DUAttack with the same number of updates and queries. The architecture of Vision Transformer and Swin Transformer significantly stand out from convolution-based networks. Swin-T showed greater robustness against UAP, SimBA, and DUAttack compared to ViT-T, while D-BADGE outperformed others across the victims as shown in Table 1b. Figure 2 shows the difference in perturbation regarding the attack methods and victim models. We observed that DUAttack crafts similar diagonal perturbations regardless of the victim network while D-BADGE crafts different appearance against ResNet20 and ViT-T. Specifically, in Figure 2d, the squares correspond to each patch of ViTs.

We measured ASR on CIFAR-10 dataset against ResNet18 and D-BADGE works as well regardless of the number of categories. For additional results, please refer to the supplementary material. We conducted a D-BADGE attack on the MNIST dataset using a toy network. Results are shown in Figure 4a. The high-contrast pixels require a



(a) DUAttack to ResNet20 (b) D-BADGE to ResNet20 (c) DUAttack to ViT-T (d) D-BADGE to ViT-T

FIGURE 2. Perturbation comparison between DUAttack [11] and D-BADGE (ours) against ResNet20 [47] and ViT-T [31].

TABLE 2. Non-targeted attack ASRs of D-BADGE on the large-scale datasets.

Dataset	ImageNet-1k [53]		ImageNet-12 [54]	
Victim	ViT	Swin	ViT	Swin
ASR	90.91	97.69	54.71	60.87

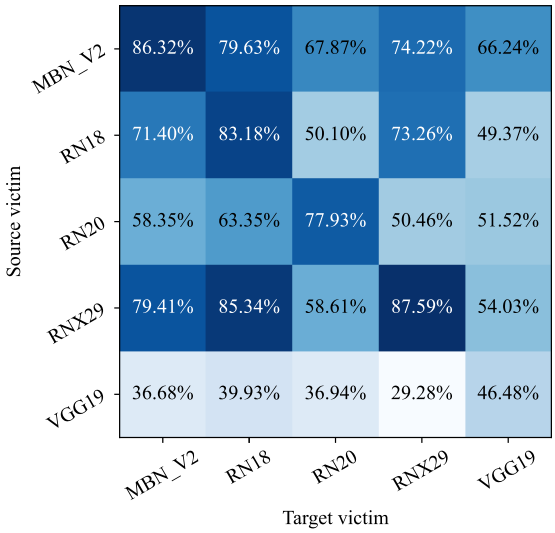


FIGURE 3. The confusion matrix of the transferability on non-targeted attack. The value of each cell refers to the ASR when fooling the target victim using a perturbation that was trained from the source victim.

higher l_2-norm limit, but the most substantial perturbation remains less visible than the real pixel value. The l_2-norms of UAP and SimBA are less than that of D-BADGE. This implies that D-BADGE has better learnability because weaker perturbation generally leads to poor ASR.

D. EVALUATION ON LARGE-SCALE DATASETS

We also examined the capability of D-BADGE on large-scale datasets that describe the real-world scenario better. For ImageNet-1k, we resized the inputs for fast training and evaluation as Zhang et al. [57] did. Table 2 depicts that our method successfully attacked both victim architectures. Specifically, our method was more effective on the smaller-size images even if the scale of the dataset is larger but, it is also capable of fooling larger-size images.

E. TRANSFERABILITY

We investigated the transferability across victim models of the proposed methods. It is well known that transferring adversarial perturbations to a network with a completely different architecture is challenging [58], [59]. This means the more a perturbation is transferable, the more the attack method used is powerful. We tried to capture the most general representation to the core operation of the network, e.g. convolution for CNNs and self-attention for transformers possible. In Figure 3, we demonstrated the transferability of our perturbations between various models. The figure summarizes the ASR of the D-BADGE attack trained on one network and evaluated on another. D-BADGE is transferable within CNNs and transformers, but not between a CNN and a transformer. The ASRs are greater than 40% among CNNs, which is much greater than the ASR of random noise with l_∞-norm $= 10.0$. The ASRs among transformers are also greater than 20%. However, the ASRs between a CNN and a transformer remain an average of 10.92%. CNNs and transformers are independent of each other, especially ViT-T. Swin Transformer is relatively more transferable to CNNs and vice versa, with an average ASR of 16.02%. In summary, D-BADGE is effectively transferable among CNNs and transformers and partially among CNNs and transformers.

