꼼꼼한 딥러닝 논문 리뷰와 코드 실습

Deep Learning Paper Review and Code Practice

나동빈(dongbinna@postech.ac.kr)

Pohang University of Science and Technology

오늘 리뷰할 논문은?

ICLR 2019

Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach

Minhao Cheng, Thong Le, Pin-Yu Chen,
Jinfeng Yi, Huan Zhang, Cho-Jui Hsieh

University of California, JD AI Research, IBM Research

나동빈

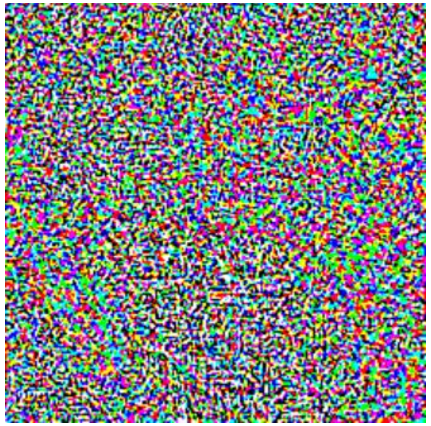
(배경 지식) Adversarial Examples

- An adversarial example can fool a deep neural network.
- An adversarial example is almost identical to original samples in human perception.
  - i.e., a norm-constrained perturbation is constrained below a specific constant ϵ.

x

$(Tabby\ Cat)$

$+ \epsilon *$



$Perturbation\ (\delta)$

$=$

x^*

$(Guacamole)$

(배경 지식) Definition of Adversarial Example

- Given an original example x_0 and a K-way multi-class classification model

$$f: \mathbb{R}^d \rightarrow \{1, \dots, K\}$$

- The attacker's goal is to generate an adversarial example x such that

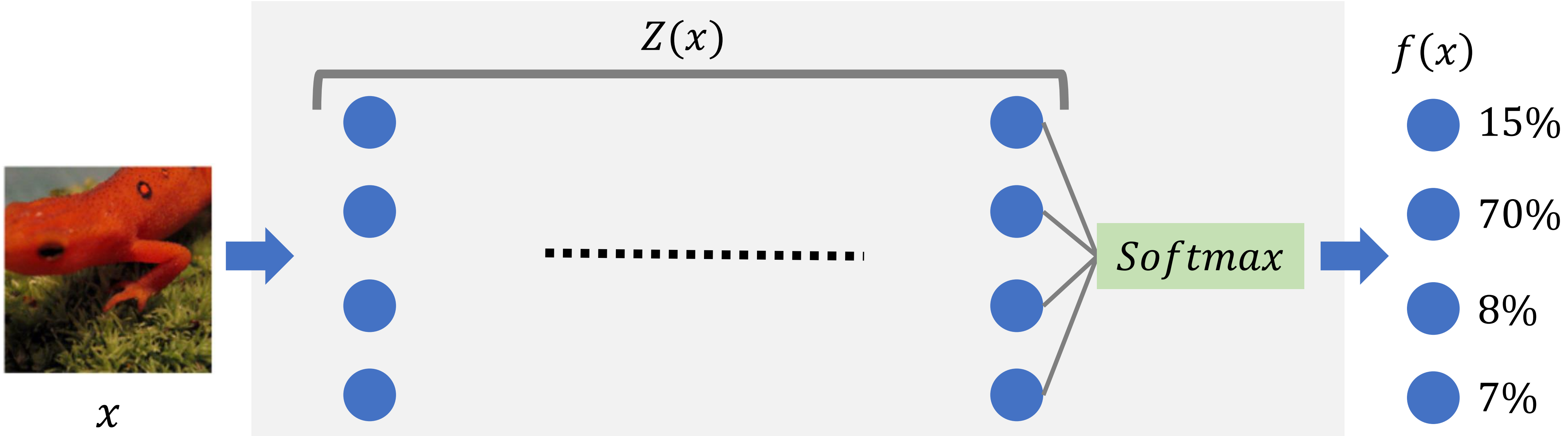
$$x \text{ is close to } x_0 \text{ and } arg\max_i f_i(x) \neq arg\max_i f_i(x_0)$$

Untargeted attack

i.e., x has a different prediction with x_0 by model f.

