Curls & Whey: Boosting Black-Box Adversarial Attacks

Yucheng Shi, Siyu Wang, Yahong Han
College of Intelligence and Computing
Tianjin University, Tianjin, China
{yucheng, syuwang, yahong}@tju.edu.cn

Abstract

Image classifiers based on deep neural networks suffer from harassment caused by adversarial examples. Two defects exist in black-box iterative attacks that generate adversarial examples by incrementally adjusting the noise-adding direction for each step. On the one hand, existing iterative attacks add noises monotonically along the direction of gradient ascent, resulting in a lack of diversity and adaptability of the generated iterative trajectories. On the other hand, it is trivial to perform adversarial attack by adding excessive noises, but currently there is no refinement mechanism to squeeze redundant noises. In this work, we propose Curls & Whey black-box attack to fix the above two defects. During Curls iteration, by combining gradient ascent and descent, we 'curl' up iterative trajectories to integrate more diversity and transferability into adversarial examples. Curls iteration also alleviates the diminishing marginal effect in existing iterative attacks. The Whey optimization further squeezes the 'whey' of noises by exploiting the robustness of adversarial perturbation. Extensive experiments on Imagenet and Tiny-Imagenet demonstrate that our approach achieves impressive decrease on noise magnitude in ℓ_2 norm. Curls & Whey attack also shows promising transferability against ensemble models as well as adversarially trained models. In addition, we extend our attack to the targeted misclassification, effectively reducing the difficulty of targeted attacks under black-box condition.

1. Introduction

The output of deep neural networks (DNNs) is highly sensitive to tiny perturbation on input images [23, 5]. Among all methods that generate adversarial examples, iterative attacks [9, 4, 27] strike a better balance between attack effect and efficiency of adversarial example generation. However, there are two severe drawbacks in current mainstream black-box iterative attacks based on substitute model [16]. In the first place, decision boundaries between models in black-box scenario are far apart [11]. Iterative trajectories have difficulties crossing decision boundary of target model with a small noise magnitude, because they are based on monotonic search along the gradient ascent direction of substitute model. This impairs adversarial examples' transferability[11]. In the second place, although noise magnitude determines the performance of attack methods, adversarial examples generated by iterative attacks contain a certain amount of redundant noises that cannot be completely removed by simply increasing the iteration number. A post-iteration refinement mechanism is needed to squeeze out the 'whey' of adversarial noises.

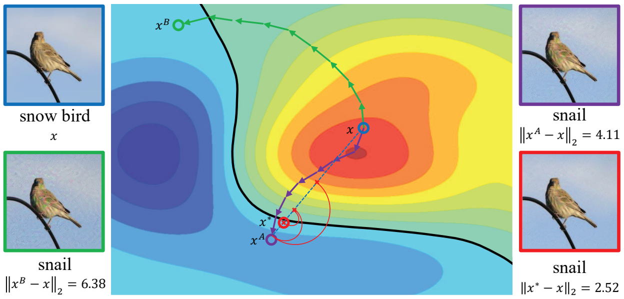


Figure 1. Iterative trajectory of Curls. Background is contour of cross entropy. The redder the color, the lower the loss. The consecutive black curve represents decision boundary between category 'snow bird' and 'snail'. Green and purple polylines represent trajectories with simply gradient ascend and Curls iteration with binary search, respectively. Blue and red rings represent the original image x and adversarial example found after binary search. Original image and three adversarial examples on both sides correspond to four rings with the same color as the image border.

In this paper, we propose Curls & Whey black-box attack. During Curls iteration, we iterate along both the gradient ascent and descent directions of substitute model's loss function, as demonstrated by green and purple polylines in Fig. 1. The dual-direction setting 'curls' up the iterative trajectories and is hence more likely to cross target model's decision boundary at a closer distance, which effectively enhances the diversity as well as transferability of adversarial examples. Diminishing marginal effect caused by monotonically adding noises along the direction of gradient ascent is also weakened. Mechanisms to refine adversarial noises (red arc in Fig. 1) and guide initial direction are included at the end and beginning of Curls iteration, respectively.

Whey optimization is applied to further squeeze the magnitude of noise by exploiting adversarial perturbation's robustness. We firstly divide adversarial perturbation into groups according to pixel value and attempt to filter out the noises of each group. Then we distill each pixel in adversarial example stochastically to squeeze out redundant noises little by little. Experiments on Imagenet [18] and Tiny-Imagenet [3] verify that our method generates adversarial examples with higher transferability and smaller perturbation in ℓ_2 norm under the same query limitation. We also systematically investigate the influence of each iterative parameter on the performance of the proposed method. In addition, our method shows strong transferability against ensemble models and adversarially trained models [24].

Targeted misclassification in black-box scenario has long been considered intractable [11], for differences on decision boundaries and classification spaces between substitute and target model hampers adversarial examples' penetration from source class to target class. Most existing iterative attacks try to solve this problem by simply replacing gradient descent in untargeted misclassification with gradient ascent towards the target class [9, 4]. In this paper, by integrating interpolation to iterative process, we boost original image into the direction towards the target category and significantly decrease the difficulty of targeted misclassification.

We summarize our contributions as follows:

(1) We bring forward Curls iteration, a black-box attack method aiming at improving diversity of iterative trajectories and transferability of adversarial examples by combining both gradient ascent and gradient descent directions.

