CGBA: Curvature-aware Geometric Black-box Attack (Supplementary Material)

In this supplementary material, in Section A, we illustrate the boundary search along a semicircular path (BSSP) and the Initialization algorithms. The experimental results on the ImageNet dataset are presented in Section B, while those on the CIFAR10 dataset against two popular classifiers are given in Section C. Moreover, we compare our proposed BSSP over binary search along the direction of the estimated normal vector (BSNV) in Section D and verify its effectiveness through both theoretical analysis and experimental evaluation. Finally, we demonstrate and compare adversarial samples and their corresponding perturbations against different classifiers in Section E.

A. Algorithms

The proposed CGBA is based on querying the boundary point along a semicircular path on a restricted 2D plane. The method to query a boundary point using BSSP is shown in Algorithm 3. This method is quite similar to the binary search. However, we conduct the boundary search on a semicircular path in a 2D plane rather than following a straight line 1D path. Starting from an adversarial and a non-adversarial point as two ends, we gradually decrease the range of the distance between adversarial and non-adversarial points on the semicircular trajectory to obtain the desired boundary point within a certain error limit. Line-3 and line-4 of the algorithm indicate the obtained unit directions $\hat{\boldsymbol{\zeta}}_c$ and $\hat{\boldsymbol{\zeta}}_{adv}$ towards a non-adversarial point $\boldsymbol{x}_c$ and an adversarial point $\boldsymbol{x}_{b_t}$ on the semicircular trajectory from source $\boldsymbol{x}_s$, respectively. Then, we obtain a perturbed point $\boldsymbol{x}_r = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_r)$ on the semicircular trajectory in the resultant direction $\hat{\boldsymbol{\zeta}}_r$, obtained from the aforementioned directions as shown in line-6, where $d(\hat{\boldsymbol{\zeta}}_r)$ is the added perturbation to follow the semicircular trajectory as given in Eq. 6. From line-8 to line-11, the query for $\boldsymbol{x}_r$ is performed to know whether $\boldsymbol{x}_r$ is adversarial or not. If $\boldsymbol{x}_r$ is adversarial, $\hat{\boldsymbol{\zeta}}_r$ is replace by $\hat{\boldsymbol{\zeta}}_{adv}$ to reduce the search range, and vice-versa. The process of reducing the range is continued until we obtain the desired $\boldsymbol{x}_{b_{t+1}}$ within a certain accuracy.

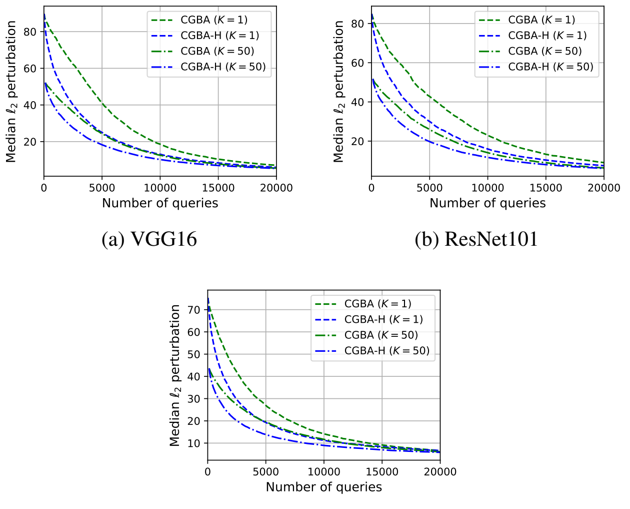
Algorithm 3: BSSP

```
1  Inputs: Source image $\boldsymbol{x}_s$, indicator function $\phi(.)$,
     non-adversarial point $\boldsymbol{x}_c$ ($\phi(\boldsymbol{x}_c) = -1$) on the
     semicircle, adversarial sample at the intersection of the
     boundary and the semicircle $\boldsymbol{x}_{b_t}$, tolerance $\epsilon = 0.0001$.
2  Output: new boundary point $\boldsymbol{x}_{b_{t+1}}$.
3  $\hat{\boldsymbol{\zeta}}_c = (\boldsymbol{x}_c - \boldsymbol{x}_s)/\|(\boldsymbol{x}_c - \boldsymbol{x}_s)\|_2$   /* direction of
     a non-adversarial point $\boldsymbol{x}_c$ on the
     semicircle from $\boldsymbol{x}_s$ */
4  $\hat{\boldsymbol{\zeta}}_{adv} = (\boldsymbol{x}_{b_t} - \boldsymbol{x}_s)/\|(\boldsymbol{x}_{b_t} - \boldsymbol{x}_s)\|_2$   /* direction
     of an adversarial point $\boldsymbol{x}_{b_t}$ on the
     semicircle from $\boldsymbol{x}_s$ */
5  while True do
6      $\hat{\boldsymbol{\zeta}}_r = (\hat{\boldsymbol{\zeta}}_c + \hat{\boldsymbol{\zeta}}_{adv})/\|(\hat{\boldsymbol{\zeta}}_c + \hat{\boldsymbol{\zeta}}_{adv})\|_2$
7      $\boldsymbol{x}_r = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_r)$   /* to obtain $\boldsymbol{x}_r$ on the
         semicircle towards $\hat{\boldsymbol{\zeta}}_r$ */
8      if $\phi(\boldsymbol{x}_r) = 1$ then
9          $\hat{\boldsymbol{\zeta}}_{adv} = \hat{\boldsymbol{\zeta}}_r$
10     else
11         $\hat{\boldsymbol{\zeta}}_c = \hat{\boldsymbol{\zeta}}_r$
12     if $\|\boldsymbol{d}(\hat{\boldsymbol{\zeta}}_{adv}) - \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_c)\|_2 \leq \epsilon$ then
13         $\boldsymbol{x}_{b_{t+1}} = \boldsymbol{x}_s + \boldsymbol{d}(\hat{\boldsymbol{\zeta}}_{adv})$
14         break
```

Algorithm 4 shows the process of finding a better initial boundary point from a set of random directions to the adversarial region. If $\boldsymbol{x}_k$ denotes any point in the adversarial region, then the direction of $\boldsymbol{x}_k$ from a source $\boldsymbol{x}_s$ can be estimated as $\boldsymbol{\Theta}_k = (\boldsymbol{x}_k - \boldsymbol{x}_s)/\|\boldsymbol{x}_k - \boldsymbol{x}_s\|_2$. For the targeted attack, we randomly choose a set of K samples $\{\boldsymbol{x}_k\}_{k=1}^{K}$ of the target class and obtain a set of K directions $\{\boldsymbol{\Theta}_k\}_{k=1}^{K}$ to the adversarial region. By using this set of random directions, we can get a better initial boundary point $\boldsymbol{x}_{b_1}$ at the cost of additional queries. We provide the explanation of the Algorithm 4 as follows.