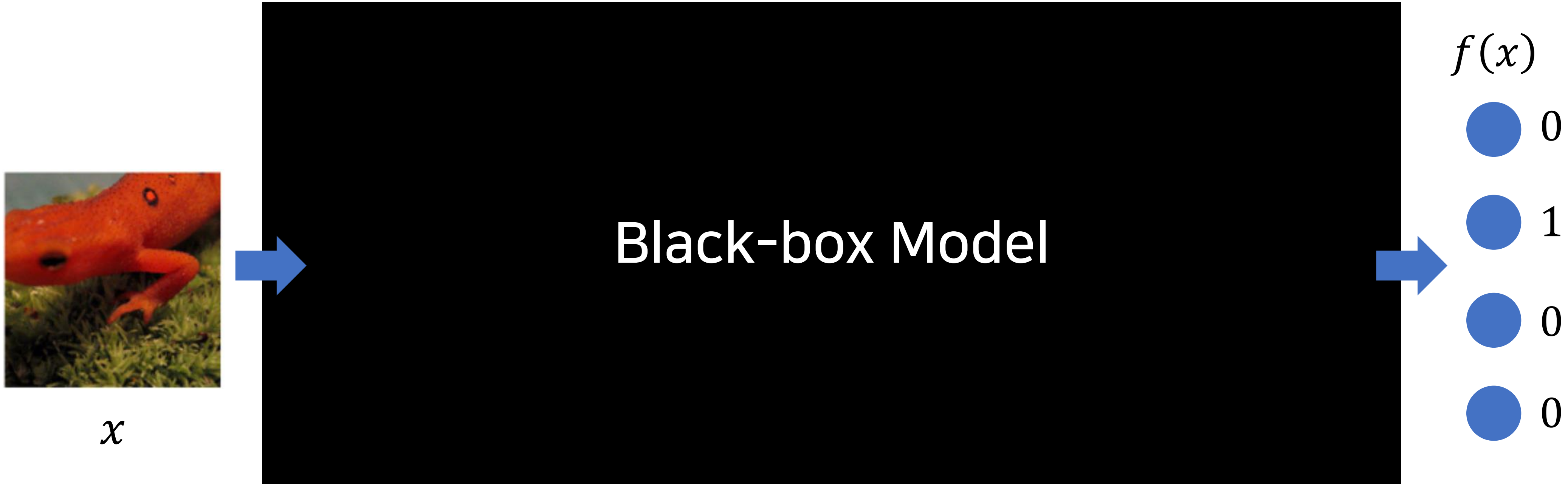
(배경 지식) Threat Model: White-box Setting

- Model information including network structure and weights is revealed to the attacker.
  - The gradient of input can be computed by back-propagation.
  - Attacker minimizes the loss function by gradient descent.



(배경 지식) Threat Model: Hard-label Black-box Setting

- The model is not known to the attacker.
  - The attacker can make a query and observe a hard-label multi-class output.
  - The attacker is not able to compute the gradient of input x by back-propagation.



x

Black-box Model

$f(x)$

0

1

0

0

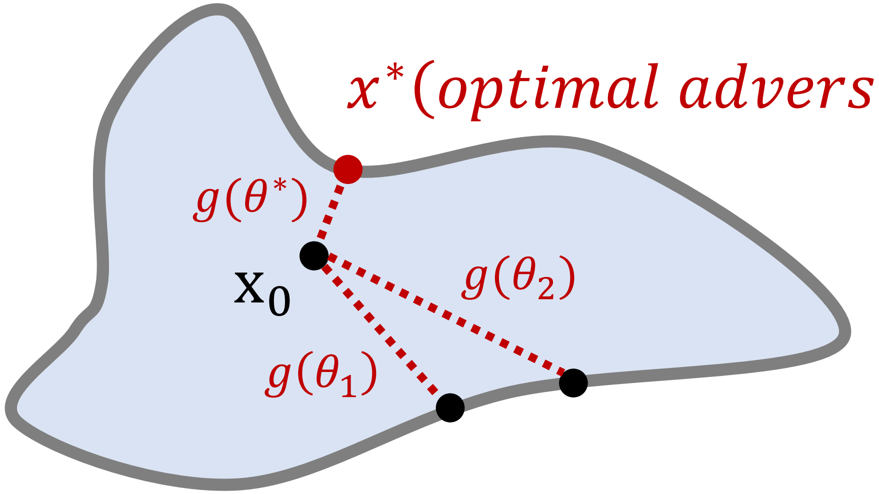
본 논문의 핵심 아이디어: A Boundary-based Reformulation

- Reformulating the hard-label black-box attack as **another optimization problem.**
- $g(\theta)$ is the distance from x_0 to the nearest adversarial example along the direction θ.

$$\theta^* = \underset{\theta}{argmin}\, g(\theta)$$

$$\text{where } g(\theta) = \underset{\lambda>0}{argmin}\left(f\left(x_0 + \lambda\frac{\theta}{\|\theta\|}\right) \neq y_0\right)$$

Untargeted attack

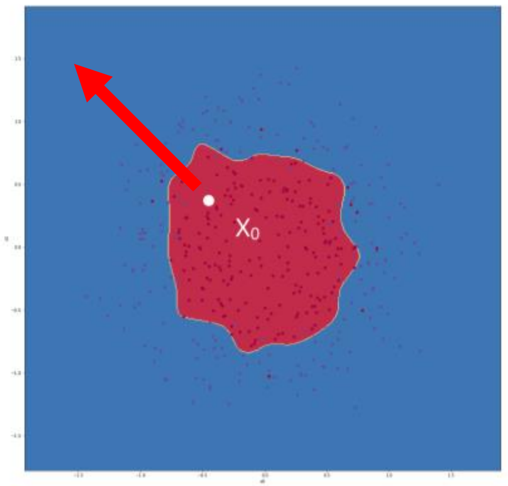


본 논문의 핵심 아이디어: A Boundary-based Reformulation (cont'd)

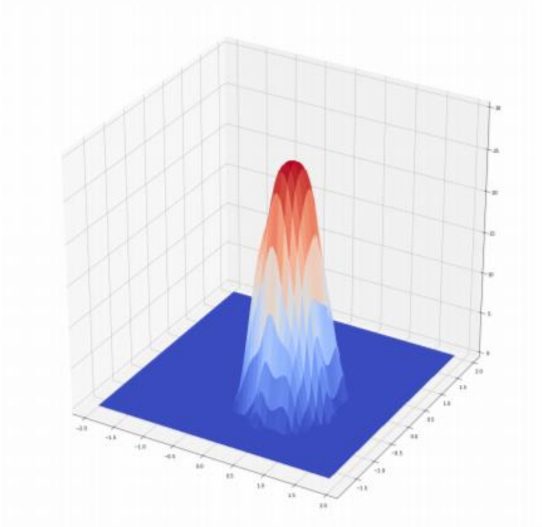
- g(θ) is continuous and hence can be easily optimized.