(2) We propose Whey optimization, the first noise-squeezing method exploiting robustness of perturbations.

(3) We expand our iterative method to targeted attacks and significantly improve attack effect of iterative methods under black-box scenario.

2. Related Work

In black-box attack, attackers can only query target model and get the score of each category [14]. One practical solution exploits transferability between two models, i.e., phenomenon that adversarial examples generated by local substitute model can fool the target model [16]. Four existing attacks are introduced in the following.

Fast Gradient Sign Method (FGSM). As a classical one-step attack, FGSM [5] finds the noise's direction by calculating the gradient of cross-entropy loss $J(x, y_T)$:

$$x' = x + \varepsilon \cdot sign(\bigtriangledown J(x, y_T)). \tag{1}$$

Iterative FGSM (I-FGSM). I-FGSM [9] splits uppper bound of noise ε into several small step size α and adds noises step by step:

$$x'_{t+1} = Clip_{x,\varepsilon}\{x'_t + \alpha \cdot sign(\bigtriangledown J(x'_t, y_T))\}. \tag{2}$$

I-FGSM possesses the highest attack effect among all current iterative attacks in white-box scenario. Its main drawback is the diminishing marginal effect of iterative steps. In other words, as the number of iterations t increases and the step size α decreases, keeping adding the iteration step has little improvement on attack effect.

Momentum Iterative FGSM (MI-FGSM). MI-FGSM [4] introduced a momentum term to make the adjustment of the noise-adding direction smoother, but the impact of diminishing marginal effect on iteration number still exists:

$$m_{t+1} = \mu \cdot m_t + \frac{\bigtriangledown J(x'_t, y_T))}{\| \bigtriangledown J(x'_t, y_T))\|}, \tag{3}$$

$$x'_{t+1} = Clip_{x,\varepsilon}\{x'_t + \alpha \cdot sign(g_{t+1})\}. \tag{4}$$

Variance-Reduced Iterative FGSM (vr-IGSM). Vr-IGSM [27] uses an averaged gradient of original image with gaussian noises to eliminate local fluctuation in substitute model and therefore improves the transferability.

$$G_{t+1} = \frac{1}{m}\sum_{i=1}^{m} \bigtriangledown J(x_t + \xi_i), \quad \xi_i \sim \mathcal{N}(0, \sigma^2 I), \tag{5}$$

$$x'_{t+1} = Clip_{x,\varepsilon}\{x'_t + \alpha \cdot sign(G_{t+1})\}. \tag{6}$$

A series of defense methods have been proposed to improve robustness of target models [15, 10, 12]. Among them, adversarial training [24] and model ensemble are two most widely-used methods. Adversarial training vaccinates against adversarial examples by including them into the training set of target model, while model ensemble reduces specific error made by single model.

3. Curls & Whey Attack

3.1. Notation

An image classifier based on DNN can be represented as $N : X^{W \times H \times C} \rightarrow Y^K$, where X represents the input space with dimension of $Width \times Height \times Channel$ and Y represents the classification space with K categories. A successful adversarial attack changes the original classification result of image classifier, i.e., the target model, after adding as little noise as possible to the original image [26]:

$$min \, \|x' - x\|_v, \quad s.t. \; N(x) \neq N(x') \, , \tag{7}$$

where v refers to the norm used to measure the noise magnitude including ℓ_1, ℓ_2 and ℓ_∞ norm. In this paper we discuss noise magnitude in ℓ_2 norm. Some existing

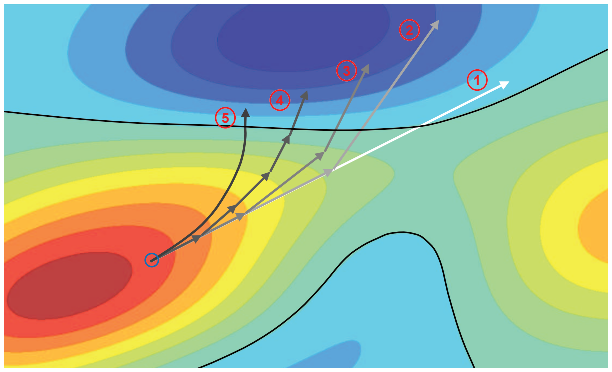


Figure 2. Diminishing marginal effect on iteration number T. The small blue ring at the bottom left represents the original image. Five polylines marked ① - ⑤ are iterative trajectories for $T = 1, 2, 3, 5, \infty$ cross the decision boundary.

works [28, 29, 4] compare the misclassification rate with a fixed ℓ_∞ norm, but we concentrate on the quality of adversarial noises generated by different attacks on one image. Here the black-box attack using substitute model [16] is used to solve the problem that the target model cannot be back propagated. The gradient information at step t refers to the gradient value of the substitute model's loss function J_{sub}, i.e., cross-entropy loss, to adversarial example x'_t.

3.2. Diminishing Marginal Effect on Iteration Steps

Iterative attacks perform well in white-box scenarios, where the transferability is guaranteed to be 100% [13]. However, when attacking against a black-box target model, the drawbacks of iterative attacks gradually expose. First of all, discrepancy on decision boundary burdens transferability between substitute model and target model [25]. Iterative attacks always step toward the direction in which loss function of substitute model increases. But there is a huge gap on classification spaces between different models. Their gradient directions may be even orthogonal to each other [11]. Therefore, simply searching for adversarial examples along the gradient ascent direction of the substitute model may no longer be suitable for black-box attacks.