While the line-3 of Algorithm 4 finds the minimum ℓ_2-norm of perturbation required to make $\boldsymbol{x}_s$ adversarial towards $\frac{\boldsymbol{\Theta}_1}{\|\boldsymbol{\Theta}_1\|_2}$, line-4 finds the adversarial image in that direction. The line-6 to line-12 is used to conduct an exhaustive search to find the direction that offers the best initial boundary point among the K directions. In finding the best initial boundary point, for a direction $\frac{\boldsymbol{\Theta}_i}{\|\boldsymbol{\Theta}_i\|_2}$, line-7 adds the current perturbation d_{best} towards $\frac{\boldsymbol{\Theta}_i}{\|\boldsymbol{\Theta}_i\|_2}$ to obtain a perturbed $\boldsymbol{x}_p$. Then, line-8 checks whether $\boldsymbol{x}_p$ is adversarial

```
Algorithm 4: Initialization
1  Inputs: Source image $\boldsymbol{x}_s$, a set of directions towards the adversarial region $\{\boldsymbol{\Theta}_k\}_{k=1}^{K}$, indicator function $\phi(.)$ of target classier output.
2  Output: Initial boundary point $\boldsymbol{x}_{b_1}$.
3  $r \leftarrow \min\{r > 0 : \phi(\boldsymbol{x}_s + r * \frac{\boldsymbol{\Theta}_1}{\|\boldsymbol{\Theta}_1\|_2}) = 1\}$   /* to find the minimum perturbation towards $\frac{\boldsymbol{\Theta}_1}{\|\boldsymbol{\Theta}_1\|_2}$ to make $\boldsymbol{x}_s$ adversarial */
4  $\boldsymbol{x}_b = \boldsymbol{x}_s + r * \frac{\boldsymbol{\Theta}_1}{\|\boldsymbol{\Theta}_1\|_2}$
5  $d_{best} = \|\boldsymbol{x}_b - \boldsymbol{x}_s\|_2$
6  for $i = 2 : K$ do
7      $\boldsymbol{x}_p = \boldsymbol{x}_s + d_{best} * \frac{\boldsymbol{\Theta}_i}{\|\boldsymbol{\Theta}_i\|_2}$
8      if $\phi(\boldsymbol{x}_p) = 1$ then
9          $\boldsymbol{x}_{b_1} \leftarrow BinarySearch(\boldsymbol{x}_s, \boldsymbol{x}_p, \phi)$
10         $d_{new} = \|\boldsymbol{x}_{b_1} - \boldsymbol{x}_s\|_2$
11         if $d_{new} < d_{best}$ then
12             $d_{best} = d_{new}$
```



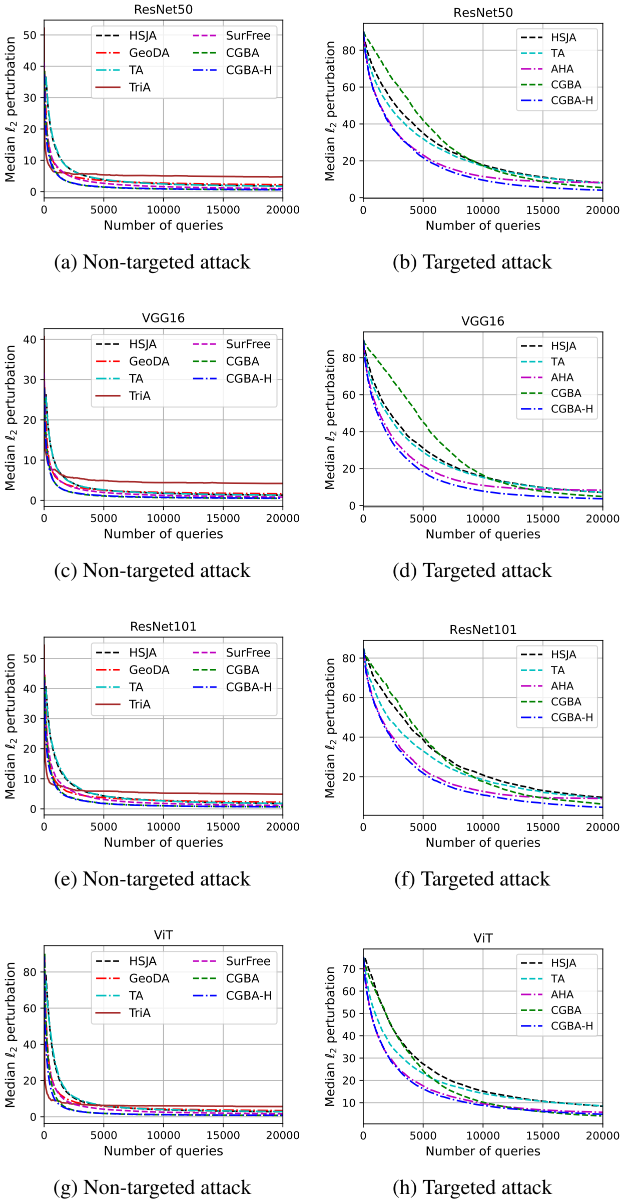
(a) VGG16

(b) ResNet101

(c) ViT

Figure 9: Performance comparison between a random initialization and the proposed initialization.

or not. If $\boldsymbol{x}_p$ is adversarial, only then we conduct a binary search to obtain a new boundary point, as shown in line-9, that further improves the obtained perturbation d_{best}. This process is continued to get the best boundary point among the given K directions. Figure 9 compares the random initialization ($K = 1$) and initialization with Algorithm 4 for $K = 50$ against VGG16, ResNet101 and ViT classifiers. It is observed that the proposed initialization algorithm finds a much better boundary point against the aforementioned classifiers and thus notably improves the performance for targeted attacks.