Untargeted attack

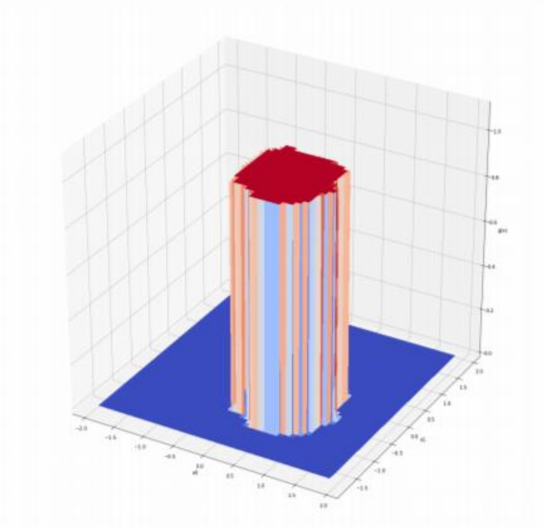
$$L\big(Z(x)\big) = max\big\{[Z(x)]_{y_0} - max_{i\neq y_0}[Z(x)]_i\,, -k\big\}$$



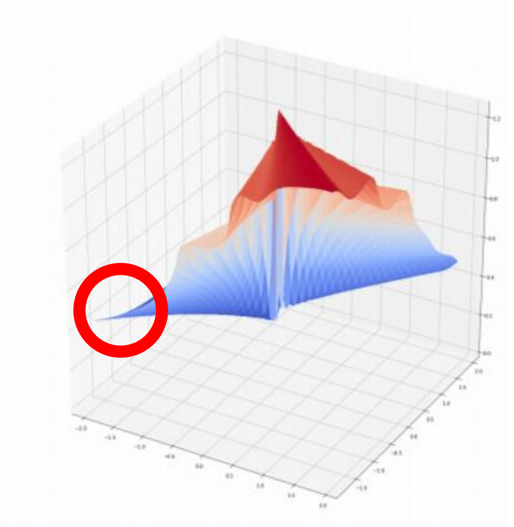
Decision boundary of $f(x)$



$L\big(Z(x)\big)$



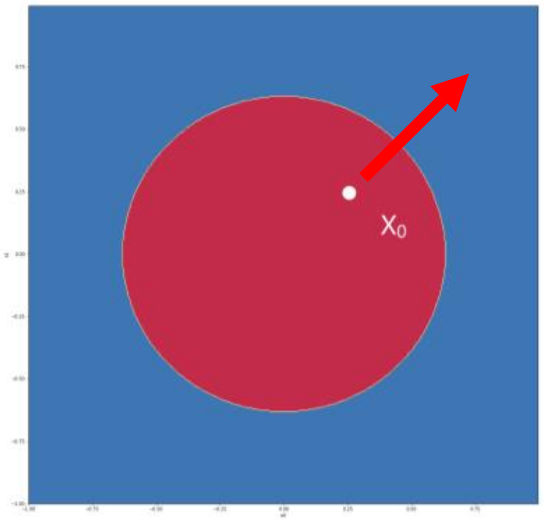
$L\big(f(x)\big)$



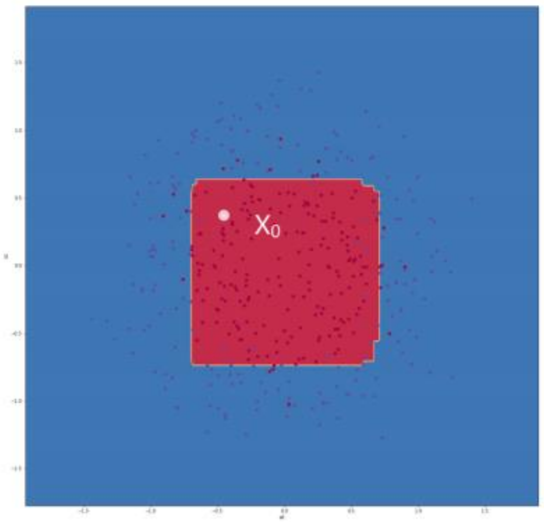
$g(\theta)$

본 논문의 핵심 아이디어: A Boundary-based Reformulation (cont'd)

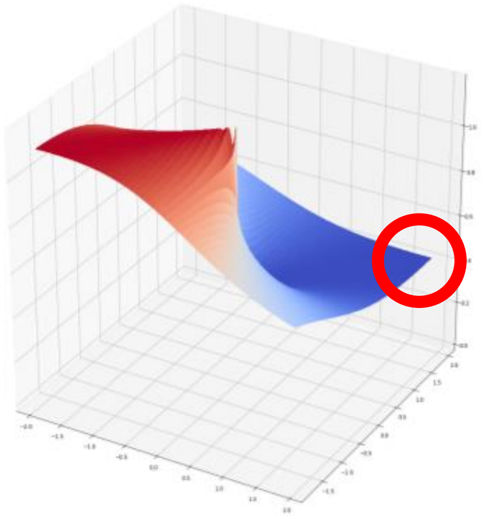
- Even if the classifier function is not continuous, $\mathrm{g}(\theta)$ is still continuous.
  - This makes it easy to apply the zeroth-order method to solve $\min\limits_{\theta} g(\theta)$.