What's more, diminishing marginal effect on the number of iterations exists. Now assume that in order to minimize the noise magnitude, the step size α of each step is inversely proportional to the total iteration numbers. In I-FGSM, when the number of iterations T increases by 1, the marginal gain for the decrease in the noise magnitude is

$$\sum_{t=1}^{T+1}\frac{1}{T+1}\cdot\bigtriangledown J_{sub}(x_t)-\sum_{t=1}^{T}\frac{1}{T}\cdot\bigtriangledown J_{sub}(x_t). \tag{8}$$

In general, as T increases and the single step size shortens, the iterative trajectory tends to be consistent and smooth and gradually converges, as shown in Fig. 2. Considering that the number of queries to the target model in black-box attack is also limited, increasing the iteration number has little effect on adversarial noise reducing if the iteration number is already high.

3.3. Curls Iteration

Iterative trajectories of current iterative attacks in black-box scenario are monotonic. First, monotonically employing gradient ascent along substitute model's loss function is more likely to bring iterative trajectories into local optimum of substitute model, rather than passing through the decision boundary of target model. Second, simply relying on transferability between substitute model and target model, but ignoring the feedback of target model after each query makes the iterative trajectories lack adaptability.

To 'curl' up and diversify the iterative trajectory may be a more cost-effective solution [19]. Fig. 1 shows one possible distribution of target model loss function. In the case that loss function rises slowly along the direction of gradient ascend, like the green trajectory, it may be possible to find a shortcut across the decision boundary from a nearby starting point, as shown by the purple polyline in Fig. 1. We abandon the monotonic search strategy base on gradient ascend to increase the diversity of iterative trajectories:

$$\begin{aligned} x'_0 &= x,\ x'_1 = Clip_{x,\varepsilon}\{x'_0 - \alpha\cdot\bigtriangledown J_{sub}(x'_0)\}, \quad (9)\\ g_{t+1} &= \begin{cases} -\bigtriangledown J_{sub}(x'_t) & J(x'_t) < J(x'_{t-1}),\\ \bigtriangledown J_{sub}(x'_t) & J(x'_t) \geq J(x'_{t-1}), \end{cases} \quad (10)\\ x'_{t+1} &= Clip_{x,\varepsilon}\{x'_t + \alpha\cdot g_{t+1}\}, \quad (11) \end{aligned}$$

where $J_{sub}(x'_t)$ and $J(x'_t)$ represent the cross entropy loss of adversarial example x'_t on the substitute model and the target model, respectively. First, update the original image for one step along the direction of gradient descent. When the cross entropy loss of current adversarial example on target model is lower than the previous step, usually the 'valley floor', i.e., the local minimum of loss function has not yet been reached. Therefore, when the loss on the target model is still declining, continue to update along the direction of gradient descend, and vice versa. We regard this 'first go down then go up' iterative method as Curls iteration.

On the basis of Curls, we introduce two heuristic strategies before and after each round of iteration. For an image, the closest adversarial examples are more likely to distribute in roughly the same direction in the feature space. Therefore, we record and update the average direction of all adversarial examples of one image, $\bar{R}$, and add a vector pointing to this direction in the first step when calculating gradients for each round:

$$\bar{R} = \frac{1}{K}\sum_{i=1}^{K} x', \quad s.t.\ N(x) \neq N(x')\,, \tag{12}$$

$$x'_1 = Clip_{x,\varepsilon}\{x'_0 + \alpha\cdot\bigtriangledown J(x'_0 + \alpha\cdot\bar{R})\}. \tag{13}$$

Algorithm 1 Curls Iteration

```
Input: Target DNN $N(x)$, substitute model $Sub(x)$
       Original image $x$ and label $y$
       Initial noise magnitude limit $\varepsilon$
       Iteration step $T$ and variance of gaussian noise $s$
       Step size $\alpha$ and binary search step $bs$
Output: Adversarial example $x'$
Initialize $\bar{R}$ and two starting points
$\bar{R} = 0, x_0^A = x, x_0^B = x$
$downhill = True$ // Set the gradient descend flag to True
for t = 0 to $T$ do
    $\xi_t^A, \xi_t^B \sim \mathcal{N}(0, s^2 I)$
    Calculate gradient on substitute model
    $g_t^A = \bigtriangledown J_{sub}(x_t^A + \xi_t^A + \alpha \cdot \bar{R})$
    $g_t^B = \bigtriangledown J_{sub}(x_t^B + \xi_t^B + \alpha \cdot \bar{R})$
    $x_{t+1}^A = \begin{cases} Clip_{x,\varepsilon}\{x_t^A - \alpha \cdot g_t^A\} & downhill = True \\ Clip_{x,\varepsilon}\{x_t^A + \alpha \cdot g_t^A\} & downhill \neq True \end{cases}$
    $x_{t+1}^B = Clip_{x,\varepsilon}\{x_t^B + \alpha \cdot g_t^B\}$
    if $downhill = True\ and\ J(x_{t+1}^A) > J(x_t^A)$ then
        $downhill = False$
    end if
    if $N(x_{t+1}^A) \neq N(x)\ or\ N(x_{t+1}^B) \neq N(x)$ then
        update $\bar{R}$ by Eqn. (12)
    end if
end for
if $N(x_T^A) \neq N(x)\ or\ N(x_T^B) \neq N(x)$ then
    $x' = \begin{cases} x_T^A & \|x_T^A - x\|_2 < \|x_T^B - x\|_2 \\ x_T^B & else \end{cases}$
    refine $x'$ by Eqn. (15)
end if
return $x'$
```