TABLE 3. Target accuracy (%) of the targeted attack for each category in the CIFAR-10 dataset.

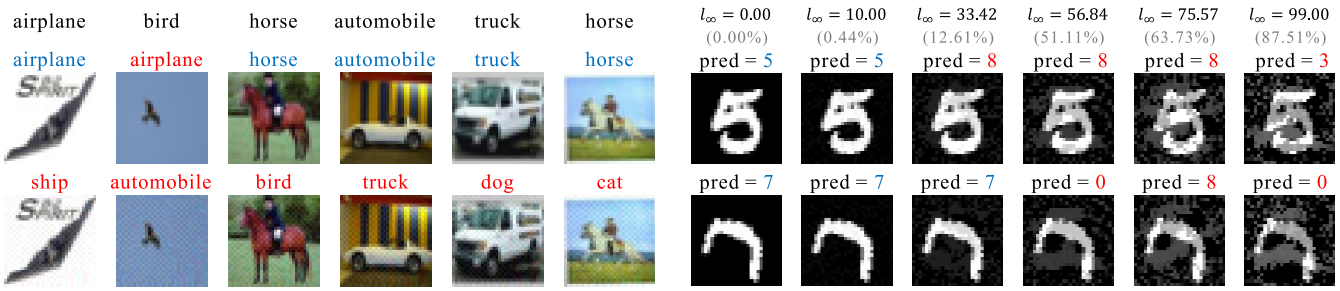
Class	RN18	RN20	VGG19	MBN_v2	RNX29
Plane	85.94	88.72	93.93	**97.45**	81.99
Car	72.99	86.91	67.55	**87.97**	86.36
Bird	87.66	56.21	**95.03**	94.81	90.41
Cat	84.57	**95.00**	86.88	86.28	94.51
Deer	10.02	10.00	**81.65**	77.72	77.44
Dog	56.08	**83.71**	9.73	67.41	74.75
Frog	91.63	86.35	95.60	**97.75**	91.46
Horse	79.78	**97.52**	88.63	91.65	92.41
Ship	80.73	10.03	**98.34**	97.22	88.70
Truck	50.88	**91.96**	73.02	90.69	82.55
Mean	**70.03**	**70.64**	**79.04**	**88.90**	**86.06**

F. TARGETED ATTACK

We conducted a targeted D-BADGE attack on five victims that were trained using the CIFAR-10 dataset. The results are displayed in Table 3. D-BADGE achieved higher target accuracy for a specific class within each victim architecture. Specifically, MVN_v2 outperformed the others, achieving the highest ASR target accuracy. VGG19 exhibited poor ASR in non-targeted attacks, yet specific target classes were still attainable with VGG19. The average target accuracy scores for all victims ranged from 70 to 85%, indicating successful attacks. The categories *Plane, cat*, and *frog* were easily targeted, while *deer, dog*, and *ship* proved to be robust in specific victim architectures. We discovered that the existence of classes with robust features can make adversarial attacks more challenging.

G. EFFECTIVENESS OF BATCH ATTACK

We evaluated ASR across various batch sizes, adjusting the number of epochs to align with the number of updates to



(a) CIFAR-10 (b) MNIST

FIGURE 4. (a) shows the adversarial examples of the CIFAR-10 dataset against ResNet18. The top row illustrates the original images, accompanied by both the ground truth and the corresponding predictions. The bottom row shows the adversarial examples with their classification results. (b) shows the adversarial examples of the MNIST dataset on various l_∞-norm constraints. The numbers in parentheses denote the ASR scores. Blue and red colors indicate correctly classified and misclassified categories, respectively, in both subfigures.

TABLE 4. Effectiveness of the batch size (BS). The number of updates was fixed in (a) and the number of epochs was fixed in (b).