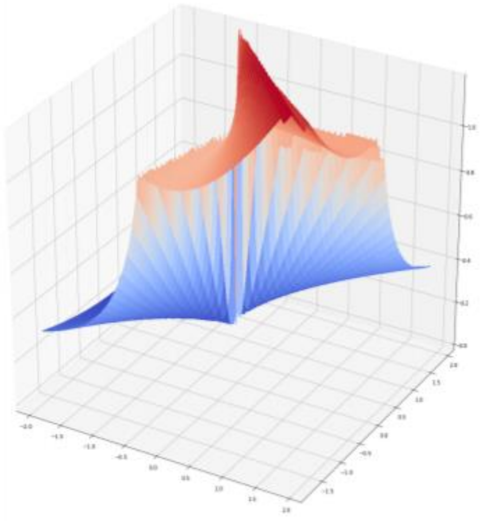
(a) Decision boundary of continuous function



(b) Decision boundary of **GBDT**

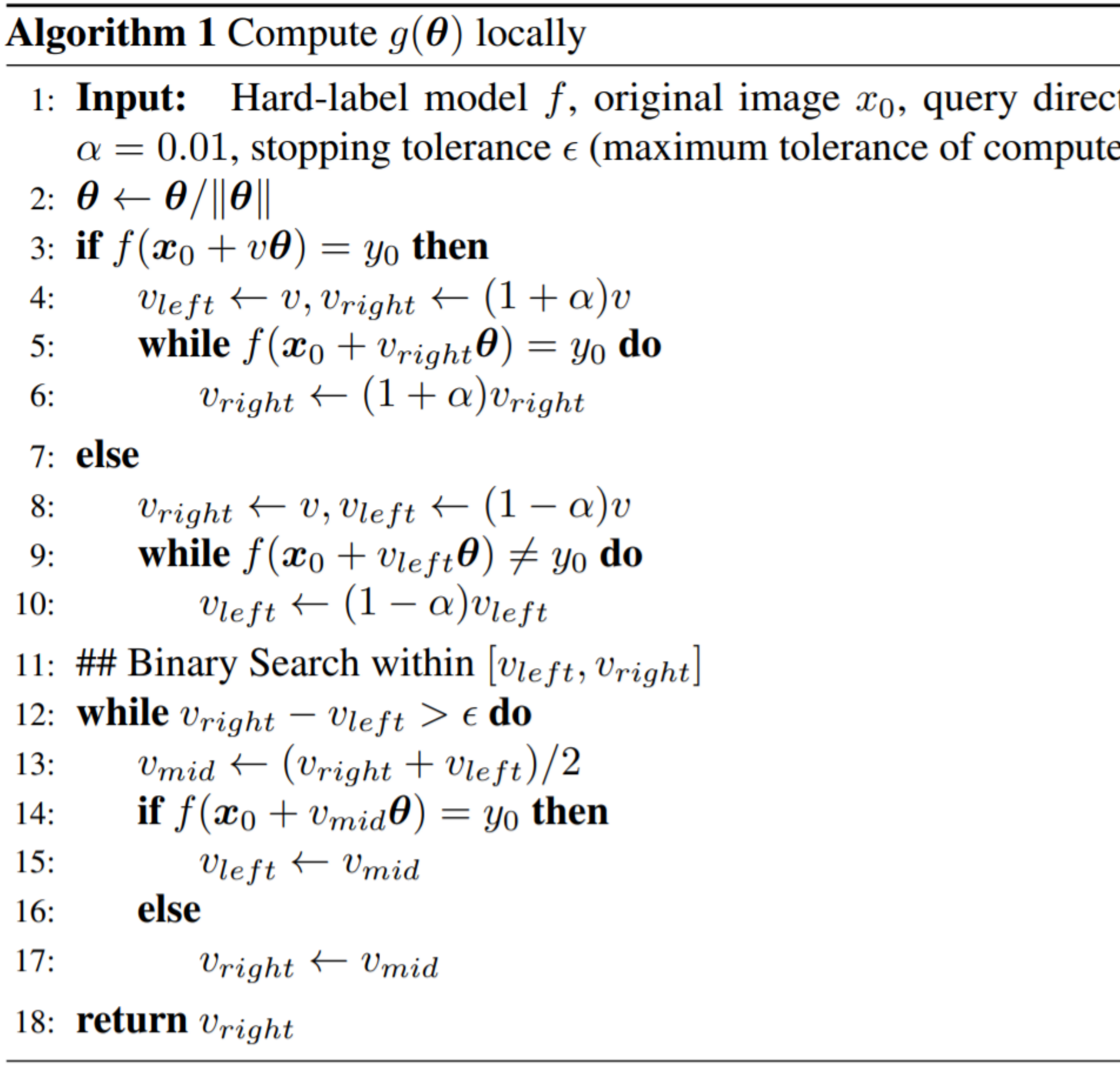


$g(\theta)$ of (a)