Since the iterative trajectory cannot be a straight line in the high-dimensional feature space, situation shown in the red arcs in Fig. 1 exists: there are adversarial examples with smaller ℓ_2 distance between the adversarial example found and original image. We perform binary search between original image x and adversarial example x' after each round to fully exploit the potential of this round:

$$L = x, R = x', \tag{14}$$

$$BS(L,R) = \begin{cases} BS(L,(L+R)/2), \\ if\ N(x) \neq N((L+R)/2), \\ BS((L+R)/2,R), \\ if \quad N(x) = N((L+R)/2). \end{cases} \tag{15}$$

In the actual implementation of Curls iteration, in order to prevent the oscillation of adversarial noise update, we do not directly determine the gradient symbol on account of target model's loss function, but divide each iterative round into two stages. In the first stage, carry out gradient descend to the original image. Once the cross entropy on target model is lower than the previous step, the second stage starts and carries out gradient ascend until the last step. At the same time, the normal iterative trajectory of direct gradient ascent is performed simultaneously. In addition, inspired by vr-IGSM [27], we add gaussian noise to image in gradient calculation process to improve the transferability. Algorithm 1 details Curls iteration.

3.4. Whey Optimization

Usually an iterative attack ends as soon as it finds adversarial example or runs out of iteration number. However, adversarial examples generated may still contain redundant 'whey' noises after iteration. Or the maximum extent to which noises can be reduced, while ensuring the adversarial example can still fool the target model [1]:

$$max(\| x' - x \|_2 - \| x^\circ - x \|_2), \quad s.t. \quad N(x') = N(x^\circ),$$

where x, x' and x° refers to original image, adversarial example found by now and the closest adversarial example to the original image, respectively.

Since binary search between x and x' is already performed, adversarial examples with less redundant noises are more likely to exist in a linearly independent direction with respect to $x' - x$. We propose Whey optimization to squeeze out the remaining 'whey' of redundant noises in black-box attack. Whey optimization maintains a balance between noise-squeezing amplitude and the number of squeezes. Squeezing excessive noises at a time may return adversarial examples to the original category. Nevertheless, an incremental squeeze makes it impossible for optimization to complete within a limited number of queries. A compromise solution is to divide adversarial noises into groups first, then try to reduce noise magnitude group by group:

$$\begin{aligned} z_0 &= x' - x, & (16) \\ z_{t+1}^{whc} &= z_t^{whc}/2, \quad s.t.\ z_t^{whc} = L(V(z_0), t), & (17) \end{aligned}$$

where z is the noise, $L(V,t)$ represents number with the t^{th} largest absolute value in pixel value set V:

$$V(z) = \{v \mid v = z^{whc},\ w \in [0,W], h \in [0,H], c \in [0,C]\}$$

W, H, C represents the width, height and channel of original image x, respectively. Whey optimization divides noise z into several groups according to the pixel value, selects one group each time in descending order, reduces all pixel value in z which equals to $L(V,t)$ by half and check whether the trimmed noises can still fool the target model.

After squeezing in groups, we perform more fine-grained squeeze. The last step of Whey optimization set the value

Algorithm 2 Whey Optimization

```
Input: Target DNN $N(x)$ and adversarial example $x'$
    Original image $x$ and label $y$
    Max attempt number for two squeeze steps, $T_1, T_2$
    Pixel value set of $x' - x$, $P$
    Random number generator over $[0, 1]$, random()
Output: Refined adversarial example $x^*$
$z = x' - x$
$t_1 = 0, t_2 = 0$
for $p$ in $P$ and $t_1 < T_1$ do // Step 1: Squeeze in groups
    Reduce the pixel value by half
        $z\,[z = p]\; / = 2$
    if $N(z) = y$ then
        Cancel the update of this step
    end if
    $t_1 = t_1 + 1$
end for
while $t_2 < T_2$ do // Step 2: Squeeze stochastically
    Generate a random mask same shape as the image
        $mask^{whc} = \begin{cases} 0 & random() \leq 0.01, \\ 1 & else. \end{cases}$
    $z = z \cdot mask$ // Element-wise product
    if $N(z) = y$ then
        Cancel the update of this step
    end if
    $t_2 = t_2 + 1$
end while
$x^* = z + x$
return $x^*$
```

of each pixel to 0 with probability of δ:

$$z_{t+1} = z_t \cdot mask_t, \tag{18}$$

$$mask^{whc} = \begin{cases} 0 & random() \leq \delta, \\ 1 & else, \end{cases} \tag{19}$$

where $mask$ is the same shape as z. Algorithm 2 gives the detail of Whey optimization.