BS	# Epochs	ASR	Time	# UDT	# Epochs	ASR	# UDT
1	4	21.03	2,931	200,000	-	-	-
32	128	83.39	3,069	199,936	800	87.29	1,250,400
64	256	86.97	4,132	199,936	800	87.42	625,600
128	512	85.99	5,485	199,680	800	87.66	312,800
256	1,025	**87.71**	**8,679**	199,875	800	**87.76**	156,800
512	2,061	87.40	15,225	199,917	800	84.06	78,400

assess the batch size's effectiveness. We observed that as the batch size increases, the attack success rate also increases, given a similar number of updates, as shown in Table 4a. However, this increased total training time, as it necessitated more inferences to the victim. In other words, a trade-off relationship exists between ASR and time in D-BADGE. Notably, the jump from 256 to 512 resulted in a dramatic increase in training time. Moreover, we fixed the the number of epochs to evaluate the query-efficiency of D-BADGE. Table 4b shows that as the batch size slightly increases, ASR increased up to the batch size of 256. The batch size greater than 256 rather decreased ASR because it catastrophically lacks the number of updates. This means the greater batch size helps the perturbation to find a better direction faster than with a smaller batch size. We determined that 256 is the optimal batch size, considering a balance between ASR and training time compared to other sizes.

H. COMPARISON OF LOSS FUNCTIONS

We also conducted experiments on score-based attacks, assuming a score could be obtained by inserting an adversarial example into the victim model. Four loss functions of Hamming Distance (HD), KLD, CE, and EMD were tested. The result is shown in Table 5. HD and KLD functioned better with decisions than scores, which is contrastive to CE and EMD. EMD worked poorly, especially with decisions. It is because EMD, intuitively speaking, calculates the amount of data that must be transferred between two distributions to make them equal. This is not a direct distance metric between two decision distributions. On the other hand, other functions

TABLE 5. Performance comparison for various loss functions. Each value means $\mu \pm \sigma$ of ASR against ResNet18. SB and DB refer to score-based and decision-based attacks respectively.

	HD (ours)	KLD	CE	EMD
SB	72.96 ± 2.99	87.69 ± 0.62	$\mathbf{88.17 \pm 0.45}$	73.42 ± 1.72
DB	$\mathbf{87.22 \pm 0.29}$	86.72 ± 0.32	86.73 ± 0.64	2.67 ± 0.11

TABLE 6. Ablation study. Batch attack, the Hamming loss function, and SPSA-AM were removed in the top three rows each.

Batch Applied	Loss HD	Loss CE	Optimization SPSA	Optimization Adam	Evaluation ASR	Evaluation # UDT
✗	✓		✓	✓	$5.79_{\pm 0.79}$	200,000
✓	✗	✓	✓	✓	$55.78_{\pm 1.73}$	199,875
✓	✓		✓	✗	$\mathbf{58.77}_{\pm 5.34}$	199,875
✓	✓		✓	✓	$57.44_{\pm \mathbf{0.63}}$	199,875

are well-known distance metrics. As a result, we obtained the best ASR with CE and HD loss in score-based and decision-based attacks, respectively. The ASR using HD on decision-based attacks is slightly lower than score-based attacks using CE but still performed with the lower variance of ASR. This suggests that D-BADGE works, at least, as well as on score-based attacks despite of catastrophic lack of information.

I. COMPARISON OF OPTIMIZATION ALGORITHMS

The optimization algorithm is a critical factor in decision-based attacks. As we combined the SPSA and the Adam optimizer, we also tested other combinations of optimization algorithms. We evaluated five combinations as shown in Figure 5. SPSA-based algorithms are basically performed better than RGF-based algorithms. Gradient correction using NAG or Adam helps consistent convergence of perturbations. We captured that SPSA-GC is more consistent than SPSA while the ASR is much lower. However, SPSA-AM still converges stably with a few ASR drops compared to SPSA.

J. ABLATION STUDY

D-BADGE consists of three factors: batch attack, the Hamming loss function, and SPSA-AM. If a perturbation

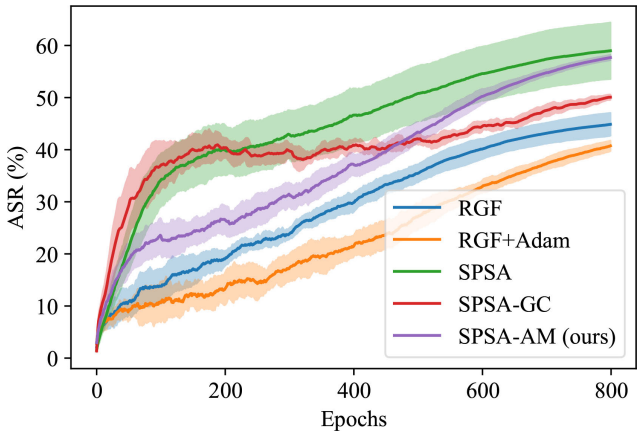


FIGURE 5. Convergence using different algorithms. Five different combinations of optimization algorithms were tested against ViT-T. The ASRs of 20 attacks were averaged and the error range shows 2σ of them.