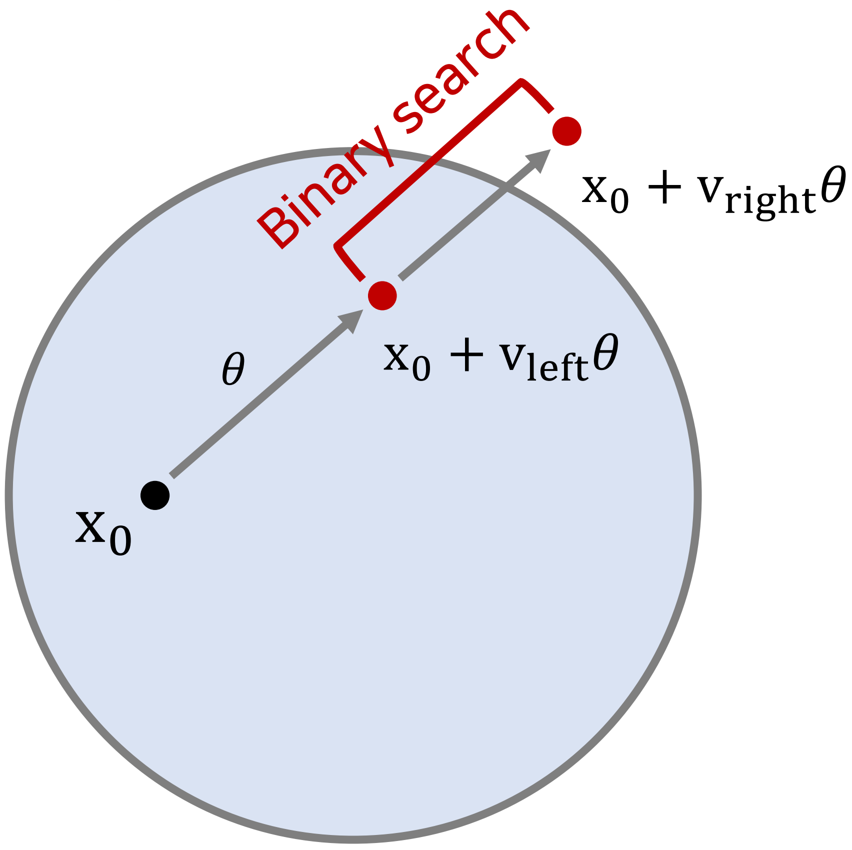
$g(\theta)$ of (b)

How to compute $g(\theta)$



Algorithm 1 Compute $g(\boldsymbol{\theta})$ locally

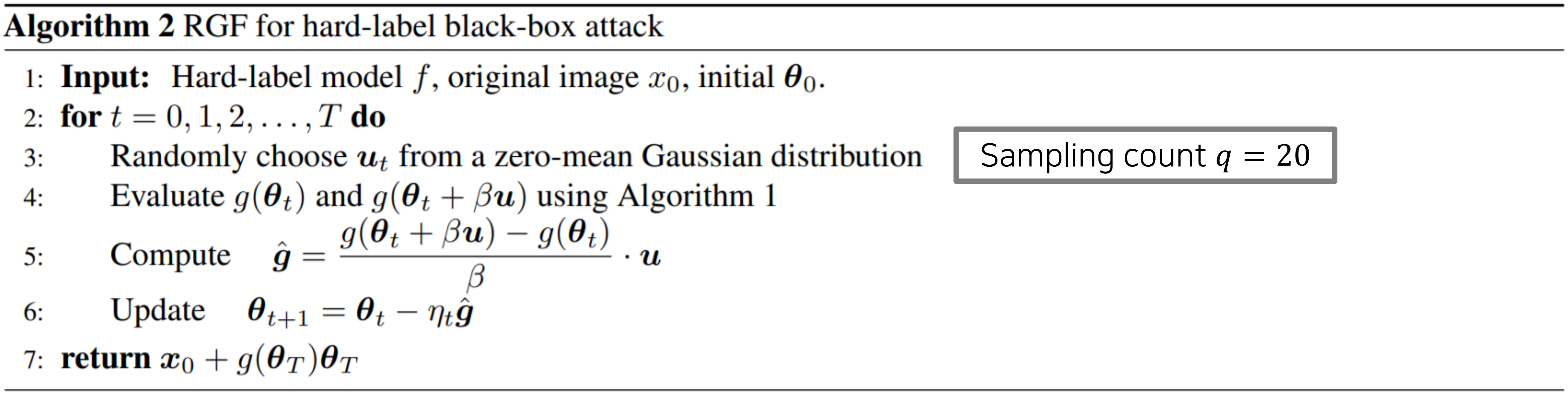
```
Input: Hard-label model $f$, original image $x_0$, query direction $\boldsymbol{\theta}$, previous value $v$, increase/decrease ratio $\alpha = 0.01$, stopping tolerance $\epsilon$ (maximum tolerance of computed error)
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}/\|\boldsymbol{\theta}\|$
if $f(\boldsymbol{x}_0 + v\boldsymbol{\theta}) = y_0$ then
    $v_{left} \leftarrow v, v_{right} \leftarrow (1+\alpha)v$
    while $f(\boldsymbol{x}_0 + v_{right}\boldsymbol{\theta}) = y_0$ do
        $v_{right} \leftarrow (1+\alpha)v_{right}$
else
    $v_{right} \leftarrow v, v_{left} \leftarrow (1-\alpha)v$
    while $f(\boldsymbol{x}_0 + v_{left}\boldsymbol{\theta}) \neq y_0$ do
        $v_{left} \leftarrow (1-\alpha)v_{left}$
## Binary Search within $[v_{left}, v_{right}]$
while $v_{right} - v_{left} > \epsilon$ do
    $v_{mid} \leftarrow (v_{right} + v_{left})/2$
    if $f(\boldsymbol{x}_0 + v_{mid}\boldsymbol{\theta}) = y_0$ then
        $v_{left} \leftarrow v_{mid}$
    else
        $v_{right} \leftarrow v_{mid}$
return $v_{right}$
```