3.5. Targeted Attack

Unlike untargeted attack, targeted attack requires not only the adversarial example be misclassified by the target model, but also it can be misclassified into the specified category. This is especially difficult in black-box attack because the decision boundaries between different models vary greatly, and the gradient direction are even orthogonal to each other [11]. Even if the update of each step is changed from gradient ascend with respect to the original category $\bigtriangledown J_{sub}(x', y_{ori})$ to gradient descend with respect to the target category $- \bigtriangledown J_{sub}(x', y_{target})$ [4], an iterative trajectory from original image is almost impossible to reach the target category space, due to the difference in gradient values between target model and substitute model.

We abandon the 'start from scratch' strategy and integrate interpolation to the iterative attack to get a better initial update direction. First, we collect a legitimate image x_T that can be classified into the target category by the target model. Second, we use binary search to find an image x'_0 between the original image x and x_T, making sure that x'_0 can also be classified into the target category. After that, we use x'_0 to guide the first gradient ascent step starting from x:

$$x'_0 = (1 - s) \cdot x + s \cdot x_T, \tag{20}$$

$$x'_1 = Clip_{x,\varepsilon}\{x - \alpha \cdot \bigtriangledown J(x'_0)\}, \tag{21}$$

$$x'_{t+1} = Clip_{x,\varepsilon}\{x'_t - \alpha \cdot \bigtriangledown J(x'_t)\}, t \geq 1, \tag{22}$$

where $0 < s < 1$ indicates the interpolation coefficient determined by binary search. In this way, we boost original example into the direction towards the target category. After the first boosting step, we continue to apply Curls&Whey attack as in untargeted attacks.

4. Experiments

4.1. Experiment Settings

All our experiments are performed on Tiny-Imagenet used in NIPS 2018 Adversarial Vision Challenge [3] and Imagenet [18], with image shape of 64 × 64 × 3 and 224 × 224 × 3, respectively. Imagenet contains 1000 image categories. We picked 10000 images from its validation set that can be correctly classified by all target models, 10 images for each category. As for Tiny-Imagenet with 200 image categories, we choose 2000 images, 10 images for each category. 8 neural network models with different structures are compared: resnet-18 [6], resnet-101, inception v3 [22], inception-resnet v2 [21], nasnet [30], densenet-161 [8], vgg19-bn [20], senet-154 [7].

We implement our black-box iterative attack on Foolbox [17] framework. In order to accurately measure the attack effect of each method, a large loop for determining ε is added outside the iterative process. For evaluation criterion, we choose the median and average size of adversarial perturbation transferred from substitute model to target model, as applied in NIPS 2018 Adversarial Vision Challenge [3]:

$$mid(Sub, N) = median(\{d(x, x^*) \mid x \in \mathbf{X}\}), \tag{23}$$

$$avg(Sub, N) = \frac{1}{N} \sum_{i=1}^{N} (\{d(x, x^*) \mid x \in \mathbf{X}\}), \tag{24}$$

$$d(x, x^*) = \|x - x^*\|_2, \tag{25}$$

where sub and N represent substitute model and target model, respectively. x is an original image in the test set $\mathbf{X}$. x^* is the adversarial example found that is closest to x. $d(x, x^*)$ returns the ℓ_2 distance between x and x^*. A

Table 1. Median and average ℓ_2 distance of adversarial perturbation crafted from pairwise attack between four models.

		resnet18		inceptionv3		inception resnet v2		nasnet	
	attack methods	median	average	median	average	median	average	median	average
resnet 18	FGSM	*0.1321*	*0.8893*	4.3085	7.4580	3.6764	5.3257	3.4187	4.5589
	I-FGSM	*0.0800*	***0.0881***	1.9686	2.9287	2.4624	3.3192	2.1865	2.9644
	MI-FGSM	*0.0866*	*0.1029*	2.3220	3.4386	2.9526	3.9267	2.0174	2.9723
	vr-IGSM	*0.0941*	*0.1120*	1.8737	2.8228	2.4803	3.4085	1.7991	2.7645
	Curls	*0.0731*	*0.1182*	1.6443	2.4739	1.8507	2.6290	1.6773	2.4919
	Curls&Whey	***0.0627***	*0.1040*	**1.1942**	**1.7387**	**1.4549**	**1.9450**	**1.3902**	**1.9696**
inception v3	FGSM	0.9944	3.6262	*0.1521*	*1.9010*	2.6171	4.9078	2.8729	4.5217
	I-FGSM	0.6699	1.8883	***0.1132***	***0.1518***	1.3415	1.9095	1.3774	2.1675
	MI-FGSM	0.8124	2.2895	*0.1283*	*0.1989*	1.6248	2.4642	1.6800	2.7336
	vr-IGSM	0.6072	1.7973	*0.1297*	*0.1834*	1.3214	2.0991	1.3569	2.3010
	Curls	0.5760	1.6781	*0.1243*	*0.2194*	1.1163	1.8997	1.2335	2.1067
	Curls&Whey	**0.5140**	**1.4941**	*0.1252*	*0.9200*	**0.9058**	**1.7913**	**0.9398**	**1.9315**
inception resnet v2	FGSM	1.6729	5.0270	4.2482	6.6191	*0.2855*	*4.5974*	4.1107	5.5487
	I-FGSM	0.7019	2.3966	1.3314	2.3834	***0.1293***	*0.3814*	1.3761	2.3732
	MI-FGSM	0.8561	2.8611	1.6342	3.0884	*0.1602*	*0.5419*	1.6594	3.0469
	vr-IGSM	0.6463	2.4453	1.3166	2.6256	*0.1640*	*0.5197*	1.3292	2.6710
	Curls	0.6040	2.0220	1.1325	1.9407	*0.1501*	*0.3450*	1.0978	1.9644
	Curls&Whey	**0.5227**	**1.2404**	**0.8431**	**1.3437**	*0.1485*	***0.3199***	**0.8483**	**1.4403**
nasnet	FGSM	3.7356	6.0550	3.5277	7.2388	3.4829	7.1657	*0.2008*	*6.3891*
	I-FGSM	1.5575	4.1401	1.5926	4.3745	1.4180	4.2968	***0.1173***	*1.8225*
	MI-FGSM	0.9518	3.0544	1.8850	3.9685	1.6458	3.7643	*0.1317*	*0.3632*
	vr-IGSM	0.5659	2.4410	1.5006	**3.2440**	1.3066	**3.1112**	*0.1371*	***0.3197***
	Curls	0.5821	2.1520	1.2719	3.9490	1.2048	4.1637	*0.1360*	*2.7491*
	Curls&Whey	**0.5543**	**1.8582**	**1.0003**	3.6760	**0.9599**	3.6069	*0.1354*	*2.5653*