TABLE 7. Attack success rate (%) of iGAT [60] against two attack methods.

Victim	Defense	DUAttack [11]	D-BADGE (ours)
ResNet20	iGAT + ADT [61]	3.91	1.72
	iGAT + DVERGE [62]	4.35	2.83
ViT-T	iGAT + ADT [61]	(Defense failure)	
	iGAT + DVERGE [62]	4.73	1.60

is updated with a single image in sequence, the direction of the update will be biased and inconsistent. A Batch attack helps to find a general, unbiased, and consistent direction for updates because the direction is not only for a single decision. This is the way how the batch attack contributes to address universal adversarial attacks and outperforms others. In Table 6, we removed the three factors to reveal the effect of the factors. It shows that batch attack is essential for successful attack and the Hamming loss function increases the ASR and make it consistent. And Adam may slightly reduce the ASR, but it achieved outstanding consistency with SPSA.

K. ROBUSTNESS AGAINST DEFENSE

We evaluated the robustness of DUAttack and D-BADGE against one of the most recent ensemble-based adversarial defense method, iGAT [60] combined with ADT [61] and DVERGE [62] (Table 7). We observed that DUAttack achieved slightly higher ASR than D-BADGE. However, the ASR of both was lower than 5%, which is almost a failure. This can be interpreted as decision-based universal attack is not powerful enough and yet to against recent defense methods. ADT with iGAT failed to train the ViT-T model. It also implies that defense methods are not robust enough, even in the white-box settings.

V. ANALYSIS AND DISCUSSION

A. COMPUTATIONAL COMPLEXITY

We analyzed the computational complexity of D-BADGE compared to others.

TABLE 8. The theoretical and empirical big-**O** complexity comparison among two image-specific attack and universal attack methods.

Method		Theoretical	Empirical
Image-specific	DeepFool [9]	TN	-
	SimBA [26]	TN	$2.40N$
Universal	DF-UAP [10]	SN	$1.84N$
	DUAttack [11]	$\frac{E}{B}N$	$0.57N$
	D-BADGE (ours)	$\frac{E}{B}N$	$0.13N$

1) IMAGE-SPECIFIC ATTACK

Image-specific attacks require to generate adversarial perturbation for each single image for a total of N images. Image-specific attack methods DeepFool [9] and SimBA [26] iterate until the perturbation converges. Suppose it iterates for T times, the time complexity big-**O** notation of both is $\mathbf{O}(N) = TN$.

2) UNIVERSAL ATTACK

Basic universal white-box attack DF-UAP [10] generates one perturbation for N images. But it has to be updated sequentially for N images each, which demands unnecessarily higher complexity ($\mathbf{O}(N) = SN$ where it iterates S times per image). DUAttack [11] is another universal attack method. It parallelizes B images as a batch, and iterates for E epochs. Consequently, the big-**O** notation becomes $\mathbf{O}(N) = \frac{E}{B}N$. D-BADGE runs similarly to DUAttack in the perspective of time complexity. As shown in Table 1, the empirically required time per image of DUAttack and D-BADGE are approximately 0.57 seconds and 0.13 seconds, respectively, due to the characteristics of the methods. We compared the big-**O** notation of the five methods in Table 8.

B. EXPERIMENTAL ASSUMPTIONS

Our experiment holds two assumptions:

1) The attacker has inputs that are independent and identically distributed (IID) along the trained data.
2) Multiple queries are guaranteed to the attacker so that the universal adversarial perturbation can be updated repeatedly.

