BOOSTING BLACK-BOX ADVERSARIAL ATTACK VIA EXPLOITING LOSS SMOOTHNESS

Hoang Tran, Dan Lu and Guannan Zhang
Oak Ridge National Laboratory, Oak Ridge, TN 37830
{tranha, lud1, zhangg}@ornl.gov

ABSTRACT

We propose to exploit smoothness of adversarial loss functions to accelerate random search for generating adversarial images in the black-box setting. Our approach stems from the observation that an adversarial loss varies more smoothly with frequency perturbation than pixel perturbation. At each iteration, we build a linear or quadratic approximation of the loss, with no additional query, around the current perturbation in the frequency domain, and use such approximation to determine the step size with a good balance between the decrease in loss and the increase in distortion. This strategy improves the performance of discrete-cosine-transform-based random search methods in which fixed step sizes are commonly used. Our experiment results on CIFAR-10 and ImageNet shows that loss smoothness can help significantly reduce the number of queries and increase the success rate.

1 INTRODUCTION

Deep neural networks (DNN) have been shown to be susceptible to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). Adversarial attacks against neural network models can be categorized into two main settings: white-box attacks (Szegedy et al., 2014; Carlini & Wagner, 2017) and black-box attacks (Narodytska & Kasiviswanathan, 2017; Chen et al., 2017). Black-box attacks search for adversarial examples by either adopting a gradient-free random search (Guo et al., 2019b; Alzantot et al., 2019), or finding a gradient surrogate from substitute networks (Papernot et al., 2016; 2017) or from model queries via finite different approximation, natural evolution strategies (NES), gradient priors, etc., (Chen et al., 2017; Bhagoji et al., 2018; Ilyas et al., 2018; 2019).

Traditionally, black-box methods require a massive amount of queries. However, recent studies have advanced several black-box approaches with significantly improved query efficiency (Tu et al., 2019; Moon et al., 2019; Li et al., 2019; Ilyas et al., 2019; Dolatabadi et al., 2020; Al-Dujaili & O'Reilly, 2020; Andriushchenko et al., 2020), and one of the most successful ℓ_2-attack methods is SimBA (Guo et al., 2019b). This method leverages the discrete cosine transform (DCT) and conducts a random search (Rastrigin, 1963) for perturbations in the low frequency DCT space (Guo et al., 2019a). A brief overview of SimBA is given in Section 2. In many evaluations with other baselines, SimBA is shown to be one of the most query efficient and competitive black-box methods. Yet, SimBA uses fixed size perturbations at every iteration, resulting in two limitations that can weaken its performance. First, when queries on both positive and negative directions do not decrease the loss (unproductive queries), SimBA does not update the perturbation. However, we argue that a good perturbation can still be extracted from those queries. Second, when a query shows a loss reduction, SimBA performs a fixed size update disregarding the margin of the loss reduction. This fixed step size may be too big (introducing unnecessary distortion), or too small (not fully exploiting the steep descent of the loss).

We propose a new strategy to address above two limitations of SimBA, based on our observation that the loss varies much more smoothly with respect to perturbations in the DCT domain than in the pixel domain. This helps us construct an accurate approximation of the loss along the DCT basis via linear or quadratic fitting *without additional query*, and we use it to find an adaptive update step size, which provides a good balance between the decrease in loss and the increase in distortion. Our method consists of two main components: an interpolation scheme to extract perturbations from unproductive queries, and an interpolation/extrapolation scheme to adaptively adjust the step size according to the magnitude of the loss reduction revealed by productive queries. We refer to our method as *Black-box Attack Based on Interpolation and Extrapolation Schemes (BABIES)*.

2 BACKGROUND

Let $f : [0,1]^d \to \mathbb{R}^K$ be a classifier with d inputs and K classes, where $f_k(\mathbf{x})$ is the predicted probability that image $\mathbf{x}$ belongs to class k. The predicted label of the image $\mathbf{x}$ is denoted by $h(\mathbf{x}) := \arg\max_{k=1,\dots,K} f_k(\mathbf{x})$. An adversary aims to generate a perturbed image, denoted by $\hat{\mathbf{x}}$, with a small perturbation that solves the following constrained optimization problem

$$\min_{\hat{\mathbf{x}}} \delta(\mathbf{x}, \hat{\mathbf{x}}) \;\text{ s.t. }\; \begin{cases} h(\hat{\mathbf{x}}) \neq h(\mathbf{x}) & \text{(untargeted)}, \\ h(\hat{\mathbf{x}}) = \hat{y} & \text{(targeted)}, \end{cases} \tag{1}$$