(a) Non-targeted attack

(b) Targeted attack

(c) Non-targeted attack

(d) Targeted attack

(e) Non-targeted attack

(f) Targeted attack

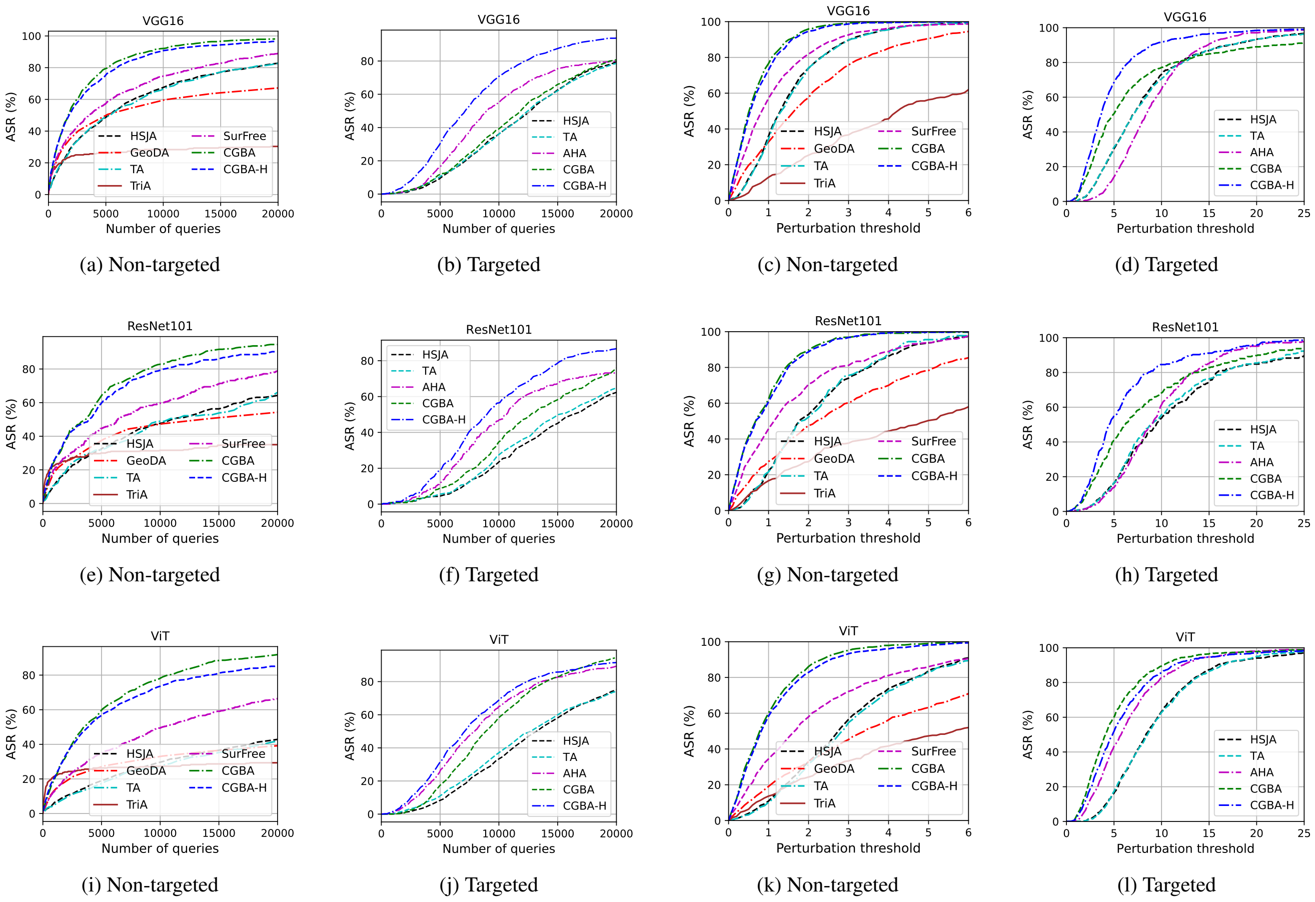
(g) Non-targeted attack

(h) Targeted attack

Figure 10: Variation of ℓ_2-norm of perturbation with the number of queries against ResNet50, VGG16, ResNet101 and ViT on ImageNet.

B. Results on ImageNet

Figure 10 depicts the variation of median ℓ_2-norms of perturbation with queries for both non-targeted and targeted attacks against ResNet50, VGG16, ResNet101 and ViT. Based on these findings, it is evident that for non-targeted attacks, both CGBA and CGBA-H outperform the baseline methods significantly. In contrast, for targeted attacks, while CGBA-H outperforms all the baselines, the obtained perturbation using CGBA is expectedly higher when the query budget is not sufficiently high due to the high cur-



(a) Non-targeted

(b) Targeted

(c) Non-targeted

(d) Targeted

(e) Non-targeted

(f) Targeted

(g) Non-targeted

(h) Targeted

(i) Non-targeted

(j) Targeted

(k) Non-targeted

(l) Targeted

Figure 11: Experimental results of ASR against VGG16, ResNet101 and ViT on ImageNet using different methods.

vature of the boundary. However, with the increase in the number of iterations and corresponding queries, the boundary point gets closer to the source image, resulting in a flatter decision boundary with respect to the viewpoint of the source image. Consequently, with a sufficient query budget, the proposed CGBA also outperforms the baselines.

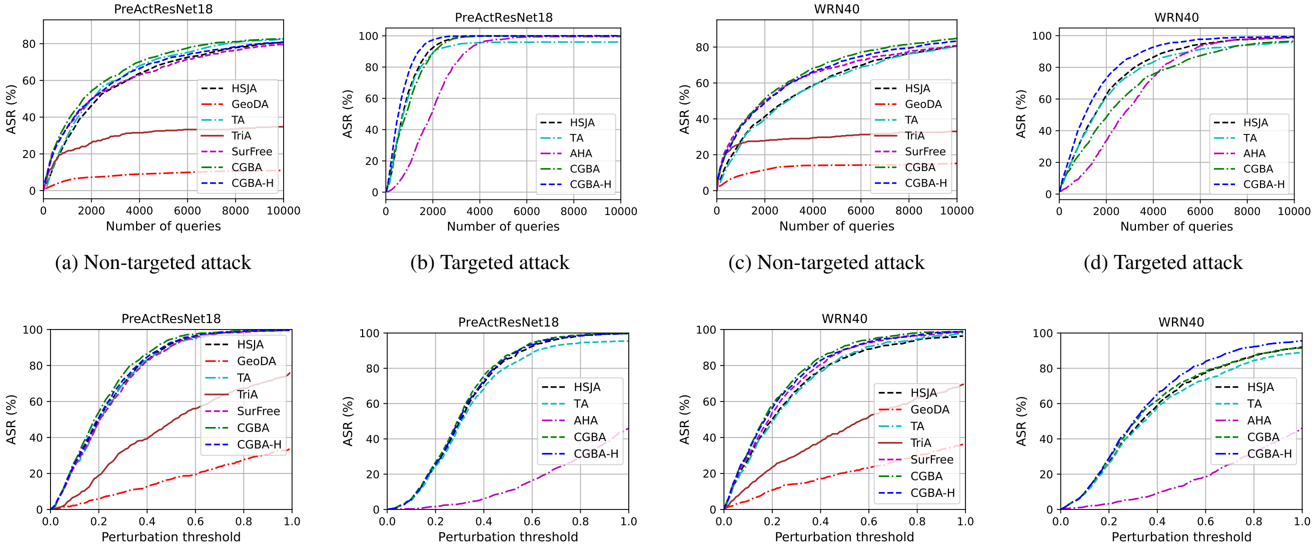
Figure 11 shows the attack success rate (ASR) comparison of the proposed methods with the baselines for the different query budgets and threshold values. The first two columns of the figure depict the obtained ASR for different query budgets with a threshold value of 2.5 for the non-targeted attack and 12 for the targeted attack, while the last two columns show the obtained ASR for different threshold values for a query budget of 20,000. From these experimental results, we observe that CGBA and CGBA-H offer significantly better performance over the baselines for these three popular classifiers, as we have observed for ResNet50.