The first assumption means D-BADGE does not address source-free adversarial attack tasks. Decision-based adversarial attacks are at the academic level, and this could be a constraint. There is a task free from the second assumption, namely, query-free adversarial attack [63], [64]. This scenario holds very tight constraints and is not necessary in the real-world scenario. Our assumptions yield real-world scenarios with minimal constraints of data distribution and queries.

C. ARTIFICIAL INTELLIGENCE ETHICS

Ethical discussion must be preceded before utilizing artificial intelligence technology, especially attack and defense methods. We believe researches about adversarial attack method are most worth as a foundation for researches about defense method and general robustness of DNNs. Both are currently in academic level. Even so, attack methods are threats in

certain scenarios such as medical data processing [65], [66] and autonomous driving [67], [68]. It is known that visual perception techniques can be used to discriminate medical harms: tumors and abdominal aortic aneurysms [66]. It will be critical if adversarial perturbations could increase false negative rate.

VI. CONCLUSION

This paper proposes a novel method named D-BADGE, which crafts image-agnostic universal perturbations in decision-based black-box attacks. The proposed method utilize decisions of mini-batch to reconstruct probability distribution. The Hamming loss function is used to optimize universal perturbation. We demonstrated that the D-BADGE can easily be applied to targeted and score-based attacks using a straight forward combination of SPSA and Adam, namely SPSA-AM. The proposed method achieved better performance with less number of updates and training time compared to white-box UAP and score-based SimBA, even with the same number of updates and the same number of queries compared to DUAttack for CNN-based and Transformer-based victim models. As it is evident that the number of queries is also a crucial factor in black-box attacks, we leave it as future work to craft UAPs with a few queries.

REFERENCES

[1] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *Int. J. Multimedia Inf. Retr.*, vol. 7, pp. 87–93, Jun. 2018.

[2] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[4] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[5] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 250–258.

[6] K.-C. Wang, Y. Fu, K. Li, A. Khisti, R. Zemel, and A. Makhzani, "Variational model inversion attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9706–9719.

[7] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, "Exploring connections between active learning and model extraction," in *Proc. 29th USENIX Conf. Secur. Symp.*, 2020, pp. 1309–1326.

[8] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *Proc. 29th USENIX Conf. Secur. Symp.*, 2020, pp. 1345–1362.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 86–94.

[11] J. Wu, M. Zhou, S. Liu, Y. Liu, and C. Zhu, "Decision-based universal adversarial attack," 2020, *arXiv:2009.07024*.

[12] J. Chen, M. Su, S. Shen, H. Xiong, and H. Zheng, "POBA-GA: Perturbation optimized black-box adversarial attacks via genetic algorithm," *Comput. Secur.*, vol. 85, pp. 89–106, Aug. 2019.

[13] P. N. Williams and K. Li, "Black-box sparse adversarial attack via multi-objective optimisation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 12291–12301.

[14] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting deep neural networks globally via universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13420–13429.

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

[16] T. Brunner, F. Diehl, M. T. Le, and A. Knoll, "Guessing smart: Biased sampling for efficient black-box adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4957–4965.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519, doi: 10.1145/3052973.3053009.

[18] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6512–6520.

[19] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1277–1294.

[20] I. Oseledets and V. Khrulkov, "Art of singular vectors and universal adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8562–8570.

[21] K. R. Mopuri, U. Ojha, U. Garg, and R. V. Babu, "NAG: Network for adversary generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 742–751.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[23] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4422–4431.

[24] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4307–4316.

[25] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh, "ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[26] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," in *Proc. 36th Int. Conf. Mach. Learn.*, Jun. 2019, pp. 2484–2493.

[27] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, pp. 527–566, Apr. 2017, doi: 10.1007/s10208-015-9296-2.

[28] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[29] V. Q. Vo, E. Abbasnejad, and D. C. Ranasinghe, "Query efficient decision based sparse attacks against black-box deep learning models," in *Proc. 10th Int. Conf. Learn. Represent.*, 2022, pp. 1–19.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[33] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.

[34] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017, *arXiv:1703.03864*.

[35] Z. Huang and T. Zhang, "Black-box adversarial attack with transferable model-based embedding," 2019, *arXiv:1911.07140*.

[36] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, 1992.