Zeroth Order Optimization

- To solve the optimization problem, the authors use **Random Gradient-Free (RGF)** method.
- In each iteration, the gradient is estimated by

$$\hat{\boldsymbol{g}} = \frac{g(\boldsymbol{\theta} + \beta \boldsymbol{u}) - g(\boldsymbol{\theta})}{\beta} \cdot \boldsymbol{u}$$



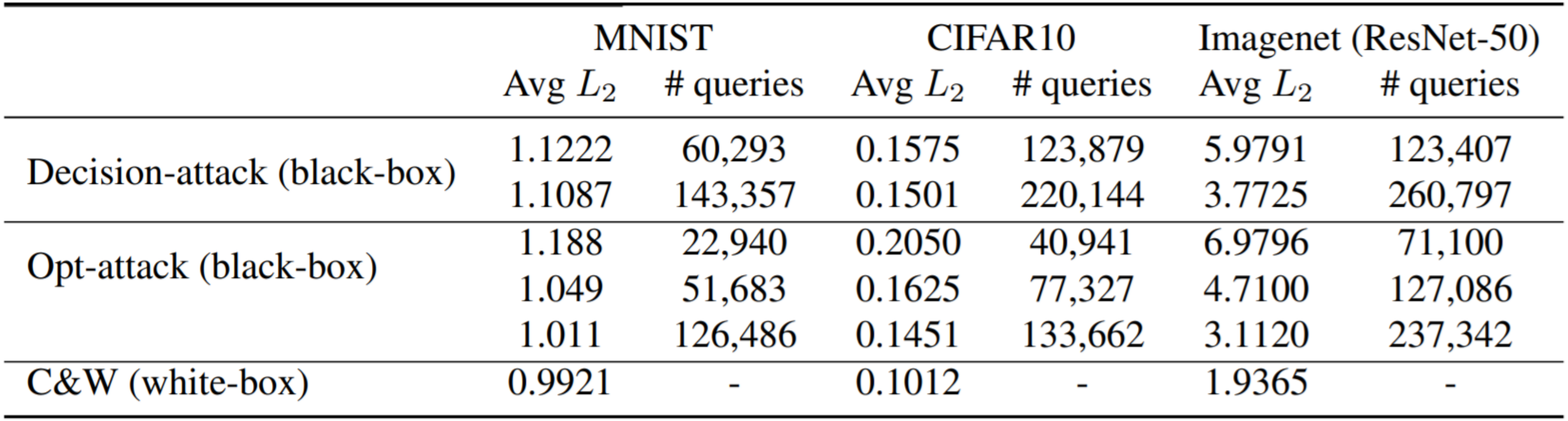
Algorithm 2 RGF for hard-label black-box attack

```
Input: Hard-label model $f$, original image $x_0$, initial $\boldsymbol{\theta}_0$.
for $t = 0, 1, 2, \ldots, T$ do
    Randomly choose $\boldsymbol{u}_t$ from a zero-mean Gaussian distribution
    Evaluate $g(\boldsymbol{\theta}_t)$ and $g(\boldsymbol{\theta}_t + \beta \boldsymbol{u})$ using Algorithm 1
    Compute $\hat{\boldsymbol{g}} = \frac{g(\boldsymbol{\theta}_t + \beta \boldsymbol{u}) - g(\boldsymbol{\theta}_t)}{\beta} \cdot \boldsymbol{u}$
    Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\boldsymbol{g}}$
return $x_0 + g(\boldsymbol{\theta}_T)\boldsymbol{\theta}_T$
```