C. Results on CIFAR10

Our proposed attacks are not restricted to high-dimensional datasets with a large number of classification labels, such as ImageNet. This section further examines their effectiveness in generating adversarial samples for a low-dimensional dataset, CIFAR10, which contains ten classification labels. To perform the experiments, rather than using the dimension-reduced subspace, we consider full-dimensional image space (full-space) to attack against PreActResNet18 [4] and WRN40 [10] using the baselines and the proposed methods. We randomly choose 1000 images for the non-targeted attack and 1000 pairs of images for the targeted attack that are correctly classified by the target classifier. The corresponding results are shown in Table. 4. From this Table, CGBA outperforms all the baselines, as expected, for the non-targeted attack against PreActResNet18. On the other hand, for the non-targeted attack against WRN40, while SurFree performs better with a very low query budget, CGBA outperforms SurFree with a sufficient query budget. However, for the targeted attack, while CGBA-H performs better with smaller query budgets, CGBA shows slightly better performance than CGBA-H with an increase in the query budget. This observation could be explained as follows. First of all, since the number

		Non-targeted							Targeted						
	Queries	1000	2500	5000	7500	10000	15000	20000	1000	2500	5000	7500	10000	15000	20000
PreActResNet18	HSJA [1]	0.531	0.279	0.203	0.174	0.161	0.146	0.140	1.771	0.682	0.421	0.337	0.304	0.268	0.252
	GeoDA [8]	2.41	1.87	1.526	1.349	1.296	1.176	1.116	-	-	-	-	-	-	-
	TA [6]	0.502	0.274	0.199	0.178	0.166	0.150	0.144	1.657	0.670	0.408	0.337	0.301	0.272	0.256
	TriA [9]	0.621	0.504	0.469	0.436	0.433	0.414	0.406	-	-	-	-	-	-	-
	SurFree [7]	0.428	0.270	0.204	0.177	0.163	0.147	0.140	-	-	-	-	-	-	-
	AHA [5]	-	-	-	-	-	-	-	4.096	1.968	1.126	1.053	1.053	1.053	1.053
	CGBA	**0.409**	**0.237**	**0.184**	**0.163**	**0.152**	**0.140**	**0.135**	2.012	0.645	0.391	**0.321**	**0.291**	**0.262**	**0.247**
	CGBA-H	0.435	0.267	1.95	0.172	0.159	0.145	0.140	**1.257**	**0.577**	**0.385**	0.329	0.296	0.266	0.251
WRN40	HSJA [1]	0.739	0.336	0.206	0.166	0.148	0.129	0.123	4.028	1.315	0.596	0.410	0.336	0.271	0.244
	GeoDA [8]	3.241	2.243	1.677	1.491	1.387	1.264	1.162	-	-	-	-	-	-	-
	TA [6]	0.714	0.334	0.209	0.169	0.149	0.132	0.125	3.736	1.229	0.571	0.396	0.330	0.272	0.250
	TriA [9]	0.949	0.697	0.625	0.574	0.541	0.528	0.501	-	-	-	-	-	-	-
	SurFree [7]	**0.493**	0.262	0.187	0.156	0.142	0.127	0.119	-	-	-	-	-	-	
	AHA [5]	-	-	-	-	-	-	-	5.372	2.709	1.359	1.081	1.044	1.041	1.041
	CGBA	0.498	**0.245**	**0.167**	**0.142**	**0.131**	**0.120**	**0.115**	6.221	1.774	0.578	0.383	0.312	**0.256**	**0.231**
	CGBA-H	0.537	0.259	0.172	0.148	0.135	0.122	0.116	**2.690**	**0.878**	**0.465**	**0.351**	**0.302**	0.257	0.238

Table 4: Median ℓ_2-norm of perturbation for different query budgets of our proposed attacks and baselines on CIFAR10 dataset.



(a) Non-targeted attack

(b) Targeted attack

(c) Non-targeted attack

(d) Targeted attack

(e) Non-targeted attack

(f) Targeted attack

(g) Non-targeted attack

(h) Targeted attack

Figure 12: Experimental results of ASR against PreActResNet18 and WRN40 on CIFAR10 using different methods.

of classes in the CIFAR10 dataset is much smaller, the adversarial region for the targeted attack on CIFAR10 is much wider in the 2D search plane as compared to the adversarial region of the ImageNet dataset with 1000 classes. Secondly, with the increase in the number of queries, the obtained boundary point is getting closer to the source image. Thus, the curvature of the boundary becomes flatter from the viewpoint of the source. Therefore, with a sufficiently large query budget, which in turn requires a large number of iterations, CGBA performs better on a low-dimensional dataset like CIFAR10. This is consistent with our previous observations that CGBA performs better on lower curvature boundaries while CGBA-H can further adapt to high curvature.

Moreover, the obtained ASR against the two classifiers is shown in Figure 12. Figures 12(a-d) demonstrate the variation of ASR with queries for a perturbation threshold of 0.3 for the non-targeted attack and 2.5 for the targeted attack. On the other hand, Figures 12(e-h) demonstrate the obtained ASR with different threshold values. To depict Figures 12(e-h), while we consider a query budget of 5000 for the non-targeted attack, we consider a query bud-

	Methods	HSJA [1]	GeoDA [8]	TA [6]	TriA [9]	SurFree [7]	AHA [5]	CGBA	CGBA-H
PreActResNet18	Non-targeted	3281.2	16007	3293.7	5216.4	2867.5	-	**2693.8**	2866.0
	Targeted	9056.2	-	8882.3	-	-	19478.3	9742.0	**7734.3**
WRN40	Non-targeted	3704.9	19119	3876.5	7025.9	2881.4	-	**2860.3**	2948.9
	Targeted	14322.7	-	13806.6	-	-	23543.7	18443.9	**11376.2**

Table 5: AUC comparison against PreActResNet18 and WRN40 for a query budget of 10000 on CIFAR10.

get of 10000 for the targeted attack. We choose two different query budgets due to the faster convergence of the non-targeted attack than the targeted attack. From these figures, it is observed that we get the expected improved performance using our proposed methods. Furthermore, the experimental results of AUC are shown in Table 5. The obtained results demonstrate the consistency in the performance of the proposed methods on different datasets.

D. BSSP versus BSNV

In this section, we compare binary search along the direction of the estimated normal vector (BSNV) and boundary search along a semicircular path (BSSP) to find a new boundary point. First, we show the experimental evaluation of these two methods in finding the new boundary point with the queries spent to estimate the normal vector on the boundary point. Then we provide a theoretical analysis to further justify the improved efficiency of BSSP in comparison with BSNV.

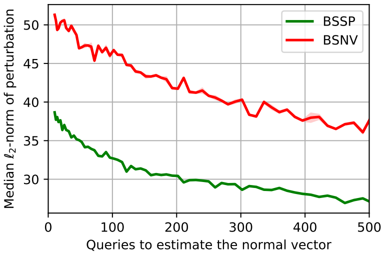


Figure 13: Impact of the normal vector estimation on the performance of BSSP and BSNV in finding a new boundary point.