where $\delta(\cdot,\cdot)$ measures the perceptual difference between the original image $\mathbf{x}$ and the adversarial $\hat{\mathbf{x}}$, and $\hat{y}$ is the target label for targeted attacks. The ℓ_2 norm $\delta(\mathbf{x}, \hat{\mathbf{x}}) := \|\mathbf{x} - \hat{\mathbf{x}}\|_2$ is used as the distortion metric. For untargeted attack, we define the adversarial loss as the probability of the original class $h(\mathbf{x})$, i.e., $L(\hat{\mathbf{x}}, h(\mathbf{x})) := f_{h(\mathbf{x})}(\hat{\mathbf{x}})$. For targeted attack towards a label $\hat{y}$, the loss is $L(\hat{\mathbf{x}}, \hat{y}) := -f_{\hat{y}}(\hat{\mathbf{x}})$. Assuming B is the maximum allowable number of queries and ρ is the maximum image distortion, the optimization problem in Eq. (1) is modified to

$$\min_{\hat{\mathbf{x}}} L(\hat{\mathbf{x}}, y) \;\text{ s.t. }\; \|\hat{\mathbf{x}} - \mathbf{x}\|_2 \leq \rho, \;\text{ queries }\; \leq B. \tag{2}$$

For the rest of the paper, we suppress the dependence of L on y and write L as $L(\hat{\mathbf{x}})$ for brevity.

Our method is built upon SimBA (Guo et al., 2019b), a random search black-box approach. SimBA (see **Algorithm 1**) finds adversarial images by iteratively updating perturbation δ using fixed step size ε. At current iterate $\mathbf{x} + \delta$, assume search direction $\mathbf{q}$, three cases could happen:

C1: Query at $\mathbf{x} - \varepsilon\mathbf{q}$ and $L(\mathbf{x}+\delta-\varepsilon\mathbf{q}) < L(\mathbf{x}+\delta) \implies$ update δ to $\delta - \varepsilon\mathbf{q}$;

C2: Query at $\mathbf{x} \pm \varepsilon\mathbf{q}$ and $L(\mathbf{x}+\delta+\varepsilon\mathbf{q}) < L(\mathbf{x}+\delta) \leq L(\mathbf{x}+\delta-\varepsilon\mathbf{q}) \implies$ update δ to $\delta + \varepsilon\mathbf{q}$;

C3: Query at $\mathbf{x} \pm \varepsilon\mathbf{q}$ and $\min(L(\mathbf{x}+\delta+\varepsilon\mathbf{q}), L(\mathbf{x}+\delta-\varepsilon\mathbf{q})) \geq L(\mathbf{x}+\delta) \implies$ not update δ.

Each iteration requires from 1 to 2 queries, and the procedure is repeated on a set Q of randomly selected directions $\mathbf{q}$ until an adversarial image is found or the prescribed query budget is met.

Algorithm 1: SimBA

Procedure: SimBA($\mathbf{x}, \hat{y}, Q, \varepsilon$)
$\delta = \mathbf{0}$, $\mathcal{L} = L(\mathbf{x}+\delta)$
while $h(\mathbf{x}+\delta) \neq \hat{y}$ **do**
  Pick $\mathbf{q} \in Q$ randomly
  for $\beta \in \{-\varepsilon, \varepsilon\}$ **do**
    $L' = L(\mathbf{x}+\delta+\beta\mathbf{q})$
    if $L' < L(\mathbf{x}+\delta)$ **then**
      $\delta = \delta + \beta\mathbf{q}$
      $\mathcal{L} = L'$
      break
return δ

DCT attack. In this work, we perform attacks on the low frequency DCT domain. We extract the set of orthogonal frequencies by the DCT, retaining only a small fraction of the lowest frequency directions to craft adversarial perturbations on that subset. The effectiveness of low frequency DCT perturbations has been demonstrated in (Sharma et al., 2019; Guo et al., 2019b). Here, our new observation is that the adversarial loss is smooth along DCT basis, of which we take advantage to deploy our interpolation/extrapolation schemes.

3 THE BABIES ALGORITHM

Intuition. Potential inefficiency of Algorithm 1 can occur in the aforementioned three cases. In **C3**, two queries at $\mathbf{x} + \delta \pm \varepsilon\mathbf{q}$ are wasted leading to zero reduction of the loss. In **C1** and **C2**, the step size ε does *not* adapt to the slope $\Delta L/\varepsilon$ (where ΔL is the size of loss reduction). This fixed ε may be either too small to fully exploit a steep slope, or too big that adds too much distortion for marginal loss reduction. Our key idea is to exploit *the smoothness of the loss* in the DCT domain to improve the usage of the above one or two queries by introducing: i) an interpolation scheme to find a good update for δ in **C3**, and ii) an interpolation/extrapolation scheme to adaptively choose a step size α (instead of relying on fixed ε) in **C1** and **C2**, without additional query.