Sampling count $q = 20$

실험 결과: Results of Untargeted Attack

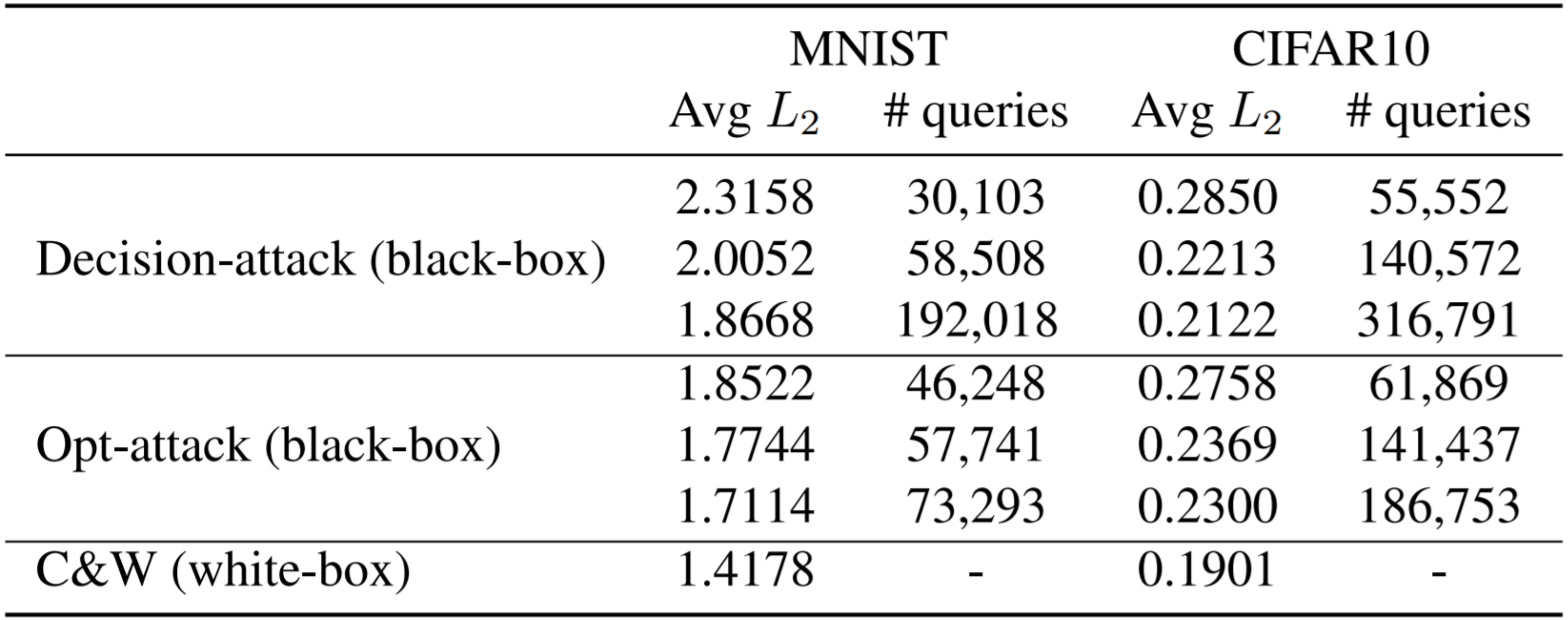
- The proposed **Opt-attack** achieves a smaller distortion than **Decision-attack (BA).**
- Compared with C&W attack, **Opt-attack** attains slightly worse distortion on MNIST and CIFAR.



	MNIST		CIFAR10		Imagenet (ResNet-50)	
	Avg L_2	# queries	Avg L_2	# queries	Avg L_2	# queries
Decision-attack (black-box)	1.1222	60,293	0.1575	123,879	5.9791	123,407
	1.1087	143,357	0.1501	220,144	3.7725	260,797
Opt-attack (black-box)	1.188	22,940	0.2050	40,941	6.9796	71,100
	1.049	51,683	0.1625	77,327	4.7100	127,086
	1.011	126,486	0.1451	133,662	3.1120	237,342
C&W (white-box)	0.9921	-	0.1012	-	1.9365	-

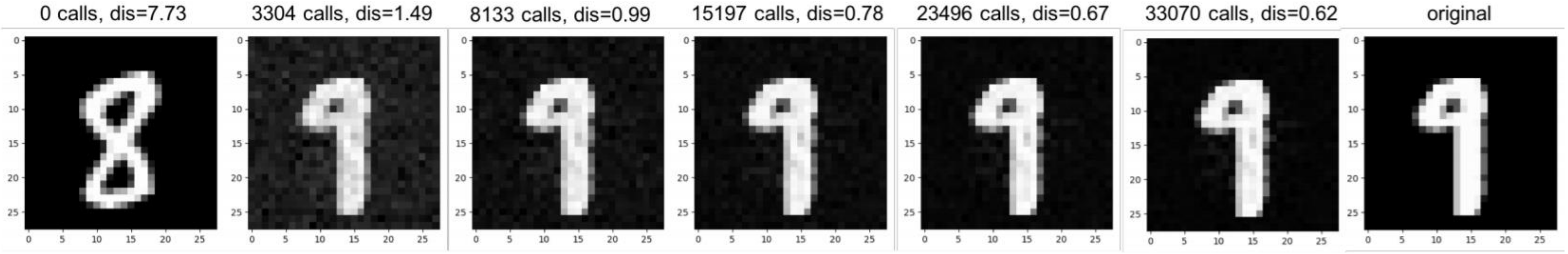
실험 결과: Results of Targeted Attack

- The proposed **Opt-attack** is better than **Decision-attack (BA)** on MNIST.
- Opt-attack has similar efficiency with Decision-attack at the first 60,000 queries on CIFAR.

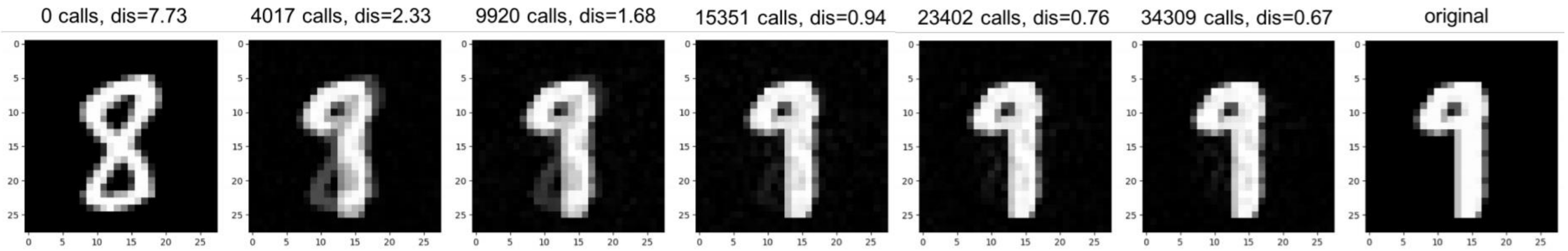


	MNIST		CIFAR10	
	Avg L_2	# queries	Avg L_2	# queries
Decision-attack (black-box)	2.3158	30,103	0.2850	55,552
	2.0052	58,508	0.2213	140,572
	1.8668	192,018	0.2122	316,791
Opt-attack (black-box)	1.8522	46,248	0.2758	61,869
	1.7744	57,741	0.2369	141,437
	1.7114	73,293	0.2300	186,753
C&W (white-box)	1.4178	-	0.1901	-