D.1. Impact of normal vector estimation

We demonstrate the impact of the normal vector estimation on the performance of BSSP and BSNV in finding a new boundary point in Figure 13. To depict this figure, we consider the non-targeted attack against ResNet50 [3] for 1000 test images from ImageNet [2]. First, we estimate the normal vector at the same initial boundary point $\boldsymbol{x}_{b_1}$ considering different numbers of queries. Then, we use both BSNV and the proposed BSSP to find a new boundary point and compare the difference in the resultant perturbations. It is clearly seen that while the performance of both methods improves with the increase of queries spent for the normal vector estimation as expected, BSSP consistently outperforms BSNV by a large margin.

D.2. Theoretical analysis

We theoretically verify the advantage of the BSSP algorithm compared to BSNV in finding a new boundary point. As BSSP is conducted in a 2D plane, we consider a hypothetical parabolic boundary in the 2D plane to perform our analysis for tractability. Let the source image $\boldsymbol{x}_s$ be located at the origin of a xy-coordinate plane spanned by $(\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{\eta}}_t)$ at iteration t, as shown in Figure 14, where $\hat{\boldsymbol{v}}_t$ is the direction of the boundary point $\boldsymbol{x}_{b_t}$ from source $\boldsymbol{x}_s$ and $\hat{\boldsymbol{\eta}}_t$ is the estimated normal vector at $\boldsymbol{x}_{b_t}$. Assume the boundary separating the benign and adversarial regions of $\boldsymbol{x}_s$ in the 2D plane is represented as a parabolic function:

$$y = \frac{x^2}{4p} + h, \tag{9}$$

whose coordinate of the vertex is at $(0, h)$ and the length of the latus rectum is $4p$. Therefore, the optimal perturbation required to make $\boldsymbol{x}_s$ adversarial is h at the given iteration t.

Let us assume the direction of the boundary point $\boldsymbol{x}_{b_t}$ w.r.t. the x-axis is δ_t and the amount of perturbation in that particular direction is $r_t = \|\boldsymbol{x}_{b_t} - \boldsymbol{x}_s\|_2$ at the t-th iteration. Thus, the projection of perturbation in the direction of x-axis and y-axis is given as $a_{x_t} = r_t \cos \delta_t$ and $a_{y_t} = r_t \sin \delta_t$, respectively. Hence, by putting the values of a_{x_t} and a_{y_t} in Eq. 9, r_t is related to δ_t as

$$r_t = \frac{2p \sin \delta_t}{\cos^2 \delta_t} \left[1 - \sqrt{1 - \frac{h}{p} \cot^2 \delta_t} \right]. \tag{10}$$

As can be inferred from Eq. 10, it is possible to find a boundary point at the direction δ_t iff $p \geq h \cot^2 \delta_t$. Now using Eq. 10, we have,

$$a_{x_t} = r_t \cos \delta_t = 2p \tan \delta_t \left[1 - \sqrt{1 - \frac{h}{p} \cot^2 \delta_t} \right] \tag{11}$$

$$a_{y_t} = r_t \sin \delta_t = 2p \tan^2 \delta_t \left[1 - \sqrt{1 - \frac{h}{p} \cot^2 \delta_t} \right]. \tag{12}$$

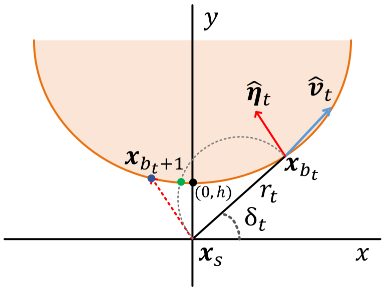


Figure 14: A parabolic boundary with vertex at $(0, h)$, where $\boldsymbol{x}_{b_t}$ represents the boundary point at t-th iteration, and the green and blue points on the boundary denote the obtained boundary point $\boldsymbol{x}_{b_{t+1}}$ using BSSP and BSNV, respectively.

In this analysis, we consider two extreme scenarios for both BSNV and BSSP: linear boundary and curved boundary such that the line towards $\hat{\boldsymbol{v}}_t$ is tangent with the boundary.

D.2.1 Finding boundary point using BSNV

The tangent of any point on the parabola is given by

$$\frac{dy}{dx} = \frac{x}{2p}. \tag{13}$$

Therefore, the slope of the line in the normal direction $\hat{\boldsymbol{\eta}}_t$ on the boundary point (a_{x_t}, a_{y_t}) can be expressed as

$$m = -\frac{2p}{a_x}. \tag{14}$$

Hence, the next boundary point $\boldsymbol{x}_{b_{t+1}}$ toward $\hat{\boldsymbol{\eta}}_t$ is located on the line $y = mx$. Let $(a_{x_{t+1}}, a_{y_{t+1}})$ be the coordinate of the boundary point $\boldsymbol{x}_{b_{t+1}}$ on the xy-coordinate plane. Then, $(a_{x_{t+1}}, a_{y_{t+1}})$ can be obtained at the intersection point of $y = mx$ and $y = \frac{x^2}{4p} + h$. Therefore, we have

$$(a_{x_{t+1}}, a_{y_{t+1}}) = \left(\frac{2h/m}{1 + \sqrt{1 - \frac{h}{pm^2}}}, \frac{2h}{1 + \sqrt{1 - \frac{h}{pm^2}}} \right), \tag{15}$$

where $m = -\frac{2p}{a_x}$.

Case 1: Linear boundary ($p = \infty$)**.** For this case, pushing the image $\boldsymbol{x}_s$ towards the normal direction $\hat{\boldsymbol{\eta}}_t$, the Eq. 15 will result in

$$(a_{x_{t+1}}, a_{y_{t+1}}) = (0, h). \tag{16}$$

Thus, we can conclude that the BSNV method finds the subsequent boundary point with optimal perturbation, $r_{t+1} = h$, if the boundary is linear and the normal vector estimate is accurate. This explains the success of qFool and GeoDA when the decision boundary can be well approximated as a plane.

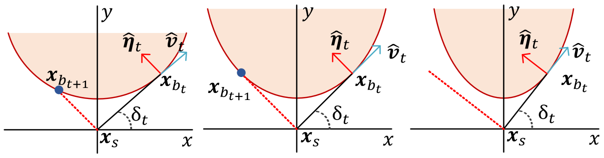


Figure 15: Obtaining a new boundary point $\boldsymbol{x}_{b_{t+1}}$ using BSNV when $\hat{\boldsymbol{v}}_t$ is tangent on the boundary at $\boldsymbol{x}_{b_t}$ for $\delta_t < 45^0$ (left), $\delta_t = 45^0$ (middle) and $\delta_t > 45^0$ (right). BSNV cannot find $\boldsymbol{x}_{b_{t+1}}$ if $\delta_t > 45^0$.