smaller ℓ_2 distance indicates a stronger attack effect and higher the transferability of generated adversarial examples.

4.2. Black-box Attack on Multiple Models

We report the median and average adversarial perturbation on Tiny-Imagenet in Table 1. In this 4×4 matrix, each element represents the result of substitute model of this row against the target model of this column over the entire 2000 images. Elements on diagonal are results of white-box attacks (marked in italics). Fig. 4 shows median perturbation on three target models when using vgg19-bn as substitute model. More experiments on Imagenet can be found in supplemental material. For each pair of substitute and target model, we compare our methods (Curls&Whey as well as Curls only) with FGSM [5] and three other iterative attacks, I-FGSM [9], MI-FGSM [4] and vr-IGSM [27]. Since ℓ_2 norm is used to measure noise magnitude, we no longer use sign function to update adversarial examples. For the fairness of comparison, the number of queries to the target model is basically equal for the iterative attacks. Table 2 reports parameters related to query number, including iterative round number T_0, iteration step T, binary search step bs, max attemp number for two squeeze steps in Whey optimization T_1 and T_2. The total query number for our method is $T_0 \times (T + bs) \times 2 + T_1 + T_2$, and $T_0 \times T$ for other iterative methods. The initial noise magnitude ε and stepsize α are 0.3 and $1/2T$, respectively. For variance of gaussian noise in vr-IGSM and our method, we set $s = 1$.

Table 2. Parameter set for experiments on two datasets.

		T_0	T	bs	T_1	T_2	Total
Tiny-Imagenet	Others	20	10	–	–	–	200
	Ours	10	4	2	40	40	200
Imagenet	Others	24	24	–	–	–	576
	Ours	14	7	3	200	100	580

It can be seen from Table 1 that Curls&Whey achieves smaller median noise magnitude in ℓ_2 norm than all other methods, and smaller average magnitude than most other methods, on black-box attacks, i.e., off-diagonal elements. With the diversification of iterative trajectories and squeeze of redundant noises, noises are reduced by 20%-30%, in some cases even 40%, over most model combinations. Curls iteration alone also outperforms existing methods in almost all black-box attacks. Due to gaussian noises in gradient-calculating process, noise magnitude of our methods are slightly higher than I-FGSM in white-box attacks, where transferability is no need to be considered. However,

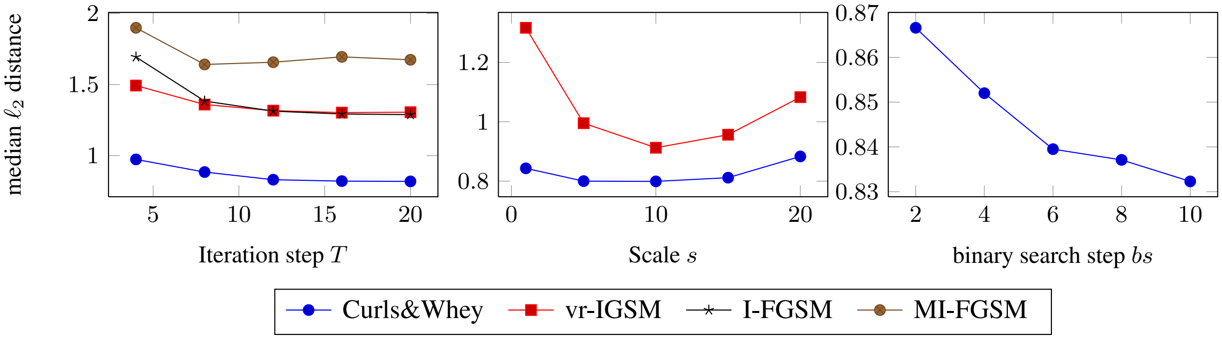


Figure 3. Median noise magnitude under different iteration steps (left), $T = (4, 8, 12, 16, 20)$, gaussian noise variance (middle), $s = (1, 5, 10, 15, 20)$ and binary search step (right), $bs = (2, 4, 6, 8, 10)$.