실험 결과: Results of Targeted Attack (cont’d)

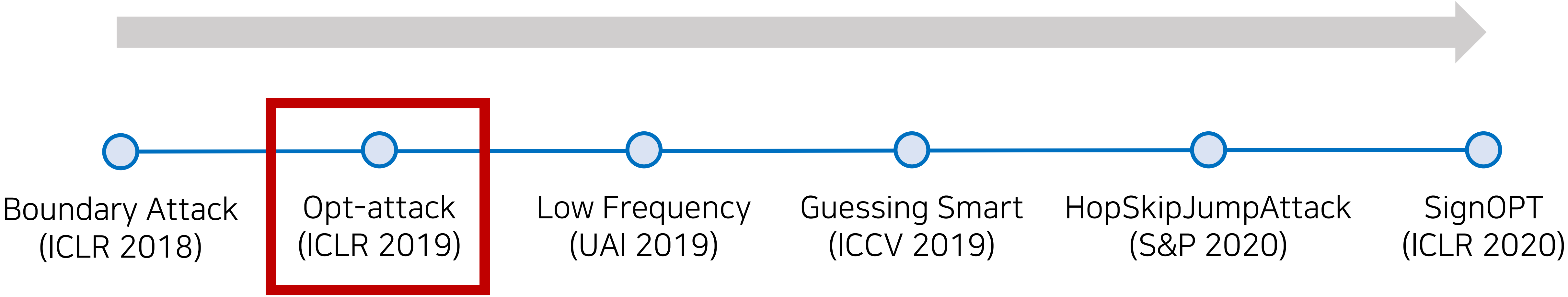


(a) Examples of targeted **Opt-attack**.



(b) Examples of targeted Decision-attack (BA).

(참고 자료) Recent Hard-label Black-box Attacks



Boundary Attack (ICLR 2018)

Opt-attack (ICLR 2019)

Low Frequency (UAI 2019)

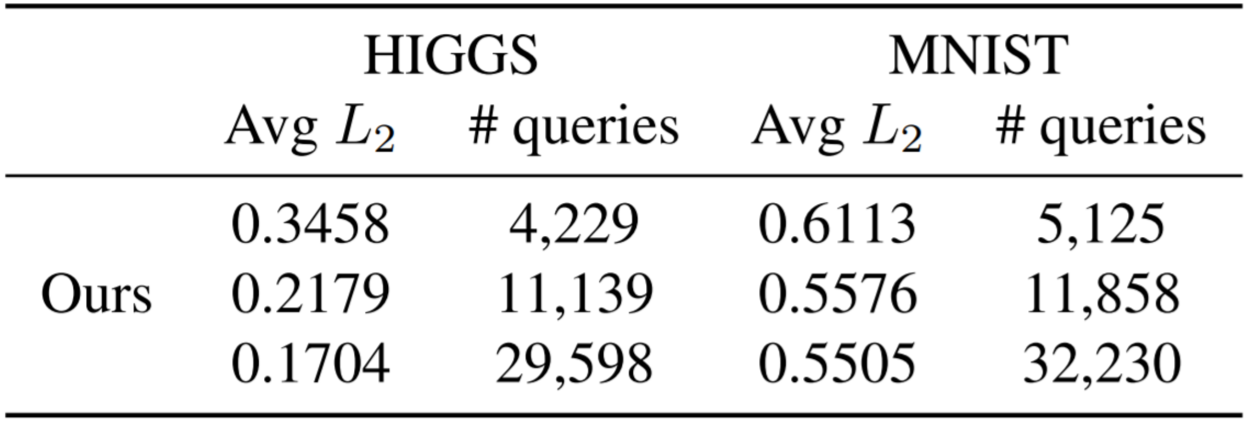
Guessing Smart (ICCV 2019)

HopSkipJumpAttack (S&P 2020)

SignOPT (ICLR 2020)

실험 결과: Attack Gradient Boosting Decision Tree (GBDT)

- The authors conduct the untargeted attack on **gradient boosting decision tree (GBDT).**
  - The GBDT is one of the discrete decision functions.
- The authors first uncover the vulnerability of GBDT models.



	HIGGS		MNIST	
	Avg L_2	# queries	Avg L_2	# queries
Ours	0.3458	4,229	0.6113	5,125
	0.2179	11,139	0.5576	11,858
	0.1704	29,598	0.5505	32,230

Conclusion

- The authors propose a generic and optimization-based hard-label black-box attack algorithm.
- The Opt-attack can be applied to discrete and non-continuous models besides neural networks.
    - The GBDT models are vulnerable under their Opt-attack.
- **Opt-attack** achieves smaller or similar distortion using 3-4 times fewer queries compared with the state-of-the-art algorithm (BA).