Case 2: Curved boundary ($p = h \cot^2 \delta_t$)**.** With this condition, the line starting from the source $\boldsymbol{x}_s$ with an angle δ_t from the x-axis is the tangent with the boundary at (a_{x_t}, a_{y_t}). From Eq. 11 and Eq. 12, we have $(a_x, a_y) = (2p \tan \delta_t, 2p \tan^2 \delta_t)$. Thus, from Eq. 14, we get the slope of line in the normal direction $\hat{\boldsymbol{\eta}}_t$ on the boundary point as

$$m = -\frac{2p}{2p \tan \delta_t} = -\cot \delta_t. \tag{17}$$

Now, if we use the BSNV method to find $\boldsymbol{x}_{b_{t+1}}$, from Eq. 15 we can find a valid boundary point $\boldsymbol{x}_{b_{t+1}}$, only if $pm^2 \geq h$. Therefore, using the condition $p = h \cot^2 \delta_t$ and Eq. 17, we get

$$\delta_t \leq 45^0. \tag{18}$$

Thus, if the line in the direction of δ_t w.r.t. x-axis is the tangent on $\boldsymbol{x}_{b_t}$, the BSNV method will find the subsequent boundary point $\boldsymbol{x}_{b_{t+1}}$ only if $\delta_t \leq 45^0$. Consider the extreme condition when $\delta_t = 45^0$, for which we have $m = -1$ and $p = h$. Therefore, from Eq. 15, the coordinate of the subsequent boundary point can be written as

$$(a_{x_{t+1}}, a_{y_{t+1}}) = (-2h, 2h). \tag{19}$$

The amount of perturbation of the obtained boundary point $\boldsymbol{x}_{b_{t+1}}$ is $r_{t+1} = 2\sqrt{2}h$, which is same as $r_t = 2\sqrt{2}p$, and the iterative querying process will not converge. So, we can conclude that if $p = h \cot^2 \delta_t$, finding the subsequent boundary point using the BSNV method converges iif $\delta_t < 45^0$ in this scenario, as shown in Figure 15.

D.2.2 Finding boundary point using BSSP

In this subsection, we theoretically analyze the amount of perturbation required to make $\boldsymbol{x}_s$ adversarial by using the BSSP method to find the boundary point in the 2-D plane spanned by $(\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{\eta}}_t)$. The boundary point $(a_{x_{t+1}}, a_{y_{t+1}})$ can be simply obtained by finding the intersection of the parabolic boundary given in Eq. 9 and the circle specified by the following equation.

$$(x - \frac{r_t}{2} \cos \delta_t)^2 + (y - \frac{r_t}{2} \sin \delta_t)^2 = \frac{r^2}{4}.$$

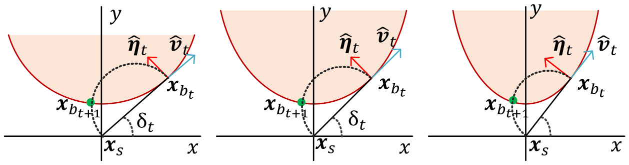


Figure 16: Obtaining a new boundary point $\boldsymbol{x}_{b_{t+1}}$ using BSSP when $\hat{\boldsymbol{v}}_t$ is tangent on the boundary at $\boldsymbol{x}_{b_t}$ for $\delta_t < 45^0$ (left), $\delta_t = 45^0$ (middle) and $\delta_t > 45^0$ (right). BSSP finds $\boldsymbol{x}_{b_{t+1}}$ irrespective of the boundary's curvature.

Thus, we have

$$a_{x_{t+1}}^2 + (\frac{a_{x_{t+1}}^2}{4p} + h)^2 - \frac{2h \cot \delta_t}{1 + \sqrt{1 - \frac{h}{p} \cot^2 \delta_t}} a_{x_{t+1}} - \frac{2h}{1 + \sqrt{1 - \frac{h}{p} \cot^2 \delta_t}} (\frac{a_{x_{t+1}}^2}{4p} + h) = 0. \quad (20)$$

Case 1: Linear boundary $(p = \infty)$**.** Under this condition, the coordinate of $\boldsymbol{x}_{b_{t+1}}$ can be calculated by solving the Eq. 20 as

$$(a_{x_{t+1}}, a_{y_{t+1}}) = (0, h). \quad (21)$$

Hence, the BSSP also finds the optimal boundary point $\boldsymbol{x}_{b_{t+1}}$ with minimum perturbation $\|\boldsymbol{x}_{b_{t+1}} - \boldsymbol{x}_s\|_2 = h$ as it is obtained by using BSNV for a linear boundary.

Case 2: Curved boundary $(p = h \cot^2 \delta_t)$**.** In this case, Eq. 20 can be written as

$$\frac{a_{x_{t+1}}^4}{16h^2} + a_{x_{t+1}}^2 - 2h \cot \delta_t a_{x_{t+1}} + h^2 = 0. \quad (22)$$

One solution of the above equation is the coordinate of the current boundary point $\boldsymbol{x}_{b_t}$. As the coefficients of the above equation are real, there must be another real solution irrespective of the value of δ_t. Thus, for a given boundary point $\boldsymbol{x}_{b_t}$, the proposed BSSP method ensures finding the subsequent boundary point $\boldsymbol{x}_{b_{t+1}}$ irrespective of the value of δ_t. As $p = h \cot^2 \delta_t$ and the curvature is related to the latus rectum $4p$, we can say, conversely, that the proposed BSSP is guaranteed to find the next boundary point no matter what the boundary curvature is, and it is depicted in Figure 16. In contrast, as we have seen, BSNV cannot find a new boundary point when $\delta_t > 45^0$ under the condition of $p = h \cot^2 \delta_t$.

As a concrete example, consider the extreme conditions that we consider for BSNV above, where the direction of $\boldsymbol{x}_{b_t}$ from $\boldsymbol{x}_s$ is a tangent at $\boldsymbol{x}_{b_t}$ and creates an angle $\delta_t = 45^0$. Therefore, Eq. 20 satisfying these conditions can be written as

$$\frac{a_{x_{t+1}}^4}{16h^2} + a_{x_{t+1}}^2 - 2h a_{x_{t+1}} + h^2 = 0. \quad (23)$$

By solving Eq. 23, we can obtain the coordinate of the subsequent boundary point $\boldsymbol{x}_{b_{t+1}}$ as

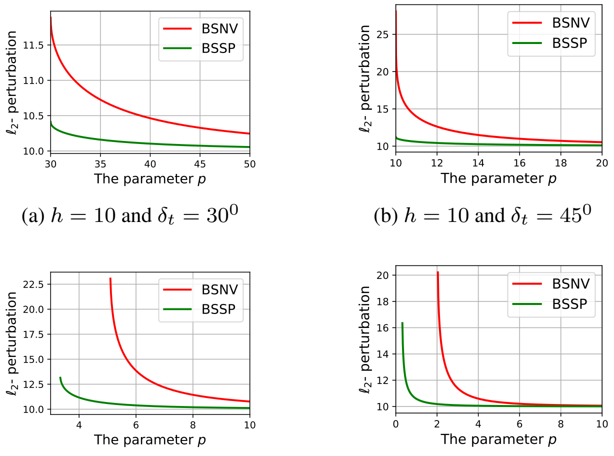
$$(a_{x_{t+1}}, a_{y_{t+1}}) = (-0.4135h, 1.0427h). \quad (24)$$