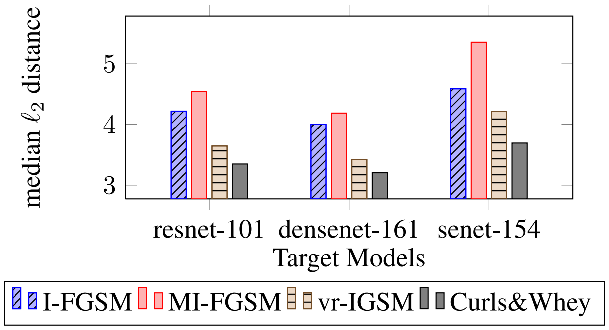


Figure 4. Median ℓ_2 distance comparison of adversarial noises generated using vgg19-bn as substitute model on Imagenet.

white-box noise of our method is still smaller than that of vr-IGSM, which validates the effectiveness of Whey optimization. Fig. 5 shows adversarial examples crafted on two datasets. Curls & Whey achieves targeted and untargeted misclassification with nearly imperceptible noises.

4.3. Ablation Study

Here we investigate influence of iteration step T, binary search step bs and variance of gaussian noise s to black-box attack effect. We use inception-resnet v2 and inception v3 as substitute and target model, respectively. Results on Tiny-Imagenet under different T, s and bs is shown in Fig. 3. As discussed in Section 3, although T is negatively correlated with noise magnitude, diminishing marginal effect exists. The noise drop of $T = 20$ relative to $T = 16$ is obviously not as great as the drop of $T = 8$ relative to $T = 4$. Our method does not simply increase the iteration number, but improve the diversity of iterative trajectories. Therefore, Curls&Whey is able to find adversarial examples with smaller ℓ_2 norm with equal queries, and use part of the query to refine adversarial noises.

Variance s is related to the transferability between substitute and target model. The higher the s, the greater the likelihood that adversarial example may transfer from one model to another highly different model. However, as the variance of gaussian noise increases, the proportion of original image in gradient calculation process will gradually decrease, resulting in decline in transferability. Therefore, a local minimum appears in the results on different s. As can be seen from Fig. 3, when using inception-resnet v2 to attack inception v3, the local optimal value of s is around 10.

Table 3. Incremental comparison on each part of Curls&Whey.

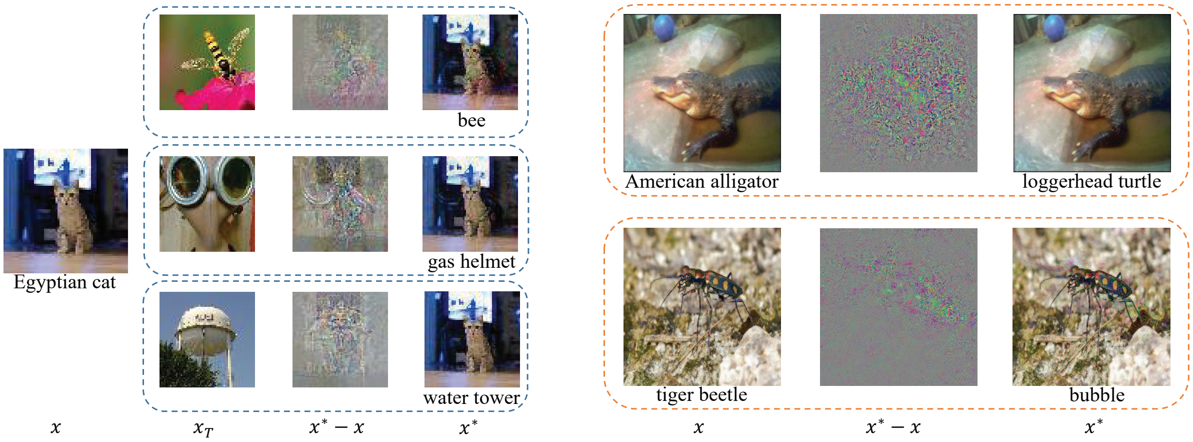
	Curls	+BS	+Whey(1)	+Whey(2)
median	1.3138	1.1111	0.9354	**0.8431**
average	2.3154	1.9039	1.4723	**1.3437**

As for binary search step, a larger bs means more binary search between the adversarial example and original image. As an auxiliary process in Curls iteration, a relatively small bs is sufficient to reduce the noises.

To verify the effectiveness of each part of our attack method, we conduct ablation experiment on Curls&Whey. As can be seen from Table 3, whether it is Curls iteration, binary search (BS), or two steps in Whey optimization, each component can effectively reduce the noise magnitude.

4.4. Targeted Attack Results

For targeted attack, we assign 5 target categories for each image and calculate the ℓ_2 distance between original image and adversarial examples of each category. We select one image from the test set that can be classified into target category for interpolation. We choose resnet18 and inceptionv3 as our substitute model and three other models as target models. As shown in Fig. 6, three existing iterative attacks have difficulties achieving targeted misclassification. Compared to three decision-based attacks, boundary attack [2], pointwise attack and vanilla interpolation [17], noise magnitude of our method is also significantly reduced. This confirms the effectiveness of integrating interpolation method into Curls & Whey attack.



(a) Targeted Attack

(b) Untargeted Attack

Figure 5. Adversarial examples generated by Curls & Whey attack. Targeted attack results on Tiny-Imagenet are shown on subplot (a). Original image x, image of target category x_T, noise $x^* - x$ and adversarial example x^* are listed from left to right. Untargeted results on Imagenet are shown on subplot (b). Classification result on target model are shown at the bottom.