Interpolation to improve C3. Here, we have the loss values at three points $\mathbf{x}_{-\varepsilon} = \mathbf{x} + \delta - \varepsilon\mathbf{q}$, $\mathbf{x}_0 = \mathbf{x} + \delta$ and $\mathbf{x}_\varepsilon = \mathbf{x} + \delta + \varepsilon\mathbf{q}$. As $\min(L(\mathbf{x}_{-\varepsilon}), L(\mathbf{x}_\varepsilon)) \geq L(\mathbf{x}_0)$ in **C3**, *a natural idea is to fit three data points with a parabola, then update the perturbation δ such that $\mathbf{x} + \delta$ is the minimizer of the parabola,* which locates within $[\mathbf{x}_{-\varepsilon}, \mathbf{x}_\varepsilon]$. We update δ and compute the loss value at the new state via interpolation (rather than direct query) as follows. Its accuracy is illustrated in Figure 2(c).

$$\delta = \delta + \frac{\varepsilon}{2}\frac{L(\mathbf{x}_\varepsilon) - L(\mathbf{x}_{-\varepsilon})}{L(\mathbf{x}_\varepsilon) - 2L(\mathbf{x}_0) + L(\mathbf{x}_{-\varepsilon})}\mathbf{q}, \quad L_{\text{int}} = L(\mathbf{x}_0) + \frac{1}{8}\frac{(L(\mathbf{x}_\varepsilon) - L(\mathbf{x}_{-\varepsilon}))^2}{L(\mathbf{x}_\varepsilon) - 2L(\mathbf{x}_0) + L(\mathbf{x}_{-\varepsilon})}. \tag{3}$$

Interpolation/extrapolation to improve C1 and C2. Here, *the idea is to construct an empirical distribution of ΔL, and use this distribution to compute an adaptive step size α for updating δ*. Let $z = \log_{10}(\Delta L)$ and $\varepsilon_{\min}, \varepsilon_{\max}$ be the lower and upper bounds of the step size. We denote by $g_n(z|\mathbf{x})$ the empirical distribution of $\log_{10}(\Delta L)$ for the image $\mathbf{x}$ at iteration n, and by $\mu_n(\mathbf{x})$ and $\sigma_n(\mathbf{x})$ the mean and the standard deviation of $g_n(z|\mathbf{x})$, respectively. Then, α is adaptively adjusted within $[\varepsilon_{\min}, \varepsilon_{\max}]$ so that it grows linearly in z in the interval $[\mu_n - c\sigma_n, \mu_n + c\sigma_n]$, in particular,

$$\alpha = \min\left\{\varepsilon_{\max}, \max\left\{\varepsilon_{\min}, \frac{z-(\mu_n - c\sigma_n)}{2c\sigma_n}\varepsilon_{\max} - \frac{z-(\mu_n + c\sigma_n)}{2c\sigma_n}\varepsilon_{\min}\right\}\right\}. \tag{4}$$

In this way, a big ΔL leads to an aggressive step, and a small ΔL leads to a conservative step. We set $c = 0$ to use the fixed step size ε for the first 10% of the iterations, during which a good empirical distribution g_n can be constructed, and $c = 2$ after that. Similar to **C3**, we interpolate or extrapolate the loss value at the new perturbed states without additional loss queries.

Algorithm 2: BABIES in Pseudocode

Procedure:
BABIES($\mathbf{x}, \hat{y}, Q, \varepsilon, \varepsilon_{\min}, \varepsilon_{\max}, c$)
$\delta = \mathbf{0}$, $\mathcal{L} = L(\mathbf{x} + \delta)$
while $h(\mathbf{x} + \delta) \neq \hat{y}$ **do**
  Pick $\mathbf{q} \in Q$ randomly
  for $\beta \in \{-\varepsilon, \varepsilon\}$ **do**
    Compute $L' = L(\mathbf{x} + \delta + \beta\mathbf{q})$
    if $L' < L(\mathbf{x} + \delta)$ **then**
      Compute the step size α using (4)
      Update $\delta = \delta + \alpha\mathbf{q}$
      Update $\mathcal{L}$ by extrapolating at $\mathbf{x} + \delta$
      Update the empirical distribution g_n
      break
    else if $L(\mathbf{x} + \delta \pm \varepsilon\mathbf{q}) \geq \mathcal{L}$ **then**
      Update δ and $\mathcal{L}$ using (3)
      Update the empirical distribution g_n
return δ

An illustrative example. We illustrate the effectiveness of the interpolation and extrapolation on an example image. Figure 1 shows attacking DCT domain (BABIES-DCT) is easier than pixel domain (BABIES-Pixel). Compared to SimBA-DCT (same setting as BABIES-DCT but without interpolation and extrapolation), BABIES-DCT's loss decays *faster*. This is due to the fact that the loss function varies smoothly in frequency domain, so that our interpolation and extrapolation schemes are accurate (verified in Figure 2).