The perturbation of the new boundary point $\boldsymbol{x}_{b_{t+1}}$ is

$$r_{t+1} = \sqrt{(-0.4135h)^2 + (1.0427h)^2} = 1.1217h.$$

Hence, under the conditions of $p = h \cot^2 \delta_t$ and $\delta_t = 45^0$, the amount of reduction in perturbation using BSSP as compared to BSNV is obtained as

$$\frac{2\sqrt{2}h - 1.1217h}{2\sqrt{2}h} = 60.3\%.$$



(a) $h = 10$ and $\delta_t = 30^0$

(b) $h = 10$ and $\delta_t = 45^0$

(c) $h = 10$ and $\delta_t = 60^0$

(d) $h = 10$ and $\delta_t = 80^0$

Figure 17: Performance Comparision for different values of parameter p.

D.2.3 Impact of different curvature

We further investigate the impact of the boundary curvature on the performance of these two search algorithms. We have already seen that these two methods achieve the same optimal performance for the linear boundary $(p = \infty)$. However, when $p = h \cos^2 \delta_t$ and $\delta_t = 45^0$, BSSP can reduce the perturbation by about 60% as compared to BSNV. Figure 17 shows the performance comparison of BSNV and BSSP for different values of p (with smaller p corresponding to higher curvature). We consider $h = 10$ and $\delta_t =$

$30^0, 45^0, 60^0$ & 80^0 to obtain the solutions of the aforementioned equations for different values of p numerically. From this figure, we observe that BSSP uniformly outperforms BSNV, with dramatic improvement when the curvature is high. Moreover, as shown clearly in Figure 17(c,d), while BSNV fails when parameter p falls below a certain threshold, BSSP remains functioning under such high curvature settings.

E. Visualizing perturbation

In this section, we visualize obtained adversarial images and corresponding perturbations for different query budgets for both non-targeted and targeted attacks against ResNet50, VGG16 and ViT. We use test images from the ILSVRC2012's validation set [2] that are correctly classified by the target classifiers. In Figure 18, the first row of each sub-figures demonstrates the source image 'Sea urchin' and crafted adversarial examples for different query budgets by using CGBA for non-targeted attacks. While the adversarial samples crafted against ResNet50 are misclassified as 'Rock-crab', the obtained perturbations against VGG16 and ViT are misclassified as 'Lionfish'. Moreover, the second row of each of the sub-figures depicts amplified (10 times) perturbations for different query budgets. Likewise, Figure 20 depicts the adversarial examples and corresponding perturbations of 'Lionfish", crafted by CGBA-H, that are misclassified as 'Sea urchin' by different classifiers. From Figures 18 and 20, we can visualize how the perturbations diminish with the increase of queries. Furthermore, Figures 19 and 21 show the difference of the obtained amplified perturbations between different classifiers. From these figures, it is observed that the crafted perturbations vary with classifiers. Because of this variation in crafted perturbations from one classifier to another, the obtained adversarial image in one classifier is not directly transferable to another in case of a decision-based attack.

References

[1] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2019. 4, 5

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 5, 8

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 3

[5] Jie Li, Rongrong Ji, Peixian Chen, Baochang Zhang, Xiaopeng Hong, Ruixin Zhang, Shaoxin Li, Jilin Li, Feiyue Huang, and Yongjian Wu. Aha! adaptive history-driven attack for decision-based black-box models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16168–16177, 2021. 4, 5

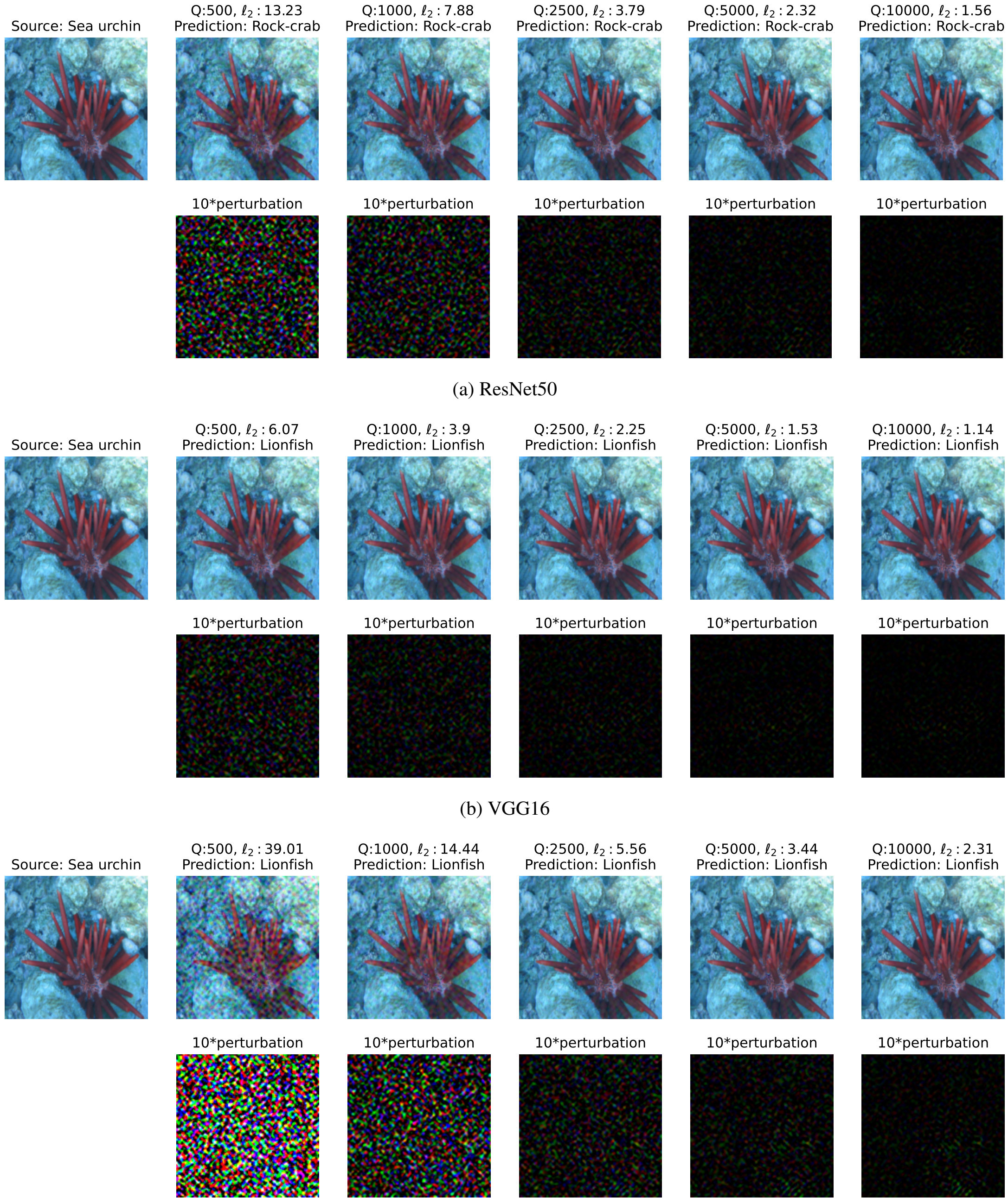
[6] Chen Ma, Xiangyu Guo, Li Chen, Jun-Hai Yong, and Yisen Wang. Finding optimal tangent points for reducing distortions of hard-label attacks. *Advances in Neural Information Processing Systems*, 34, 2021. 4, 5