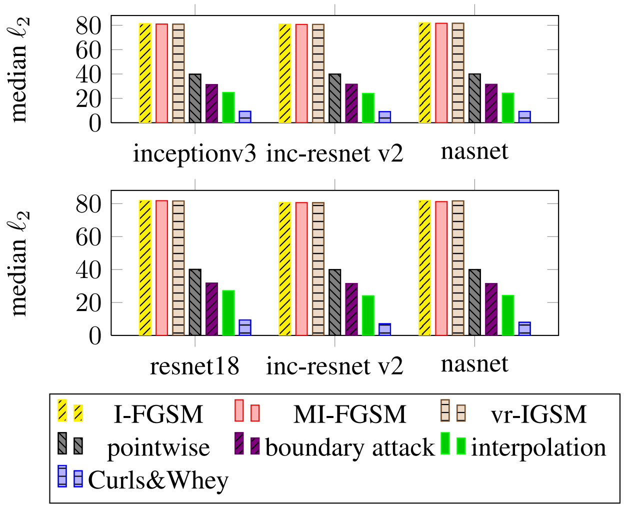


Figure 6. Median ℓ_2 distance comparison of targeted adversarial noises generated using resnet18 (up) and inceptionv3 (down) as substitute model on Tiny-Imagenet.

4.5. Attack on Defence and Ensemble Models

Adversarial training [24] and model ensemble are two widely used defend methods. In Table 4, we use resnet18 as substitute models to attack two adversarially trained models (inceptionv3 and inception-resnet v2) and ensemble model consisting of three models. Although defence methods increase the difficulty of adversarial attack compared with Table 1, the noise magnitude of adversarial examples built by Curls & Whey is still much lower than other attacks.

Table 4. Median and average ℓ_2 distance of adversarial perturbation against adversarially trained models and ensemble model.

target model	attack methods	median	average
inceptionv3(adv)	FGSM	6.5812	9.1681
	I-FGSM	2.8839	3.76
	MI-FGSM	3.8039	4.6529
	vr-IGSM	3.2752	4.1449
	Curls&Whey	**2.0633**	**2.6349**
inc-resnet v2(adv)	FGSM	4.7029	6.2954
	I-FGSM	3.3195	3.9606
	MI-FGSM	3.9919	4.9481
	vr-IGSM	3.3829	4.2706
	Curls&Whey	**2.2852**	**2.7884**
inceptionv3+ inc-resnet v2+ nasnet	FGSM	4.5826	5.9755
	I-FGSM	2.7742	3.595
	MI-FGSM	3.5819	4.5227
	vr-IGSM	3.0785	4.0499
	Curls&Whey	**2.0321**	**2.6187**

5. Conclusion

We propose Curls & Whey, a new black-box attack containing Curls iteration and Whey optimization, to diversify the iterative trajectory and squeeze the adversarial noises respectively. In addition, we integrate interpolation to iterative attack to reduce the difficulty of targeted attacks in black-box scenario significantly. Experimental results on Tiny-Imagenet and ImageNet demonstrate that compared to existing iterative attacks, Curls & Whey generates adversarial examples with smaller ℓ_2 distance and stronger transferability against a variety of target models.

Acknowledgements

This work is supported by the NSFC (under Grant 61876130, U1509206).

References

[1] Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017. 4

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 7

[3] Wieland Brendel, Jonas Rauber, Alexey Kurakin, Nicolas Papernot, Behar Veliqi, Marcel Salathé, Sharada P Mohanty, and Matthias Bethge. Adversarial vision challenge. *arXiv preprint arXiv:1808.01976*, 2018. 2, 5

[4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, Jianguo Li, and Jun Zhu. Boosting adversarial attacks with momentum. *CVPR*, 2018. 1, 2, 3, 5, 6

[5] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 6

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 5

[8] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CVPR*, pages 2261–2269, 2017. 5

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017. 1, 2, 6

[10] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *ICCV*, pages 5775–5783, 2017. 2

[11] Yanpei Liu, Xinyun Chen, Cheng Chih Liu, and Dawn Xiaodong Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. 1, 2, 3, 5

[12] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 2

[13] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. 3

[14] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016. 2

[15] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. 2

[16] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *AsiaCCS*, 2017. 1, 2, 3

[17] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017. 5, 7

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 5

[19] Tegjyot Singh Sethi and Mehmed Kantardzic. Data driven exploratory attacks on black box classifiers in adversarial domains. *Neurocomputing*, 289:129–143, 2018. 3

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[21] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 5

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 5

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, abs/1312.6199, 2014. 1

[24] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. *CoRR*, abs/1705.07204, 2017. 2, 8

[25] Florian Tramèr, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. The space of transferable adversarial examples. *CoRR*, abs/1704.03453, 2017. 3

[26] Beilun Wang, Ji Gao, and Yanjun Qi. A theoretical framework for robustness of (deep) classifiers against adversarial examples. *ICLR Workshop*, 2017. 2

[27] Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018. 1, 2, 4, 6

[28] Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity. *arXiv preprint arXiv:1803.06978*, 2018. 3

[29] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Computer Vision–ECCV 2018*, pages 471–486. Springer, 2018. 3

[30] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017. 5