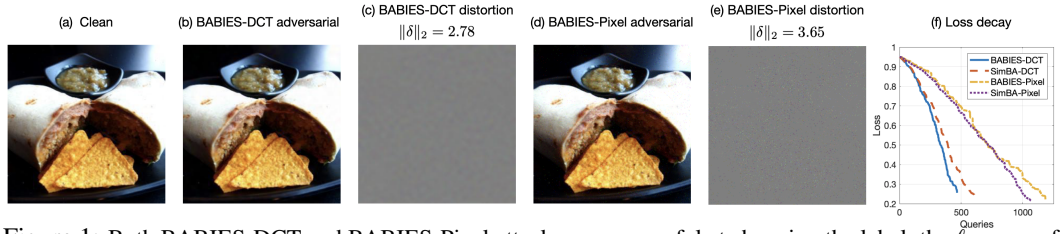

Figure 1: Both BABIES-DCT and BABIES-Pixel attacks are successful at changing the label; the ℓ_2 norm of BABIES-DCT is smaller than that of BABIES-Pixel. The two DCT attacks achieve faster loss decay than the pixel attacks, indicating that the DCT domain is easier to attack for this image. BABIES-DCT's loss decays *faster* than SimBA-DCT's loss, but BABIES-Pixel's loss decays *slower* than SimBA-Pixel's loss, because the loss function is smoother in the frequency domain than in the pixel domain, as illustrated in Figure 2.

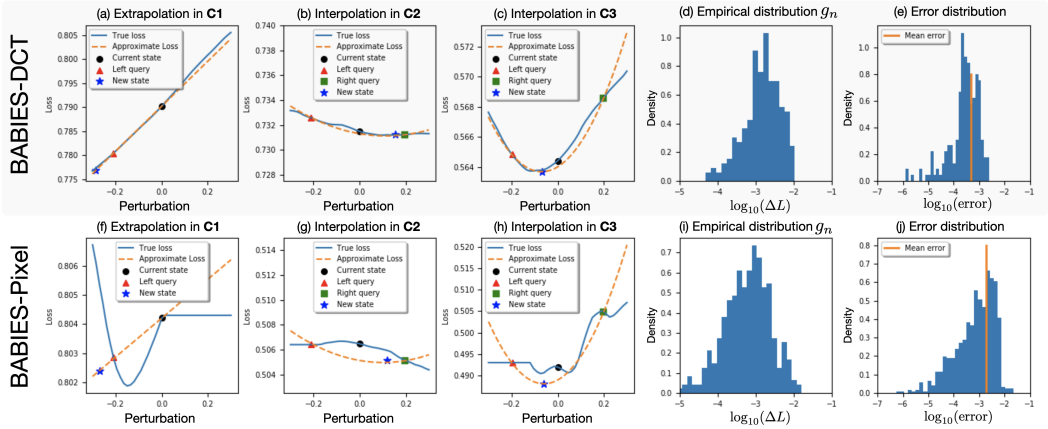

Figure 2: (a, f), (b,g), (c, h) illustrate the loss landscapes corresponding to the cases **C1**, **C2**, **C3**, respectively. The loss is much smoother in the frequency domain than in the pixel domain. BABIES-DCT can exploit such smoothness to build accurate interpolation shown in (b,c) and extrapolation shown in (a), while similar procedures for BABIES-Pixel (f,g,h) are not as successful. The average interpolation/extrapolation error of BABIES-DCT (shown in (e)) is about one order of magnitude smaller than that of BABIES-Pixel (shown in (j)).

4 EXPERIMENTAL EVALUATION

We compare BABIES with the following established algorithms: Bandits-TD ℓ_2 attack (Ilyas et al., 2019), SimBA-DCT (Guo et al., 2019b) and ℓ_2-Square Attack (Andriushchenko et al., 2020) on *targeted* attacks for CIFAR-10 and ImageNet. The following standard metrics are used to evaluate attack performance: the mean and median number of queries of successful attacks (**Avg. QY** and **Med. QY**), the success rate (**SR**), and the resulting average distortion in ℓ_2 norm (**Avg.** ℓ_2). Experiment setup, ablation study and additional results on *untargeted* attacks are included in Appendix.

Evaluating interpolation/extrapolation (Figure 3). We evaluate the effectiveness of the proposed interpolation and extrapolation schemes by comparing BABIES-DCT and SimBA-DCT. The only difference between the two methods is that BABIES-DCT uses interpolation and extrapolation. Inception_v3 is used as the target model. The result in Figure 3 sends a consistent message as Figure 2 (d,e) that these schemes can significantly improve the growth of the success rate. The distribution of the approximation error shown in Figure 3 (b,e) indicate that the interpolation/extrapolation is sufficiently accurate to guide the random search. In addition, the dramatic variation of ΔL demonstrates the necessity of adaptive step size to control the growth of the total distortion.

Aggregate statistics (Table 1 & 2). For CIFAR-10 in Table 1, our method shows a very strong performance for the Inception_v3 model, where it outperforms the others in all three metrics and requires much fewer number of queries (32% less than those from the next baseline). The evaluated methods are much more comparable for VGG13, with Square Attack having the best performance overall. The advantage of BABIES-DCT is also well demonstrated in targeted attack on ImageNet (Table 2). With comparable average ℓ_2 distortion and success rate, BABIES-DCT achieves about 18% average query reduction compared to Square Attack, the second best in this case.