[7] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10430–10439, 2021. 4, 5

[8] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8446–8455, 2020. 4, 5

[9] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *ECCV*, 2022. 4, 5

[10] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3



(a) ResNet50

(b) VGG16

(c) ViT

Figure 18: Obtained adversarial images and corresponding amplified perturbations with different query budgets against different classifiers using non-targeted CGBA.

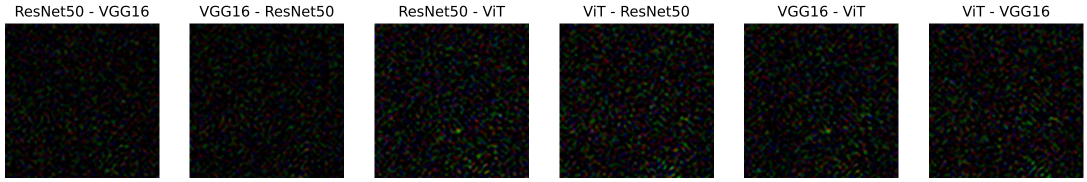
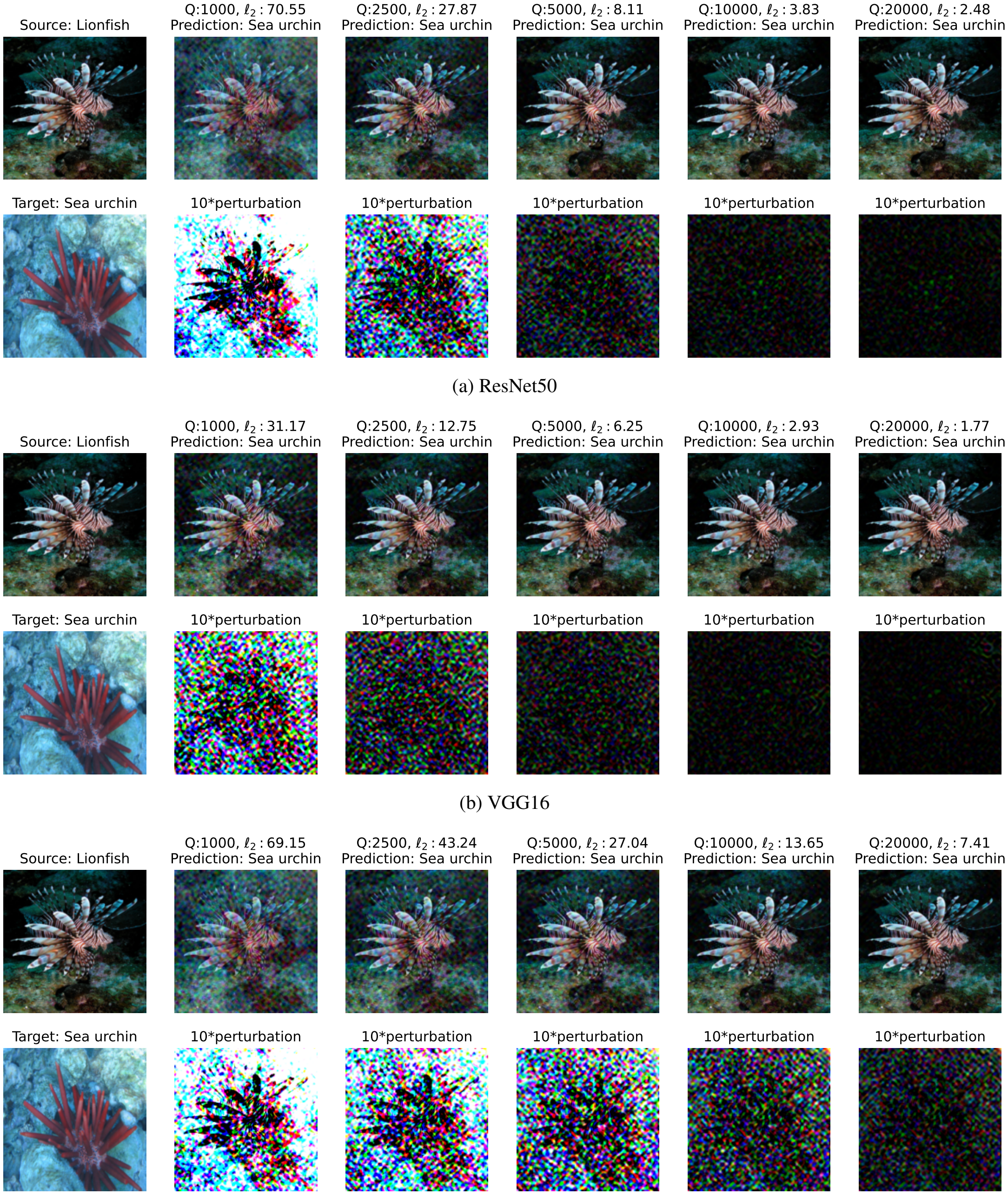


Figure 19: Difference in crafted amplified perturbation in Figure 18 between different classifiers for a query budget of 5000.



(a) ResNet50

(b) VGG16

(c) ViT

Figure 20: Obtained adversarial images and corresponding amplified perturbations with different query budgets against different classifiers using targeted CGBA-H.

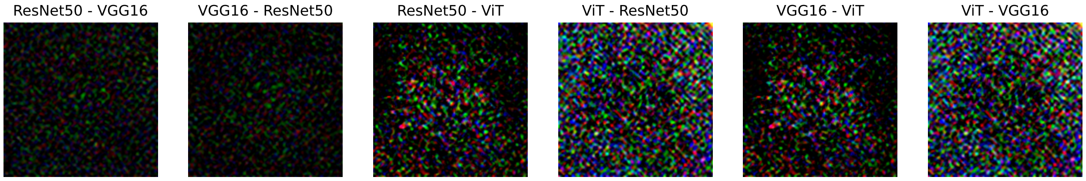


Figure 21: Difference in crafted amplified perturbation in Figure 20 between different classifiers for a query budget of 10000.