[37] J. L. Maryak and D. C. Chin, "Global random optimization by simultaneous perturbation stochastic approximation," in *Proc. Amer. Control Conf.*, vol. 2, 2001, pp. 756–762.
[38] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord, "Adversarial risk and the dangers of evaluating against weak attacks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5025–5034.
[39] C. Oh, H. Hwang, H.-Y. Lee, Y. Lim, G. Jung, J. Jung, H. Choi, and K. Song, "BlackVIP: Black-box visual prompting for robust transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24224–24235.
[40] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Proc. USSR Acad. Sci.*, vol. 269, pp. 543–547, Feb. 1983.
[41] N. Tursynbek, A. Petiushko, and I. Oseledets, "Geometry-inspired top-k adversarial perturbations," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 4059–4068.
[42] M. Ye, J. Chen, C. Miao, H. Liu, T. Wang, and F. Ma, "PAT: Geometry-aware hard-label black-box adversarial attacks on text," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023, pp. 3093–3104.
[43] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," 2020, *arXiv:2010.01736*.
[44] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
[45] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. theory*, vol. 8, no. 1, pp. 21–28, 1962.
[46] Z. Yu, D. W. C. Ho, and D. Yuan, "Distributed randomized gradient-free mirror descent algorithm for constrained optimization," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 957–964, Feb. 2022.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.
[49] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[50] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
[51] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998.
[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[54] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 14900–14912.
[55] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[57] J. Zhang, B. Li, C. Chen, L. Lyu, S. Wu, S. Ding, and C. Wu, "Delving into the adversarial robustness of federated learning," 2023, *arXiv:2302.09479*.
[58] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, "Feature importance-aware transferable adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7619–7628.
[59] S. Wu, Y.-A. Tan, Y. Wang, R. Ma, W. Ma, and Y. Li, "Towards transferable adversarial attacks with centralized perturbation," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 6, pp. 6109–6116.
[60] Y. Deng and T. Mu, "Understanding and improving ensemble adversarial defense," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–13.
[61] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4970–4979.
[62] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, and H. Li, "DVERGE: Diversifying vulnerabilities for enhanced robust generation of ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 5505–5515.
[63] H. Zhuang, Y. Zhang, and S. Liu, "A pilot study of query-free adversarial attack against stable diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 2384–2391.
[64] G. Chen, Y. Zhang, Z. Zhao, and F. Song, "QFA2SR: Query-free adversarial transfer attacks to speaker recognition systems," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 2437–2454.
[65] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, pp. 1287–1289, Mar. 2019.
[66] Y. Jung, S. Kim, J. Kim, B. Hwang, S. Lee, E. Y. Kim, J. H. Kim, and H. Hwang, "Abdominal aortic thrombus segmentation in postoperative computed tomography angiography images using bi-directional convolutional long short-term memory architecture," *Sensors*, vol. 23, no. 1, p. 175, 2022.
[67] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-adaptive YOLO for object detection in adverse weather conditions," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 2, pp. 1792–1800.
[68] H. Zhang, L. Xiao, X. Cao, and H. Foroosh, "Multiple adverse weather conditions adaptation for object detection via causal intervention," *IEEE Trans. Pattern Anal. Mach. Intell.*

GEUNHYEOK YU received the B.S. degree from the Department of Software, Gachon University, Seongnam-si, South Korea, in 2022, and the M.S. degree from the Department of Software Convergence, Kyung Hee University, Yongin-si, South Korea, in 2023, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and machine learning, which spans over perturbation-based visual representation learning for machine intelligence.

MINWOO JEON received the B.S. degree from the Department of Software Convergence, Kyung Hee University, Yongin-si, South Korea, in 2023, where he is currently pursuing the master's degree. His research interest includes computer vision.

HYOSEOK HWANG (Member, IEEE) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, South Korea, in 2004, the M.S. degree in robotics program, in 2009, and the Ph.D. degree from the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017. From 2009 to 2018, he was with the Samsung Advanced Institute of Technology (SAIT), Samsung Electronics, as a Senior Researcher. He is currently an Associate Professor with the Department of Software Convergence, Kyung Hee University, Yongin-si, South Korea. His research interests include computer vision and machine learning, which spans over 3D perception and reconstruction for intelligent robots and autonomous systems.