Table 1: Comparison on targeted attacks for CIFAR-10

	Inception v3				**VGG13**			
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2	Avg. QY	Med. QY	SR	Avg. ℓ_2
Bandits	795	538	79.8%	3.65	682	298	94.7%	**2.78**
Square-Attack	293	148	85.3%	**3.63**	**215**	99	**97.7%**	2.79
SimBA-DCT	251	150	89.5%	3.74	414	194	80.6%	2.96
BABIES-DCT	**172**	**82**	**91.7%**	3.71	297	**96**	93.9%	2.89

Table 2: Comparison on targeted attacks for ImageNet

	Inception v3				**ResNet50**			
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2	Avg. QY	Med. QY	SR	Avg. ℓ_2
Bandits	15362	14365	82.5%	12.5	13991	10356	88.5%	9.3
Square-Attack	15211	12192	95.0%	11.9	7506	5805	99.5%	9.1
SimBA-DCT	16419	12934	90.0%	**10.8**	8761	6244	95.5%	**8.8**
BABIES-DCT	**12856**	**10744**	**97.5%**	11.7	**6115**	**5376**	**100.0%**	9.1

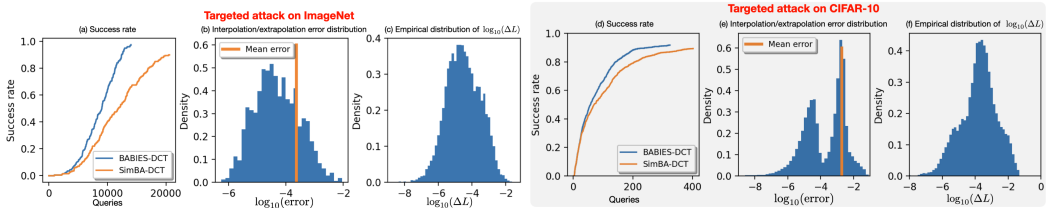

Figure 3: Comparison between BABIES and SimBA to show the effectiveness of the interpolation/extrapolation for targeted attacks. The error distribution in (b, e) show a good accuracy of interpolation/extrapolation, which leads to faster growth of the success rates for BABIES shown in (a, d). The distribution of $\log_{10}(\Delta L)$ in (c,f) shows the necessity of adaptive step size to control the growth of the total distortion.

5 ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract ERKJ352, ERKJ369; and by the Artificial Intelligence Initiative at the Oak Ridge National Laboratory (ORNL). ORNL is operated by UT-Battelle, LLC., for the U.S. DOE under Contract DEAC05- 00OR22725.

REFERENCES

Abdullah Al-Dujaili and Una-May O'Reilly. Sign bits are all you need for black-box attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SygW0TEFwH.

Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani Srivastava. Genattack: Practical black-box attacks with gradient-free optimization, 2019.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search, 2020.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

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

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pp. 1526, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352024. doi: 10.1145/3128572.3140448. URL https://doi.org/10.1145/3128572.3140448.

Hadi Mohaghegh Dolatabadi, Sarah M. Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b6cf334c22c8f4ce8eb920bb7b512ed0-Abstract.html.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In Amir Globerson and Ricardo Silva (eds.), *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pp. 1127–1137. AUAI Press, 2019a. URL http://proceedings.mlr.press/v115/guo20a.html.

Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2484–2493. PMLR, 2019b. URL http://proceedings.mlr.press/v97/guo19a.html.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2137–2146, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/ilyas18a.html.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BkMiWhR5K7.

A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3866–3876. PMLR, 2019. URL http://proceedings.mlr.press/v97/li19g.html.

Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4636–4645. PMLR, 2019. URL http://proceedings.mlr.press/v97/moon19a.html.

N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, 2017. doi: 10.1109/CVPRW.2017.172.

Nicolas Papernot, P. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *ArXiv*, abs/1605.07277, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506519, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349444. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

L. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton Remote Control*, 24:1337–1342, 1963.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/recht19a.html.

Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the Effectiveness of Low Frequency Perturbations. *arXiv e-prints*, art. arXiv:1903.00073, February 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

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

Chun-Chen Tu, Pai-Shun Ting, P. Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, C. Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *AAAI*, 2019.

A APPENDIX

A.1 EXPERIMENT SETUP

We provide details of how to setup the numerical experiments in Section 4. For the CIFAR-10 (Krizhevsky, 2009), we select 1,000 correctly labeled images (scale to $[0, 1]$) and test on two pre-trained classifiers: Inception_v3 (Szegedy et al., 2016) and VGG13, taken from the repository `https://github.com/huyvnphan/PyTorch_CIFAR10`. We set $\varepsilon = 2.0$, $\varepsilon_{\min} = 1.0$, $\varepsilon_{\max} = 3.0$ and $c = 2$ for BABIES, and constrain the perturbation's ℓ_2 norm with an additional ℓ_2 projection step. For ImageNet (Recht et al., 2019), we randomly select 200 correctly labeled images from the ImageNetV2 (Recht et al., 2019) and attack pre-trained Inception_v3 and ResNet50 classifiers, downloaded from PyTorch. We set $\varepsilon = 0.2$ for SimBA-DCT and BABIES-DCT, as suggested in (Guo et al., 2019b). The other hyper-parameters of BABIES-DCT are $\varepsilon_{\min} = 0.15$, $\varepsilon_{\max} = 0.25$ and $c = 2$. The maximum number of queries is set to be 3072 for CIFAR-10 and $10,000$ and $50,000$ for ImageNet untargeted and targeted attacks, respectively.

A.2 RESULTS ON UNTARGETED ATTACKS (TABLES 3 & 4).

Table 3 reports the comparison results evaluated in the untargeted attack on CIFAR-10. For the Inception_v3 model, BABIES-DCT significantly outperforms the other baselines in query efficiency. Our method requires 42% fewer queries compared to strongest baseline method (SimBA-DCT). Our success rate (95.4%) is second to Square Attack (96.8%). For the VGG13 model, BABIES-DCT is comparable to Square Attack and require significantly fewer queries than SimBA-DCT and Bandits. However, SimBA-DCT achieve the highest success rate. For ImageNet test case (Table 4), Square Attack has the overall best performance for both Inception_v3 and ResNet50. Even though Square Attack only achieves 92.5% success rate for Inception_v3, it requires much fewer queries on average to generate a successful attack.

Table 3: Comparison on *untargeted* attacks for CIFAR-10

	Inception v3				**VGG13**			
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2	Avg. QY	Med. QY	SR	Avg. ℓ_2
Bandits	398	58	94.3%	**2.35**	236	98	98.4%	**1.94**
Square-Attack	130	44	**96.8%**	2.40	143	51	98.1%	2.00
SimBA-DCT	110	39	89.3%	2.37	366	189	**99.9%**	1.95
BABIES-DCT	**64**	**19**	95.4%	2.39	**137**	**27**	97.5%	2.09

Table 4: Comparison on *untargeted* attacks for ImageNet

	Inception v3				**ResNet50**			
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2	Avg. QY	Med. QY	SR	Avg. ℓ_2
Bandits	3813	1904	72.5%	**4.98**	1395	810	99.5%	5.01
Square-Attack	**1932**	**832**	92.5%	4.99	**1234**	**534**	99.0%	4.98
SimBA-DCT	2770	1715	97.5%	5.35	1565	1246	**100%**	4.38
BABIES-DCT	2591	1536	**98.5%**	5.36	1391	1172	**100%**	**4.13**

A.3 QUALITATIVE RESULTS (FIGURE 4).

Since the distortion metric is only an approximation of the imperceptibility, we would like to compare how imperceptible the adversarial images are to the human eye. For that purpose, we selected four images from the *targeted* attack (on Inception_v3) experiment to explain our observations. The clean images and the distorted images are shown in Figure 4. It is easy to see that different methods lead to different types of distortion. Even though Bandits is less efficient in our experiments, it generates the most imperceptible adversarial images with comparable ℓ_2 norms. The adversarial images from BABIES-DCT and SimBA-DCT exhibit noticeable wave-like distortions for some images, especially when the background color is light. Square Attack generates more noticeable sharp distortions, because the distortion mass is concentrated in a set of small squares. One possible reason is that Square Attack almost merely searches for perturbations near the boundary of the ℓ_2 ball, which reduces the chance of finding good but small perturbations near the center of the ℓ_2 ball.

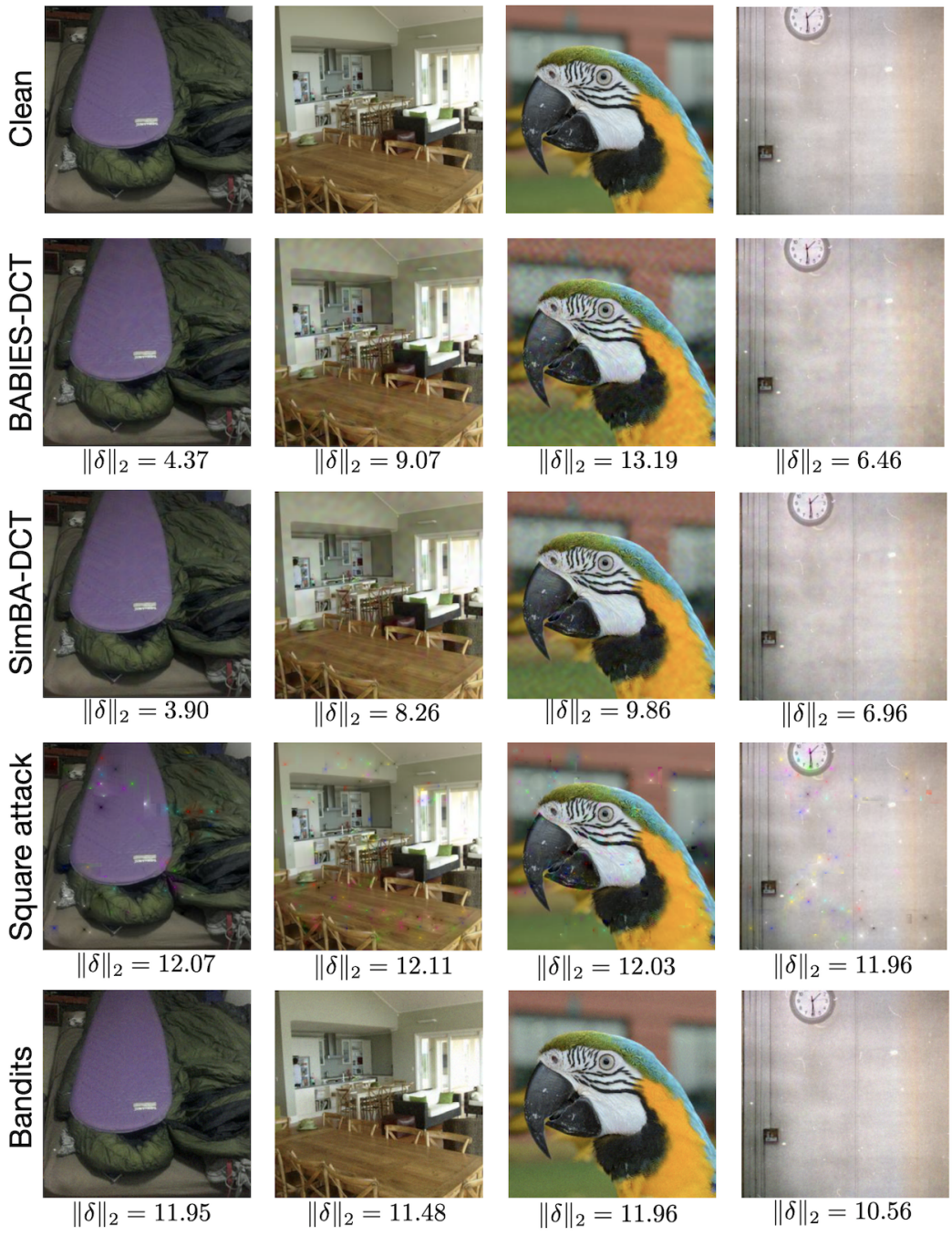

Figure 4: Qualitative comparison of the imperceptibility of distortion. The distorted images are selected from the targeted attack (on Inception_v3) experiment. Bandits produces the most imperceptible perturbations. The wave-like distortions from SimBA-DCT and BABIES-DCT are noticeable for some images. Square attack generates in general more noticeable distortions compared with the other methods.

A.4 Influence of the random seed on our algorithm

Since our algorithm is essentially a random search algorithm, it is necessary to demonstrate that the performance of our method does not vary dramatically with the change of the random seed. To this end, we use the case ***attacking Inception_v3 for ImageNet*** to test the robustness of our algorithm

with respect to the random seed. We use the same settings as in Section 4 and only change the random seed. For untargeted and targeted attacks, we run our algorithm with 20 randomly generated random seeds. The testing results are given in Table 5. We can see that our algorithm performs stably when changing the random seed. The success rate varies within 2%. The difference between the maximum and minimum numbers for both Avg.QY and Med.QY is around $2\% \sim 3\%$ of the mean values, where the standard deviation of those quantities are smaller than 2%. Thus, our idea of exploiting the smoothness of loss to accelerate random search is a statistically effective approach.

Table 5: Results on influence of the random seed (Inception_v3 for ImageNet)

Untargeted attack											
Avg.QY			Med.QY			SR			Avg.ℓ_2		
Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
2575	2551	2602	1543	1530	1569	98.8%	98.5%	99.0%	5.50	5.47	5.54
Targeted attack											
Avg.QY			Med.QY			SR			Avg.ℓ_2		
Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
12620	12432	13001	10667	10470	10827	98.2%	97.5%	99.0%	11.71	11.68	11.76

A.5 THE EFFECTIVENESS OF INTERPOLATION AND EXTRAPOLATION

To separately illustrate the effectiveness of the interpolation scheme in Eq. (3) and the interpolation/extrapolation scheme in Eq. (4), we test the performance of our algorithm by turning off one of the schemes and compare with Algorithm 2 (both schemes are turned on) and SimBA-DCT (both schemes are turned off). We use the same setting as in Section 4 except that we turn off either the interpolation and the extrapolation scheme in Section 3.

The results are shown in Table 6. We can see that the both schemes make contributions to the improvement of BABIES-DCT over SimBA-DCT. The interpolation scheme proposed in Eq. (3) makes a bigger contribution than the scheme proposed in Eq. (4). Taking the inception_v3 result in Table 7 as an example, BABIES-DCT-full reduces the Avg.QY by about 21%, compared with SimBA-DCT. The very simple interpolation scheme proposed in in Eq. (3) itself reduces the Avg.QY by 18%, and the scheme proposed in Eq. (4) only reduces the Avg.QY by 7%. This means that when both loss queries at each iteration are bigger than the current loss value from the previous iteration, i.e., the case **C3**, it is statistically more likely that the current loss value is NOT the local minimum, so that exploiting the smoothness to update the current does make significant improvement.

Table 6: Results on effectiveness of interpolation and extrapolation for **untargeted** attacks on ImageNet. SimBA-DCT: the original SimBA in Algorihm 1, BABIES-DCT-interp: only turning on the scheme in Eq. (3); BABIES-DCT-extrap: only turning on the scheme in Eq. (4); BABIES-DCT-full: turning of both schemes, i.e., Algorithm 2.

Inception v3				
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2
SimBA-DCT	2770	1715	97.5%	5.35
BABIES-DCT-interp	2683	1608	98.5%	5.62
BABIES-DCT-extrap	2701	1636	98.5%	5.66
BABIES-DCT-full	2591	1536	98.5%	5.36
ResNet50				
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2
SimBA-DCT	1565	1246	100%	4.38
BABIES-DCT-interp	1443	1187	100%	4.57
BABIES-DCT-extrap	1464	1205	100%	4.55
BABIES-DCT-full	1391	1172	100%	4.13

Table 7: Results on effectiveness of interpolation and extrapolation for **targeted** attacks on ImageNet. SimBA-DCT: the original SimBA in Algorihm 1, BABIES-DCT-interp: only turning on the scheme in Eq. (3); BABIES-DCT-extrap: only turning on the scheme in Eq. (4); BABIES-DCT-full: turning of both schemes, i.e., Algorithm 2.

Inception v3				
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2
SimBA-DCT	16419	12934	90.0%	10.8
BABIES-DCT-interp	13344	11119	97.5%	11.9
BABIES-DCT-extrap	15210	12140	97.0%	12.1
BABIES-DCT-full	12856	10744	97.5%	11.7
ResNet50				
Attack	Avg. QY	Med. QY	SR	Avg. ℓ_2
SimBA-DCT	8761	6244	95.5%	8.8
BABIES-DCT-interp	6343	5497	100%	8.8
BABIES-DCT-extrap	7385	5893	100%	9.2
BABIES-DCT-full	6115	5376	100.0%	9.1

A.6 Additional discussion

The BABIES method and SimBA essentially belong to the category of *minimum norm attack* methods, while Square Attack and Bandits are *maximum allowable attack* methods. Our method could produce adversarial images with distortions noticeably larger than the average distortion, but it could also find successful adversarial images with very small distortions. In contrast, we observe that Square Attack and Bandits turn to push the distortion to the maximum allowable one in the very early stage of the searching process, which reduces the chance of finding good attacks near the center of the ℓ_2 sphere but guarantees the ℓ_2 constraint is always satisfied. Therefore, since both type of methods have their advantages, the experimental comparison in this paper only serves the purposes of demonstrating the significance of exploiting loss smoothness.

There are several possible directions to pursue in the future research. One is to investigate the loss smoothness in other spaces, e.g., replacing DCT with wavelet transform. In fact, the idea of Square Attack makes Haar wavelet transform a good candidate to study. An advantage of using wavelet transform is that wavelet is only supported locally, which means perturbing a wavelet mode will result in a smaller distortion than perturbing a globally supported cosine basis. Another area for improvement is to perturb multiple DCT modes within each iteration for more efficient exploration. We leave these directions for future work.

A.7 Information on the codes in the supplementary material

We provide the codes to reproduce the results of BABIES and the baselines in the supplementary material. The codes for BABIES and SimBA algorithms can be found in the folder `BABIES`. The codes for Bandits and Square Attack are put in folders `Bandits` and `Square_Attack` respectively. For baseline methods, we use the default hyperparameter values suggested by the authors of those methods. Statistical results generated from running these codes will be saved in the subfolder `results` in each corresponding folder.

For CIFAR-10 test case, the attacked images, their correct and targeted labels will be loaded from file `cifar_testset.pth` folder `CIFAR10/data`. For convenience, we also include these images in `imgs` folder, as well as their labels in `.txt` files. The pretrained classifiers for CIFAR-10 was acquired from `https://github.com/huyvnphan/PyTorch_CIFAR10`. Due to their large size, we do not include these in the supplementary. For the codes to run properly, the users can download the model files `vgg13_bn.pt` and `inception_v3.pt` from the above source and put them into the `CIFAR10/models/state_dicts`.

For ImageNet test case, the attacked images will be loaded from folder `ImageNet/data/imgs`. Their correct and targeted labels are from files `class2image.txt` and `target_label.txt`

respectively. The pretrained classifiers are acquired from `torchvision.models` and will be downloaded automatically once the codes are run.

Our codes were tested on GPU with:

- `Python 3.7.9`,
- `PyTorch 1.7.1`,
- `torchvision 0.8.2`.

Further information on running each particular algorithm can be found in `README.txt` files in each corresponding folder.